

HP QuickTest Professional

ソフトウェア・バージョン : 10.00

ユーザーズ・ガイド 第 2 巻

製造部品番号 : T6511-99019

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 - 2008 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。

<http://www.hp.com/go/hpsoftwaresupport>

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。

- 関心のある内容の技術情報の検索
- 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

この目次は、『HP QuickTest Professional ユーザーズ・ガイド』の第1巻と第2巻両方の章を含みます。

本書について	xix
本ガイドの構成	xx
対象読者	xxii
QuickTest Professional オンライン・ドキュメント	xxiii
その他のオンライン・リソース	xxv

第1部：QUICKTEST PROFESSIONAL の概要（第1巻）

第1章：はじめに	3
QuickTest を使用したテスト	5
テスト・プロセスについて	6
エキスパート・ビューでのプログラミング	12
関数と関数ライブラリについて	12
Quality Center を使ったテスト・プロセスの管理	13
Business Process Testing について	14
必要なアクセス権の設定	15
サンプル・サイトの使用方法	16
ライセンス情報の変更	16
QuickTest ソフトウェアの更新	16
第2章：QuickTest の概要	19
QuickTest の開始	20
QuickTest ウィンドウ	23
キーワード・ビュー	27
エキスパート・ビュー	29
関数ライブラリ	30
スタート・ページ	31
ActiveScreen	33
[使用可能なキーワード] 表示枠	34
データ・テーブル	35
[デバッグ ビューア] 表示枠	35
[情報] 表示枠	37

[欠落リソース] 表示枠	38
[プロセス ガイダンス] 表示枠	39
[リソース] 表示枠	40
[テスト フロー] 表示枠	41
[タスク] 表示枠	42
QuickTest コマンドの使用方法	43
QuickTest Professional プログラム・フォルダの参照	71
製品情報の表示	75

第 II 部：テスト・オブジェクトを使用した作業（第 1 巻）

第 3 章：テスト・オブジェクト・モデルについて	81
テスト・オブジェクト・モデルについて	81
テスト・オブジェクト・モデルの概念の適用	85
オブジェクト・リポジトリの種類について	90
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	98
[オブジェクト スパイ] ダイアログ・ボックス	101
第 4 章：オブジェクト認識の設定	107
オブジェクトの認識の設定について	107
[オブジェクトの認識] ダイアログ・ボックスについて	109
スマート認識の設定	123
ユーザ定義のテスト・オブジェクト・クラスの割り当て	133
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	137
ローカルまたは共有 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	138
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	153
オブジェクト・リポジトリからのオブジェクトの削除	156
オブジェクトの検索	157
認識プロパティの保守	164
第 6 章：テストのオブジェクト・リポジトリの使用法	183
[オブジェクト リポジトリ] ウィンドウについて	184
[オブジェクトのプロパティ] ダイアログ・ボックス	199
共有オブジェクト・リポジトリの関連付けの管理	201
リポジトリ・パラメータ値の割り当て	204
実行セッション中のテスト・オブジェクトでの作業	208
第 7 章：オブジェクト・リポジトリの管理	211
オブジェクト・リポジトリの管理について	212
オブジェクト・リポジトリ・マネージャ	214

オブジェクト・リポジトリを使った作業	221
共有オブジェクト・リポジトリでのオブジェクトの管理	227
リポジトリ・パラメータを使用した作業	233
オブジェクトの詳細の変更	238
テスト・オブジェクトの検索	243
結合操作の実行	244
インポート操作およびエクスポート操作の実行	245
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	248
第 8 章：共有オブジェクト・リポジトリの結合	251
共有オブジェクト・リポジトリの結合について	252
オブジェクト・リポジトリ結合ツールについて	254
オブジェクト・リポジトリ結合ツールのコマンドの使用法	260
標準設定の定義	267
2つのオブジェクト・リポジトリの結合	271
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	273
結合の統計情報の表示	280
オブジェクトの矛盾について	281
オブジェクトの矛盾の解決	284
[ターゲットリポジトリ] 表示枠に対するフィルタの設定	286
特定のオブジェクトの検索	287
ターゲット・オブジェクト・リポジトリの保存	289
第 9 章：共有オブジェクト・リポジトリの比較	291
共有オブジェクト・リポジトリの比較について	292
オブジェクト・リポジトリ比較ツールについて	294
オブジェクト・リポジトリ比較ツールのコマンドの使用	298
オブジェクトの相違点について	301
色の設定の変更	302
オブジェクト・リポジトリの比較	304
比較統計情報の表示	306
[リポジトリ] 表示枠に対するフィルタ処理	308
オブジェクト・リポジトリ・ビューの同期	309
特定のオブジェクトの検索	310

第 III 部：テストの設計（第 1 巻）

第 10 章：テストの作成 – 概要	315
テストの作成について	315
使用する方法論の決定 – キーワード駆動型または記録	317
テストについて	319
テストの拡張	321
QuickTest での相対パスの使用	322

第 11 章：テストの管理	327
新規テストの作成	328
既存のテストを開く	328
テストの保存	330
テストの携帯用コピーの作成	332
テストの圧縮	338
テストの解凍	338
テストの印刷	339
第 12 章：キーワード駆動型の方法論を使用したテストの作成	341
キーワード駆動型の方法論について	342
キーワード駆動型の方法論の使用	348
キーワード駆動型の方法論の実装のサンプル	356
第 13 章：記録メカニズムを使用したテストの作成	365
テストの記録について	366
テストの記録	368
記録モードの選択	372
ActiveScreen を使った作業	380
第 14 章：キーワード・ビューを使った作業	387
キーワード・ビューを使った作業について	388
キーワード・ビュー	389
QuickTest のオブジェクト階層について	395
テストへの標準ステップの追加	396
テストへのその他のタイプのステップの追加	411
ステップの部分の変更	415
コメントの使用	415
アクションステップの管理	416
キーワード・ビューでのキーボード・コマンドの使用	419
キーワード・ビューの表示オプションの定義	420
キーワード・ビューでのステップ要素の表示プロパティ	425
キーワード・ビューのブレークポイントを使った作業	427
第 15 章：アクションを使った作業	429
アクションを使った作業について	430
グローバル・データ・シートとアクション・データ・シートの使用	433
[テスト フロー] 表示枠の使用	435
キーワード・ビューのアクション・ツールバーの使用	439
新規アクションの作成	441
アクションを使った作業についてのガイドライン	443
アクションのプロパティの設定	445
アクションのネスト	458
アクションの分割	459

アクション名の変更	461
テストからのアクションの削除	464
アクション・テンプレートの作成	466
第 16 章：高度なアクション機能を使用した作業	469
高度なアクション機能を使用した作業について	469
既存のアクションへの呼び出しの挿入	470
アクション・パラメータの設定	477
アクション・パラメータの使用	480
アクションの呼び出しのプロパティの設定	485
アクション情報の共有	490
エキスパート・ビューのアクションの構文について	492
アクションの終了	495

第 IV 部：テストの拡張（第 1 巻）

第 17 章：チェックポイントについて	499
チェックポイントについて	499
テストへの新しいチェックポイントの追加	500
既存のチェックポイントのテストへの追加	501
チェックポイントの種類について	505
第 18 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	511
オブジェクトのプロパティ値の検査について	511
標準チェックポイントの作成	512
[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	514
[画像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	518
チェックポイントの変更	521
第 19 章：ビットマップの検査	523
ビットマップの検査について	523
ビットマップ比較の微調整	524
ビットマップ・チェックポイントの作成と変更	526
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	530
第 20 章：テーブルの検査	537
テーブルの検査について	537
テーブル・チェックポイントの作成	538
[テーブル チェックポイントのプロパティ]	
ダイアログ・ボックスについて	542
テーブル内容の検査	544
テーブル・プロパティの検査	554

テーブル・チェックポイントの変更	556
第 21 章：テキストの検査	559
テキストの検査について	559
テキスト・チェックポイントの作成	560
テキスト領域チェックポイントの作成	563
[テキスト/テキスト領域チェックポイントのプロパティ]	
ダイアログ・ボックス	565
テキストまたはテキスト領域チェックポイントの変更	578
テキストを検査するための標準チェックポイントの作成	579
第 22 章：データベースの検査	583
データベースの検査について	583
データベースの検査の作成	584
[データベース 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	589
データベース・チェックポイントの変更	599
第 23 章：XML の検査	601
XML の検査について	601
XML チェックポイントの作成	603
XML テスト・オブジェクト操作チェックポイントの XML 階層の更新 (WebService テスト・オブジェクトの場合のみ)	622
XML チェックポイントの変更	630
XML チェックポイントの結果の確認	630
XML オブジェクトとメソッドの使用によるテスト内容の拡張	631
第 24 章：値のパラメータ化	633
値のパラメータ化について	634
ステップおよびチェックポイントの値のパラメータ化	636
テスト・パラメータとアクション入力パラメータの使用	643
データ・テーブル・パラメータの使用	647
環境変数パラメータの使用	653
乱数パラメータの使用	663
パラメータ化したテストの例	665
データ・ドライバによるテストのパラメータ化	670
第 25 章：値の出力	677
値の出力について	677
出力値の作成	678
プロパティ値の出力	683
出力タイプおよび設定の指定	690
テキスト値の出力	697
テーブル値の出力	708
データベース値の出力	722

XML 値の出力.....	727
XML テスト・オブジェクト操作出力値ステップの XML 階層の更新 (WebService テスト・オブジェクトの場合のみ).....	740
既存の出力値のテストへの追加.....	745
第 26 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.....	749
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.....	749
[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.....	750
テキスト認識のガイドライン.....	754
テキスト認識および開発環境.....	756
使用事例シナリオ：画像内のテキストの検査.....	758
第 27 章：値の設定.....	763
値の設定について.....	763
定数およびパラメータ値の設定.....	764
正規表現の概要と使い方.....	769
正規表現の定義.....	772
第 28 章：プログラミング・ロジックを含むステップの追加.....	781
プログラミング・ロジックを含むステップの追加について.....	781
ステップ・ジェネレータを使用したステップの挿入.....	783
条件ステートメントの使用.....	801
ループ・ステートメントの使用.....	806
テストに対する「With」ステートメントの生成.....	809
メッセージの生成.....	815
コメントの追加.....	818
テストの同期化.....	819
第 V 部：関数およびその他のプログラミング・タスクの定義（第 2 巻）	
第 29 章：エキスパート・ビューおよび 関数ライブラリ・ウィンドウを使用した作業.....	829
エキスパート・ビューおよび関数ライブラリ・ウィンドウを 使用した作業について.....	830
エキスパート・ビューの理解と使用.....	831
エキスパート・ビューおよび関数ライブラリ内での操作.....	846
VBScript の基本的な構文の理解.....	856
プログラムの記述の使用.....	865
プログラムによるアプリケーションの実行と終了.....	877
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.....	878

認識プロパティ値の取得と設定	887
ネイティブ・プロパティおよび操作へのアクセス	889
DOS コマンドの実行	891
Windows API を使用したテストおよび関数ライブラリの拡張	891
実行セッション中に報告するステップの選択	894
第 30 章：エキスパート・ビューおよび	
関数ライブラリ・ウィンドウのカスタマイズ	897
[エキスパート ビュー] および関数ライブラリ・ウィンドウの	
カスタマイズについて	898
エディタの動作のカスタマイズ	899
エレメントの見映えのカスタマイズ	902
編集コマンドのカスタマイズ	903
第 31 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	907
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	907
関数ライブラリの管理	909
関連付けられている関数ライブラリを使用した作業	920
関数定義ジェネレータの使用方法	924
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	939
ユーザ定義関数の使い方のヒント	945
テストからの外部定義された関数の実行	947

第 VI 部：テストの実行と分析（第 2 巻）

第 32 章：テストの実行	951
テストの実行について	952
テスト全体の実行	953
テストの一部の実行	954
[実行] ダイアログ・ボックス：[結果の場所] タブ	958
[実行] ダイアログ・ボックス：[入力パラメータ] タブ	960
オプション・ステップの使用	961
テスト・バッチの実行	963
第 33 章：実行セッション結果の表示	967
実行セッション結果の表示について	967
[テスト結果] ウィンドウ	969
実行セッションの結果の表示	978
実行結果の削除	1000
実行セッション中に検出された不具合の送信	1009
テスト結果での WinRunner テスト・ステップの表示	1013
テスト結果の表示のカスタマイズ	1015
第 34 章：実行セッション結果の分析	1019
テスト結果に含まれるスマート認識情報の分析	1019

チェックポイントの結果の表示	1022
パラメータ化された値と出力値の結果の表示	1046
[システム モニタ] の結果表示	1054

第 VII 部 : テストの保守およびデバッグ (第 2 巻)

第 35 章 : テストと関数ライブラリのデバッグ	1061
テストと関数ライブラリのデバッグについて	1062
デバッグ・セッションの速度調節	1064
シングル・ステップ・コマンドの使用	1064
[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	1068
実行セッションの一時停止	1070
ブレークポイントの使用	1070
[デバッグ ビューア] 表示枠	1073
実行エラーの処理	1084
アクションまたは関数のデバッグの練習	1086
第 36 章 : テストの保守	1091
テストが失敗する理由	1092
メンテナンス実行ウィザードを使ったテストの実行	1094
更新モード・オプションを使ったテストの更新	1111

第 VIII 部 : QUICKTEST IDE を使用した作業 (第 2 巻)

第 37 章 : QuickTest ウィンドウのレイアウト	1123
QuickTest ウィンドウのレイアウト変更	1123
ツールバーとメニューのカスタマイズ	1133
複数のドキュメントを使った作業	1146
第 38 章 : リソースの管理	1149
[リソース] 表示枠	1149
第 39 章 : テストへのキーワードの追加	1155
[使用可能なキーワード] 表示枠について	1155
第 40 章 : QuickTest のタスクとコメントの管理	1159
タスクおよび TODO コメントを使った作業	1159
[タスク] 表示枠	1160
[タスク エディタ] ダイアログ・ボックス	1167
第 41 章 : 欠落リソースの処理	1169
欠落リソースの処理について	1169
欠落アクションの処理	1172
欠落している環境変数ファイルの処理	1177
欠落している関数ライブラリの処理	1178

欠落している共有オブジェクト・リポジトリの処理	1180
欠落している回復シナリオの処理	1181
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	1184
第 42 章： データ・テーブルを使った作業	1185
データ・テーブルを使った作業について	1185
グローバル・シートおよびアクション・シートを使った作業	1187
データ・テーブルの保存	1189
データ・テーブルの編集	1190
Quality Center でのデータ・テーブル・ファイルの使用	1200
データベースからのデータのインポート	1201
データ・テーブルでの数式の使用	1204
データ・テーブルのスクリプト・メソッドの使用	1208
第 43 章： プロセス・ガイダンスを使った作業	1209
[プロセス ガイダンス] 表示枠	1210
プロセス・ガイダンスの開始	1212
利用できるプロセスのリストの管理	1213
[プロセス ガイダンス管理] ダイアログ・ボックス	1214

第 IX 部： QUICKTEST の設定（第 2 巻）

第 44 章： グローバル・テスト・オプションの設定	1219
グローバル・テスト・オプションの設定について	1219
[オプション] ダイアログ・ボックスの使用	1220
テストの一般オプションの設定	1223
テストのフォルダ・オプションの設定	1226
ActiveScreen オプションの設定	1229
テストの実行オプションの設定	1243
第 45 章： 個別のテストのオプション設定	1251
[テストの設定] ダイアログ・ボックスの使用	1252
テストのためのプロパティの定義	1255
テストのための実行設定の定義	1259
テストのためのリソース設定の定義	1263
テストのパラメータの定義	1269
テストのための環境設定の定義	1271
テストのための回復シナリオ設定の定義	1279
テストのためのシステム監視の有効化	1284
第 46 章： 実行セッション中にテスト・オプションを設定するための Setting オブジェクトの使用	1289
実行セッション中のテスト・オプションの設定について	1289
テスト・オプションの設定	1290

テスト・オプションの取得	1292
テスト実行の制御	1292
テスト実行設定の追加と削除	1293

第 X 部：高度なテスト機能を使用した作業（第 2 巻）

第 47 章：仮想オブジェクトの学習	1297
仮想オブジェクトの学習について	1297
仮想オブジェクトについて	1299
仮想オブジェクト・マネージャについて	1300
仮想オブジェクトの定義	1302
無効な仮想オブジェクト定義の削除と	1315
第 48 章：回復シナリオの定義と使用	1317
回復シナリオの定義と使用について	1317
回復シナリオを使用するタイミングの決定	1319
回復シナリオの定義	1320
回復シナリオ・ウィザードについて	1324
回復シナリオの管理	1351
テストへの回復シナリオの関連付け	1355
プログラムによる回復メカニズムの制御	1361
第 49 章：QuickTest Script Editor を使った作業	1363
QuickTest Script Editor について	1364
[QuickTest Script Editor] ウィンドウについて	1366
QuickTest Script Editor・ウィンドウのカスタマイズ	1367
[フロー] 表示枠について	1369
[リソース] 表示枠について	1372
表示領域について	1375
テストを使った作業	1377
関数ライブラリを使った作業	1381
第 50 章：QuickTest 操作のオートメーション	1387
QuickTest 操作の自動化について	1388
QuickTest オートメーション・スクリプトを使用する条件	1389
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	1390
QuickTest オートメーション・スクリプトの基本要素の学習	1392
オートメーション・スクリプトの生成	1393
QuickTest オートメーション・リファレンスの使用方法	1394

第 XI 部：QUALITY CENTER を使用した作業（第 2 巻）

第 51 章：Quality Center との統合	1397
Quality Center を使用した作業について	1398

Quality Center との接続と切断.....	1399
QuickTest の Quality Center との統合.....	1404
Quality Center プロジェクトへのテストの保存.....	1406
Quality Center プロジェクトからテストを開く.....	1407
テンプレート・テストを使用した作業.....	1412
Quality Center プロジェクトに格納されているテストの QuickTest からの実行.....	1419
Quality Center テストの実行に関する設定.....	1421
第 52 章： リソースおよび依存関係モデルの使用.....	1429
リソースおよび依存関係モデルの用語.....	1430
リソースおよび依存関係モデルについて.....	1431
アセットの依存関係を使用する利点.....	1433
Quality Center でのリソースおよび依存関係モデルの使用.....	1435
第 53 章： QuickTest アセットのバージョンの表示と比較.....	1443
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.....	1444
QuickTest アセット比較ツール.....	1446
QuickTest アセット・ビューア.....	1456
第 54 章： バージョン・コントロールを使ったアセット管理.....	1461
Quality Center でのアセットのバージョン管理.....	1462
アセットのバージョンの履歴の表示.....	1470
ベースラインの履歴の表示.....	1472
バージョンの履歴とベースラインの履歴.....	1476
第 55 章： Quality Center 9.x でのバージョン・コントロールを 使った作業.....	1477
バージョン・コントロール・サポート付きの Quality Center 9.x プロジェクトからテストを開く.....	1478
QuickTest でのテストのバージョン管理.....	1478
第 XII 部： その他の HP 製品を使用した作業（第 2 巻）	
第 56 章： Business Process Testing を使用した作業.....	1489
Business Process Testing での作業について.....	1489
Business Process Testing での役割について.....	1490
Business Process Testing のテスト方法について.....	1494
第 57 章： WinRunner を使用した作業.....	1501
WinRunner を使用した作業について.....	1501
WinRunner テストの呼び出し.....	1502
WinRunner 関数の呼び出し.....	1505

第 58 章：HP のパフォーマンス・テストおよび Business Availability Center 製品を使用した作業	1511
HP のパフォーマンス・テストおよび Business Availability Center 製品を使用した作業について	1512
QuickTest のパフォーマンス・テストおよび Business Availability Center の使用	1513
パフォーマンス・テスト製品または Business Process Monitor で 使用する QuickTest テストの設計	1514
パフォーマンス・テストまたは Business Process Monitor での テストの挿入と実行	1516
トランザクションの測定	1518
サイレント・テスト・ランナーの使用	1523

第 XIII 部：付録（第 2 巻）

付録 A：各アドイン用にサポートされているチェックポイントおよび 出力値	1531
サポートされているチェックポイント	1532
サポートされている出力値	1534
付録 B：FAQ（よくある質問）	1537
テストの作成	1537
エキスパート・ビューでのプログラミング	1539
動的なコンテンツを使った作業	1541
Web に関する高度な問題	1543
標準 Windows 環境	1546
テストの保守	1547
ローカライズされたアプリケーションのテスト	1549
QuickTest のパフォーマンスの向上	1550
付録 C：カスタムのプロセス・ガイダンス・パッケージの作成	1555
プロセス・ガイダンス・パッケージについて	1555
パッケージ設定ファイルについて	1556
データ・ファイルの作成	1558
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	1559
付録 D：ビットマップ・チェックポイントのカスタマイズ	1561
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	1562
カスタム・ビットマップ・コンペアラの開発	1565
チュートリアル：カスタム・コンペアラの作成	1574
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	1585
索引	I-1

本書について

QuickTest Professional ユーザーズ・ガイドへようこそ。本ガイドでは、QuickTest を使用してアプリケーションをテストする方法について説明します。テストの作成・デバッグ・実行の方法と、テスト・プロセス中に検出された不具合の報告の方法を順を追って説明します。

本章では、次の項目について説明します。

- ▶ 本ガイドの構成 (xx ページ)
- ▶ 対象読者 (xxii ページ)
- ▶ QuickTest Professional オンライン・ドキュメント (xxiii ページ)
- ▶ その他のオンライン・リソース (xxv ページ)

本ガイドの構成

QuickTest Professional の印刷版は 2 つの巻で構成されています。QuickTest Professional のインストールに含まれている本ガイドの PDF 版およびコンテキスト・センシティブ・ヘルプ版では、両巻の情報が 1 つのファイルに納められています。

本ガイドは、以下の部で構成されています。

第 1 巻

第 I 部 QuickTest Professional の概要

QuickTest の概要と、テスト・プロセスの主要な手順を説明します。

第 II 部 テスト・オブジェクトを使用した作業

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 テストの設計

テストを計画および作成する方法、アクションを使用する方法について説明します。

第 IV 部 テストの拡張

チェックポイント、パラメータ、および出力値を挿入し、正規表現を使用する方法について説明します。

第 2 巻

第 V 部 関数およびその他のプログラミング・タスクの定義

QuickTest で、エキスパート・ビューを使用してテストを拡張する方法、エキスパート・ビューと関数ライブラリ・ウィンドウをカスタマイズする方法、およびユーザ定義関数と関数ライブラリを使用する方法について説明します。

第 VI 部 テストの実行と分析

テストを実行し、結果を分析する方法について説明します。

第 VII 部 テストの保守およびデバッグ

実行セッションを制御して、テスト・スクリプトおよび関数ライブラリ内の不具合を特定し切り分ける方法について説明します。

第 VIII 部 QuickTest IDE を使用した作業

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について説明します。

第 IX 部 QuickTest の設定

グローバルおよびローカルの QuickTest テスト・オプションを変更する方法、および実行セッション中にテスト・オプションを設定する方法について説明します。

第 X 部 高度なテスト機能を使用した作業

仮想オブジェクトおよび回復シナリオを使用する方法について説明します。また、より強力なスクリプトを作成するための複数のプログラミング・テクニック、および QuickTest の操作を自動化する方法について説明します。

第 XI 部 Quality Center を使用した作業

HP Quality Center との統合および Quality Center を使った作業について説明します。Quality Center には、テストの実行、結果の収集、結果の分析、不具合の追跡、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。

第 XII 部 その他の HP 製品を使用した作業

テストを実行し、HP の企業向け Microsoft Windows アプリケーション機能テスト・ツールである WinRunner でコンパイルされたモジュールの関数を呼び出す方法について説明します。また、この部では、QuickTest を Business Process Testing と組み合わせて使用方法や QuickTest を、HP の品質集中管理ソリューションである Quality Center (以前の TestDirector) と連携させる方法についても説明します。さらに、この部では HP パフォーマンス・テスト製品やアプリケーション管理製品で使用できるように QuickTest テストを設計するための注意事項についても説明します。

第 XIII 部 付録

よくある質問に関する情報を提供し、カスタマイズされたプロセス・ガイダンス・パッケージを作成する方法について説明します。

対象読者

本ガイドは、すべてのレベルの QuickTest Professional ユーザを対象としています。読者は、機能テストの概念およびプロセスについてある程度理解していて、テストするアプリケーションの側面を把握している必要があります。

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・テストングで使用する資産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) では、次の QuickTest COM および XML リファレンスに関するマニュアルが含まれています。
 - 『**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) では、VBScript, Script Runtime, および Windows Script Host を含む Microsoft VBScript のマニュアルが含まれています。

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.demout.com> です。**[スタート]** > **[プログラム]** > **[QuickTest Professional]** > **[Sample Applications]** > **[Mercury Tours Web Site]** 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>

ようこそ

第 V 部

関数およびその他のプログラミング・ タスクの定義

第 29 章

エキスパート・ビューおよび関数ライブラリ・ウィンドウを使用した作業

QuickTest のテストは、Microsoft のプログラミング言語である VBScript で記述されたステートメントによって構成されています。エキスパート・ビューは、VBScript に慣れているテスト担当者がキーワード・ビューの代わりに使える機能です。QuickTest では VBScript を使用して関数ライブラリも作成できます。

本章では、エキスパート・ビューでの作業方法を説明し、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テストおよび関数ライブラリを拡張する方法を示します。

本章の内容

- ▶ エクスパート・ビューおよび関数ライブラリ・ウィンドウを使用した作業について (830 ページ)
- ▶ エクスパート・ビューの理解と使用 (831 ページ)
- ▶ エクスパート・ビューおよび関数ライブラリ内での操作 (846 ページ)
- ▶ VBScript の基本的な構文の理解 (856 ページ)
- ▶ プログラム的記述の使用 (865 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 (877 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 (878 ページ)
- ▶ 認識プロパティ値の取得と設定 (887 ページ)
- ▶ ネイティブ・プロパティおよび操作へのアクセス (889 ページ)
- ▶ DOS コマンドの実行 (891 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 (891 ページ)
- ▶ 実行セッション中に報告するステップの選択 (894 ページ)

エキスパート・ビューおよび関数ライブラリ・ウィンドウを使用した作業について

エキスパート・ビューでは、アクションを VBScript として表示できます。VBScript に慣れていれば、プログラミングを通じてステートメントの追加と更新を行い、テストおよび関数ライブラリを拡張できます。これにより、テストの機能と柔軟性を向上させることができます。また、[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、テストまたは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、783 ページ「ステップ・ジェネレータを使用したステップの挿入」を参照してください。

エキスパート・ビューに表示されたテストまたは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。エキスパート・ビューでの印刷の詳細については、339 ページ「テストの印刷」を参照してください。関数ライブラリの印刷の詳細については、918 ページ「関数ライブラリの印刷」を参照してください。

エキスパート・ビューの理解と使用

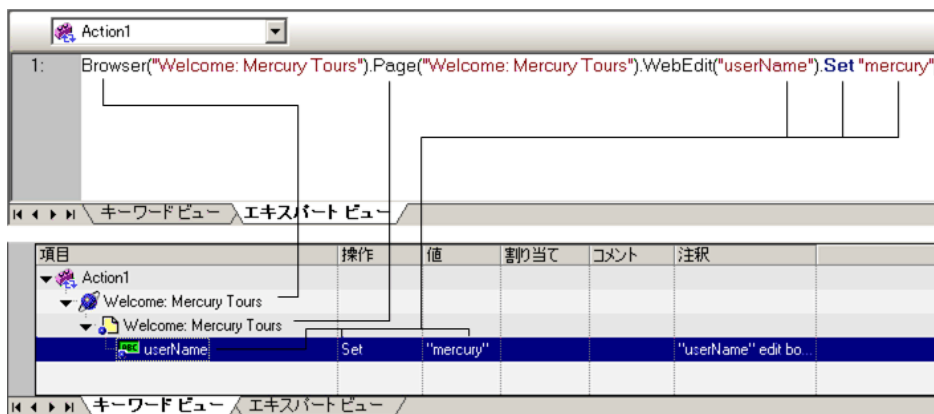
VBScript ステートメントの作業を行うには、キーワード・ビューの代わりに、エキスパート・ビューでテストの作業を行うことができます。それぞれのビューを切り替えることが可能です。ビューを切り替えるには、QuickTest ウィンドウのテスト表示枠の最下部にある [エキスパート ビュー] タブまたは [キーワード ビュー] タブを選択します。

エキスパート・ビューでの作業

エキスパート・ビューには、キーワード・ビューと同じステップおよびオブジェクトが表示されますが、表示される形式が異なります。

- ▶ キーワード・ビューでは、各ステップの情報とともに、オブジェクトの階層がアイコン形式のテーブルとして表示されます。詳細については、第 14 章「キーワード・ビューを使った作業」を参照してください。
- ▶ エキスパート・ビューでは、各ステップが VBScrip の行として表示されます。オブジェクト・ベースのステップでは、VBScript のステートメントがオブジェクト階層を定義します。

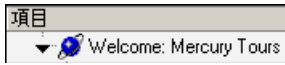
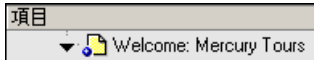
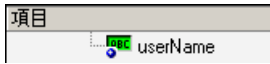

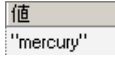
次の図は、同じオブジェクト階層をエキスパート・ビューで表示した場合とキーワード・ビューで表示した場合を示します。



エキスパート・ビューの VBScript の各行は、テストの各ステップに相当します。上の例は、ユーザがエディット・ボックスに「mercury」という名前を挿入するテストのステップを表しています。ステップの階層では、サイト名、

ページ名、ページ内のオブジェクトのタイプと名前、オブジェクトに対して実行された操作名を確認できます。

次の表は、同じステップの各部がキーワード・ビューとエキスパート・ビューのそれぞれでどのように表現されるかを示します。

キーワード・ビュー	エキスパート・ビュー	説明
	Browser ("Welcome: Mercury Tours")	ブラウザ・テスト・オブジェクトの名前は Welcome:Mercury Tours 。
	Page ("Welcome: Mercury Tours")	現在のページの名前は Welcome:Mercury Tours 。
	WebEdit ("userName")	オブジェクトのタイプは WebEdit 。操作の対象となる エディット・ボックスの名前は userName 。
	Set	対象エディット・ボックス で実行されるメソッドは Set 。
	"mercury"	[username] エディット・ ボックスに挿入される値は mercury 。

エキスパート・ビューでは、オブジェクトの記述は、オブジェクトのタイプに続いて、括弧内に表示されます。オブジェクトのリポジトリに格納されているすべてのオブジェクトにとって、オブジェクトの記述としては、オブジェクトの名前で十分です。次の例では、オブジェクトのタイプは **Browser** で、オブジェクトの名前は「**Welcome: Mercury Tours**」です。

Browser ("Welcome:Mercury Tours")

ヒント : テスト・オブジェクトおよび操作の名前は、大文字小文字が区別されません。

オブジェクトの階層では、オブジェクトはピリオドで区切られます。次の例では、**Browser** と **Page** は同じ階層構造内の 2 つの別々のオブジェクトです。

```
Browser("Welcome:Mercury Tours").Page("Welcome:Mercury Tours")
```

オブジェクトに対して実行される操作（メソッド）は、常にステートメントの末尾に、操作に関係する値が後ろに続く形式で表示されます。次の例では、**Set** メソッドを使って、「**userName**」エディット・ボックスに「**mercury**」という文字列を挿入しています。

```
Browser("Welcome:Mercury Tours").Page("Welcome: Mercury Tours").  
WebEdit("userName").Set "mercury"
```

QuickTest はアプリケーション内のオブジェクトに基づいてアプリケーションを処理します。テストに追加したステップは、アプリケーション内のオブジェクトに対して実行される操作に対応します。

QuickTest 内のオブジェクトは、環境によって分けられます。**QuickTest** の標準設定では、標準の Windows 環境のオブジェクトがサポートされます。追加の環境を使用するには、**QuickTest** を開いたときに、アドイン・マネージャで該当する **QuickTest** アドインをロードします。

ほとんどのオブジェクトには、操作が対応付けられています。たとえば、**Back** メソッドは **Browser** オブジェクトに関連付けられています。

オブジェクトと、それに関連付けられている操作とプロパティの完全なリストを参照するには、**[ヘルプ]** > **[QuickTest Professional ヘルプ]** を選択し、**[目次]** タブで **[QuickTest Object Model]** を開きます。

操作を実行するステップを追加する方法の詳細については、836 ページ「エキスパート・ビューまたは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。

VBScript を使用する方法の詳細については、856 ページ「**VBScript** の基本的な構文の理解」を参照してください。


チェックポイント・ステートメントおよび出力ステートメントについて

QuickTest で、ページ、テキスト文字列、テーブル、およびそのほかのオブジェクトを対象としたチェックポイントおよび出力値を作成できます。キーワード・ビューでチェックポイントまたは出力値を作成すると、QuickTest によって、対応する VBScript 行がエキスパート・ビューに作成されます。チェックポイントの実行には **Check** メソッドが使用され、出力値ステップの実行には **Output** メソッドが使用されます。

たとえば、次のステートメントでは、QuickTest によって「New York」という文字列が検査されます。

```
Browser("Mercury Tours").Page("Flight Confirmation").Check
Checkpoint("New York")
```

キーワード・ビューでは、対応するステップが次のように表示されます。

	操作	値	注釈
 Flight Confirmation: Mercury	Check	Checkpoint("New York")	"Flight Confirmation: Mercury" Web page

注：

- ▶ チェックポイントの詳細は、対応する [チェックポイントのプロパティ] ダイアログ・ボックスで設定され、検査対象のオブジェクトとともに保存されます。出力値の詳細は、対応する [出力値のプロパティ] ダイアログ・ボックスで設定され、値を出力する元となるオブジェクトとともに保存されます。エキスパート・ビューに表示されるステートメントは、格納されている情報への参照です。したがって、エキスパート・ビューに手作業でチェックポイント・ステートメントおよび出力値ステートメントを挿入することはできません。また、エキスパート・ビューから別のテストに **Checkpoint** ステートメントおよび **Output** ステートメントをコピーすることもできません。
- ▶ チェックポイントの挿入と変更の詳細については、第 17 章「チェックポイントについて」を参照してください。出力値の挿入と変更の詳細については、第 25 章「値の出力」を参照してください。

パラメータ指定について

QuickTest を使って、テストの値をパラメータ化することにより、テストを拡張できます。「**パラメータ**」とは、外部のデータ・ソースまたはジェネレータから値が割り当てられる変数です。

キーワード・ビューでパラメータを作成すると、対応する VBScript 行がエキスパート・ビューに作成されます。

たとえば、メソッド引数の値をデータ・テーブル・パラメータとして定義した場合、次の構文を使用してデータ・テーブルから値が取得されます。

Object_Hierarchy.Method DataTable (parameterID, sheetID)

項目	説明
Object_Hierarchy	テスト・オブジェクトの階層定義。ピリオドで区切られた 1 つ以上のオブジェクトの並びから成ります。
Method	パラメータ化されたオブジェクトを対象に QuickTest によって実行されるメソッドの名前。
DataTable	データ・テーブルを表す予約済みオブジェクト。
parameterID	値の取得先となるデータ・テーブル内のカラムの名前。
sheetID	値が格納されているシートの名前。パラメータがグローバル・パラメータである場合は、dtGlobalSheet が sheetID となります。

たとえば、Mercury Tours サイト用にテストを作成しているときに、目的地として「San Francisco」を選択するとします。エキスパート・ビューで、次のステートメントがテストに挿入されます。

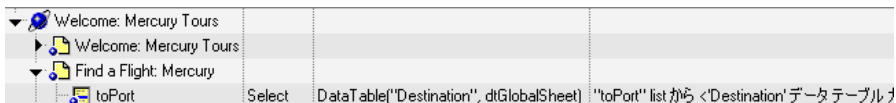
```
Browser("Welcome: Mercury").Page("Find a Flight:").WebList("toPort").
    Select "San Francisco"
```

ここで、目的地の値をパラメータ化し、データ・テーブル内に「**Destination**」カラムを作成したとします。前のステートメントが次のように変更されます。

```
Browser("Welcome: Mercury").Page("Find a Flight:").WebList("toPort").
    Select DataTable("Destination",dtGlobalSheet)
```

ここで、**Select** はメソッド名、**DataTable** はデータ・テーブルを表すオブジェクト、**Destination** はデータ・テーブルのカラムの名前、**dtGlobalSheet** はデータ・テーブルのグローバル・シートの名前です。

キーワード・ビューでは、このステップが次のように表示されます。



パラメータ値の使用方法和定義方法の詳細については、第 24 章「値のパラメータ化」を参照してください。

エキスパート・ビューまたは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、783 ページ「ステップ・ジェネレータを使用したステップの挿入」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、エキスパート・ビューまたは関数ライブラリで入力するときに、変数、テスト・オブジェクト、操作、プロパティ、またはコレクション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、837 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、エキスパート・ビューまたは関数ライブラリで 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、845 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)

エキスパート・ビューまたは関数ライブラリ、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、テスト・オブジェクト、操作、プロパティ、またはコレクションをドロップダウン・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 30 章「エキスパート・ビューおよび関数ライブラリ・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、エキスパート・ビューまた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効にします。
- ▶ 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いか、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たびに、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。

- ▶ **利用可能なテスト・オブジェクト**：テスト・オブジェクト・クラスに続いて開き括弧「(」を入力すると、オブジェクト・リポジトリ内の該当するクラスのすべてのテスト・オブジェクトのリストが表示されます。オブジェクト・リポジトリに、当該クラスに一致するオブジェクトが1つだけある場合には、そのオブジェクトの名前が開き括弧の後ろに、引用符で囲まれた状態で **QuickTest** によって自動的に挿入されます。たとえば、「**Page**(」と入力すると、**QuickTest** によってオブジェクト・リポジトリ内のすべての **Page** テスト・オブジェクトのリストが表示されます。
- ▶ **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、テストまたは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テスト・オブジェクト、操作、プロパティ、コレクション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 [ActiveScreen] または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, 889 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.
```

このテストの編集では、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.
```

このテストの編集では、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。エキスパート・ビューでは、Java または ActiveX オブジェクトで作業するときに、QuickTest によって、アプリケーションのオブジェクトから特定の引数に対する利用可能な値のリストが動的に取得されます。利用可能な値を QuickTest で取得するには、アプリケーションが起動されていて、関連するオブジェクトが表示されている必要があります。たとえば、QuickTest によって、特定の Java リスト・オブジェクトで項目のリストを取得し、**Select** メソッドの **Item** 引数に対する利用可能な値として表示できます。

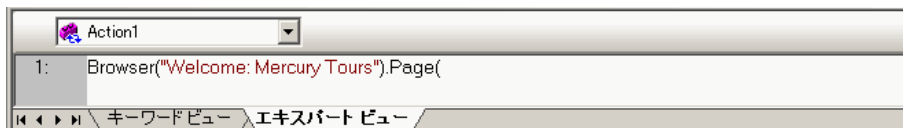
注：記録セッション中にテストを編集する場合、利用可能な引数値はアプリケーションから取得されません。

- ▶ **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。

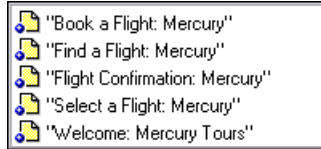
ヒント：CTRL キーを押しながら SPACE キーを押すと、追加が可能な対応するテスト・オブジェクト、操作、プロパティ、コレクション、VBScript 関数、ユーザ定義関数、VBScript 定数、およびユーティリティ・オブジェクトのリストが表示されます。このリストは、オブジェクト・リポジトリにまだ追加されていないオブジェクトを入力した場合でも表示されます。テストに関数が含まれている場合、またはテストが関数ライブラリと関連付けられている場合は、その関数もリストに表示されます。

エキスパート・ビューまたは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。

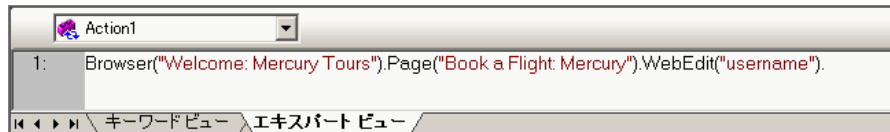
- 1 [ステートメントの自動補完を行う] オプションが選択されていることを確認します（[ツール] > [表示オプション] > [一般] タブ）。
- 2 次の手順のいずれかを実行します。
 - ▶ 関数ライブラリで作業している場合は、手順 4 に進みます。
 - ▶ エキスパート・ビューで作業している場合は、オブジェクトに続いて開き括弧「(」を入力します。



オブジェクト・リポジトリに、当該タイプに一致するオブジェクトが 1 つだけある場合には、そのオブジェクトの名前が開き括弧の後ろに、引用符で囲まれた状態で QuickTest によって自動的に挿入されます。当該タイプのオブジェクトがオブジェクト・リポジトリに複数存在する場合には、QuickTest によってそれらがリストに表示されます。

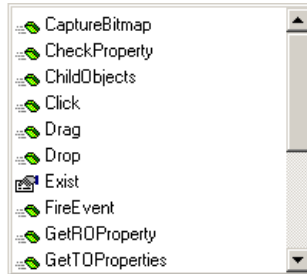


- 3 リスト内のオブジェクトをダブルクリックするか、矢印キーを使ってオブジェクトを選択し、ENTER キーを押します。QuickTest によって、オブジェクトがステートメントに挿入されます。
- 4 次の手順のいずれかを実行します。
 - ▶ エキスパート・ビューで作業している場合は、操作を実行する対象となるオブジェクトの後に、ピリオド (.) を入力します。



- ▶ 関数ライブラリで作業している場合は、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")`

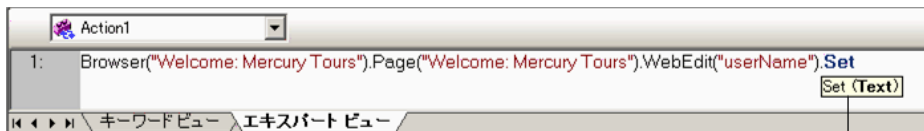
- 5 オブジェクト記述の後ろには, ("username"). のようにピリオド (.) を入力します。オブジェクトに対して利用できる操作およびプロパティのリストが表示されます。



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。
-

- 6 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。次に示すエキスパート・ビューの例のように、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。



ステートメントの完了のツールチップ

上の例では、**Set** メソッドには **Text** という引数が 1 つあります。この引数名は、ボックスの中に挿入するテキストを表します。

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。

- 7 表示される構文に従って、操作の後ろに操作の引数を入力します。



注：エキスパート・ビューでステップを追加した後、追加した新しいステップをキーワード・ビューに表示することができます。エキスパート・ビューに追加したステートメントに構文エラーがある場合、**[キーワードビュー]** を選択したときに、それらのエラーが**[情報]** 表示枠に表示されます。詳細については、862 ページ「VBScript 構文エラーの処理方法」を参照してください。

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、856 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

エキスパート・ビューまたは関数ライブラリで 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、899 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、856 ページ「VBScript の基本的な構文の理解」を参照してください。

エキスパート・ビューおよび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、エキスパート・ビューまたは関数ライブラリ内の特定の行に移動できます。また、エキスパート・ビューまたは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、長いアクションや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。

注：テストが対象の場合、エキスパート・ビューにはアクションが 1 つのみ表示されます。この項で説明する操作機能は、テスト全体ではなく、現在選択されているアクションを対象とするものです。

[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、アクション内または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、エキスパート・ビューおよび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 30 章「エキスパート・ビューおよび関数ライブラリ・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・ボックスを使ってエキスパート・ビューまたは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 [エキスパート ビュー] タブをクリックするか、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。



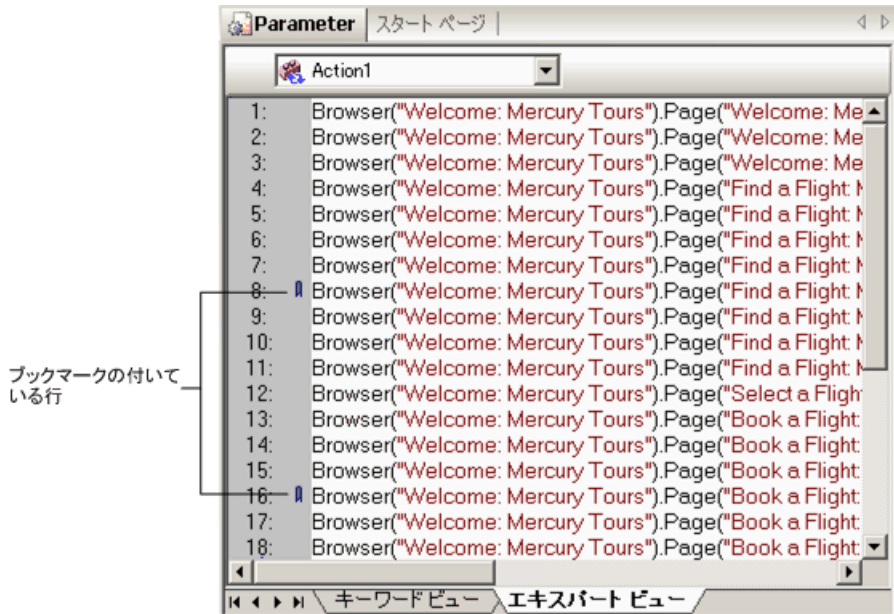
- 3 [行番号] ボックスに移動先の行番号を入力し、[OK] をクリックします。指定した行の先頭にカーソルが移動します。

ブックマークの使用

ブックマークを使って、アクションまたは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。テストの中では、ブックマークは特定のアクション内でのみ使用できます。アクション間を移動するときには維持されず、テストまたは関数ライブラリとともに保存されません。

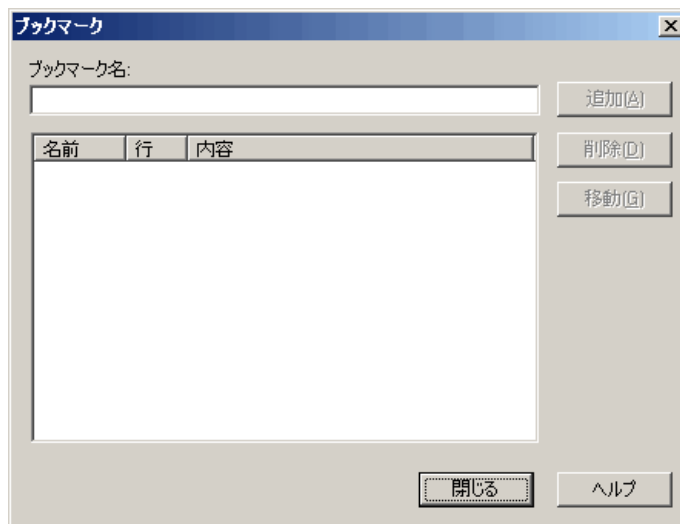
ブックマークを割り当てると、エキスパート・ビューまたは関数ライブラリ内で選択した行の左に、アイコンが付加されます。[ブックマーク] ダイアログ・ボックスの [移動] ボタンを使用して、ブックマークの付いている行に移動できます。


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 **[エキスパート ビュー]** タブをクリックするか、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、エキスパート・ビューまたは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 **[エキスパート ビュー]** タブをクリックするか、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 が現在のアクションまたは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

エキスパート・ビュー内の現在のアクションの中または関数ライブラリ内で、検索する文字列を指定できます。また、ページ・チェックポイントの **[HTML ソースの編集]** ダイアログ・ボックスおよび **[HTML タグの編集]** ダイアログ・ボックスのほか、**["With" ステートメント作成結果]** ダイアログ・ボックスの中の文字列も検索できます。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。

["With" ステートメント作成結果] ダイアログ・ボックスの詳細については、809 ページ「テストに対する「With」ステートメントの生成」を参照してください。ページ・チェックポイントの詳細については、『**HP QuickTest Professional アドイン・ガイド**』のページ・チェックポイントに関する項を参照してください。

文字列を検索するには、次の手順を実行します。

- 1 エクスパート・ビューまたは関数ライブラリで、次のいずれかを実行します。



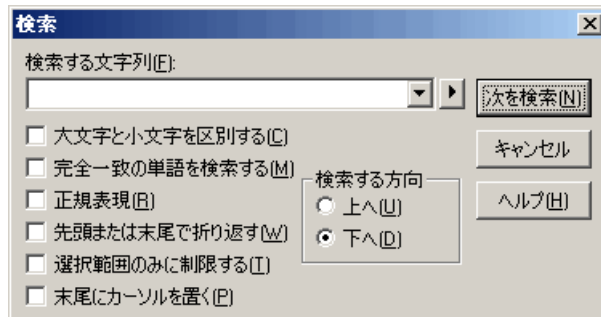
- ▶ **[検索]** ボタンをクリックします。
- ▶ **[編集]** > **[検索]** を選択します。


ヒント：エキスパート・ビューでは、次のいずれかを実行できます。

[編集] > **[詳細設定]** > **[スクリプトに "With" を追加]** を選択してから CTRL キーを押しながら F キーを押します。

[ページチェックポイントのプロパティ] ダイアログ・ボックスで、**[HTML ソースの編集]** または **[HTML タグの編集]** を選択してから右クリックし、表示されたダイアログ・ボックスで **[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、855 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、検索がアクション、ダイアログ・ボックス、または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]**：アクション、ダイアログ・ボックス、または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 アクション、ダイアログ・ボックス、または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 現在のアクション、ダイアログ・ボックスの中、あるいは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

エキスパート・ビュー内の現在のアクションまたは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。**[HTML ソースの編集]** ダイアログ・ボックスおよび **[HTML タグの編集]** ダイアログ・ボックス内の文字列を検索して置換することも可能です。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。

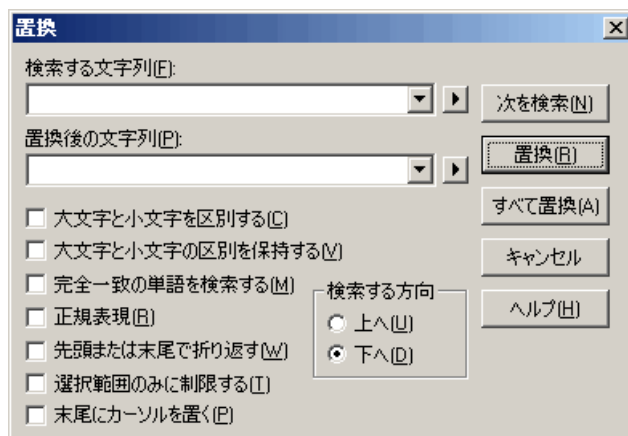
- 1 エクスパート・ビューまたは関数ライブラリで、次のいずれかを実行します。




- ▶ **[置換]** ボタンをクリックします。
- ▶ **[編集]** > **[置換]** を選択します。

ヒント: [ページチェックポイントのプロパティ] ダイアログ・ボックスで、**[HTML ソースの編集]** または **[HTML タグの編集]** を選択してから右クリックし、表示されたダイアログ・ボックスで **[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、855 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。

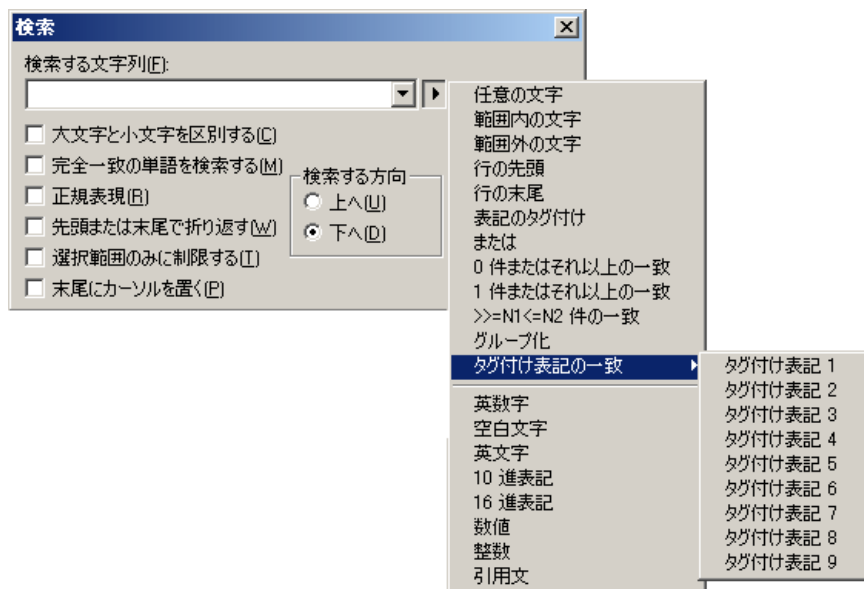
- 5 次の任意のオプションを選択して検索結果の絞込みを行うことも可能です。
- ▶ **[大文字と小文字を区別する]**：検索の際に大文字と小文字を区別します。
[大文字と小文字を区別する] を選択した場合、大文字小文字が、**[検索する文字列]** ボックスに入力した文字列と正確に一致する対象のみが QuickTest によって検索されます。
 - ▶ **[大文字と小文字の区別を保持する]**：**[検索する文字列]** に指定した文字列について、全部小文字、全部大文字、先頭のみ大文字、大文字小文字入り混じりのそれぞれを調べます。**[置換後の文字列]** に指定した文字列は、大文字小文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、検索がアクション、ダイアログ・ボックス、または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]**：アクション、ダイアログ・ボックス、または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アクション、ダイアログ・ボックス、または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アクション、ダイアログ・ボックス、または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 現在のアクションまたはダイアログ・ボックスの中、あるいは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。現在のアクションまたはダイアログ・ボックス、あるいは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、769 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の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

VBScript は簡単に覚えられる強力なスクリプト言語です。VBScript を使用すれば、これまでにプログラミングの経験がなくても、オブジェクトを対象とした簡単な作業から複雑な作業まで実行できます。

この項では、VBScript ステートメントを使用して QuickTest テストまたは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。加えて、エキスパート・ビューからキーワード・ビューに切り替えるときに、**[情報]** 表示枠のドキュメント内に構文エラーがあれば、QuickTest によってそれらがリスト表示されます。構文エラーを修正してなくさない限り、キーワード・ビューに切り替えることはできません。詳細については、862 ページ「VBScript 構文エラーの処理方法」を参照してください。



ヒント：現在のドキュメントの構文は、**[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。テストが開いている場合は、すべてのアクションの構文が確認されます。関数ライブラリが開いている場合は、ライブラリ・スクリプトの構文が確認されます。

エキスパート・ビューまたは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、858 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、859 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、861 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、861 ページ「VBScript テキストの書式設定」および 879 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テストまたは関数ライブラリを拡張する方法の詳細については、878 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

テストまたは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。次の例は、単純な値のための変数を定義する方法を示します。

```
MyVar = Browser("Mercury Tours").Page("Mercury Tours").  
    WebEdit("username").GetTOProperty("type")
```

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、テストまたは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 **passengers** 変数を宣言し、現在のアクションまたは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 passengers  
passengers = Browser("Mercury Tours").Page("Find Flights").  
    WebEdit("numpassengers").GetROProperty("value")
```

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アクションまたは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。チェックポイントの戻り値を取得したい場合にも、チェックポイント名を括弧で囲む必要があります。

ヒント：テストまたは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戻すため、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 RunAction("BookFlight", oneIteration)  
または  
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、**Check** メソッドがチェックポイントの値を返すため、メソッドの引数を括弧で囲む必要があります。

```
a = Browser("MyBrowser").Page("MyPage").Check (CheckPoint("MyProperty"))
```

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

エキスパート・ビューまたは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、テストおよび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、879 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。899 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

エキスパート・ビューで [キーワード ビュー] タブを選択すると、更新された情報が QuickTest によってキーワード・ビューに表示されます。新規または更新された VBScript ステートメントに構文エラーがある場合、ステータス・バーの右側に「エラー」というテキストが赤で点滅し、ステータス・バーに、スクリプトの構文エラーの情報について [情報] 表示枠を確認する必要があることを示すエラー・メッセージが表示されます。QuickTest は、すべての構文エラーが修正されるまで、キーワード・ビューにドキュメントを表示できません。

[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。

ヒント:



- ▶ 現在のドキュメントの構文は、**【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】** を選択することで、いつでも確認できます。テストが開いている場合は、すべてのアクションの構文が確認されます。関数ライブラリが開いている場合は、ライブラリ・スクリプトの構文が確認されます。
- ▶ 『**Microsoft VBScript Language Reference**』（英語版）では、VBScript の構文エラーを次のように定義しています。「ある VBScript ステートメントの構造が VBScript スクリプト言語の 1 つ以上の文法規則に違反した結果として生じるエラー」。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 のリファレンスを参照してください（**【ヘルプ】 > 【QuickTest Professional ヘルプ】 > 【VBScript Reference】**）。

[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。

詳細	項目	アクション	行
① Expected end of statement	RegExpression	Action1	2
① Expected statement	RegExpression	Action1	4
① Expected statement	RegExpression	Action1	6
① Expected ']'	RegExpression	Action1	7
① Expected end of statement	RegExpression	Action1	8
① Expected ']'	RegExpression	Action1	9

〔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〕	<p>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。</p> <p>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'</p> <p>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。</p>
〔項目〕	<p>問題のステートメントが含まれているテストまたは関数ライブラリの名前。</p>
〔アクション〕	<p>問題のステートメントが含まれているアクションの名前。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。</p>
〔行〕	<p>構文エラーが含まれている行。行の番号は各アクションまたは関数ライブラリの先頭から数えられます。</p>

〔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。

プログラムの記述の使用

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**プログラムの記述**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。プログラムの記述は、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の記述または **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はできません。こうした状況で、プログラムの記述を使うことで **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。

プログラムの記述には 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の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の記述の直接入力

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 はプログラムの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。正規表現の詳細については、769 ページ「正規表現の概要と使い方」を参照してください。

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の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の記述を使用する必要があります。プログラムの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の記述を使用しているため、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）プログラムの記述を使用しているため、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の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ため、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ていますが、親オブジェクトがプログラムの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 回のテストまたは 1 つの関数ライブラリで同じプログラムの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。

たとえば、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
```

さらに別の方法として、**With** ステートメントを使うこともできます。

```
With Window("Text:=Myfile.txt - Notepad")
  .Move 50, 50
  .WinEdit("AttachedText:=Find what:").Set "hello"
  .WinButton("Caption:=Find next").Click
End With
```

With ステートメントの詳細については、885 ページ「With ステートメント」を参照してください。

プログラムの記述のための 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。正規表現の詳細については、769 ページ「正規表現の概要と使い方」を参照してください。

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。

たとえば、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の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の記述を使用する必要があります。プログラムの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の記述を使用しているため、このステートメントは使用できません。

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）プログラムの記述を使用しているため、このステートメントは使用できません。

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の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ため、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、プログラムの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の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の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の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の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の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。

プログラムの記述での 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。

プログラムの記述によるチェックの実行

プログラムの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。

プログラムの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、プログラムの記述を使って **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、889 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、883 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

QuickTest では、[記録と実行環境設定] ダイアログ・ボックスを使って、テストの実行開始時にアプリケーションを起動するように指定したり、テスト対象アプリケーションを手作業で起動したりできます。また、テスト対象アプリケーションを起動または終了するステートメントをテストに挿入することもできます。

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、**SystemUtil.Run** ステートメントを使います。これはテストに複数のアプリケーションが含まれていて、[記録と実行環境設定] ダイアログ・ボックスで **[開かれている Window ベースのアプリケーションすべてでテストを記録して実行する]** チェック・ボックスを選択している場合に特に便利です。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ができます。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
```

注：

- ▶ 実行するアプリケーションを [記録と実行環境設定] ダイアログ・ボックスを使用して指定した場合、QuickTest はテストに **SystemUtil.Run** ステートメントを追加しません。
 - ▶ **InvokeApplication** メソッドは、実行可能ファイルのみを開くことができ、主に下位互換性を維持するために使用します。
-

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。

コメント、フロー制御、その他の VBScript ステートメントの使用

QuickTest では、テストまた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、テストまたは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。テストおよび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。

キーワード・ビューでこれらのプログラミングの概念を使用する方法の詳細については、第 28 章「プログラミング・ロジックを含むステップの追加」を参照してください。

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。関数ライブラリでテストまたは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、テストおよび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、コメントはエキスパート・ビューおよび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、902 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
/	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となります。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
```

With ステートメント

With ステートメントで、同じ親階層を持つ連続するステートメントをグループ化することによって、スクリプトが短くなり、読み書きと編集がしやすくなります。

注： **With** ステートメントを実行する場合、QuickTest により、最初のステートメントを実行する前にアプリケーション内のオブジェクトが識別されますが、各ステートメントを実行する前に再識別されることはありません。**With** ステートメントによって参照されているオブジェクトが、何らかの方法で **With** ステートメントの実行中にアプリケーション内で更新、再描画、または変更される場合、このことが、テストの実行に影響する場合があります。次のステートメントを実行する前にアプリケーション内のオブジェクトを再識別するように QuickTest に指示するには、**RefreshObject** テスト・オブジェクト操作を呼び出すステートメントを追加します。**RefreshObject** の操作の詳細については、『**HP QuickTest Professional Object Model Reference**』（英語版）を参照してください。

With ステートメントの構文は、次のとおりです。

With object
statements
End With

項目	説明
object	オブジェクトを返すオブジェクトまたは関数
statements	オブジェクトに対して実行する 1 つ以上のステートメント

たとえば、次のようなスクリプトがあったとします。

```
Window("Flight Reservation").WinComboBox("Fly From:").Select "London"
Window("Flight Reservation").WinComboBox("Fly To:").Select "Los Angeles"
Window("Flight Reservation").WinButton("FLIGHT").Click
Window("Flight Reservation").Dialog("Flights Table").WinList("From").
    Select "19097 LON "
Window("Flight Reservation").Dialog("Flights Table").WinButton("OK").Click
```

これは、次のスクリプトで置き換えることができます。

```
With Window("Flight Reservation")
    .WinComboBox("Fly From:").Select "London"
    .WinComboBox("Fly To:").Select "Los Angeles"
    .WinButton("FLIGHT").Click
    With .Dialog("Flights Table")
        .WinList("From").Select "19097 LON "
        .WinButton("OK").Click
    End With 'Dialog("Flights Table")
End With 'Window("Flight Reservation")
```

エキスパート・ビューで **With** ステートメントを入力しても、キーワード・ビューにはまったく影響しません。

注 : **With** ステートメントは手作業で入力することもできますが、記録時に **With** ステートメントを自動生成させたり、既存のテストに基づいて **With** ステートメントを生成させたりもできます。詳細については、809 ページ「テストに対する「With」ステートメントの生成」を参照してください。

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。

テストまたは関数を実行すると、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。テストまたは関数ライブラリ内の **GetTOPProperty**、**GetTOPProperties**、および **SetTOPProperty** 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。

GetTOPProperty および **GetTOPProperties** 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。

SetTOPProperty 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、**SetTOPProperty** 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、836 ページ「エキスパート・ビューまたは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、プログラムの記述の中で「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、890 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、プログラムの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の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
```

プログラムの記述の詳細については、865 ページ「プログラムの記述の使用」を参照してください。

DOS コマンドの実行

QuickTest テストまたは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、テストおよび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、テストまたは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）にテストまたは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の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、テストまた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 30 章

エキスパート・ビューおよび関数ライブラリ・ウィンドウのカスタマイズ

[エキスパートビュー] でのテストの表示方法および、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、[エキスパートビュー] および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

- ▶ [エキスパートビュー] および関数ライブラリ・ウィンドウのカスタマイズについて (898 ページ)
- ▶ エディタの動作のカスタマイズ (899 ページ)
- ▶ エLEMENTの見映えのカスタマイズ (902 ページ)
- ▶ 編集コマンドのカスタマイズ (903 ページ)

[エキスパート ビュー] および関数ライブラリ・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、[エキスパート ビュー] および関数ライブラリ・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、[エキスパート ビュー] および関数ライブラリ・ウィンドウでの、スクリプトや関数ライブラリの表示方法を変更できます。また、スクリプトや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、スクリプトまたは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、339 ページ「テストの印刷」および 918 ページ「関数ライブラリの印刷」を参照してください。

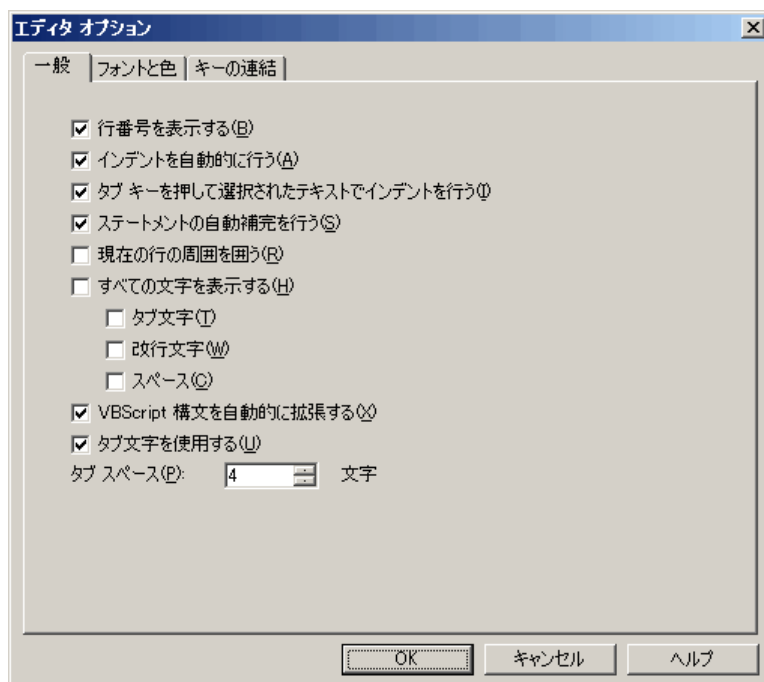
[エキスパート ビュー] の使用方法の詳細については、第 29 章「エキスパート・ビューおよび関数ライブラリ・ウィンドウを使用した作業」を参照してください。関数ライブラリを使った作業の詳細については、第 31 章「ユーザ定義関数および関数ライブラリを使用した作業」を参照してください。

エディタの動作のカスタマイズ

[エキスパート ビュー] および関数ライブラリ・ウィンドウにおける スクリプトや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。[エキスパート ビュー] の使用方法の詳細については、第29章「エキスパート・ビューおよび関数ライブラリ・ウィンドウを使用した作業」を参照してください。関数ライブラリを使った作業の詳細については、第31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 [エキスパート ビュー] または関数ライブラリ・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]	スクリプトまたは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。
[TAB 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、エキスパート・ビューまたは関数ライブラリ、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、テスト・オブジェクト、メソッド、プロパティ、またはコレクションをドロップダウン・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、837 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。

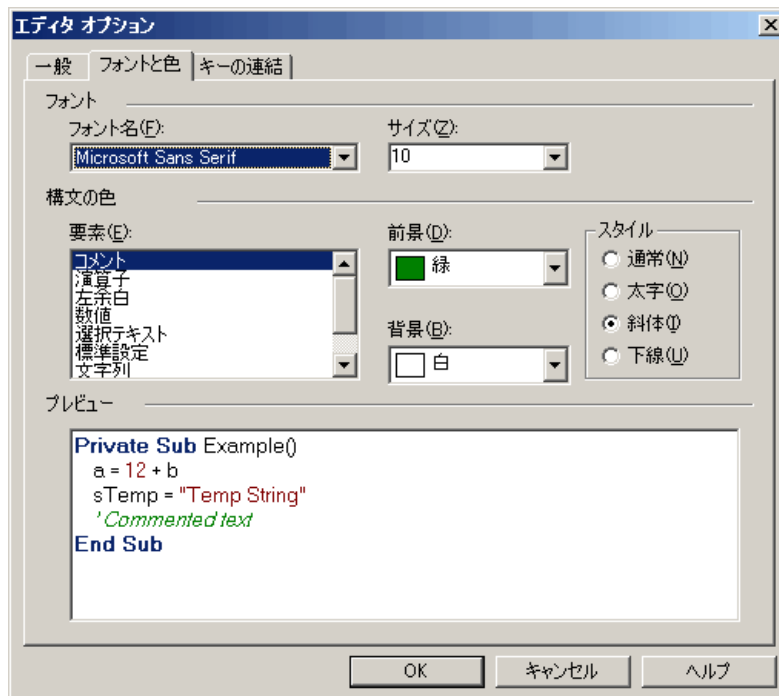
4 [OK] をクリックし、変更を適用してダイアログ・ボックスを閉じます。

エレメントの見映えのカスタマイズ

QuickTest のテストおよび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 のテストおよび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 [エキスパート ビュー] または関数ライブラリ・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、テストまたは関数ライブラリ内のエレメントが、[エキスパート ビュー] または関数ライブラリ・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、テストまたは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。

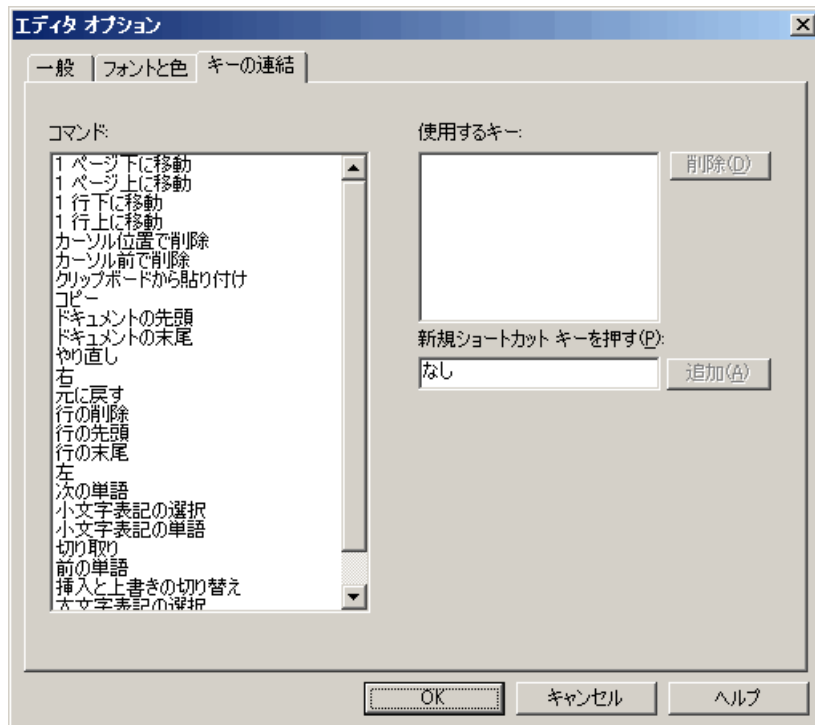
編集コマンドのカスタマイズ

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、47 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 [エキスパート ビュー] または関数ライブラリ・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 31 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、その関数をテストから呼び出せます。

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 (907 ページ)
- ▶ 関数ライブラリの管理 (909 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 (920 ページ)
- ▶ 関数定義ジェネレータの使用法 (924 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 (939 ページ)
- ▶ ユーザ定義関数の使い方のヒント (945 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

テスト内で何度も使用する必要があるコード・セグメントがある場合は、ユーザ定義関数を作成するとよい場合があります。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、テストの簡潔になり、設計、理解、保守が容易になります。QuickTest エンジニアが関連するキーワード（または操作）をアクションに挿入することで、そのアクション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、924 ページ「関数定義ジェネレータの使用法」および 939 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せん。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）または直接テスト内のアクションに格納できます。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、862 ページ「VBScript 構文エラーの処理方法」および 856 ページ「VBScript の基本的な構文の理解」を参照してください。

関数を関数ライブラリに格納し、その関数ライブラリをテストに関連付ければ、その関数ライブラリのパブリック関数をテストで呼び出すことができます。詳細については、920 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、[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、[エキスパートビュー] では手動で入力できません。

関数をテスト・アクションに格納した場合、その関数はそのアクション内からのみ呼び出すことができます。ほかのアクションまたはテストから呼び出すことはできません。これは、特定のアクションの外側から関数を使用できないようにする場合に役に立ちます。

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、924 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、その関数をテストのアクション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テストに関連付けることによって、ほかのユーザでも、テストに直接コードを追加せずに、複雑な操作（テスト・ステップへの if/then ステートメントとループの追加や、ユーティリティ・オブジェクトを使った作業など）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、第 29 章「エキスパート・ビューおよび関数ライブラリ・ウィンドウを使用した作業」を参照してください。

注：QuickTest では、テストを開くと、Quality Center プロジェクトに保存されている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、別のユーザが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を変更した場合、または、外部エディタ（QuickTest ではないもの）を使用してリソースを変更した場合、その変更はテストを閉じて再度開くまでテストに実装されません。これに対して、関数ライブラリなどファイル・システムに保存されている外部リソースに適用した変更は、ただちに実装されます。これらのファイルは直接アクセスされ、テストを開いたときにローカル・コピーとして保存されないためです。

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、897 ページ「エキスパート・ビューおよび関数ライブラリ・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。

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、915 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、916 ページ「関数ライブラリの編集」および 917 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テスト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。
(Quality Center プロジェクトに保存されている関数ライブラリを選択する場合は、関連付けられている関数ライブラリを開くために、QuickTest が該当するプロジェクトに接続されている必要があります)。

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 プロジェクトまたはファイル・システムに関数ライブラリを保存できます。

また、関数ライブラリは、保存だけを目的として、テストへの添付ファイルとして保存することもできます。関数呼び出しをこの関数ライブラリからテストに挿入するには、最初に関数ライブラリをテストに関連付ける必要があります。

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** プロジェクトまたはファイル・システムに関数ライブラリを保存します。

関数ライブラリをテストへの添付ファイルとして **Quality Center** プロジェクトに保存するには、次の手順を実行します。

- 1 上記のステップ 1 および 2 を繰り返し、**Quality Center** プロジェクトに接続されていることを確認します。
- 2 [関数ライブラリを保存] ダイアログ・ボックスのサイドバーで、[**Quality Center テスト計画**] をクリックします。ダイアログ・ボックスのタイトル・バーが [関数ライブラリを添付として保存] に変わります。
- 3 関数ライブラリを添付するテストを参照してダブルクリックします。テストは [対象] パスに最後の項目として表示されます。
- 4 [保存] をクリックします。関数ライブラリはテストへの添付ファイルとして保存されます。

注： 添付された関数ライブラリに呼び出しを挿入するには、テストに関連付ける必要があります。詳細については、922 ページ「テストとの関数ライブラリの関連付け」を参照してください。

開いている QuickTest ドキュメント間でのフォーカスの移動

テストが開いている間、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、同じドキュメント（テストまたは関数ライブラリ）、またはテスト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 **[エキスパートビュー]** または関数ライブラリで、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、アクション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、856 ページ「VBScript の基本的な構文の理解」を参照してください。
 - ▶ テストに関連付けられているすべての関数ライブラリの構文をチェックするには、[テストの設定] ダイアログ・ボックスの [リソース] 表示枠にある [構文チェック] ボタンをクリックします（[ファイル] > [設定] > [リソース]）。詳細については、1263 ページ「テストのためのリソース設定の定義」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です。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、911 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（テス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、関数ライブラリをテストに関連付けて、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、テストを実行し、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、1061 ページ「テス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。

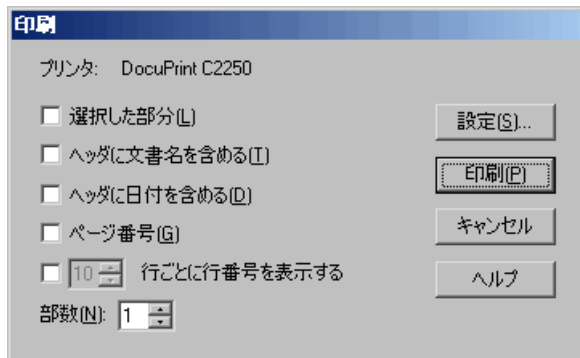
関数ライブラリの印刷

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、そのファイルをテストに関連付けることができます。これにより、ユーザは、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、そのテス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テスト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・ファイルに関数ライブラリとして使えます。

[テストの設定] ダイアログ・ボックス ([**ファイル**] > [**設定**] > [**リソース**] ノード) で、新規のすべてのテストに関連付ける標準の関数ライブラリを指定できます。テストが作成されると、標準関数ライブラリのリストがテストに統合されます。したがって、[テストの設定] ダイアログ・ボックスの標準関数ライブラリ・リストに変更を加えても、既存のテストには影響しません。

既存のテストに関連付けられている関数ライブラリのリストは [リソース] 表示枠または [テストの設定] ダイアログ・ボックスで編集できます。詳細については、1149 ページ「[リソース] 表示枠」、および 1263 ページ「テストのためのリソース設定の定義」を参照してください。

注：

- ▶ 関連関数ライブラリの中にある関数のほかに、任意の関数ライブラリ（または VBScript ファイル）に含まれている関数を任意のアクションから **ExecuteFile** 関数を使って直接呼び出すことができます。関連関数ライブラリに **ExecuteFile** 関数を挿入することもできます。詳細については、947 ページ「テストからの外部定義された関数の実行」を参照してください。
 - ▶ **ExecuteFile** ステートメントを使用して呼び出されるファイルおよび当該ファイルに含まれている関数をデバッグすることはできません。また、**ExecuteFile** ステートメントを含んでいるテストをデバッグする場合、実行マークが正しく表示されないことがあります。
-

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

関数ライブラリがファイル・システムまたは Quality Center プロジェクトに格納されているかどうかに関係なく、テストに関数ライブラリを関連付けることができます。ただし、ビジネス・プロセス・テストで関数ライブラリを使用する計画がある場合は、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。

Quality Center および関連関数ライブラリを使用するときには、[テストの設定] ダイアログ・ボックスの [リソース] 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。

ファイル・システムから Quality Center プロジェクトに既存の関数ライブラリを追加する場合、実際にはそのファイルのコピーがプロジェクトに追加されます。したがって、これらの関数ライブラリのどちらか一方（ファイル・システム内または Quality Center プロジェクト内）を後で変更した場合、もう一方の関数ライブラリは影響を受けません。

テスト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、開いているテスト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、現在開いているテスト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、923 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。

[リソース] 表示枠を使用して関数ライブラリをテストに関連付けるには、次の手順を実行します。

- 1 [リソース] 表示枠で、ツリー内の **[関連付けられた関数ライブラリ]** ノードを右クリックし、**[関数ライブラリの関連付け]** を選択します。**[関数ライブラリを開く]** ダイアログ・ボックスが開きます。
- 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。

関数ライブラリがテストに関連付けられ、ツリー内の **[関連付けられた関数ライブラリ]** ノードにノードとして表示されます。


開いている関数ライブラリをテストに関連付けるには、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テストが 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、テスト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、909 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、またはファイル・システムに保存します。詳細については、912 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 'Test' 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 'Test' に関連付ける]** を選択します。QuickTest によって、開いているテスト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

テストに対する関連付けられている関数ライブラリのリストは、[Test Settings] ダイアログ・ボックスの [リソース] 表示枠、または [リソース] 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。

[テストの設定] ダイアログ・ボックスの [リソース] 表示枠で、テスト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 [テストの設定] ダイアログ・ボックス ([**ファイル**] > [**設定**]) で、ナビゲーション・バーの [**リソース**] ノードをクリックします。
- 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。QuickTest に参照ボタンが表示されます。このボタンを使用して、ファイル・システム内の関数ライブラリを参照できます。Quality Center プロジェクトに接続している場合は、ファイル・パスに **[QualityCenter]** が追加され、Quality Center プロジェクトとファイル・システムのいずれかで関数ライブラリを参照できます。



ヒント : Quality Center プロジェクトからファイルを追加する必要がある、なおかつ Quality Center に接続されていない場合は、SHIFT キーを押したまま **[追加]** ボタンをクリックします。QuickTest によって **[QualityCenter]** が追加され、パスを手動で入力できるようになります。その場合には、**[QualityCenter]** の後にスペースを入れてください。次に例を示します。**[QualityCenter] Subject\Tests** 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。

- 3 テスト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。




[テストの設定] ダイアログ・ボックスの [リソース] 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、1263 ページ「テストのためのリソース設定の定義」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。
- ▶  [テストの設定] ダイアログ・ボックスの [リソース] 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、1263 ページ「テストのためのリソース設定の定義」を参照してください。

関数定義ジェネレータの使用法

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テストに関連付けることができます。また、定義はエキスパート・ビュー内のテスト・スクリプトに直接挿入することもできます。最後に、内容（コード）を追加して関数を完成させます。

注：関数をエキスパート・ビューに直接挿入した場合、テストは特定のアクションのどこからでもその関数にアクセスできるようになります。

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、キーワード・ビューの **[操作]** カラムに、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ステップ・ジェネレータまたはキーワード・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、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、ステップ・ジェネレータの **[ステップについてのコメント]** ボックスと、キーワード・ビューの **[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。定義を関数ライブラリに挿入した場合、関連付けられているテストは関数にアクセスできます。関数を **[エキスパート ビュー]** 内のテストに直接挿入した場合、関数は、特定のアクション内からのみ呼び出すことができます。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。

- 1 926 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。
- 2 928 ページ「関数の定義」の説明に従って、関数を定義します。
- 3 929 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。

- 4 933 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、934 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、936 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。
- 7 936 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。
- 8 937 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またはテスト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、ドキュメント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。

The dialog box '関数定義ジェネレータ' (Function Definition Generator) is shown with the following fields and options:

- 関数定義 (Function Definition):**
 - 名前(N): [Text input field]
 - タイプ(T): [Function] (dropdown menu)
 - 対象(O): [Public] (dropdown menu)
- 引数 (Arguments):**
 - Buttons: +, ×, ↑, ↓
 - Table:

名前	成功モード
- 登録 (Registration):**
 - テストオブジェクトに登録する(R)
 - テストオブジェクト(O): [Text input field] 操作(O): [Text input field]
 - 標準設定操作として登録する(O)
- 追加情報 (Additional Information):**
 - 詳細(D): [Text input field]
 - ドキュメント(M): [Text input field]
- プレビュー (Preview):**

```
Public Function
'TODO: add function body here
End Function
```
- 別の関数定義を挿入する(S)
- Buttons: OK, キャンセル, ヘルプ

関数定義ジェネレータを開いた後、928 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テスト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せん。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、何をやる操作なのかは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]（この関数ライブラリに関連付けられている任意のテスト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] とともに [Private] とともに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、933 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、ステップ・ジェネレータでそのテスト・オブジェクトを選択したときに、およびキーワード・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。また、IntelliSense でも関数を選択できるようになります。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、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、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。

実行時に、QuickTest によって、指定された関数がまずテストの中で検索され、次に [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検索されます。指定したテストまたは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、テストまた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 つのテス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` エンジンアキーマワード・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、テストに関連付けられているテスト・スクリプトまたは関数ライブラリに追加された後にのみ、テス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)。

関数定義	
名前(N):	VerifyProperty
タイプ(T):	Function
対象(O):	Public
<input checked="" type="checkbox"/> テストオブジェクトに登録する(R)	
テストオブジェクト(O):	Link
操作(O):	VerifyProperty
<input type="checkbox"/> 標準設定操作として登録する(O)	

引数:	
名前	成功モード
test_object	リファレンス

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。





引数:	
名前	成功モード
test_object	値
prop_name	値
expected_value	値

たとえば、929 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。

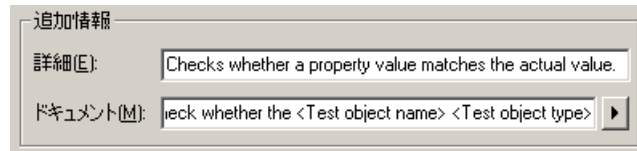
関数への説明の追加

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、ステップ・ジェネレータの **ステップについてのコメント**] ボックスと、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 に **[注釈]** カラムだけが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、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.

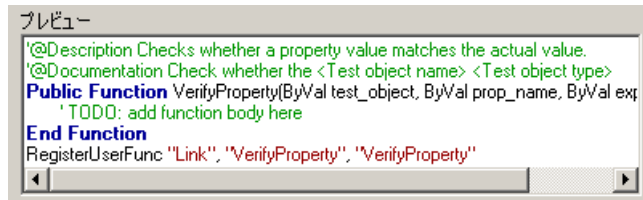
- 2 **[ドキュメント]** ボックスに、ステップ・ジェネレータの **[ステップについてのコメント]** ボックスまたはキーワード・ビューの **[注釈]** カラムに表示させるテキストを入力します。**[注釈]** のテキストには引数を含めることができます。それには、 をクリックして使用する引数を選択します。

[**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。



```
プレビュー
'@Description Checks whether a property value matches the actual value.
'@Documentation Check whether the <Test object name> <Test object type>
Public Function VerifyProperty(ByVal test_object, ByVal prop_name, ByVal exp
'TODO: add function body here
End Function
RegisterUserFunc "Link", "VerifyProperty", "VerifyProperty"
```

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。

- 2 928 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、関数ライブラリに関連付けられているテストは、関数にアクセスできます。関数をテスト（[エキスパート ビュー] 内）に直接挿入した場合、テストには、特定のアクション内の任意の場所からの関数呼び出しを含めることができます。

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、関数ライブラリまたはテスト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。

注：アクション内で関数を直接定義すると、その関数はそのアクションの中でのみ呼び出せます。

- 2 関数ライブラリまたはテストの中で、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、必要に応じてテストのステータスを変更したりできます。

- 3 コードを関数ライブラリに挿入した場合は、その関数ライブラリをテストに関連付けて、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、テストがその関数にアクセスできると、QuickTest エンジニアがその関数を表示し、使用できることを確認する必要があります。詳細については、920 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

929 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ステップ・ジェネレータ、またはキーワード・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、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、まずテストまたは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テストまたは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

ユーザ定義関数の使用範囲をローカル・アクションに限定するには、関数をテストに直接書き込みます。ユーザ定義関数を多数のアクションやテストで使えるようにするには、関数を関連関数ライブラリに格納します（推奨）。ローカルのアクションと関連関数ライブラリの中に同じ名前の関数が存在する場合、QuickTest ではアクション内で定義されている関数が使用されます。

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ができます。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、テストから呼び出されるたびに、**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、テストの実行中、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。

注：メソッドを登録する（そしてアクションが終っても登録を解除しない）外部アクションを呼び出した場合、登録したメソッドは、そのアクションを呼び出したテストでそれ以降も使用できます。

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。 注 ：QuickTest の予約済みオブジェクト（たとえば DataTable , Environment , Reporter など）のメソッドとしてユーザ定義メソッドを登録することはできません。
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	テストから呼び出すユーザ定義関数の名前。この関数はテストまたは任意の関連付けられている関数ライブラリに置けません。
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・ビューまたはステップ・ジェネレータでテスト・オブジェクトを選択すると、 [操作] カラム（キーワード・ビュー）または [操作] ボックス（ステップ・ジェネレータ）に標準設定のメソッドが自動的に表示されます。

ヒント：登録する関数が関数ライブラリ内で定義されている場合は、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テス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、そのメソッドはテストの終わりまで、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。

メソッドの登録解除は、再利用可能なアクションに `QuickTest` のメソッドをオーバーライドする登録メソッドが含まれている場合に特に重要です。たとえば、呼び出し先のアクションの中で直接定義されている関数を使うメソッドの

登録を解除しない場合、登録したメソッドが後のアクションで呼び出されると、呼び出し元のテストが失敗します。これは、関数の定義を見つけることができないことが原因です。

この場合、登録した関数が関数ライブラリ内で定義されていれば、呼び出し元のテストは正常に動作できます（関数ライブラリが呼び出し元のテストに関連付けられている場合）。しかし、呼び出し元のテストの作者は、呼び出し先のアクションに登録された関数が含まれていることに気付かずに、後のアクションで QuickTest の通常の動作を想定して、登録されたメソッドを使う可能性があります。そして、予期しない結果が生じることも考えられます。

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。
- ▶ 登録する関数が関数ライブラリ内で定義されている場合は、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テストでメソッドをすぐに利用できるようになります。
- ▶ QuickTest は各実行セッションの開始時にすべての登録メソッドを消去します。
- ▶ [ステップから実行] や [ステップからデバッグ] などの部分的な実行またはデバッグ・オプションを使用して、(関数ライブラリの中でなく) テスト・ステップ内のメソッド登録の後の位置からテストの実行を開始した場合、QuickTest はメソッド登録を認識しません。これは、登録が現在の実行セッションの開始よりも前に行われるためです。
- ▶ テス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、ステートメントをテス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。**Option Explicit** ステートメントは、制限なしにアクション・スクリプトの中で直接使用することができます。
- ▶ 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、テス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テスト・ステップは  アイコンで示されます。その後、その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。
- ▶ テス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、テストを再度開くまで反映されません。
- ▶ 同じ名前を持つ 2 つ以上の関数がテスト・スクリプトまたは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 (QuickTest は、関数ライブラリを検索する前に、テスト・スクリプト内で関数を検索します)。混乱を避けるために、1 つのテス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的な名前を付けてください。
- ▶ 再利用可能なアクションの中でメソッドを登録した場合には、アクションの終わりにそのメソッドの登録を解除することを (そして必要があれば次のアクションの初めに再登録することを) 強くお勧めします。そうすることで、そのアクションを呼び出すテストがメソッドの登録による影響を受けないようにします。
- ▶ 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])。

テストからの外部定義された関数の実行

関数ライブラリ (VBScript ファイル) をテストに関連付けず、なおかつテスト内のアクションまたはほかの関数ライブラリから関数、サブルーチン、クラスなどを呼び出せるようにするには、アクションに **ExecuteFile** ステートメントを挿入します。

テストを実行すると、**ExecuteFile** ステートメントは関数ライブラリ内のすべてのグローバル・コードを実行して、そのファイル内のすべての定義がアクションのスクリプトのグローバル・スコープから利用できるようにします。

注： **ExecuteFile** ステートメントを使用して呼び出されるファイルおよび当該ファイルに含まれている関数をデバッグすることはできません。また、**ExecuteFile** ステートメントを含んでいるテストをデバッグする場合、実行マークが正しく表示されないことがあります。

ヒント： 作成するすべてのアクションに同じ **ExecuteFile** ステートメントを含めなければ、ステートメントをアクション・テンプレートに追加します。詳細については、466 ページ「アクション・テンプレートの作成」を参照してください。

外部定義関数を実行するには、次の手順を実行します。

- 1 標準の VBScript 構文を使って VBScript ファイルを作成します。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]）を参照してください。
- 2 作成したファイルを、テストを実行するコンピュータからアクセスできる任意のフォルダに保存します。

- 3 次の構文を使用して、テストのアクションに **ExecuteFile** ステートメントを追加します。

ExecuteFile FileName

ここで **FileName** は、作成した VBScript ファイルの絶対パスまたは相対パスです。

- 4 必要に応じて、アクション内で、指定した VBScript ファイルにある関数やサブルーチンを使用します。

注：

- ▶ **ExecuteFile** ステートメントでは、VBScript の **ExecuteGlobal** ステートメントが利用されます。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。
 - ▶ **ExecuteFile** ステートメントをアクション内で実行するとき、ファイル内の関数は現在のアクションからのみ呼び出せます。テスト全体で VBScript ファイル内の関数を利用できるようにするには、[テストの設定] ダイアログ・ボックスの [リソース] 表示枠の関連関数ライブラリのリストにファイル名を追加します。詳細については、920 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。
-

第 VI 部

テストの実行と分析

第 32 章

テストの実行

テストの作成後は、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

- ▶ テストの実行について (952 ページ)
- ▶ テスト全体の実行 (953 ページ)
- ▶ テストの一部の実行 (954 ページ)
- ▶ [実行] ダイアログ・ボックス : [結果の場所] タブ (958 ページ)
- ▶ [実行] ダイアログ・ボックス : [入力パラメータ] タブ (960 ページ)

テストの実行について

テストを実行すると、それに含まれるステップが QuickTest によって実行されます。テス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 33 章「実行セッション結果の表示」を参照してください。

テストにグローバルなデータ・テーブル・パラメータが含まれている場合は、データ・テーブルのデータ行ごとに 1 回ずつテストが実行されます。テストに、現在のアクションのデータ・シートに対応するデータ・テーブル・パラメータが格納されている場合は、そのアクション・データ・シートのデータ行ごとに 1 回ずつ、そのアクションが実行されます。また、テスト全体または特定のアクションについて、最初の反復だけを実行するのか、すべての反復を実行するのかを指定することもできます。あるいは、データ・セットの指定した範囲について反復を実行するように指定することも可能です。テスト反復の詳細については、第 45 章「個別のテストのオプション設定」を参照してください。データ・テーブル・パラメータの詳細については、第 15 章「アクションを使った作業」を参照してください。

テス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。特定のステップを「オプション」に指定すれば、そのステップが成功しなかった場合に、実行が停止するのではなくステップがバイパスされるようにすることができます。テストを更新して、テスト・オブジェクト記述、チェックポイントの期待値、ActiveScreen の画像や値を変更できます。

動的な要素を持つオブジェクトについてテス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。

一連のテストを設定して順次実行するには、QuickTest Test Batch Runner を使用します。詳細については、963 ページ「テスト・バッチの実行」を参照してください。

WinRunner ユーザの方へ： QuickTest テストの実行中に、WinRunner テストを実行したり、WinRunner のコンパイル済みモジュールから関数を呼び出したりできます。詳細については、第 57 章「WinRunner を使用した作業」を参照してください。

テスト全体の実行

QuickTest では、テス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。ステップまたはアクションを指定して、そこから（またはそこまで）テストを実行するには、**[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、テストを最初からまたは最後まで実行せずに、テス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、954 ページ「テストの一部の実行」を参照してください。

テストを開始するには、**[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、定義されたテストのパラメータの値を入力します。

テストを実行するには、次の手順を実行します。

- 1 テストをまだ開いていない場合は、**[ファイル]** > **[開く]** > **[テスト]** を選択します。

ヒント：最近使用したテストを開く場合は、**[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、958 ページ「**[実行]** ダイアログ・ボックス：**[結果の場所]** タブ」および 960 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 33 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 44 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、1243 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、別のテストを開いたり、新規のテストを作成したりするなど）、実行セッションは中断します。

テストの一部の実行

テス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、テス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、テストをテス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、1068 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。

エキスパート・ビューでは、**[ステップから実行]** オプションを使用して、選択したステップからアクションの最後までテストを実行できます。このモードで **[ステップから実行]** を使用すると、すべての反復が無視されます。ただし、アクションにほかのアクションがネストされている場合、ネストされているアクションが指定の回数だけ反復されます。

[キーワードビュー] では、**[ステップから実行]** オプションを使用して、選択したステップからテストの最後までテストを実行できます（選択されているステップが再利用可能なアクションの一部でない場合。テストが継続場所を認識するために、再利用可能なアクションをテストから呼び出す必要があるため）。このモードで **[ステップから実行]** を使用すると、すべての反復が実行されます。最初の反復は、選択したステップからテストの最後まで実行され、それ以降のすべての反復は、テストの最初から実行されます。

テスト内の単一のアクションを実行するには、**[現在のアクションの実行]** オプションを使用します。**[現在のアクションの実行]** を使用すると、すべての反復が無視されます。ただし、アクションにほかのアクションがネストされている場合、ネストされているアクションが指定の回数だけ反復されます。

ヒント：

- ▶ テストの反復を 1 回だけ実行するには、**[テストの設定]** ダイアログ・ボックスの **[実行]** 表示枠で、**[反復なしで実行する]** を選択します。
 - ▶ テスト内の特定のポイントまでテストを実行するには、ブレークポイントを挿入します。その場合、テストは選択したステップまたはアクションからブレークポイントまで実行されます。ブレークポイントの詳細については、1071 ページ「**ブレークポイントの設定**」を参照してください。
-

アクションの詳細については、第 15 章「アクションを使った作業」を参照してください。

選択したステップからアクション全体、あるいはテストまたはアクションを実行するには、次の手順を実行します。

- 1 アプリケーションが、実行するアクションまたは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、テストの実行を開始するアクションまたはステップを選択します。
 - ▶ [テストフロー] 表示枠でアクションを選択します。
 - ▶ キーワード・ビューで、ステップまたはアクションの行を選択して強調表示します。
 - ▶ エキスパート・ビューで特定の行にカーソルを置きます。

選択したステップまたはアクションが、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] または [現在のアクションの実行] を選択するか、右クリックして [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。
- 4 958 ページ「[実行] ダイアログ・ボックス : [結果の場所] タブ」および 960 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。

注：アクションの範囲内でテストの一部を実行する場合は、[実行] ダイアログ・ボックスの [入力パラメータ] タブに、テスト・パラメータではなくアクションのパラメータを指定する必要があります。詳細については、477 ページ「アクション・パラメータの設定」を参照してください。

- 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 33 章「実行セッション結果の表示」を参照してください。

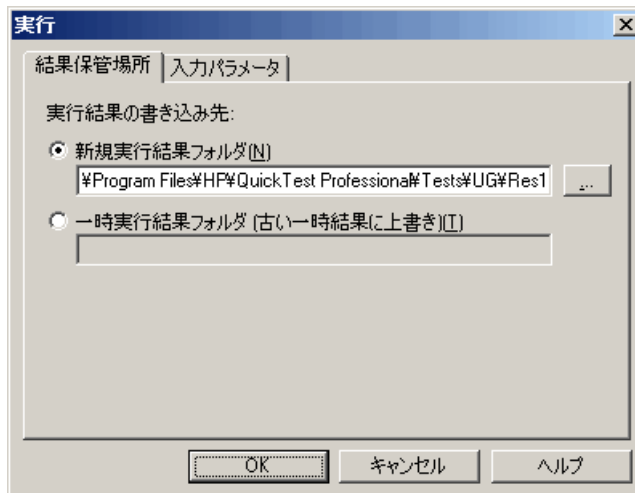
[テスト結果] サマリには、[**ステップから実行**] または [**現在のアクションの実行**] オプションを使用してテストが実行されたことを示す注記が表示されません。

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 44 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	<p>概念の概要 : 951 ページ「テストの実行」</p> <p>主要なタスク :</p> <ul style="list-style-type: none"> ▶ 953 ページ「テスト全体の実行」 ▶ 954 ページ「テストの一部の実行」 ▶ 1094 ページ「メンテナンス実行ウィザードを使ったテストの実行」 ▶ 1068 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。



注：Quality Center プロジェクトからテストを実行する場合，[**新規実行結果フォルダ**] オプションは表示されず，代わりに [プロジェクト名]，[実行名]，[テストセット] および [インスタンス] の各オプションが表示されます。詳細については，1419 ページ「Quality Center プロジェクトに格納されているテストの QuickTest からの実行」を参照してください。

【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せん。標準設定では，QuickTest テストの結果はテスト・フォルダに格納されません。

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 のテス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。

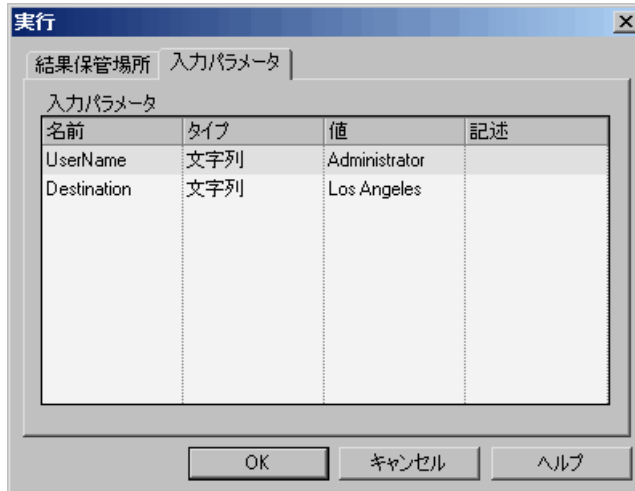
- ▶ [**一時実行結果フォルダ**]：一時フォルダに実行結果を保存します。このオプションによって，このフォルダに保存されている結果はすべて上書きされません。

注：QuickTest では，すべてのテス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	<p>概念の概要 : 951 ページ「テストの実行」</p> <p>主要なタスク :</p> <ul style="list-style-type: none"> ▶ 953 ページ「テスト全体の実行」 ▶ 954 ページ「テストの一部の実行」 ▶ 1094 ページ「メンテナンス実行ウィザードを使ったテストの実行」 ▶ 1068 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」 <p>その他の関連項目 : 961 ページ「その他の参照情報」</p>

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、テス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 [テストの設定] ダイアログ・ボックスの標準設定の値が使用されます。

注：アクションの範囲内でテストの一部を実行する場合は（[オートメーション] > [ステップから実行] オプションまたは [オートメーション] > [現在のアクションの実行] オプションを使用）、[実行] ダイアログ・ボックスの [入力パラメータ] タブに、テスト・パラメータではなくアクションのパラメータを指定する必要があります。

その他の参照情報

<p>関連する概念</p>	<ul style="list-style-type: none"> ▶ テストのパラメータの設定の詳細については、716 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、第 24 章「値のパラメータ化」を参照してください。
----------------------	---

オプション・ステップの使用

オプション・ステップは、実行セッションが正常に終了するために必ずしも必要ではないステップです。たとえば、テスト対象アプリケーションによってログイン・ウィンドウにユーザ名とパスワードを入力するよう求められるために、テストの作成中にログイン・ステップを追加するとします。この特定のアプリケーションはユーザ・ログインの詳細を記憶しているため、アプリケーションを起動するたびにログインする必要はありません。実行セッション中には、ユーザ名とパスワードの入力は要求されません。これは、以前に入力した情報がアプリケーションに保持されているためです。この場合、ログイン情報を入力するために追加したステップは必要ではないので、オプション・ステップとして指定する必要があります。


実行セッション中、オプション・ステップのオブジェクトがアプリケーションに存在しない場合、QuickTestはこのステップを無視し、テストの実行を続行します。実行セッションが完了すると、実行されなかったステップに関するメッセージが表示されますが、このステップが原因で実行が失敗することはありません。

ただし、実行セッション中、QuickTestがオブジェクト・リポジトリ内でオプション・ステップのオブジェクトを見つけられない場合（たとえば、オブジェクト名がテスト内で変更されてもオブジェクト・リポジトリ内では変更されていない場合や、オブジェクトがオブジェクト・リポジトリから削除された場合）、必要なオブジェクトを一覧表示したエラー・メッセージが表示され、実行は失敗します。

記録セッション中、QuickTestは、特定のダイアログ・ボックスを開くステップにオプションとして自動的に印を付けます（特定のダイアログ・ボックスの一覧は、963 ページ「標準設定のオプション・ステップ」を参照してください）。

また、ステップをオプションとして手動で指定できます。たとえば、条件ステートメントを追加したり回復シナリオを使用したりして自動的にボタンをクリックするか、ENTER キーを押すか、またはステップにログイン情報を入力できます。詳細については、801 ページ「条件ステートメントの使用」および 1317 ページ「回復シナリオの定義と使用」を参照してください。

オプション・ステップの設定

キーワード・ビューでオプション・ステップを設定するには、右クリックして[オプションステップ]を選択します。[オプションステップ]アイコンが、選択したステップの横に追加されます。

Welcome: Mercury Tours			
	userName	Set	"Nicole" "userName" edit box に "Nicole" を入力する。
	password	SetSecure	"47ce40d2145e..." 暗号化された文字列 "47ce40d2145eb49088fa85c9c214"
	Sign-In	Click	"Sign-In" image をクリックする。
	Find a Flight: Mercury	Check	Checkpoint("Fin..." "Find a Flight: Mercury" Web page が Web コンテンツアクト

オプション・ステップをエキスパート・ビューで追加するには、VBScript ステートメントの先頭に OptionalStep を追加します。次に例を示します。

```
OptionalStep.Browser("Browser").Dialog("AutoComplete").WinButton("Yes").Click
```

エキスパート・ビューでの作業の詳細については、第 29 章「エキスパート・ビューおよび関数ライブラリ・ウィンドウを使用した作業」を参照してください。

標準設定のオプション・ステップ

標準設定では、次のダイアログ・ボックスまたはメッセージ・ボックスがオプション・ステップとみなされます。

ダイアログ・ボックス / メッセージ・ボックスのタイトル・バー
[AutoComplete]
[ファイルのダウンロード]
[Internet Explorer]
[Netscape]
[ネットワーク パスワードの入力]
[エラー]
[セキュリティの警告]
[セキュリティ情報]
[セキュリティ警告]
Username and Password Required

テスト・バッチの実行

複数のテストを連続して実行するには、**Test Batch Runner** を使用します。各テストの結果は、それぞれの標準の場所に保存されます。

Test Batch Runner を使用して、同じ一連のテストを後で簡単に実行できるように、テストのリストを作成して、**.mtb** ファイルとして保存しておくことができます。また、バッチを実行する際に、テストをバッチ・リストに追加したり、バッチ・リストから除外したりできます。

注：

- ▶ Test Batch Runner でテストを実行できるようにするには、[オプション] ダイアログ・ボックスの [実行] 表示枠で、**[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・ボックスを選択する必要があります。詳細については、第 44 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。
- ▶ Test Batch Runner は、ファイル・システムに含まれるテストだけを使用できます。バッチ実行で Quality Center に保存されたテストを含める場合は、まずファイル・システムにテストを保存しなければなりません。
- ▶ **[停止]** ボタンをクリックすると、いつでもバッチ実行を終了できます。

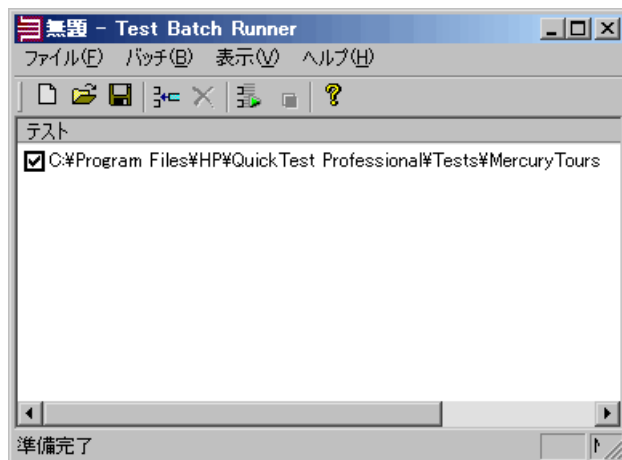


テスト・バッチを設定して実行するには、次の手順を実行します。

- 1 **[スタート]** メニューで **[プログラム]** > **[QuickTest Professional]** > **[Tools]** > **[Test Batch Runner]** を選択します。[Test Batch Runner] ダイアログ・ボックスが開きます。
- 2 **[追加]** ボタンをクリックするか、または **[バッチ]** > **[追加]** を選択します。**[テストを開く]** ダイアログ・ボックスが開きます。



- 3 テスト・バッチ・リストに追加するテストを選択し、**[開く]** をクリックします。選択したテストがバッチ・リストに追加されます。



- 4 リストに追加するそれぞれのテストについて、手順 3 を繰り返します。標準設定では、選択した各テストはリストの最後に追加されます。

リストの最後以外の位置にテストを挿入するには、テストを追加する位置の 1 つ下のテストを選択します。テストを追加すると、選択したテストの上にテストが挿入されます。



テストをリストから削除するには、**[削除]** ボタンをクリックするか、**[バッチ]** > **[削除]** を選択します。

テストをリストに追加した後で、次回のバッチ実行中にそのテストを実行しないようにするには、テスト名の横にあるチェック・ボックスをクリアします。



- 5 バッチ・リストを保存する場合は、**[保存]** ボタンをクリックするか、**[ファイル]** > **[上書き保存]** を選択し、リストの名前を入力します。ファイルの拡張子は **.mtb** です。



- 6 テスト・バッチを実行する準備ができたなら、**[実行]** ボタンをクリックするか、**[バッチ]** > **[実行]** を選択します。QuickTest をまだ開いていない場合には自動的に開き、テストが順次実行されます。バッチの実行が完了したら、各テストの結果が保存される標準のフォルダ (**<テスト・フォルダ> \res#\report**) でテスト結果を参照できます。

テスト結果の詳細については、第 33 章「実行セッション結果の表示」を参照してください。

第 33 章

実行セッション結果の表示

テス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。

本章の内容

- ▶ 実行セッション結果の表示について (967 ページ)
- ▶ [テスト結果] ウィンドウ (969 ページ)
- ▶ 実行セッションの結果の表示 (978 ページ)
- ▶ 実行結果の削除 (1000 ページ)
- ▶ 実行セッション中に検出された不具合の送信 (1009 ページ)
- ▶ Quality Center プロジェクトへの自動による不具合の送信 (1011 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。データ・テーブル・パラメータを含まないテストの場合は、[テスト結果] ウィンドウには単一のテスト反復が表示されます。

テストにデータ・テーブル・パラメータが含まれていて、テストの設定が複数の反復を実行するように設定されている場合、[テスト結果] ウィンドウには、テスト実行の反復ごとの詳細が表示されます。結果はテストのアクション別にグループ化されます。

注： [テストの設定] ダイアログ・ボックスの [実行] 表示枠の中で、テストを 1 つまたはすべての反復で実行するように設定します。詳細については、1259 ページ「テストのための実行設定の定義」を参照してください。

テス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
- ▶ テス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テスト全体のアプリケーションの状態のムービー・クリップ
- ▶ テストの各段階で、成功または失敗した各ステップとチェックポイントの詳細な説明
- ▶ テストのために監視したシステム・カウンタ
- ▶ テストのための Quality Center 情報（テストが Quality Center から実行された場合、または Quality Center に保存されているテストが QuickTest から実行され、結果を 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、1243 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せん。[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、テストのサマリ、ステップの詳細、アプリケーションの静止画像、アプリケーションのムービー、またはシステム・カウンタの結果が含まれます。

[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。

次の例では、3つの反復を含むテストの結果を示します。



前の例に示すテストには、実行結果ツリーに示されているように3つの反復が含まれます。テストの結果は、テストのアクションごとにまとめられています。

[テスト結果] ウィンドウには、次の主要要素があります。






- ▶ **テスト結果タイトル・バー**：テス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**：テスト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、975 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**：実行結果ツリーにテスト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、971 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、972 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、989 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：テストのために有効化されたシステム・カウンタの結果を示す線グラフが表示されます。[システム モニタ] 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、1054 ページ「[システム モニタ] の結果表示」を参照してください。
- ▶ **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、977 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**が表示されません。実行結果ツリーには、テスト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。テストにチェックポイントが含まれていない場合、アイコンは表示されないことに注意してください。
- ▶  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（ルートのアクションまたはテストまで）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、アクションまたはテス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。
- ▶  失敗したために無視された省略可能なステップを示します。これによってテストが失敗することはありません。

- ▶  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。
- ▶  **実行時データ・テーブル**を表示します。これは、データ・テーブル・パラメータまたはテストの実行中に取得されたデータ・テーブルの出力値を含むテストを実行するとき使用する値を示すテーブルです。
- ▶  **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テス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 36 章「テストの保守」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、テス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] タブにテス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] タブにはテストの名前、結果名、実行セッションの開始日時と終了日時、反復の回数、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。

- ▶ 反復にチェックポイントが含まれている場合、起こりうる結果は **[成功]** または **[失敗]** です。反復にチェックポイントが含まれていない場合、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。テストに 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
- ▶ テストが**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、テス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、テス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、1091 ページ「テストの保守」を参照してください。







- ▶ テストが、Quality Center から実行された場合、または Quality Center に保存されているテストが QuickTest から実行され、ユーザが結果を Quality Center に保存することを選択した場合、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。

注：テスト・セットとは、特定のテスト目標を達成するために選択されたテストで構成されたグループです。たとえば、アプリケーションのユーザ・インタフェースや、ストレス下でのアプリケーションのパフォーマンスをテストするテスト・セットを作成できます。テスト・セットの定義は、Quality Center のテスト実行モードで行います。詳細については、Quality Center のドキュメントを参照してください。

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、979 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、996 ページ「テスト結果の印刷」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、985 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、1399 ページ「Quality Center との接続と切断」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、1009 ページ「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。
	[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、987 ページ「結果ステップの検索」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
	[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 (987 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 (987 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、982 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、982 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テストのステップを強調表示します。この機能は、アクション、反復、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、984 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、27 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、1243 ページ「テストの実行オプションの設定」を参照)。

また、現在のテストの以前の実行結果や、ほかのテストの結果を表示することもできます。テスト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。

- ▶ 979 ページ「特定の実行を表示するためのテスト結果の表示」
- ▶ 982 ページ「実行結果ツリーの操作」
- ▶ 983 ページ「結果の詳細の表示」
- ▶ 984 ページ「QuickTest 内でのステップへの移動」
- ▶ 985 ページ「テスト結果のフィルタリング」
- ▶ 987 ページ「結果ステップの検索」
- ▶ 988 ページ「Quality Center からのテスト実行の結果の表示」
- ▶ 989 ページ「アプリケーションの静止画像とムービーの表示」
- ▶ 994 ページ「テスト結果のプレビュー」
- ▶ 996 ページ「テスト結果の印刷」
- ▶ 997 ページ「テスト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

現在のテストに関して保存されている結果を表示できます。また、ほかのテストに関して保存されている結果も表示できます。結果はファイル・システムまたは Quality Center で検索できます。

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。



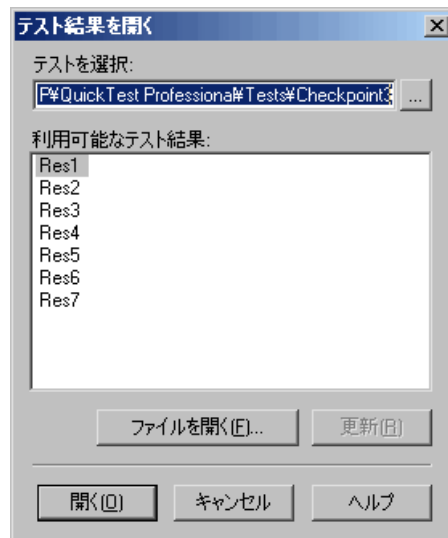
QuickTest ウィンドウで **[結果]** ボタンをクリックするか、**[オートメーション]** > **[結果]** を選択します。

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、または現在のテス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。



現在のテス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：

- ▶ 指定したテス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。
-

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、ファイル・システムに保存されている QuickTest テストの結果はテスト・フォルダに格納されます。テストを実行するとき、[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、958 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。

QuickTest テストが Quality Center に保存されている場合、結果は Quality Center のテスト・フォルダに保存されます。実行セッションの結果の保存場所は変更できません。

テストごと、または結果ファイルごとに結果を検索できます。

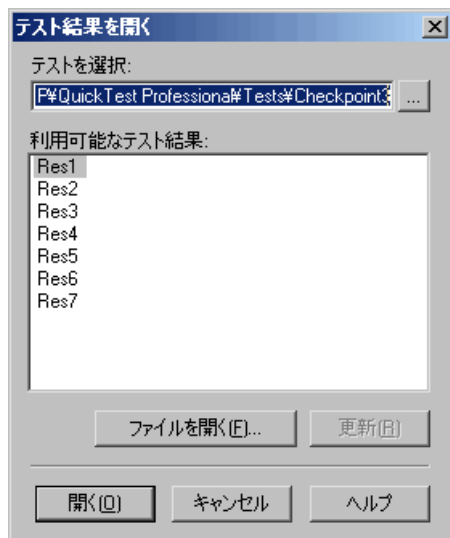
テストごとに結果を検索するには、次の手順を実行します。



- 1 (任意) テスト結果が、Quality Center に保存されている場合、[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、テス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、結果を表示するテストの場所を選択します。テス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、表示するテス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。

Quality Center での作業の詳細については、第 51 章「Quality Center との統合」を参照してください。

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 テス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。

注：

- ▶ 標準設定では、テストの結果ファイルは<テスト> ¥ <結果名> ¥Report に保存されます。
 - ▶ QuickTest Professional バージョン 6.5 およびそれ以前のバージョンの結果ファイルは、.qtp という拡張子付きで保存されます。標準設定では、[結果ファイルの選択] ダイアログ・ボックスでは拡張子が .xml の結果ファイルのみが表示されます。[結果ファイルの選択] ダイアログ・ボックスで拡張子が .qtp の結果ファイルが表示されるようにするには、[ファイルの種類] ボックスで「**Test Results (*.qtp)**」を選択します。
-

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、987 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

個々の反復、アクション、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、およびテスト実行中に監視したシステム・カウンタの結果を表示できる [システム モニタ] タブも表示されます。詳細については、989 ページ「アプリケーションの静止画像とムービーの表示」、1054 ページ「[システム モニタ] の結果表示」、および 1243 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、実行が成功した反復、アクション、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反復、アクション、およびステップは、[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。

- ▶ 成功しなかったが、テス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。

注：[警告] と表示されているステップを含むテスト、反復、またはアクションは、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。

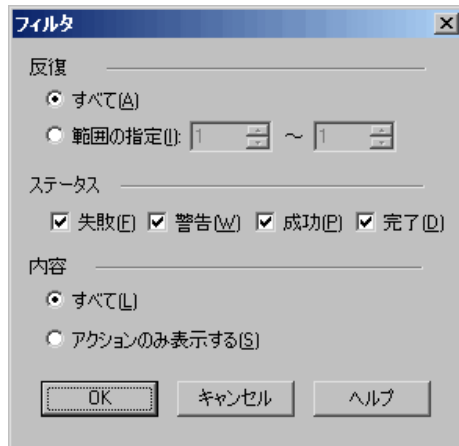
ステップに移動するには、次の条件を満たす必要があります。

- ▶ 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テストに対して開かれている。
- ▶ テストが、[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、アクション、反復実行時データ・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。

- ▶ ステップが **LoadAndRunAction** ステートメントを使用して実行されたアクションの一部ではない。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。
- ▶ テストが実行セッションの前に保存された。
- ▶ テストが [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、1243 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。



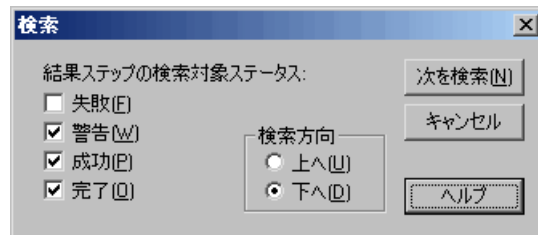
次のオプションを使用できます。

オプション	説明
反復	<p>(このオプションはテストでのみ使用できます。)</p> <ul style="list-style-type: none"> ▶ [すべて] : すべての反復のテスト結果を表示します。 ▶ [範囲の指定 X ~ Y] : 指定した範囲のテスト反復のテスト結果を表示します。
ステータス	<ul style="list-style-type: none"> ▶ [失敗] : 失敗したステップのテスト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) に関するテスト結果を表示します。 ▶ [成功] : 失敗したステップのテスト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) に関するテスト実行結果を表示します。
内容	<p>(このオプションはテストでのみ使用できます。)</p> <ul style="list-style-type: none"> ▶ [すべて] : テストのすべてのノードからすべてのステップを表示します。 ▶ [アクションのみ表示する] : テスト内のアクション・ノードを表示します (アクション・ノードの特定のステップではありません)。

注：エキスパート・ビューで Reporter.Filter ステートメントを使用し、選択したステップの保存を有効または無効にしたり、「**失敗**」と「**警告**」ステータスのステップだけを保存したりできます。実行セッション情報の保存の詳細については、894 ページ「実行セッション中に報告するステップの選択」または『**HP QuickTest Professional Object Model Reference**』（英語版）を参照してください。Reporter.Filter ステートメントは、上記で説明した [フィルタ] ダイアログ・ボックスとは異なります。Reporter.Filter ステートメントは [テスト結果] に保存するステップを指定するのに対し、[フィルタ] ダイアログ・ボックスは、表示するステップを決定するものです。

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 からのテスト実行の結果の表示

Quality Center から QuickTest テストが含まれるテスト・セットを実行すると、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テストを実行します。すべての実行結果は、実行されたテストの標準設定の場所に保存されます。

QuickTest のテスト実行の結果は、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。

テストが、Quality Center から実行された場合、または Quality Center に保存されているテストが QuickTest から実行され、ユーザが結果を Quality Center に保存することを選択した場合、実行結果には、969 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]** : テストが実行された Quality Center サーバの名前を指定します。
- ▶ **[プロジェクト名]** : テストが実行された 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]** : テスト・セットの場所を指定します。
- ▶ **[テストのインスタンス]** : テスト・セッ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、テスト・セットに同じテストが 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。

フライトwoParam 結果サマリ

テスト: フライトwoParam

結果名: Run_2-4_17-18-13

タイムゾーン: 東京 (標準時)

サーバ名: http://cinderella:8080/qcbin

プロジェクト名: DEFAULTバージョン

実行開始: 2009/02/04 - 17:15:18

実行終了: 2009/02/04 - 17:15:41

テストのバージョン: 5

テスト セット: Root¥QT¥QTPテストセット

テストのインスタンス: 1

Quality Center に保存されているテストは、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

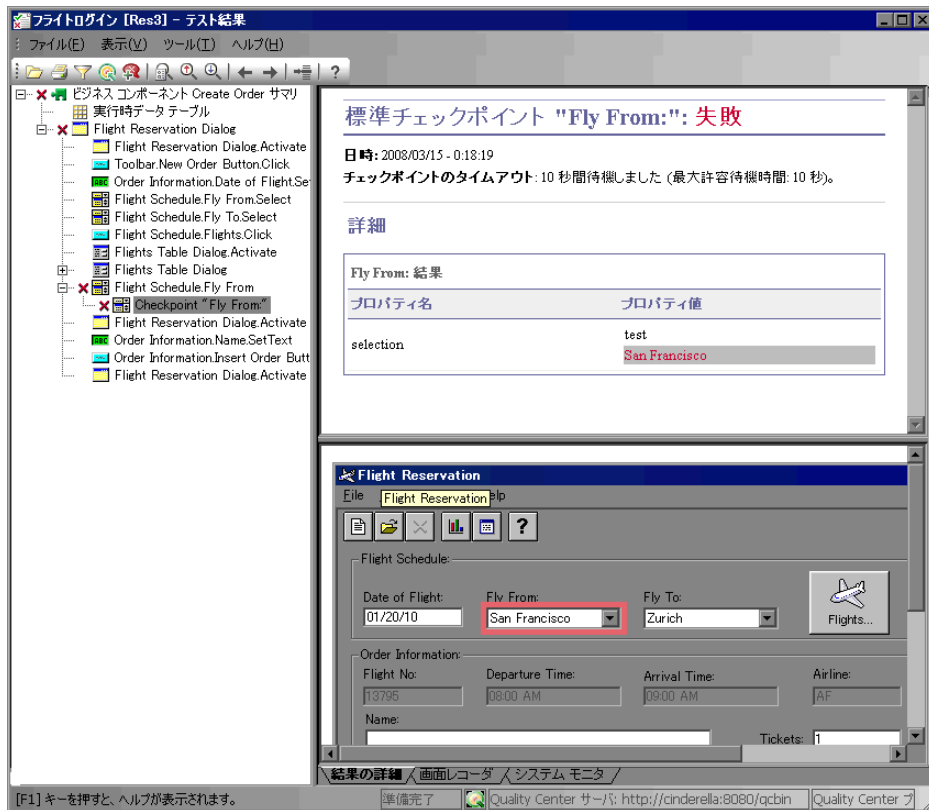
QuickTest Professional では、実行セッション中にアプリケーションの静止画像とムービーをキャプチャできます。キャプチャされたファイルは、[テスト結果] ウィンドウで表示できます。右側の表示枠の [結果の詳細] および [画面レコーダ] タブでは、静止画像とテキストの詳細、またはアプリケーションのムービーを表示できます。

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 Professional Object 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、1243 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。



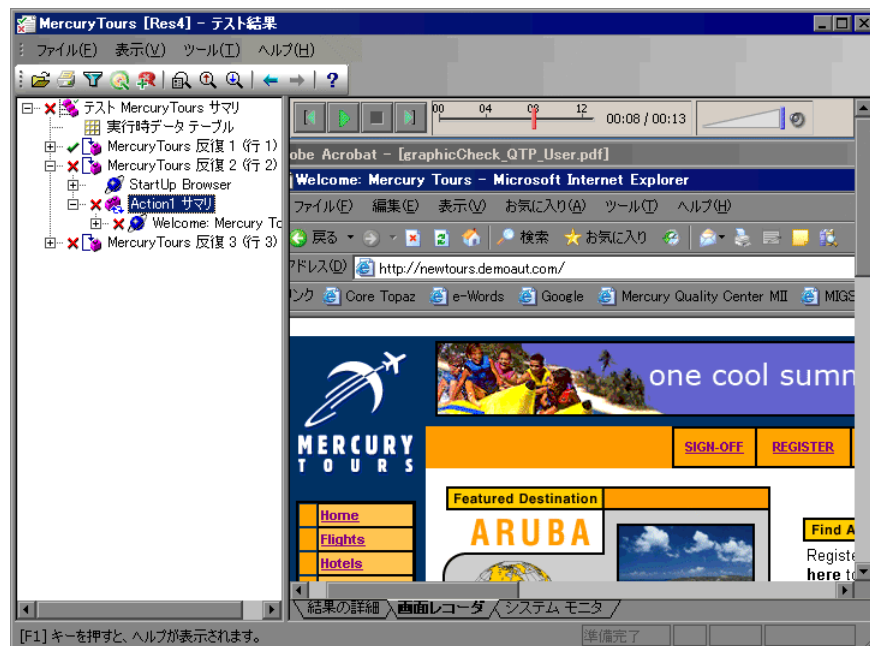
強調表示したステップにエラーがない場合、右側の表示枠には結果詳細が表示され、画面キャプチャは表示されません。

[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、1246 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。

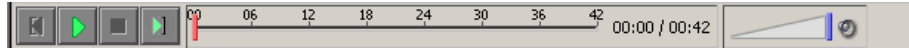
実行セッションのムービーの表示

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せん。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、テス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、1243 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、1132 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

テスト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、テスト結果ファイルのサイズが小さくなります。テスト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 (993 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 <テスト名> [<実行結果名>] と命名され、テスト結果フォルダに保存されません。
- 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 (993 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。

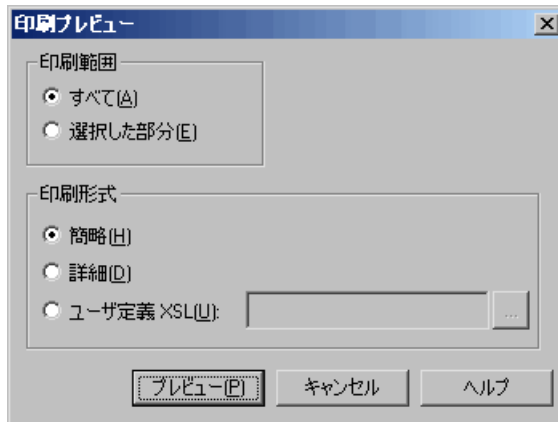
テスト結果のプレビュー

テスト結果は、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 以降で作成したテスト結果にのみ使用できます。

テスト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] : テスト全体のテスト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テスト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、1015 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、画面にテスト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。

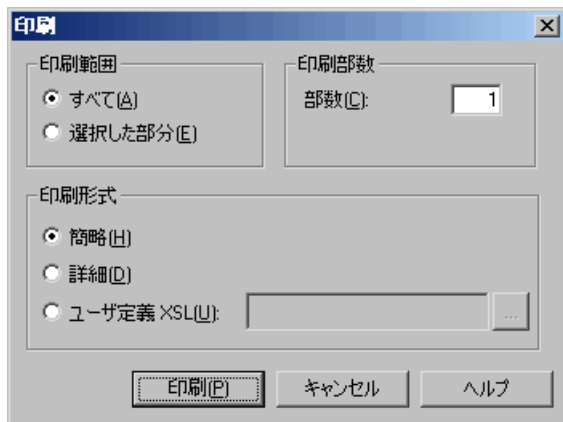
テスト結果の印刷

[テスト結果] ウィンドウから、テスト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。

テスト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]** : テス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テスト結果情報を印刷します。
- 3 **[印刷部数]** で印刷するテスト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、1015 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 以降で作成したテスト結果にのみ使用できます。

- 5 **[印刷]** をクリックして、選択したテスト結果情報を標準の Windows プリンタに出力します。

テスト結果のエクスポート

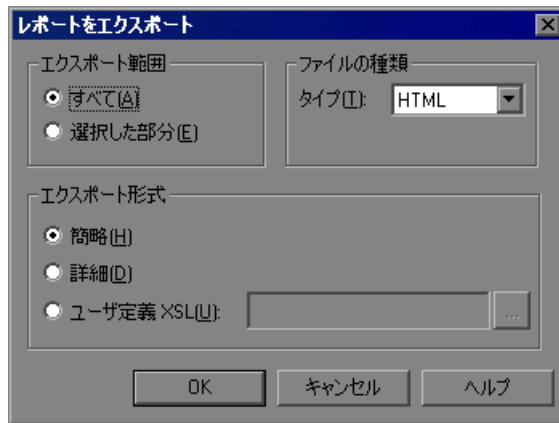
テスト結果の詳細を HTML、PDF、または DOC ファイルにエクスポートできます。これにより、QuickTest 環境を使用できなくても、テスト結果を表示できます。たとえば、テスト結果を収めたファイルを、QuickTest をインストールしていない第三者に電子メールで送信できます。エクスポートするレポートの形式を選択できます。また、ユーザ定義のレポートの作成やエクスポートも行えます。テスト結果をエクスポートする場合、**[結果の詳細]** タブの情報がレポートに含まれます。**[画面レコーダ]** または **[システム モニタ]** の結果をエクスポートするには、これらのタブで特定のエクスポート・オプションを使用します。詳細については、991 ページ「実行セッションのムービーの表示」および 1054 ページ「**[システム モニタ]** の結果表示」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。

テスト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] : テス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テスト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、1015 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 以降で作成したテスト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<テスト名> [**<実行結果名>**] という名前が付けられ、テスト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テスト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。

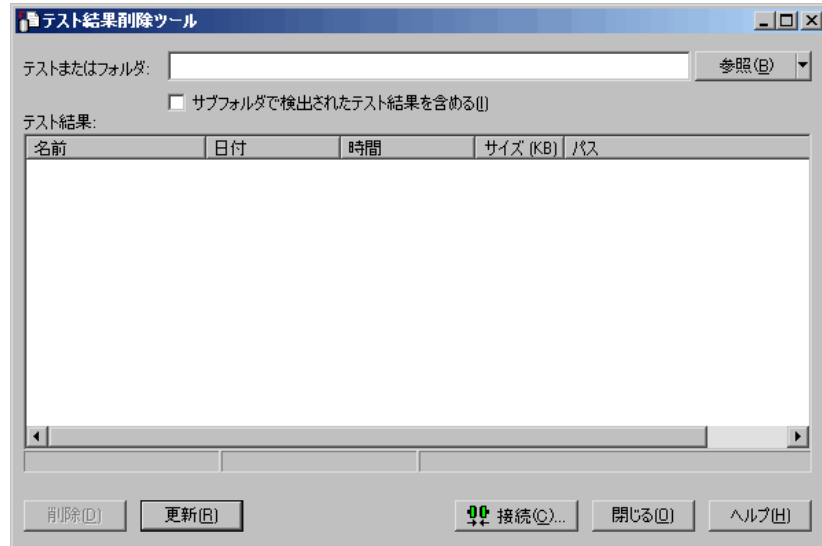
テスト結果削除ツールを使った結果の削除

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。

削除対象の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テスト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、テスト結果を削除するパスを指定します。ファイル・システムで作業する場合は、テストまたはフォルダを指定できます。Quality Center 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。

テストまたはフォルダを参照するには、[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、削除するテスト結果の場所を選択します。削除するフォルダまたは特定のテスト結果を参照して選択し、[開く] をクリックします。

注：テスト結果を 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、1399 ページ「Quality Center との接続と切断」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テスト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。

指定したテストまたはフォルダのテスト結果が、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び替えられます。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 削除するテスト結果を選択します。Windows の標準的な選択操作の方法を使用して、削除するテスト結果を複数選択できます。
- 5 **[削除]** をクリックします。選択したテスト結果が、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なります。

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テス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 のテス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、769 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 のテス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 < テス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テストまたはテスト・パスを設定します。テスト名、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ると、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で使用します。

次の例では、2005 年 11 月 1 日より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る必要があります。

このオプションは必ず **-Silent** オプションと組み合わせで使用します。

実行セッション中に検出された不具合の送信

テストで失敗した各ステップの不具合を自動的に Quality Center プロジェクトに送信するよう、QuickTest に指示できます。また、特定のステップの不具合を QuickTest の [テスト結果] ウィンドウから Quality Center に直接手作業で送信できます。これらのオプションは、Quality Center プロジェクトに接続している場合にのみ使用できます。

Quality Center および QuickTest の使用方法の詳細については、第 51 章「Quality Center との統合」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。

Quality Center プロジェクトへの手動による不具合の送信

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。

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、第 51 章「Quality Center との接続と切断」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。テストおよびチェックポイント（存在する場合）に関する基本情報は詳細領域に含まれています。

Operating system : Windows 2000
Test path : C:\Program Files\Mercury Interactive\QuickTest Professional\Tests\tutorial\Recording on PREDATOR The CheckPoint 'Flight Details' Failed

- 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。

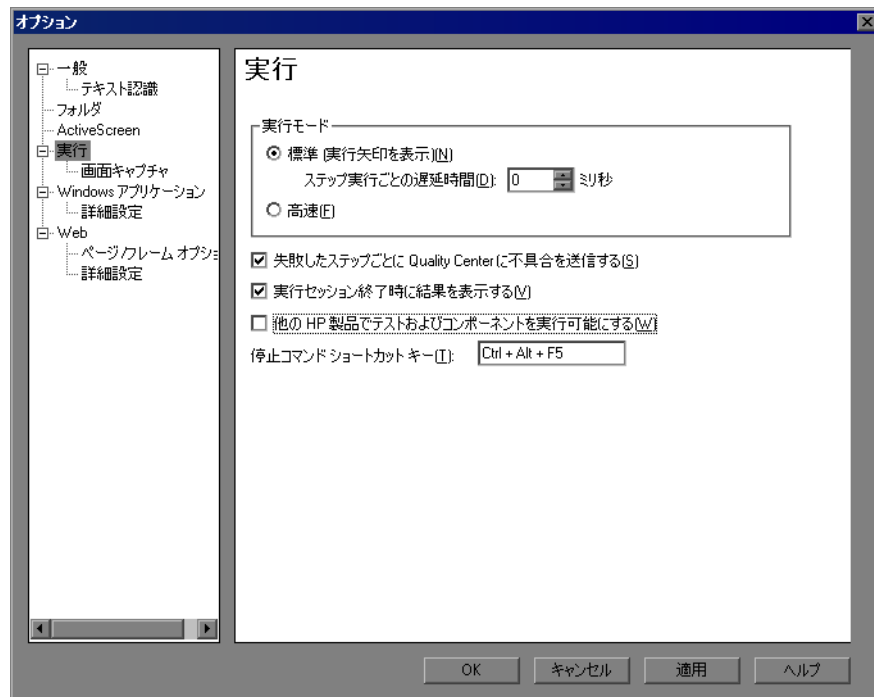
Quality Center プロジェクトへの自動による不具合の送信

テストで失敗した各ステップの不具合を自動的に [Quality Center への接続] ダイアログ・ボックス ([ファイル] > [ツール] > [Quality Center への接続] で表示されます) で指定した Quality Center プロジェクトへ送信するよう、QuickTest に指示できます。不具合を Quality Center プロジェクトに自動的に送信できるのは、テスト結果が Quality Center に保存されている場合にに限られます。

不具合が自動的に Quality Center に送信されるようにするには、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] 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。
- 2 [実行] ノードをクリックします。



- 3 [失敗したステップごとに Quality Center に不具合を送信する] チェック・ボックスを選択します。
- 4 [OK] をクリックして [オプション] ダイアログ・ボックスを閉じます。

次に、不具合ごとに Quality Center に送信される情報の例を示します。

```
This defect was added automatically by QuickTest Professional

Standard Checkpoint "Flight Details_4" failed

Test name: Recording
Test location: C:\Program Files\Mercury Interactive\QuickTest Professional
\Tests\Tutorial\Recording on BINDER
Action name: Action1

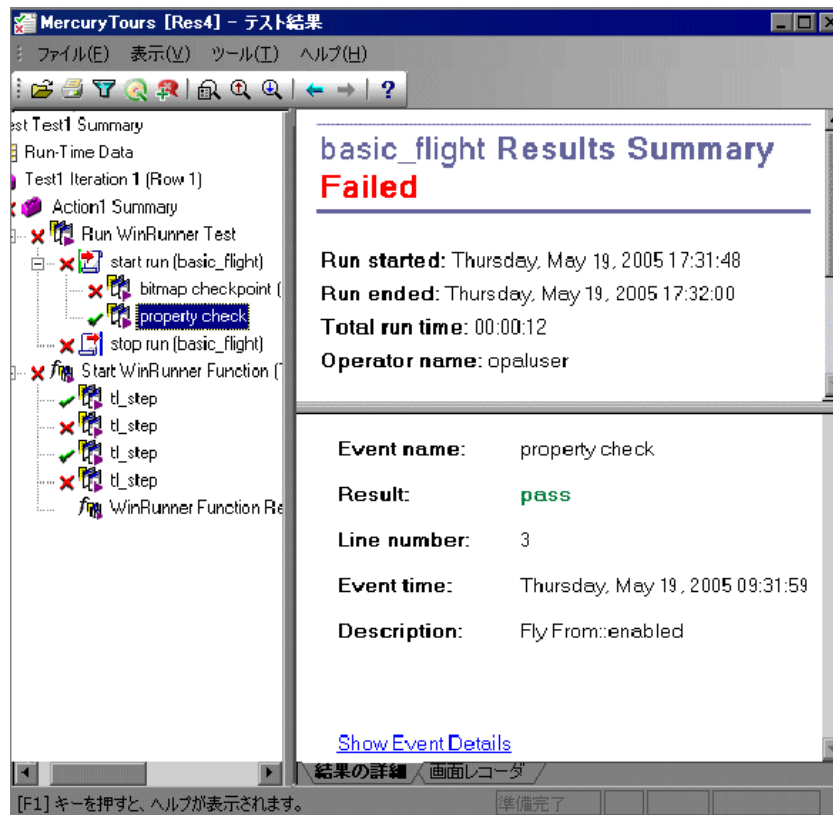
Operating system : Windows 2000
Host: BINDER




Additional Information:
Verification type: String Content.
Settings: Exact match - ON; Ignore space - ON; Match case - OFF.
Results: Checked 28 cells;
Succeeded: 27;
Failed: 1
```


テスト結果での WinRunner テスト・ステップの表示

QuickTest テストに WinRunner テストへの呼び出しが含まれている場合は、QuickTest の [テスト結果] ウィンドウ内で WinRunner ステップの詳細結果を表示できます。

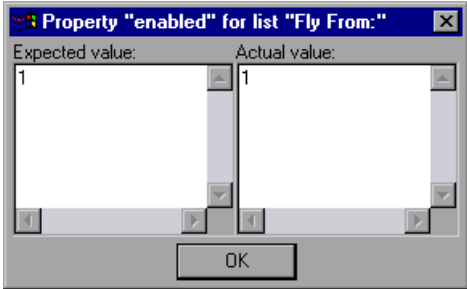
QuickTest テスト結果の左側の表示枠には、通常 WinRunner の結果に含まれている WinRunner イベントごとのノードが表示されます。WinRunner テストのイベントまたは関数呼び出しに対応するノードを選択すると、右側の表示枠に呼び出された WinRunner テストまたは関数のサマリーと、選択したイベントの詳細が表示されます。



WinRunner テストの開始と終了は、結果ツリーにテスト実行  アイコン付きで表示されます。WinRunner イベントは、WinRunner  アイコンで表示されます。WinRunner 関数の呼び出しは、 アイコンで表示されます。

WinRunner テストのステップを選択すると、右上の表示枠に WinRunner テストの結果のサマリが表示されます。サマリには、テストの開始と終了時間、総実行時間、オペレータ名、テスト中に実行されたチェックポイントの結果のサマリが含まれています。

右下の表示枠には次の情報が表示されます。

オプション	説明
[イベント名]	選択したステップの名前です。
[結果]	ステップのステータス（成功または失敗）です。
[行番号]	WinRunner テスト内のステップの行番号です。
[イベント時間]	イベントが実行された時間です。
[詳細]	<p>選択したステップに関する追加情報が表示されます。その後にはそのステップの WinRunner の詳細へのリンクが表示されます。</p> <p>たとえば、プッシュ・ボタンの enabled プロパティを検査する GUI チェックポイントへのリンクをクリックすると、次に示すような [WinRunner] ダイアログ・ボックスが表示されます。</p>  <p>注： 選択したステップの WinRunner による詳細情報を表示するには、コンピュータに WinRunner をインストールする必要があります。</p>

QuickTest から WinRunner テストおよび関数を実行する方法の詳細については、第 57 章「WinRunner を使用した作業」を参照してください。

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示する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、977 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、ある要素タグにはアクション名が、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、アクション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、996 ページ「テスト結果の印刷」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、997 ページ「テスト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 34 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。

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（1019 ページ）
- ▶ チェックポイントの結果の表示（1022 ページ）
- ▶ パラメータ化された値と出力値の結果の表示（1046 ページ）

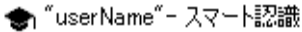
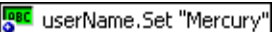
テスト結果に含まれるスマート認識情報の分析

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

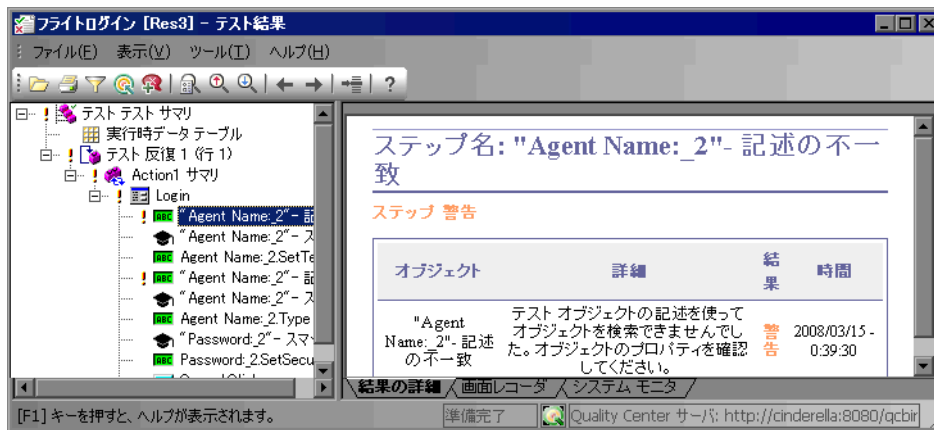
QuickTest において、学習した記述に一致するオブジェクトが見つからず、スマート認識によってオブジェクトが見つかった場合、テスト結果に警告ステータスおよび次の情報が表示されます。

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。  "userName"- 記述の不一致	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。

結果ツリー	結果の詳細
<p>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。</p> 	<p>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で、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なります。</p>
<p>実際に実行されたステップ。例を次に示します。</p> 	<p>実行されたステップの詳細を示す通常の結果。</p>

スマート認識メカニズムの詳細については、第 4 章「オブジェクト認識の設定」を参照してください。

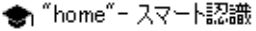
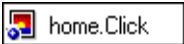
次の図に、学習した記述プロパティ値の 1 つが変更された後で、スマート認識を使って AgentName WebEdit オブジェクトを認識したテストの結果を示します。



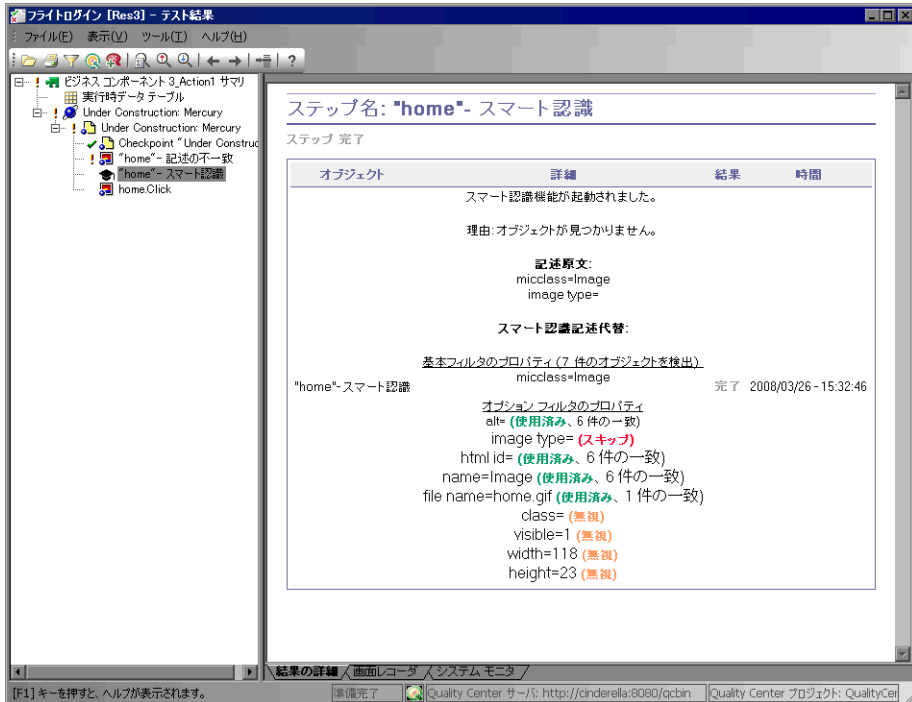
スマート認識 – 学習した記述に複数のオブジェクトが一致する場合

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
<p>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。</p> 	<p>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で、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なります。</p>
<p>実際に実行されたステップ。例を次に示します。</p> 	<p>実行されたステップの詳細を示す通常の結果。</p>

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テス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、テストは失敗し、[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

テス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、テーブルなどの期待値を、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。

テストを実行すると、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、テストは失敗します。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] ウィンドウでテストのテスト結果を表示します。詳細については、978 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、1246 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。

- ▶ 1024 ページ「標準チェックポイントの結果の分析」
- ▶ 1025 ページ「テーブル・チェックポイントとデータベース・チェックポイントの結果の分析」
- ▶ 1027 ページ「ビットマップ・チェックポイント結果の分析」
- ▶ 1029 ページ「テキスト・チェックポイントまたはテキスト領域チェックポイントの結果の分析」
- ▶ 1031 ページ「XML チェックポイント結果の分析」
- ▶ 1042 ページ「アクセシビリティ・チェックポイント結果の分析」

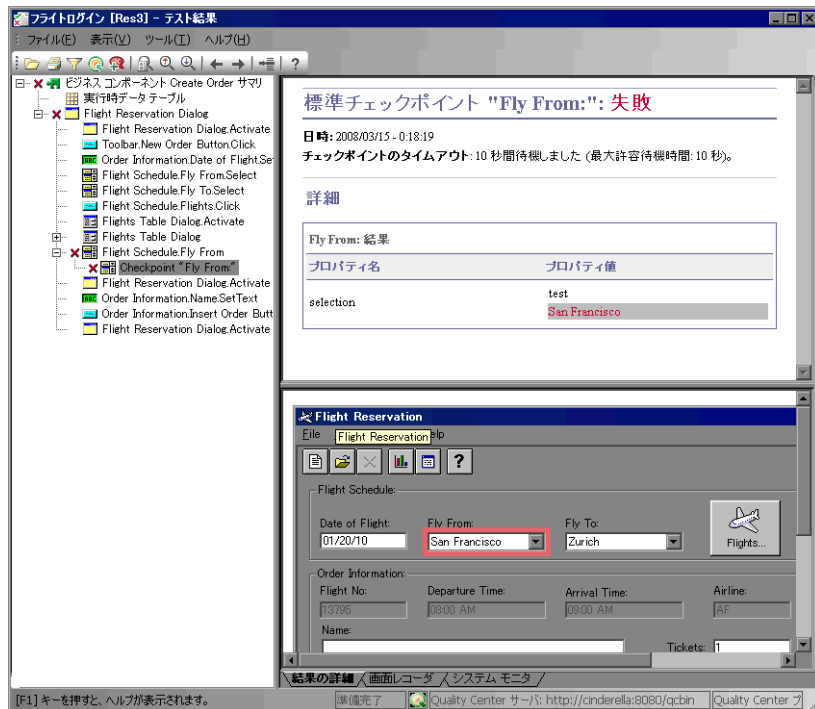
- 3 [ファイル] > [終了] を選択して、[テスト結果] ウィンドウを閉じます。

チェックポイントの詳細については、第17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テストに追加すると、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、511 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、1022 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。フライトの出発地の期待値は「**test**」ですが、実際の値は「**San Francisco**」です。

テーブル・チェックポイントとデータベース・チェックポイントの結果の分析

テストにテーブル・チェックポイントを追加することで、アプリケーションのテーブル内のセルに、指定したテキスト文字列が表示されているかどうか検査できます。テストにデータベース・チェックポイントを追加して、アプリケーションからアクセスするデータベースの内容を検査できます。

テーブル・チェックポイントとデータベース・チェックポイントの結果の表示は似ています。テストを実行すると、QuickTestにより、チェックポイントの期待結果と、実行セッションでの実際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。

テーブル・チェックポイントおよびデータベース・チェックポイントの詳細については、第 20 章「テーブルの検査」および第 22 章「データベースの検査」を参照してください。

テーブル・チェックポイントまたはデータベース・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、1022 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。

標準チェックポイント "DEPART": 失敗

日時: 2009/02/04-17:37:33
 チェックポイントのタイムアウト: 10 秒間待機しました (最大許容待機時間: 10 秒)。

詳細

検証タイプ: 文字列コンテンツ。設定: 完全に一致する単語のみ - ON; スペースを無視 - ON; 大文字と小文字の区別 - OFF。結果: 確認されたセルの数 40; 成功: 39; 失敗: 1

	1	2	3	4
1	DEPAR			
2	SELECT	✗ FLIGHT	DEPAR	STOPS
3		Blue Sk	5:03	non-sto
4	Price: \$			
5		Blue Sk	7:10	non-sto
6	Price: \$			
7		Pangaea	9:17	non-sto
8	Price: \$			
9		Unified	11:24	non-sto
10	Price: \$			

結果の詳細 / 画面レコーダ / システム モニタ

[F1] キーを押すと、ヘルプが表示されます。 準備完了

「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、チェックポイントに指定した検証設定、チェックポイントで成功/失敗した個々のテーブル・セルまたはデータベース・レコードの数が表示されます。

チェックポイントに失敗すると、「結果の詳細」タブの下の表示枠に、チェックポイントによって検査されたテーブル・セルまたはデータベース・レコードが表示されます。検査されたセルの値やレコードは黒で、検査されなかったセルの値やレコードは灰色で表示されます。チェックポイントが失敗となったセルやレコードは、失敗 ✗ アイコンで示されます。



下の表示枠にある **[次の不一致]** ボタンをクリックして、チェックポイントが失敗となった次のテーブル・セルまたはデータベース・レコードを強調表示できます。



下の表示枠にある **[値の比較]** ボタンをクリックして、選択したテーブル・セルまたはデータベース・レコードの期待値と実際の値を表示できます。

ビットマップ・チェックポイント結果の分析

テストにビットマップ・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。テストを実行する場合、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 19 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、1022 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、1246 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、524 ページ「ビットマップ比較の微調整」を参照してください。

テキスト・チェックポイントまたはテキスト領域チェックポイントの結果の分析

テストにテキスト・チェックポイントまたはテキスト領域チェックポイントを追加することで、テキスト文字列がアプリケーション内の適切な場所に表示されていることを検査します。テストを実行すると、QuickTest により、チェックポイントの期待結果と、実行セッションでの実際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。テキストおよびテキスト領域チェックポイントの詳細については、第 21 章「テキストの検査」を参照してください。

テキスト・チェックポイントまたはテキスト領域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、1022 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。

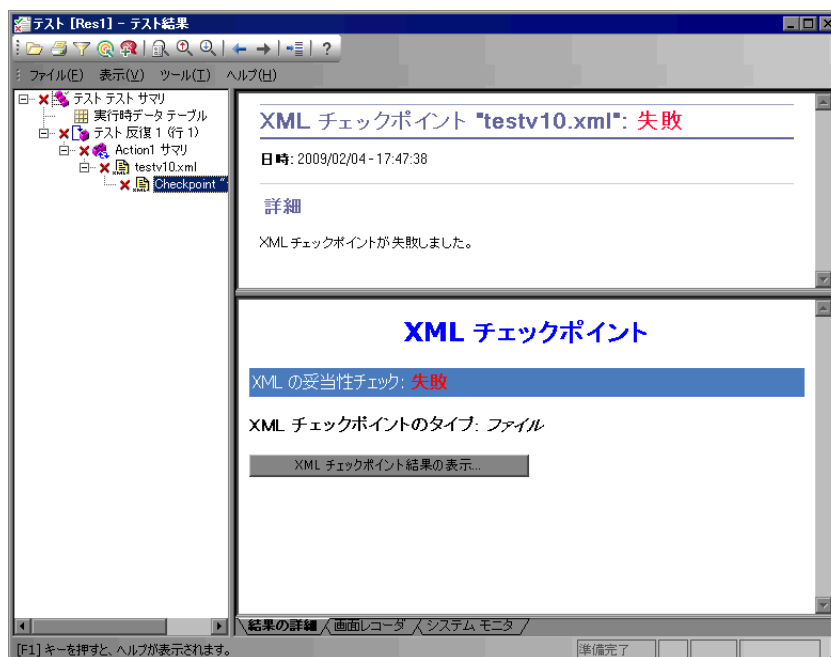


[結果の詳細] タブの上の表示枠にはチェックポイント・ステップの結果が表示されます。ここにはチェックポイントのステータス（成功または失敗）、チェックポイントが実行された日時、使用されたチェックポイント・タイムアウトの間隔（必要に応じて）などが表示されます。また、検査された期待テキストと実際のテキスト、およびチェックポイントに指定した検証設定が表示されます。

XML チェックポイント結果の分析

XML チェックポイントをテストに追加することで、XML ドキュメントまたはファイル内のデータおよび構造に予定外の変更が生じていないかどうかを検証できます。テストを実行すると、QuickTest により、チェックポイントの期待結果と、実行セッションでの実際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。XML チェックポイントの詳細については、第 23 章「XML の検査」を参照してください。

XML チェックポイントの結果の要約は [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、1022 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。

[結果の詳細] タブの下の表示枠には、スキーマ検証 (適切な場合) の詳細とチェックポイント結果のサマリが表示されます。スキーマ検証が失敗した場合は、失敗の理由も表示されます。

チェックポイントが失敗した場合は、[結果の詳細] タブの下の表示枠で **[XML チェックポイント結果の表示]** をクリックして、チェックポイントで実行された各検査の詳細を表示できます。[XML チェックポイント結果] ウィンドウが開き、そのチェックポイントの失敗の詳細が表示されます。

注：標準設定では、チェックポイントが成功した場合、**[XML チェックポイント結果の表示]** ボタンは利用できません。詳細な結果を利用できるかどうかは、[オプション] ダイアログ・ボックスの [実行] > [画面キャプチャ] 表示枠の **[静止画像キャプチャをテスト結果へ保存]** 設定によって異なります。詳細については、1246 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。

[XML チェックポイント結果] ウィンドウについて

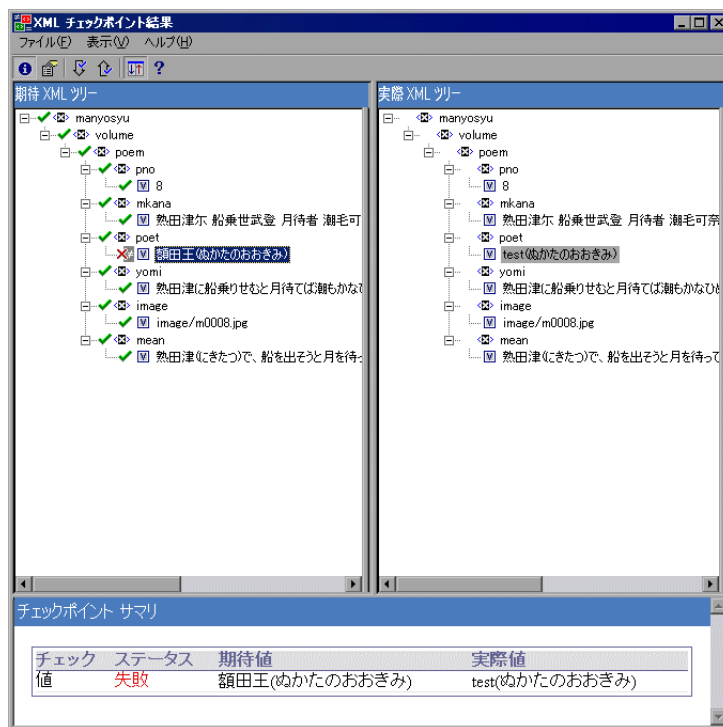
[テスト結果] ウィンドウの **[XML チェックポイント結果の表示]** ボタンをクリックすると、[XML チェックポイント結果] ウィンドウに XML ファイルの階層構造が表示されます。

期待 XML ツリー表示枠には、期待結果として XML チェックポイントに格納されている要素、属性、および値が表示されます。

実際 XML ツリー表示枠には、実際の結果が（実行セッション中に XML ドキュメントが実際に表示されたとおりに）表示されます。

[チェックポイント サマリ] 表示枠には、期待結果の表示枠で選択されている項目に対して実行された検査の結果情報が表示されます。

[XML チェックポイント結果] ウィンドウを開くと、[チェックポイントサマ
リ] 表示枠には期待結果の表示枠で最初に検査が行われた項目についての結果
のサマリが表示されます。



[XML チェックポイント結果] ウィンドウの操作

[XML チェックポイント結果] ウィンドウには、メニューとツールバーが表示され、これらを使用して XML チェックポイントの結果に関するさまざまなコンポーネントを操作できます。

次に示すコマンドまたはツールバー・ボタンを使用して、XML チェックポイントの結果を操作できます。



- ▶ **チェックポイントのサマリを表示** : XML ツリー内の要素を選択して [チェックポイントのサマリを表示] ボタンをクリックするか, [表示] > [チェックポイントのサマリ] を選択します。チェックポイント・サマリ表示枠が [XML チェックポイント結果] ウィンドウの下部に表示されます。チェックポイント・サマリ表示枠には, 要素のどの部分の検査で成功または失敗したかに関する詳細が表示されます。

次は, ある XML ファイル内の **cities** という要素についてチェックポイント・サマリを表示した場合の例です。

The screenshot shows the 'XML Checkpoint Results' window. It is divided into three main sections:

- 期待 XML ツリー (Expected XML Tree):** Shows a tree structure with nodes like 'manyosyu', 'voluma', 'poem', 'pno', '8', 'mkana', 'poet', 'yomi', 'image', and 'mean'. Each node has a green checkmark icon, indicating successful validation.
- 実際 XML ツリー (Actual XML Tree):** Shows a similar tree structure, but with some nodes having a blue 'X' icon, indicating a validation failure. For example, the 'poet' node contains 'test(おかたのおおきみ)' instead of the expected text.
- チェックポイント サマリ (Checkpoint Summary):** A table at the bottom showing the results of the checks.

チェック	ステータス	期待値	実際値
属性チェック	成功	詳細については、属性テーブルを参照してください。	
次のタイプの子数 <Any Child>	成功	6	6



- ▶ **属性詳細の表示** : XML ツリーで, 属性の検査が行われた要素を選択します。[属性詳細の表示] ボタンをクリックするか, [表示] > [属性の詳細] を選択します。[XML チェックポイント結果] ウィンドウの下部の, [期待属性値] 表示枠と [実際属性値] 表示枠にそれぞれ, 属性検査の詳細が表示されます。

次は、XML Web ページまたはフレーム内の < Action > 要素に関する属性の詳細が表示されている例です。[期待属性値] 表示枠には、各属性について名前、期待値、および属性検査の結果ステータスが表示されます。

[実際属性値] 表示枠には、属性の名前および検査実行時に検出された実際の値が表示されます。

The screenshot shows the 'XML チェックポイント結果' (XML Checkpoint Results) window. It is divided into two main columns: '期待 XML ツリー' (Expected XML Tree) on the left and '実際 XML ツリー' (Actual XML Tree) on the right. Below each tree is a table of '期待属性値' (Expected Attributes) and '実際属性値' (Actual Attributes).

期待 XML ツリー (Expected XML Tree):

- man'yosyu
 - volume
 - poem
 - pno
 - 8
 - mkana
 - 熟田津尔 船兼世武登 月待者 潮毛可
 - poet
 - 額田王(ぬかたのおおきみ)
 - yomi
 - 熟田津に船乗りせむと月待てば潮もかな
 - image
 - image/m0008.jpg
 - mean
 - 熟田津(きたつ)で、船を出そうと月を待

実際 XML ツリー (Actual XML Tree):

- man'yosyu
 - volume
 - poem
 - pno
 - 8
 - mkana
 - 熟田津尔 船兼世武登 月待者 潮毛可奈比沼 今者許
 - poet
 - test(ぬかたのおおきみ)
 - yomi
 - 熟田津に船乗りせむと月待てば潮もかなひぬ今は漕ぎ出
 - image
 - image/m0008.jpg
 - mean
 - 熟田津(きたつ)で、船を出そうと月を待っていると、いよいよ

期待属性値 (Expected Attributes):

名前	値	結果
no	1	成功

実際属性値 (Actual Attributes):

名前	値
no	1



▶ 次のチェックを検索 : [表示] > [次のチェックを検索] を選択するか、[次のチェックを検索] ボタンをクリックすると、XML ツリー内で次に検査が行われた項目に直接ジャンプします。



▶ 前のチェックを検索 : [表示] > [前のチェックを検索] を選択するか、[前のチェックを検索] ボタンをクリックすると、XML ツリー内で検査が行われた 1 つ前の項目に直接ジャンプします。



- ▶ **次のエラーを検索** : [表示] > [次のエラーを検索] を選択するか, [次のエラーを検索] ボタンをクリックすると, XML ツリー内の次のエラーに直接ジャンプします。



- ▶ **前のエラーを検索** : [表示] > [前のエラーを検索] を選択するか, [前のエラーを検索] ボタンをクリックすると, XML ツリー内の前のエラーに直接ジャンプします。



- ▶ **ツリーの同時スクロール** : [表示] > [ツリーの同時スクロール] を選択するか, [ツリーの同時スクロール] ボタンをクリックすると, [期待 XML ツリー] と [実際 XML ツリー] の両方の XML ツリーを同期させながらスクロールできます。このオプションが選択されていると, [期待 XML ツリー] と [実際 XML ツリー] は, ツリー構造内をたどっていくときに同時にスクロールします。このオプションが選択されていないと, 一度に 1 つのツリーだけスクロールできます。

- ▶ **複数行の値の表示** : [XML チェックポイント結果] ウィンドウで任意の要素の値をダブルクリックして [要素の値] ダイアログ・ボックスを開き, 複数行の編集コントロールに値を表示できます。詳細については, 1041 ページ「[要素の値] ダイアログ・ボックス」を参照してください。



- ▶ **ヘルプトピック** : [ヘルプ] > [ヘルプトピック] を選択するか, [ヘルプトピック] ボタンをクリックすると, [XML チェックポイント結果] ウィンドウのヘルプが表示されます。

サンプルの XML チェックポイントの結果の検証

以下では XML チェックポイントのサンプル・シナリオを 4 つ紹介します。それぞれの例では, 実際の XML ドキュメントに発生した変更を示し, XML チェックポイントの結果で見つかった問題の原因を特定する方法について説明し, 該当する [XML チェックポイント結果] ウィンドウを示します。

シナリオ 1 :

次の例では **airline** という要素タグが **airlines** に変更されています。XML チェックポイントによってタグ構造に対するこの変更が検出されています。**airline** 要素の子要素の検査も、親レベルの要素に不一致があったために失敗しています。

問題のあった要素の詳細を閲覧するには、[期待 XML ツリー] の中で **airline** タグを選択し、[表示] > [チェックポイント サマリ] を選択します。[XML チェックポイント結果] ウィンドウの下側の表示枠にチェックポイント・サマリが表示されます。

「指定された要素が見つかりません」というテキストは、XML ドキュメント内の **airline** 要素タグが変更されたことを示しています。

The screenshot shows the 'XML チェックポイント結果' (XML Checkpoint Results) window. It is divided into three main sections:

- 期待 XML ツリー (Expected XML Tree):** Shows a tree structure for 'st-Setting'. The 'airline' element is highlighted with a red 'X' icon, indicating a failure. Other elements like 'customer_name', 'departure_date', and 'departure_time' are marked with green checkmarks.
- 実際 XML ツリー (Actual XML Tree):** Shows the actual XML tree. The 'airline' element is replaced by 'airlines', which is marked with a blue checkmark.
- チェックポイント サマリ (Checkpoint Summary):** A table summarizing the results of the checks.

チェック	ステータス	期待値	実際値
属性チェック	失敗	詳細については、属性テーブルを参照してください。	
次のタイプの子数 <任意の子>	失敗	0	<指定された要素が見つかりません。>

シナリオ 2 :

次の例は、orders 要素タグに関連付けられている属性が、期待値 orders1 から orders2 という新しい値に変化しています。

問題のあった属性の詳細を表示するには、問題のあった要素を [期待 XML ツリー] の中で選択し、[表示] > [属性の詳細] を選択します。[期待属性値] 表示枠と [実際属性値] 表示枠が [XML チェックポイント結果] ウィンドウの下部に表示されます。

[期待属性値] 表示枠と [実際属性値] 表示枠を使用して、問題の発生した属性を特定し、不一致が生じている値を確認できます。

The screenshot shows the 'XML チェックポイント結果' (XML Checkpoint Results) window. It is divided into four main sections:

- 期待 XML ツリー (Expected XML Tree):** Shows a tree structure where the 'orders' element under 'NewDataSet' is highlighted with a red 'X' icon, indicating a discrepancy.
- 実際 XML ツリー (Actual XML Tree):** Shows the same tree structure, but the 'orders' element is highlighted with a blue icon, indicating it is the current focus.
- 期待属性値 (Expected Attribute Values):** A table showing the expected value for the 'diffgr:jd' attribute as 'orders1'.
- 実際属性値 (Actual Attribute Values):** A table showing the actual value for the 'diffgr:jd' attribute as 'orders2'.

期待属性値		
名前	結果	
diffgr:jd	失敗	

実際属性値	
名前	値
diffgr:jd	orders2

シナリオ 3 :

次の例は、**total** 要素の実際の値が実行時に変更され、チェックポイントが失敗しています。

問題のあった値の詳細を閲覧するには、問題のあった要素を [期待 XML ツリー] の中で選択し、[表示] > [チェックポイント サマリ] を選択します。[XML チェックポイント結果] ウィンドウの下側の表示枠にチェックポイント・サマリが表示されます。



[チェックポイント サマリ] 表示枠を使用して、**total** 要素の期待値と実際の値とを比較できます。

The screenshot shows a window titled "XML チェックポイント結果" (XML Checkpoint Results). It contains two side-by-side tree views: "期待 XML ツリー" (Expected XML Tree) on the left and "実際 XML ツリー" (Actual XML Tree) on the right. The "total" element in both trees is highlighted, with the expected value 442.41 and the actual value 642.41. Below the trees is a "チェックポイント サマリ" (Checkpoint Summary) table.

チェック	ステータス	期待値	実際値
値	失敗	442.41	642.41

シナリオ 4 :

次は、total 要素の値がパラメータ化され、その値の内容によってこの反復でチェックポイントが失敗した例です。

値アイコン  にシャープ記号が表示され 、値がパラメータ化されていることを示しています。

問題のあった値の詳細を閲覧するには、問題のあった要素を [期待 XML ツリー] の中で選択し、[表示] > [チェックポイント サマリ] を選択します。[XML チェックポイント結果] ウィンドウの下側の表示枠にチェックポイント・サマリが表示されます。チェックポイントの結果を分析する手順は、値がパラメータ化されていても同じです。

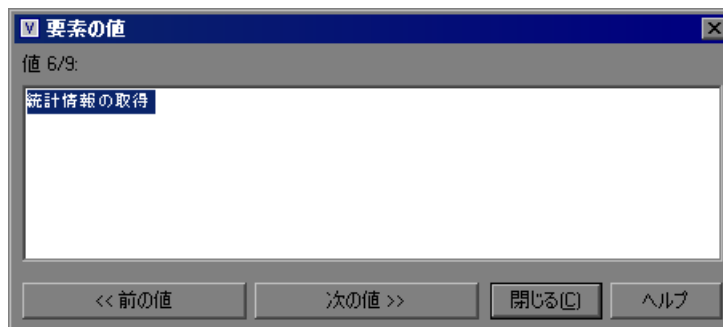
[チェックポイント サマリ] 表示枠を使用して、total 要素の期待値と実際の値とを比較できます。



[要素の値] ダイアログ・ボックス

説明	[XML チェックポイント結果] ウィンドウの要素の値を複数行の編集ウィンドウに表示できます。また、[期待 XML ツリー] または [実際 XML ツリー] の値の間で移動できます。
使用方法	[XML チェックポイント結果] ウィンドウで要素の値をダブルクリックします。
詳細	概念の概要 : 1031 ページ「XML チェックポイント結果の分析」

次に [要素の値] ダイアログ・ボックスの画像を示します。



[要素の値] ダイアログ・ボックスのオプション

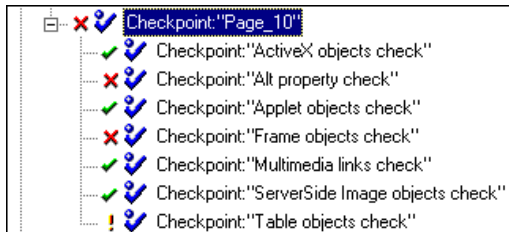
オプション	説明
[値 x/y]	[期待 XML ツリー] または [実際 XML ツリー] 内で選択した値の順位を示します。
編集ウィンドウ	複数行のウィンドウに要素または属性の完全な値が表示されます。
[前の値]	[XML チェックポイント結果] ウィンドウに、要素の値をさかのぼって表示できます。このボタンをクリックすると、[期待 XML ツリー] または [実際 XML ツリー] に次の値が表示されます。

オプション	説明
[次の値]	[XML チェックポイント結果] ウィンドウに、次の要素の値を表示できます。このボタンをクリックすると、[期待 XML ツリー] または [実際 XML ツリー] に次の値が表示されます。

アクセシビリティ・チェックポイント結果の分析

テストにアクセシビリティ・チェックポイントを含めると、検査した各アクセシビリティ・オプションの結果が [テスト結果] ウィンドウに表示されます。

実行結果ツリーには、各チェックポイントで検査された各アクセシビリティ・オプションの個々のステップが表示されます。たとえば、すべてのアクセシビリティ・オプションを選択した場合、アクセシビリティ・チェックポイントの実行結果ツリーは、次のように表示されます。



テスト結果の詳細には、W3C の「Web Content Accessibility Guidelines」に準拠していない可能性のある Web サイトの部分をも特定するのに役立つ情報が含まれています。各検査について提供される情報は、W3C 要件に基づいています。

注：アクセシビリティ・チェックポイントに関する W3C の「Web Content Accessibility Guidelines」の一部については、次の項で引用または要約します。ただし、ここに記載する情報は網羅的なものではありません。Web サイトが W3C の「Web Content Accessibility Guidelines」に準拠しているかどうかを検査する場合は、<http://www.w3.org/TR/WAI-WEBCONTENT/> で完全なドキュメントを参照してください。

アクセシビリティ・チェックポイントの詳細については、『**HP QuickTest Professional アドイン・ガイド**』の Web オブジェクトのテストの項を参照してください。

ActiveX オブジェクトのチェック

W3C の「Web Content Accessibility Guidelines」のガイドライン 6 では、新しいテクノロジーがサポートされていない場合や無効になっている場合でも、そのページに確実にアクセスできることを要求しています。[ActiveX オブジェクトのチェック] を選択した場合、選択したページまたはフレームに ActiveX オブジェクト（マルチメディアを含む）が含まれているかどうか検査されます。ActiveX オブジェクトが含まれていない場合、チェックポイントは成功となります。ページまたはフレームに ActiveX オブジェクトが含まれている場合は、警告と ActiveX オブジェクトのリストが結果に表示されるので、ActiveX がサポートされていないブラウザでのこれらのページのアクセシビリティを検査できます。例を次に示します。

ActiveX オブジェクトのチェック	
オブジェクト タグ	オブジェクト名
OBJECT	MSXML3
OBJECT	MQC

Alt プロパティのチェック

W3C の「Web Content Accessibility Guidelines」のガイドライン 1.1 では、すべての非テキスト要素に代替テキストを用意することを要求しています。[Alt Property Check] では、このガイドラインで Alt プロパティが必要とされるオブジェクトに実際にこの属性があるかどうか検査されます。選択したフレームまたはページに該当オブジェクトがない場合、またはすべての該当オブジェクトに必要な属性がある場合、チェックポイントは成功となります。プロパティが必要とされる 1 つ以上のオブジェクトにこのプロパティがない場合、テストは失敗となります。テスト結果の詳細には、属性が欠けているオブジェクトのリストが表示されます。例を次に示します。

Alt プロパティのチェック		
オブジェクト タグ	オブジェクト名	Alt 値
IMG	Mercury Tours	Mercury Tours
IMG	html	[なし]
IMG	boxad1	[なし]
IMG	banner2	[なし]
IMG	Featured Destination: Aruba	Featured Destination: Aruba

キャプチャされたページまたはフレームが [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表示されるので、[Alt プロパティのチェック] リストに表示されたオブジェクトを確認できます。

アプレットのチェック

[アプレットのチェック] を利用すると、検査対象のページまたはフレームで Java アプレットまたは Java アプリケーションを検索することで、新しいテクノロジーがサポートされていないか無効になっている場合でも、ページに確実にアクセスできる (W3C の「Web Content Accessibility Guidelines」のガイドライン 6) かどうかを検査できます。ページまたはフレームに Java アプレットまたは Java アプリケーションが含まれていない場合、チェックポイントは成功となります。それ以外の場合は、警告と Java アプレットおよび Java アプリケーションのリストが検査結果として表示されます。例を次に示します

アプレット オブジェクトのチェック	
オブジェクト タグ	オブジェクト名
APPLET	VApplet.class

フレーム・タイトルのチェック

W3C の「Web Content Accessibility Guidelines」のガイドライン 12.1 では、フレームの識別およびナビゲーションを容易にするように、各フレームにタイトルを付けることを要求しています。[Frame Titles Check] を選択すると、Frame オブジェクトと Page オブジェクトに TITLE タグがあるかどうか検査されます。選択したページまたはフレーム、およびその中のすべてのフレームにタイトルがある場合、チェックポイントは成功となります。選択したページまたは 1 つ以上のフレームにタグがない場合、テストは失敗となります。テスト結果の詳細には、タグが欠けているオブジェクトのリストが表示されます。次に例を示します。

フレーム タイトルのチェック			
オブジェクト クラス	オブジェクト タグ	オブジェクト名	タイトルの値
Frame	IFRAME	theFrame	[なし]
Page		Test.com	Test.com

キャプチャされたページまたはフレームが [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表示されるので、[フレーム タイトルのチェック] リストに表示されたフレームを確認できます。

マルチメディア・リンクのチェック

W3C の「Web Content Accessibility Guidelines」のガイドライン 1.3 および 1.4 では、マルチメディア表現の視覚的なトラックには、その表現と同期した音声による説明を付けることを要求しています。また、ガイドライン 6 では、新しいテクノロジーがサポートされていない場合や無効になっている場合でも、そのページに確実にアクセスできることを要求しています。[Multimedia Links Check] は、マルチメディア・オブジェクトへのリンクを識別するので、必要に応じて代替リンクがあることを確認できます。ページまたはフレームにマルチメディア・リンクが含まれない場合、チェックポイントは成功となります。それ以外の場合は、警告とマルチメディア・リンクのリストが検査結果として表示されます。

サーバ側の画像のチェック

W3C の「Web Content Accessibility Guidelines」のガイドライン 1.2 では、サーバ側の画像マップのアクティブな領域ごとに、テキストによる代替リンクを用意することを要求しています。ガイドライン 9.1 では、使用可能な幾何形状で領域を定義できない場合を除き、サーバ側の画像マップではなく、クライアント側の画像マップを用意することを推奨しています。[Server-Side Image Check] を選択した場合、選択したページまたはフレームにサーバ側の画像が含まれているかどうか検査されます。サーバ側の画像が含まれていない場合、チェックポイントは成功となります。ページまたはフレームにサーバ側の画像が含まれている場合、警告とサーバ側の画像のリストが検査の結果として表示されるので、それぞれがガイドライン要件を満たしているかどうかを確認できます。例を次に示します

サーバ側画像のチェック	
オブジェクト クラス	オブジェクト名
Image	Navigation

テーブルのチェック

W3C の「Web Content Accessibility Guidelines」のガイドライン 5 では、アクセス可能なブラウザやその他のユーザ・エージェントによる変換に必要なマークアップを確実にテーブルに含めることを要求しています。ガイドラインでは、テーブルの主な使用目的は真の表形式データを表示するためであり、テーブルを線形化した場合でも意味が通る場合を除いて、レイアウト目的でテーブルを使用することは避けるべきであることが強調されています。TH, TD, THEAD, TFOOT, TBODY, COL, および COLGROUP が推奨タグです。これらのタグを使用すれば、ユーザはユーザ・エージェントを利用して、音響手段、音声出力、または点字画面を介して、テーブル・セル間をナビゲートしたり、ヘッダやその他のテーブル・セル情報にアクセスしたりできます。

[テーブルのチェック] では、選択したページまたはフレームにテーブルが含まれているかどうかを検査されます。サーバ側の画像が含まれていない場合、チェックポイントは成功となります。ページまたはフレームにテーブルが含まれている場合、警告とテーブルのタグ構造の視覚的表現が検査の結果として表示されます。例を次に示します。

Table objects check		
Object Class	Object Name	Table Structure
WebTable	Table 1	

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。また、実行時データ・テーブルの内容を表示することもできます。

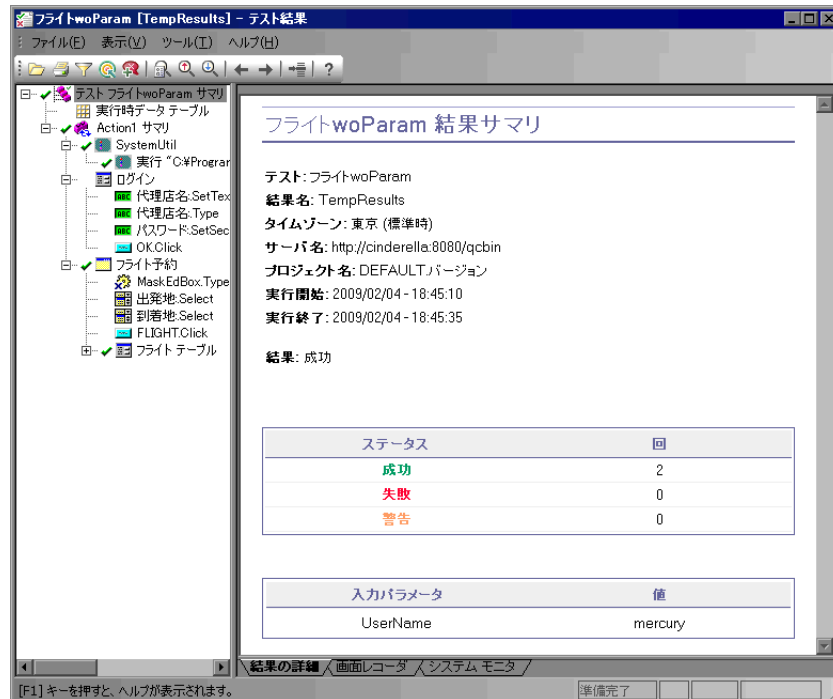
[テスト結果] ウィンドウでのパラメータ化された値の表示

「**パラメータ**」とは、外部のデータ・ソースまたはジェネレータから値が割り当てられる変数です。[テスト結果] ウィンドウにテス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 テストのテスト結果を [テスト結果] ウィンドウに表示します。詳細については、978 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、パラメータ化された値を含むテストまたはアクションの分岐をクリックするします。

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」を使ってアクション用に定義された入力パラメータ「UserName」を示します。

テストでのパラメータの定義と使用については、第 24 章「値のパラメータ化」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 テストのテスト結果を [テスト結果] ウィンドウに表示します。詳細については、978 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 25 章「値の出力」を参照してください。

XML 出力値のステップの結果の表示の詳細については、1049 ページ「XML 出力値の結果の分析」を参照してください。

実行時データ・テーブルの表示

テストをデータ・テーブル・パラメータまたはデータ・テーブル出力値ステップで実行すると、[実行時データ テーブル] には、使用されたパラメータ化された値と、実行中にデータ・テーブルに保存された出力値が表示されます。実行時データ・テーブルの内容は、[テスト結果] ウィンドウで見ることができます。

実行時データ・テーブルを表示するには、次の手順を実行します。

- 1 テストのテスト結果を [テスト結果] ウィンドウに表示します。詳細については、978 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、「**実行時データ テーブル**」を強調表示します。



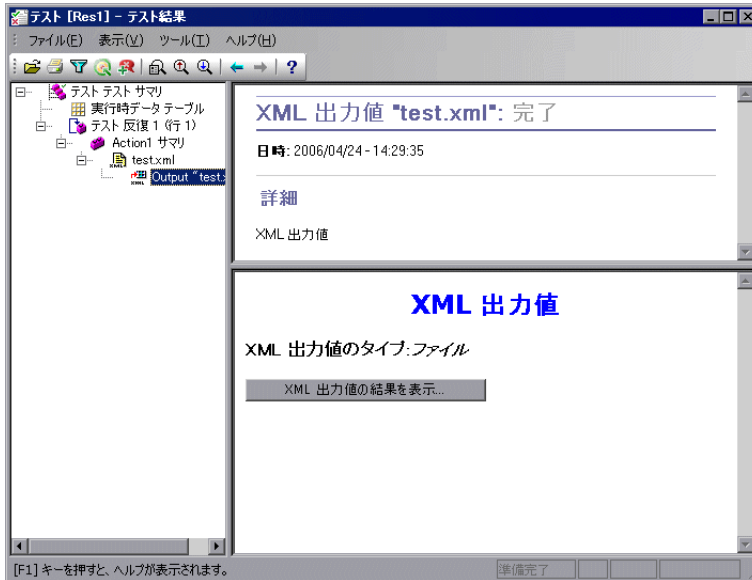
前の例では、実行時データ・テーブルには、パラメータ化されたフライトの出発地の値が含まれています。

実行時データ・テーブルの詳細については、第 42 章「データ・テーブルを使った作業」を参照してください。

XML 出力値の結果の分析

アプリケーションで使用している XML ドキュメントのテストに要素または属性値を出力できます。出力値の詳細については、727 ページ「XML 値の出力」を参照してください。

XML 出力値の結果の要約は [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、1022 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。



右側の表示枠の [結果の詳細] タブには、出力値の結果のサマリが表示されます。[XML 出力値の結果を表示] をクリックして [XML 出力値の結果] ウィンドウを開き、詳細な結果を表示できます。

注：標準設定では、[XML 出力値の結果を表示] はエラーが発生した場合のみ使用できます。詳細な結果を得られるかどうかは、[オプション] ダイアログ・ボックスの [実行] > [画面キャプチャ] 表示枠の [静止画像キャプチャをテスト結果へ保存] 設定によって異なります。詳細については、1246 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。

XML 出力値の結果の詳細については、1051 ページ「[XML 出力値の結果] ウィンドウについて」を参照してください。

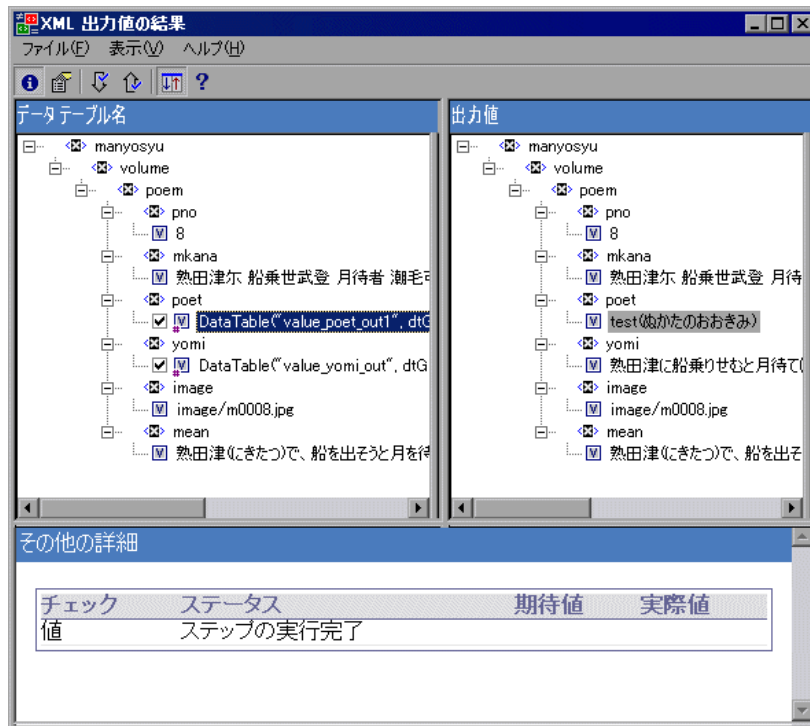
[XML 出力値の結果] ウィンドウについて

[テスト結果] ウィンドウの [XML 出力値の結果の表示] ボタンをクリックすると、[XML 出力値の結果] ウィンドウに XML ファイルの階層構造が表示されます。

[データ テーブル名] 表示枠には、XML 出力値の設定（データ・テーブル出力値の出力用に選択した XML の構造とデータ・テーブル・カラム名）が表示されます。

出力値表示枠には、実際の XML ツリー（XML ドキュメントやファイルの実際の見え方や実行中に出力された実際の値）が表示されます。

[その他の詳細] 表示枠には、選択された項目の結果情報が表示されます。



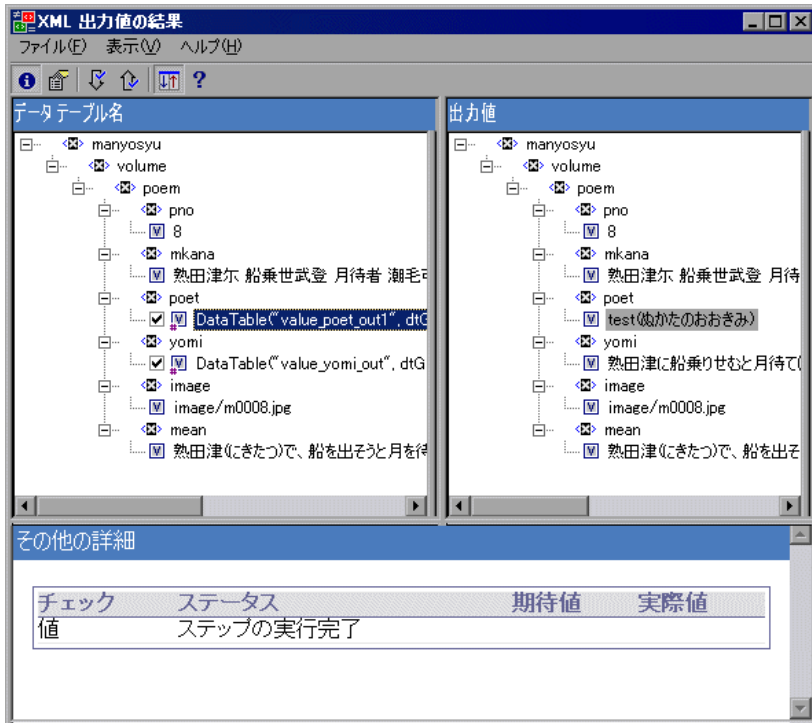
[XML 出力値の結果] ウィンドウの操作

[XML 出力値の結果] ウィンドウには、メニューとツールバーが表示され、これらを使用して XML 出力値の結果に関するさまざまなコンポーネントを操作できます。

次に示すコマンドまたはツールバー・ボタンを使用して、XML 出力値の結果を操作できます。



- ▶ **出力値のサマリを表示**：XML ツリー内の要素を選択して [出力値のサマリを表示] ボタンをクリックするか、[表示] > [出力値のサマリ] を選択します。[その他の詳細] 表示枠が [XML 出力値の結果] ウィンドウの下部に表示されます。この表示枠には、選択された要素の出力値、属性、値に関する詳細が表示されます。





- ▶ **属性詳細の表示** : XML ツリーで、属性が値として出力された要素を選択します。[属性の詳細] ボタンをクリックするか、[表示] > [属性の詳細] を選択します。[XML 出力値の結果] ウィンドウの下部の、期待属性値表示枠と実際属性値表示枠にそれぞれ、属性の出力値の詳細が表示されます。

[期待属性値] 表示枠には、各属性名と期待値、または出力値名が表示されます。[実際属性値] 表示枠には、属性の名前と各属性の実行セッション中の実際値が表示されます。

The screenshot shows the 'XML 出力値の結果' window with the following content:

データテーブル名

- manyosyu
 - volume
 - poem
 - pno: 8
 - mkana: 熟田津尔 船乗世武登 月待者 潮毛可奈比
 - poet: test(ぬかたのおおきみ)
 - yomi: 熟田津に船乗りせむと月待てば潮もかなひぬ
 - image: image/m0008.jpg
 - mean: 熟田津(きたつ)で、船を出そうと月を待

出力値

- manyosyu
 - volume
 - poem
 - pno: 8
 - mkana: 熟田津尔 船乗世武登 月待者 潮毛可奈比
 - poet: test(ぬかたのおおきみ)
 - yomi: 熟田津に船乗りせむと月待てば潮もかなひぬ
 - image: image/m0008.jpg
 - mean: 熟田津(きたつ)で、船を出そうと月を待

期待属性値

	名前	値
1	no	1

実際属性値

	名前	値
1	no	1



- ▶ **次の出力値を検索** : [表示] > [次の出力値を検索] を選択するか, [次の出力値を検索] ボタンをクリックすると, XML ツリー内の次の出力値に直接ジャンプします。



- ▶ **前の出力値を検索** : [表示] > [前の出力値を検索] を選択するか, [前の出力値を検索] ボタンをクリックすると, XML ツリー内の 1 つ前の出力値に直接ジャンプします。



- ▶ **次のエラーを検索** : [表示] > [次のエラーを検索] を選択するか, [次のエラーを検索] ボタンをクリックすると, XML ツリー内の次のエラーに直接ジャンプします。



- ▶ **前のエラーを検索** : [表示] > [前のエラーを検索] を選択するか, [前のエラーを検索] ボタンをクリックすると, XML ツリー内の前のエラーに直接ジャンプします。



- ▶ **ツリーの同時スクロール** : [表示] > [ツリーの同時スクロール] を選択するか, [ツリーの同時スクロール] ボタンをクリックすると, [データ テーブル名] ツリーと [出力値] ツリーのスクロールを同期できます。

このオプションが選択されていると, [データ テーブル名] と [出力値] は, ツリー構造内をたどっていくときに同時にスクロールします。このオプションが選択されていないと, 一度に 1 つのツリーだけスクロールできます。



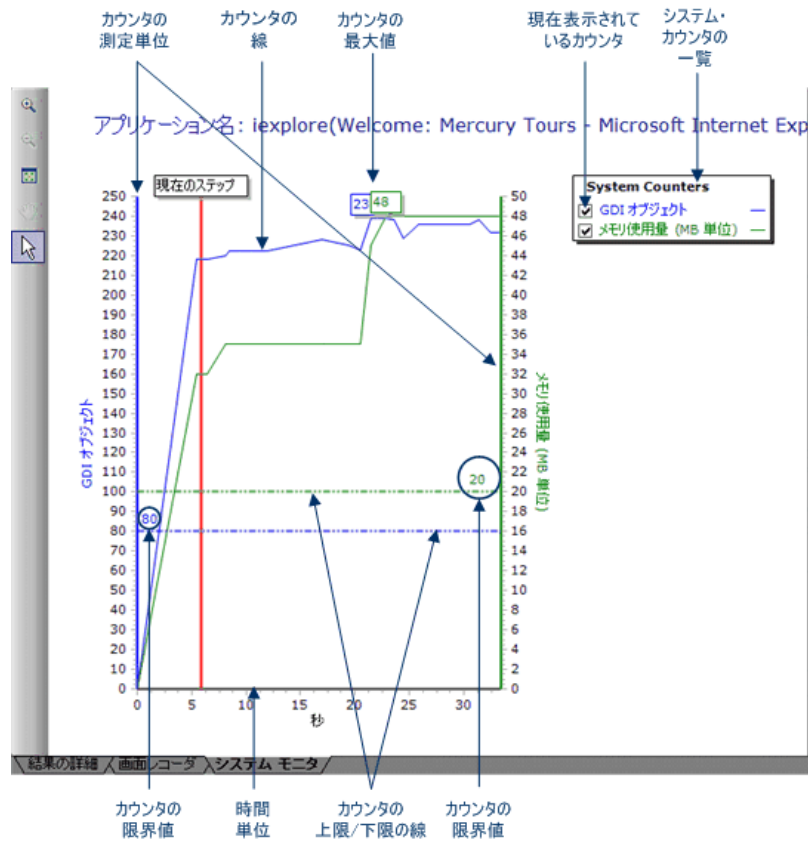
- ▶ **ヘルプトピック** : [ヘルプ] > [ヘルプトピック] を選択するか, [ヘルプトピック] ボタンをクリックすると, [XML 出力値の結果] ウィンドウのヘルプが表示されます。

[システム モニタ] の結果表示

[テスト結果] ウィンドウの [システム モニタ] タブには, テストのために監視したシステム・カウンタが表示されます。ローカル・システム監視は, テストを目的としてのみ有効化できます。

テストのためにローカル・システム監視を有効にする方法の詳細については, 1284 ページ「テストのためのシステム監視の有効化」を参照してください。

[システム モニタ] タブには、システム・カウンタの結果が線グラフで表示されます。



[システム モニタ] タブ

[システム モニタ] タブには、次の情報が表示されます。

- ▶ **[アプリケーション名]**：システム・カウンタの監視対象となったアプリケーションの名前。
- ▶ **[システム カウンタ リスト]**：監視したアプリケーションのシステム・カウンタのリスト。

- ▶ **[現在表示されているカウンタ]**：線グラフに現在表示されているカウンタのリスト。[システム モニタ] タブに表示されるカウンタは、一度に 2 個までです。表示するカウンタを変更するには、現在選択されているカウンタの 1 つまたは両方のチェック・ボックスをクリアして、必要なカウンタのチェック・ボックスを選択します。
- ▶ **[カウンタの測定単位]**：該当するカウンタのパフォーマンスの測定単位。
- ▶ **[カウンタの最大値]**：実行セッション中にカウンタが達した最大値。
- ▶ **[現在のステップ]**：実行結果ツリーで現在強調表示されているステップを示すグラフ上のポイント。
- ▶ **[カウンタの上限 / 下限の線]**：該当するカウンタの上限を視覚的に示した線（設定されている場合）。[テストの設定] ダイアログ・ボックスの [ローカル システム モニタ] 表示枠で定義します。設定した場合、カウンタがこの上限を超えるとステップが失敗します。最初にカウンタの上限を超えたステップのみが失敗します。2 つ目以降のステップがカウンタの上限を超えても影響を受けません。
- ▶ **[カウンタの限界値]**：該当するカウンタの上限値（設定されている場合）。[テストの設定] ダイアログ・ボックスの [ローカル システム モニタ] 表示枠で定義します。設定した場合、カウンタがこの上限を超えるとステップが失敗します。最初にカウンタの上限を超えたステップのみが失敗します。2 つ目以降のステップがカウンタの上限を超えても影響を受けません。
- ▶ **[時間単位]**：実行セッションの時間単位（秒単位）。






[システム モニタ] タブの色

各カウンタはグラフ内で色分けされています。カウンタの色は、次の項目を示しています。

- ▶ **[システム カウンタ リスト]** 内のカウンタの名前。
- ▶ **[カウンタの線]**
- ▶ **[カウンタの測定単位]**
- ▶ **[カウンタの上限 / 下限の線]**
- ▶ **[カウンタの限界値]**
- ▶ **[カウンタの最大値]**

【システム モニタ】 タブのツールバー

【システム モニタ】 タブのツールバーには、次のボタンが表示されます。

ボタン	用途
	拡大表示するには、 【拡大表示】 ボタンをクリックしてからグラフ上の任意の位置をクリックします。また、グラフ上の領域をクリックしてドラッグすることでその領域を拡大表示することもできます。
	縮小表示するには、 【縮小表示】 ボタンをクリックしてからグラフ上の任意の位置をクリックします。
	縮小表示してグラフ全体を表示するには 【グラフの全体を表示】 ボタンをクリックします。このボタンは、グラフが拡大表示されていないときは無効になります。
	【移動】 ボタンをクリックしてからグラフ上でクリックしてドラッグすると、左右にスクロールできます。このボタンは、グラフが拡大表示されていないときは無効になります。
	あるポイントを現在のステップとして選択するには、 【矢印】 ボタンをクリックしてからグラフ上の該当する場所をダブルクリックします。 【現在のステップ】 インジケータが新しい場所に移動して、ステップが実行結果ツリーで強調表示されます。また、グラフの 【Counter Line】 上の任意のポイントにカーソルを移動して、そのポイントの 【Counter Line】 の値を表示することもできます。

【システム モニタ】 タブの結果のエクスポート

データは、【システム モニタ】 タブから、**テキスト**（.csv または .txt）、**Excel**、**XML**、または **HTML** 形式のファイルにエクスポートできます。

システム・モニタ・データをエクスポートするには、次の手順を実行します。

【ファイル】 > **【システム モニタ データをファイルへエクスポート】** を選択し、エクスポート先データのファイル名とファイルの種類を選択します。

第 VII 部

テストの保守およびデバッグ

第 35 章

テス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、テスト、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。

本章の内容

- ▶ テストと関数ライブラリのデバッグについて (1062 ページ)
- ▶ デバッグ・セッションの速度調節 (1064 ページ)
- ▶ シングル・ステップ・コマンドの使用 (1064 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 (1068 ページ)
- ▶ 実行セッションの一時停止 (1070 ページ)
- ▶ ブレークポイントの使用 (1070 ページ)
- ▶ [デバッグ ビューア] 表示枠 (1073 ページ)
- ▶ 実行エラーの処理 (1084 ページ)
- ▶ アクションまたは関数のデバッグの練習 (1086 ページ)

テストと関数ライブラリのデバッグについて

テス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テストに関連付け、次にそのテス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、テス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、テス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、テス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、テス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、テス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テス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检查したり、テス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、954 ページ「テストの一部の実行」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、989 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。

テス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・モードでテストを実行するには、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）。
- ▶ テス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、917 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、テス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、別のテストを開いたり、新規のテス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。
- ▶ ExecuteFile ステートメントを使用してファイルを呼び出した場合、ファイルまたは当該ファイル内に含まれている関数はデバッグできません。また、ExecuteFile ステートメントを含んでいるテストをデバッグする場合、実行マークが正しく表示されないことがあります。
- ▶ QuickTest では、テストを開くと、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、テストを閉じて再度開くまでテストで認識されません（外部リソースとはテス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）。

これに対して、関数ライブラリなどファイル・システムに保存されている外部リソースに適用した変更は、ただちに実装されます。これらのファイルは直接アクセスされ、テストを開いたときにローカル・コピーとして保存されないためです。

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。テス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、1243 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

[ステップイントウ] 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、テス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。

[ステップイントウ]

[ステップイントウ] では、アクティブなテス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。現在のステップが別のアクションまたは関数を呼び出すと、呼び出されたアクションまたは関数が QuickTest ウィンドウに表示され、テストまたは関数ライブラリは、呼び出されたアクションまたは関数の最初の行で一時停止します。

[ステップイントウ] 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] > [ステップイントウ] を選択し、[ステップイントウ] 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]

[**ステップイントウ**] を使用して関数ライブラリのアクションまたは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、呼び出されたアクションまたは関数の終了まで実行が継続され、呼び出し元のテス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、アクティブなテス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、別のアクションやユーザ定義関数が呼び出されている場合は、呼び出し先のアクションまたは関数の全体が実行されますが、呼び出し先のアクションまたは関数スクリプトは 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テス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 イントウ], [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、(テストから) 実行するには、次の手順を実行します。

<p>サンプルの関数ライブラリおよびテストを作成する</p>	<ol style="list-style-type: none"> 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。 <pre>public Function myfunc() msgbox "one" msgbox "two" msgbox "three" End Function</pre> 3 SampleFL.qfl という名前で、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します (詳細については、912 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] > [テスト] を選択して新しいテストを開きます。 5 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。 6 [ファイル] > [ライブラリ 'SampleFL.qfl' を 'テスト' に関連付ける] を選択して関数ライブラリをテストに関連付けます。 7 作成したテストのタブをクリックして、フォーカスを対象に移動します。 [エキスパート ビュー] タブをクリックしてエキスパート・ビューを表示し、次の行を正確に入力します。 <pre>myfunc myfunc myfunc endOfTest="true"</pre>
--------------------------------	--

<p>関数ライブラリを テストから実行し、 [ステップ イントウ]、 [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</p>	<p>8 F9 キー ([ブレークポイントの設定 / 解除]) を押して、テス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、1071 ページ「ブレークポイントの設定」を参照してください。</p> <p>9 テストを実行します。テストは次のブレークポイントで一時停止します。</p> <p>10 F11 キー ([ステップイントウ]) 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。</p> <p>11 F11 キー ([ステップイントウ]) 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。</p> <p>12 [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。</p> <p>13 F11 キー ([ステップイントウ]) (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テストの 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。</p> <p>14 F11 キー ([ステップイントウ]) 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。</p> <p>15 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、実行矢印はテス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)。</p> <p>16 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、実行矢印はテス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、実行矢印はテストの次のステップをポイントします。</p>
---	---

[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テストまたはアクション（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]

テストまたはアクション（エキスパート・ビューのみ）の最初から、あるいはテストまたはアクション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テストまたはアクションのデバッグを開始する場合、そのステップまでテストまたはアクションを実行し、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、テストまたはアクションの編集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。
- ▶ [テストフロー] 表示枠で、実行を停止するアクションを右クリックし、ショートカット・メニューから **[QuickTest ステップまで実行]** を選択します。これにより、そのアクションの最初のステップで実行が停止されます。

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]

テストまたはアクション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。テストまたはアクションの編集集中にテストまたはアクション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。
 - ▶ [テストフロー] 表示枠で、実行を開始するアクションを右クリックし、ショートカット・メニューから **[ステップからデバッグ]** を選択します。これにより、そのアクションの最初のステップで実行が開始されます。

[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、958 ページ「[実行] ダイアログ・ボックス: [結果の場所] タブ」および 960 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。**[ステップイントウ]**、**[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 33 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 44 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。一時停止されたテス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。

ブレイクポイントの使用

ブレイクポイントを使用すると、テス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。ブレイ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。ブレイ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、ブレイクポイントからテストまたは関数ライブラリの実行を再開できます。ブレイクポイントは、現在の QuickTest セッションにのみ適用され、テストまたは関数ライブラリには保存されません。

ブレイ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・コマンドを使ってテス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。

ブレイクポイントを設定し、そのブレイ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。ブレイクポイントの使用が終了したら、テストまたは関数ライブラリからブレイ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、実行セッションをテス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。

- ▶ テス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、テス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、テス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、テス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、テス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、テス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

現在のテス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。
- ▶ テス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

<p>説明</p>	<p>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。</p> <ul style="list-style-type: none"> ▶ テスト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。
<p>使用方法</p>	<p>[表示] > [デバッグ ビューア] メニュー・コマンドを選択します。</p>
<p>重要な情報</p>	<p>次の状況の場合は、実行セッションを一時停止できます。</p> <ul style="list-style-type: none"> ▶ 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。
<p>詳細</p>	<p>概念の概要 : 1062 ページ「テストと関数ライブラリのデバッグについて」</p> <p>主要なタスク : 1086 ページ「アクションまたは関数のデバッグの練習」</p>

【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、1074 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**：現在のアクションのメイン・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、1079 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、1082 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。

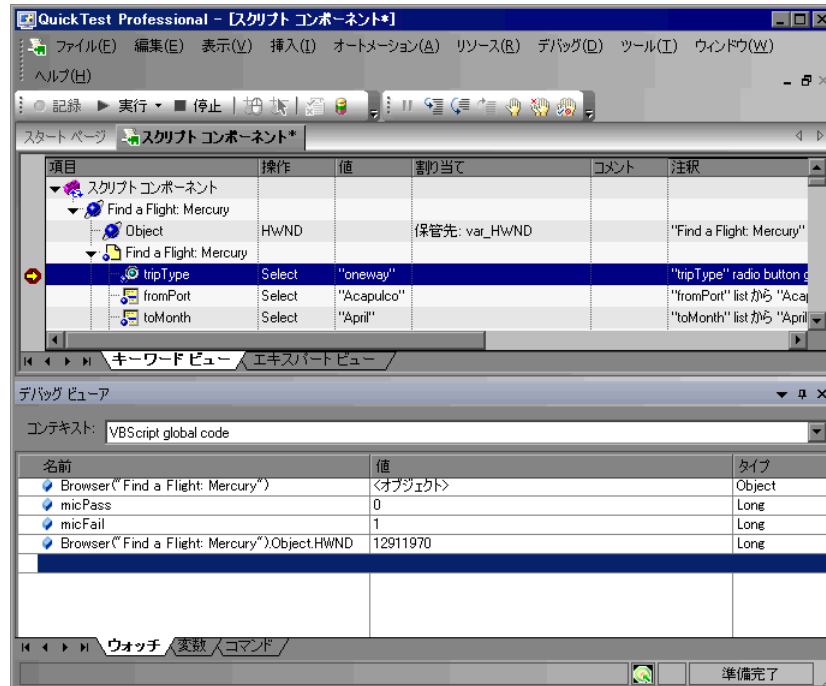
【デバッグ ビューア】 表示枠：【ウォッチ】 タブ

説明	<p>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、およびテストまたは関数ライブラリの VBScript 式を表示できます。</p> <p>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。</p>
使用方法	<p>【表示】 メニュー > 【デバッグ ビューア】 項目 > 【ウォッチ】 タブ</p>
詳細	<p>主要なタスク：1077 ページ「【デバッグ ビューア】 表示枠の【ウォッチ】 タブの使用」</p> <p>その他の関連項目：</p> <ul style="list-style-type: none"> ▶ 1073 ページ「【デバッグ ビューア】 表示枠」 ▶ 1086 ページ「アクションまたは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、テス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、一時停止されたアクション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、1079 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	<p>[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。</p> <ul style="list-style-type: none"> ▶ 実行セッションが、テス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、一時停止されたアクション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	<p>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、1077 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。</p> <p>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。テストまたは関数ライブラリで、予期しない動作が発生する可能性があります。</p>

[値] カラム	<p>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。</p> <p>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。</p> <p>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。</p> <p>[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。</p>
[タイプ] カラム	<p>評価後の式の値の種類 (Integer や String など)。</p> <p>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)。</p>

[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、テストまたは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、またはテスト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。

テストまたは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、1074 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。テストまたは関数ライブラリで、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：エキスパート・ビューまたは関数ライブラリから [ウォッチ] タブに式を追加できます。

[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。

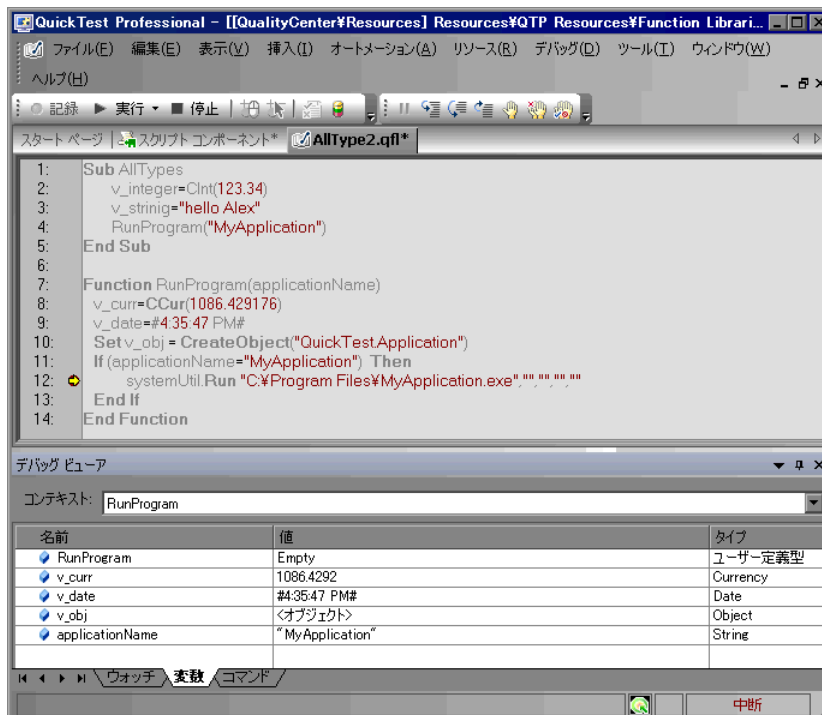
[ウォッチ] タブで削除する行を選択し、キーボードで **Delete** キーを押します。

[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] タブには現在のアクションのメイン・スクリプト内、またはテストまた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。テストまたは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： <ul style="list-style-type: none"> ▶ 1073 ページ「[デバッグ ビューア] 表示枠」 ▶ 1086 ページ「アクションまたは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。



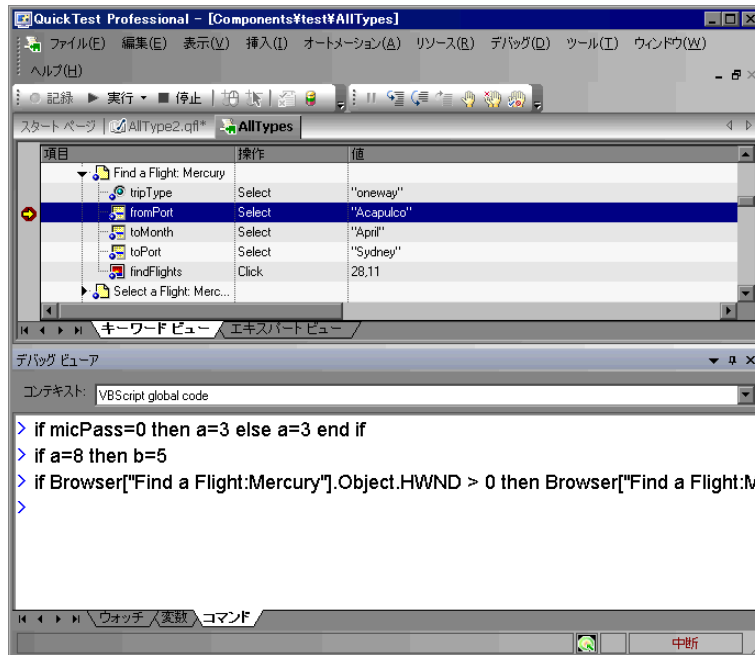
デバッグ・ビューアの [変数] タブの詳細

項目	説明
[コンテキスト] ボックス	<p>このタブに表示される変数のコンテキストを示します。</p> <ul style="list-style-type: none"> ▶ 実行セッションが、テス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、一時停止されたアクション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

<p>説明</p>	<p>実行セッションが一時停止された場合、このタブでは、テストまたは関数ライブラリの VBScript コードの行を実行できます。</p> <p>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。</p> <ul style="list-style-type: none"> ▶ 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。
<p>使用方法</p>	<p>[表示] メニュー > [デバッグ ビューア] 項目 > [コマンド] タブ</p>
<p>詳細</p>	<p>その他の関連項目：</p> <ul style="list-style-type: none"> ▶ 1073 ページ「[デバッグ ビューア] 表示枠」 ▶ 1086 ページ「アクションまたは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。

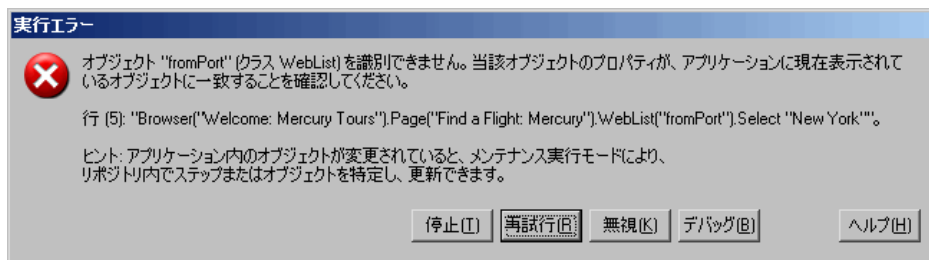
- ▶ **右クリックのショートカット・メニュー** : [コマンド] タブの内容の編集に使用できるコマンドが表示されます。
- ▶ **[切り取り]**, **[コピー]**, および **[貼り付け]** コマンドを使用すると、クリップボードを使用してコマンド履歴からテキストをコピーしたり、アクティブなコマンド・ラインを編集したりできます。
- ▶ **[すべてクリア]** コマンドを使用すると、コマンド履歴をすべて消去できます。

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、テストおよびテス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、テス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、テス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、1094 ページ「メンテナンス実行ウィザードを使ったテストの実行」を参照してください。

アクションまたは関数のデバッグの練習

テストまたは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アクションまたは関数を作成するとします。アクションまたは関数にブレークポイントを追加することで、テストまたは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、テストまたは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：新しいアクションまたは関数を作成する

テストを開いて新しいアクションを挿入します。または、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。アクションの挿入の詳細については、第 15 章「アクションを使った作業」を参照してください。関数での作業の詳細については、第 31 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。

エキスパート・ビューまたは関数ライブラリで、次のように VBScript コードを入力します。

Expert View	Function Library
<pre>Dim a a="hello" b="me" MsgBox a</pre>	<pre>Function SetVariables Dim a a="hello" b="me" MsgBox a EndFunction</pre>

[エキスパート ビュー] の詳細については、第 29 章「エキスパート・ビューおよび関数ライブラリ・ウィンドウを使用した作業」を参照してください。

注：エキスパート・ビューで作業している場合は、手順 4 に進みます。関数ライブラリで作業している場合は、手順 2 と手順 3 に進みます。

手順 2 : (関数ライブラリの場合のみ) 関数ライブラリをテストに関連付ける

- 1 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 '<テスト名>' に関連付ける] を選択します。QuickTest によって、関数ライブラリがテストに関連付けられます。

手順 3 : (関数ライブラリの場合のみ) テス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、新しいステップを挿入し、エキスパート・ビューで次のように入力します。

SetVariables

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、1071 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 : テストの実行を開始する

テストを実行します。テス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、テストが変数 **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。テストを使って作業している場合は、**a** がこの時点までに初期化された唯一の変数であるため、変数 **a** (値は「**hello**」) のみ表示されます。関数ライブラリで作業している場合は、**SetVariables** (値は **Empty**) と **a** (値は「**hello**」) の両方が表示されます。変数 **b** が宣言される前にテストが停止したため、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、テストの実行を続行します。テス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、テス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、テストを再び実行します。テス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、前のテス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テスト実行を完了させます。

第 36 章

テストの保守

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テス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、テストのチェックポイントの期待値を更新する必要がある場合もあります。本章では、QuickTest のツールを使ってテストを更新および保守する方法を説明します。

本章の内容

- ▶ テストが失敗する理由 (1092 ページ)
- ▶ メンテナンス実行ウィザードを使ったテストの実行 (1094 ページ)
- ▶ 更新モード・オプションを使ったテストの更新 (1111 ページ)

テストが失敗する理由

テストは、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。

場合によってはテス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、テストが失敗する場合があります。その場合は、QuickTest のテストを更新して、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、テス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テス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、テス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、テス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、テス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・リポジトリ内でテス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・リポジトリでテス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、テストで使用できます。

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テス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。

テス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があります。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テス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。

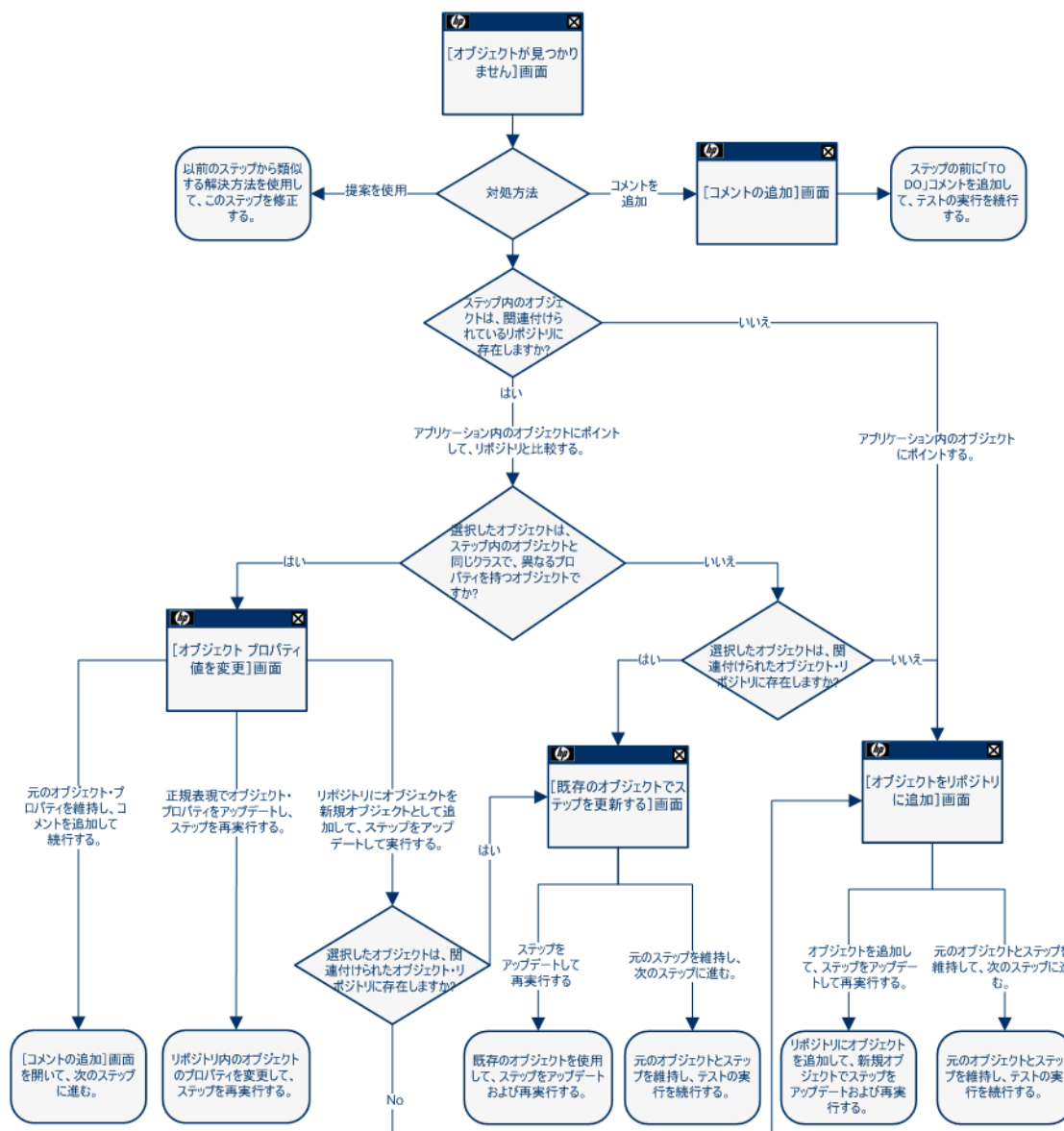
メンテナンス実行ウィザードを使ったテストの実行

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、テストを保守できます。

問題	解決策
<p>テス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。</p>	<p>メンテナンス実行ウィザードを使用すると、テス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。</p> <p>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。</p> <p>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、テス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。</p> <p>また、失敗したステップの前のテストにコメントを追加することもできます。</p>
<p>テス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。</p>	<p>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。</p> <p>また、失敗したステップの前のテストにコメントを追加することもできます。</p>
<p>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。</p>	<p>スマート認識を使用してオブジェクトを認識すると、テス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、123 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。</p>

メンテナンス実行モードでテストを実行すると、テストが実行された後、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、テス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、テス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テストを実行するには、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、1243 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テストを実行できません。
- ▶ アプリケーションが変更されていて、QuickTest によってオブジェクトを認識できない場合に、メンテナンス実行ウィザードが起動される場合があります。QuickTest がオブジェクトの表示を待ち始めてからオブジェクトが見つからないと判断するまでの時間を短くしたい場合があります。このような場合は、[テストの設定] ダイアログ・ボックスの [実行] 表示枠でオブジェクト同期化タイムアウトを変更できます。指定したタイムアウトが、アプリケーション内のオブジェクトを読み込むのに十分な長さであることを確認してください。詳細については、1259 ページ「テストのための実行設定の定義」を参照してください。

メンテナンス実行モードの終了後、その設定を前の通常のテスト実行時の値に戻したい場合があります（QuickTest の標準設定値は 20 秒です）。

- ▶ 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、167 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テストを更新した後は、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。

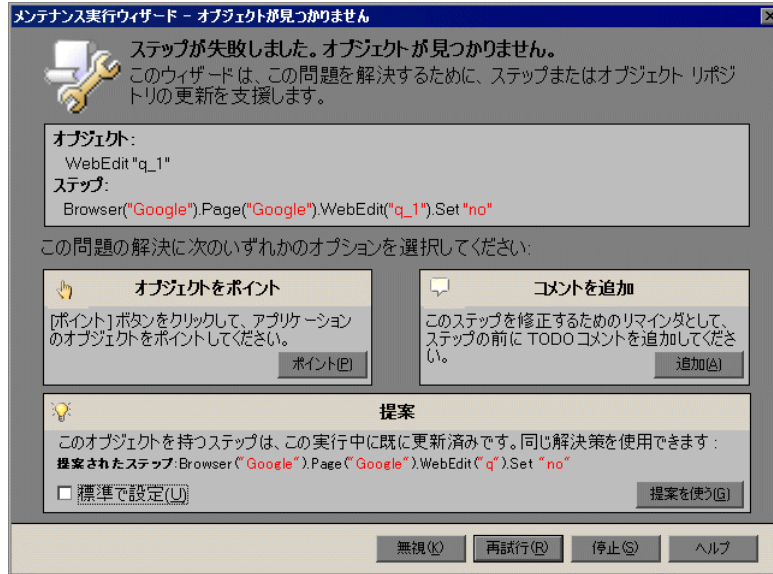
テストをメンテナンス実行モードで実行するには、次の手順を実行します。

- 1 テストを開き、[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、958 ページ「[実行] ダイアログ・ボックス：[結果の場所] タブ」および 960 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 33 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 44 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

テス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。

- ▶ 1104 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」
- ▶ 1106 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」
- ▶ 1101 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。

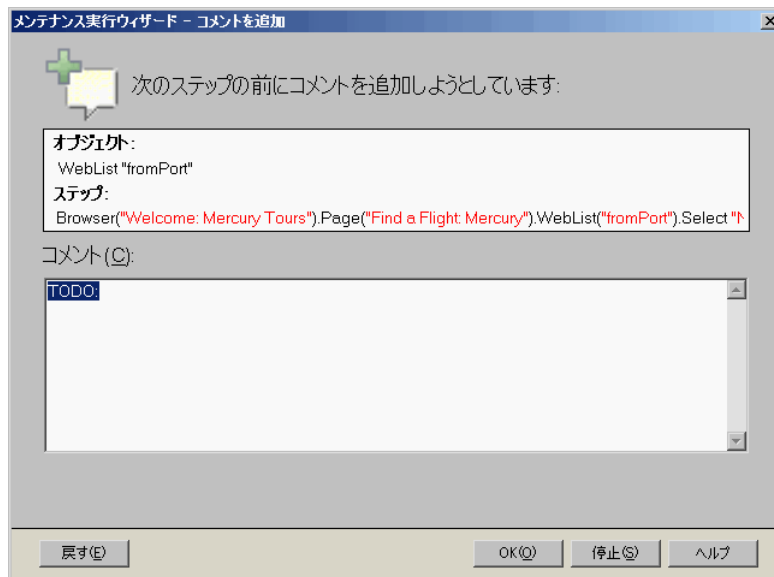
- ▶ **[無視]** : テストの現在のステップを無視し、テストの残りに対してメンテナンス実行ウィザードの実行を継続します。これは、問題が QuickTest のテストではなく、テスト中のアプリケーションにある場合に使用できます。

注： [無視] をクリックする前に、アプリケーションがテス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、テス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、テス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、またはテス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、テスト内に、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	<p>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、769 ページ「正規表現の概要と使い方」を参照してください。</p> <p>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。</p>

[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：

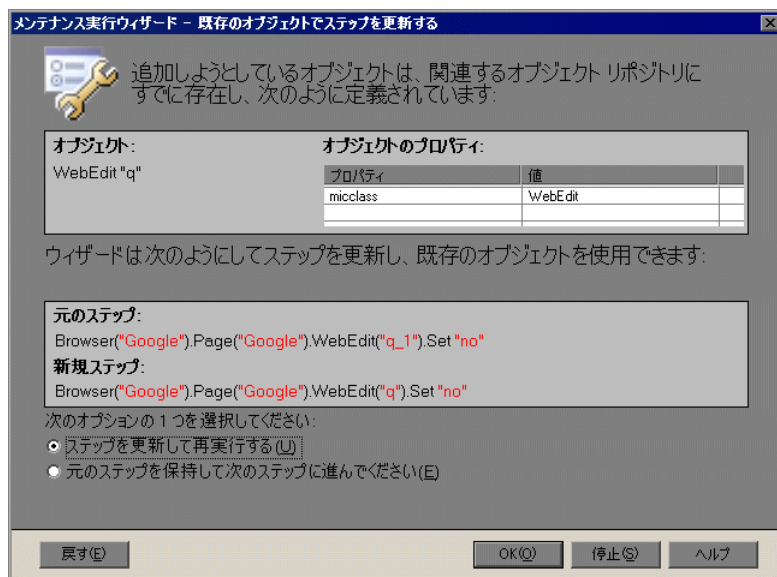
- ▶ 指定したオブジェクトがオブジェクト・リポジトリ内のオブジェクトとは異なる親オブジェクトとプロパティ値を持っている場合は、[オブジェクトプロパティ値を変更]画面が2回開きます。最初の画面では、オブジェクト・リポジトリ内のオブジェクトの親オブジェクトを更新して、指定したオブジェクトの親オブジェクトと一致させることができます。2つ目の画面では、オブジェクト・リポジトリ内のオブジェクトを更新して、指定したオブジェクトと一致させることができます。
 - ▶ メンテナンス実行ウィザードでは、ローカル・オブジェクト・リポジトリのみが変更されます。新しいオブジェクトを共有オブジェクト・リポジトリに含めるには、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、244 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、テス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、テス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。

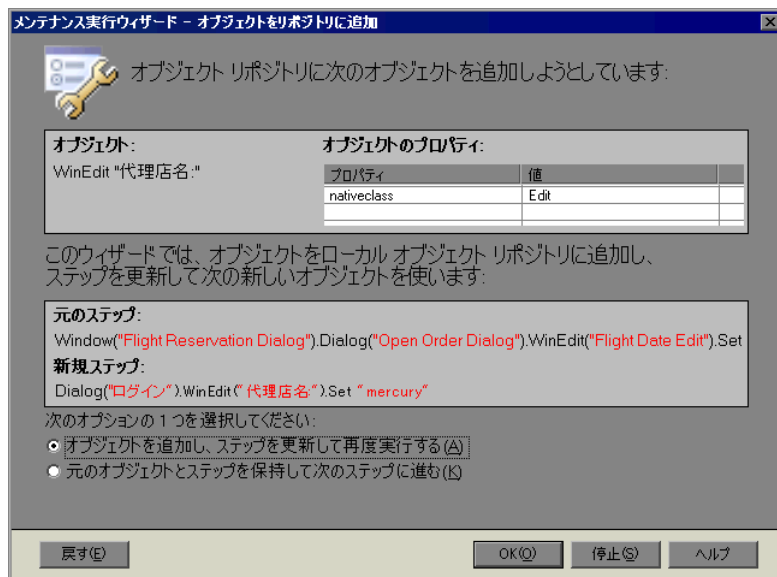
画面の下部には、**[戻す]** ボタンが表示されます。このボタンをクリックすると、**[オブジェクトが見つかりません]** 画面に戻り、アプリケーション内の別のオブジェクトを指定したり、このステップに対する別の措置を選択したりできます。

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、テス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：

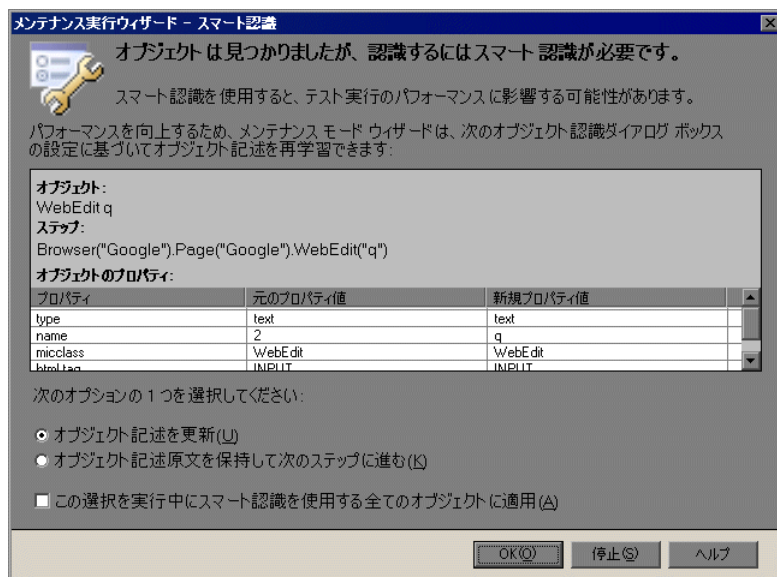
- ▶ メンテナンス実行ウィザードでは、ローカル・オブジェクト・リポジトリのみが変更されます。新しいオブジェクトを共有オブジェクト・リポジトリに含めるには、オブジェクト・リポジトリ・マネージャを使用します。詳細については、244 ページ「結合操作の実行」を参照してください。
- ▶ メンテナンス実行ウィザードでは、テストから元のステップは削除されません。元のステップはコメントに変換され、更新されたステップがその下に追加されます。

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、テス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、123 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、テス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。

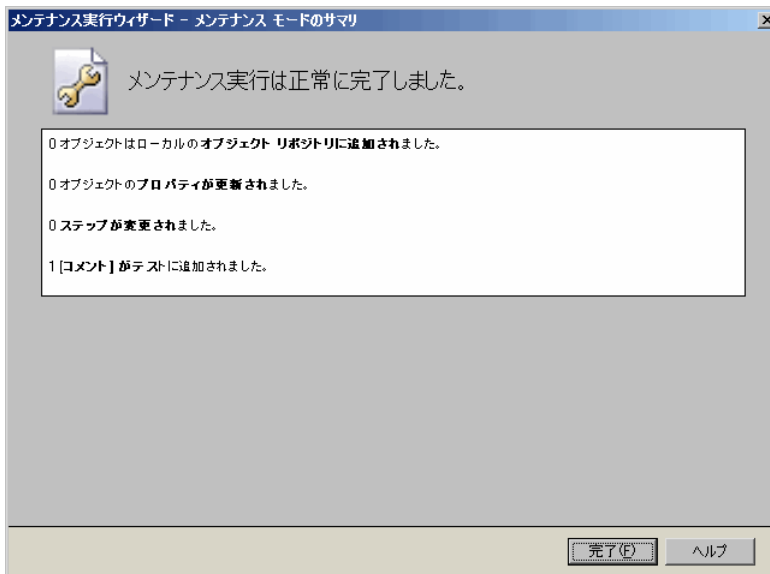
セクション	説明
[オブジェクト]	認識するのにスマート認識メカニズムを必要としたアプリケーション内のオブジェクト。
ステップ	オブジェクトが参照されるテストのステップ。
[オブジェクトのプロパティ]	<p>[プロパティ]：古いオブジェクト記述および新しいオブジェクト記述のプロパティのリスト。</p> <p>[元のプロパティ値]：[プロパティ] カラムのプロパティの元の値。値を持たないプロパティは、元のオブジェクト記述の一部ではありません。</p> <p>[新規プロパティ値]：[プロパティ] カラムのプロパティの新しい値。</p>

[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、テス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テス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、969 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 44 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テストの更新

更新モードでテストを実行すると、QuickTest によってテストが実行され、テスト・オブジェクト記述、ActiveScreen の画像と値、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。テストを保存すると、更新されたデータが以降の実行に使用されます。

QuickTest によるテストの更新時には、選択した実行オプションに従って、テストの反復が 1 回だけと、テスト内の各アクションの反復が 1 回実行されます。アクションの詳細については、第 15 章「アクションを使った作業」を参照してください。

注：

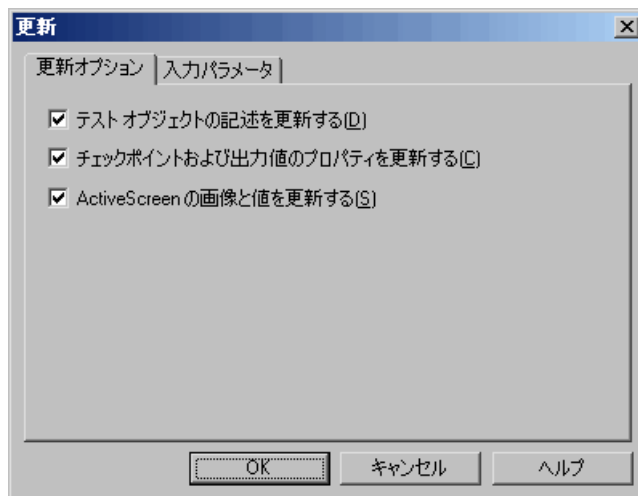
- ▶ 更新モードでテストを実行した場合、データ・テーブルのデータや環境変数など、パラメータ化された値は更新されません。パラメータ化された値および環境変数の詳細については、第 24 章「値のパラメータ化」を参照してください。更新モードでは、オブジェクト・リポジトリ内の既存のオブジェクト記述のプロパティ値は変更されません。オブジェクトのプロパティ値をアプリケーションと一致するように修正するには、メンテナンス実行モードを使用します。詳細については、1094 ページ「メンテナンス実行ウィザードを使ったテストの実行」を参照してください。
 - ▶ QuickTest によるテストの更新時には、更新対象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、次回テス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テストを更新した後は、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。

▶ 実行 ▾

- 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，1115 ページ「[更新オプション] タブについて」および 960 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。テストの更新中は，ステータス・バーで「更新」というテキストが点滅します。

テストが実行され，テスト・オブジェクト記述，ActiveScreen 情報，チェックポイントの期待値（それぞれユーザの選択に応じて）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 33 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 44 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。

次に例を示します。

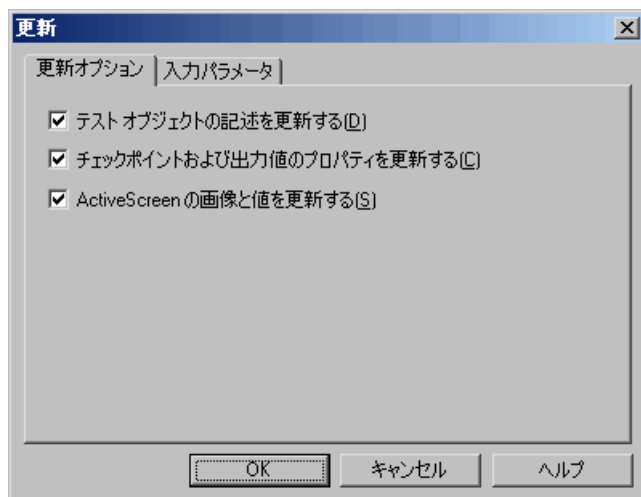
ステップ名: Welcome: Mercury Tours-記述の更新

ステップ 完了

オブジェクト	詳細	結果	時間
Welcome: Mercury Tours-記述の更新	テスト オブジェクトの以前の記述: Mercury Class = Browser CreationTime = 0	完了	2008/03/11 - 11:01:16
	テスト オブジェクトの新しい記述: Mercury Class = Browser		

[更新オプション] タブについて

[更新オプション] タブでは、テストのどの側面（テスト・オブジェクト記述、チェックポイントの期待値、ActiveScreen の画像と値、あるいはそのすべて）を更新するかを指定できます。テストを保存すると、更新されたテス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（可能な場合は、バージョン・コントロールをサポートした Quality Center プロジェクトにチェック・インする）、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）を使用してテス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。テス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英語版を対象にテス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フランス語版でも同じテス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テストを正しく実行できるようになります。

ヒント：正常に実行されるテストがあるが、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] を選択してテストを実行すると、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・テストで使用される 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。
 - ▶ テストに WinRunner テストへの呼び出しが含まれ、テストと期待結果フォルダの両方の書き込み権限を持っている場合は、**[チェックポイントのプロパティを更新する]**を選択すると、WinRunner テストのチェックポイントの期待値も更新されます。WinRunner テストを更新しない場合は、WinRunner テストを呼び出す行をコメント・アウトできます。WinRunner テストの呼び出しの詳細については、1502 ページ「WinRunner テストの呼び出し」を参照してください。コメント行の詳細については、818 ページ「コメントの追加」を参照してください。
 - ▶ 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、523 ページ「ビットマップの検査」を参照してください。
-
- ▶ **[ActiveScreen の画像と値を更新する]**：テストを記録した時点以降にアプリケーションに生じた可能性のある変更をすべて反映する場合や、ActiveScreen が想定どおりに表示されない場合に、ActiveScreen の画像とプロパティ値が更新されます。たとえば、テストを記録してからアプリケーションのダイアログ・ボックスが変更されたとします。その場合、テストを更新して、ActiveScreen のダイアログ・ボックスの外観とそのプロパティを更新できます。

また、このオプションを使用して、ActiveScreen に保存および表示される情報の量を増加または減少させることができます。[キャプチャのレベル] スライダーを変更し（[ツール] > [オプション] > [ActiveScreen] ノード）、**[ActiveScreen の画像と値を更新する]** チェック・ボックスを選択した状態で更新モードでテストを実行します。QuickTest によって、新しい設定に基づいて ActiveScreen で保存および表示される情報の量が更新されます。詳細については、1229 ページ「ActiveScreen オプションの設定」を参照してください。

第 VIII 部

QuickTest IDE を使用した作業

第 37 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 (1123 ページ)
- ▶ ツールバーとメニューのカスタマイズ (1133 ページ)
- ▶ 複数のドキュメントを使った作業 (1146 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、テスト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、1132 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。

- ▶ 1124 ページ「表示枠の移動」
- ▶ 1129 ページ「表示枠の表示と非表示」
- ▶ 1131 ページ「ツールバーのフローティングとドッキング」
- ▶ 1131 ページ「QuickTest ウィンドウの標準のレイアウトの復元」
- ▶ 1132 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。

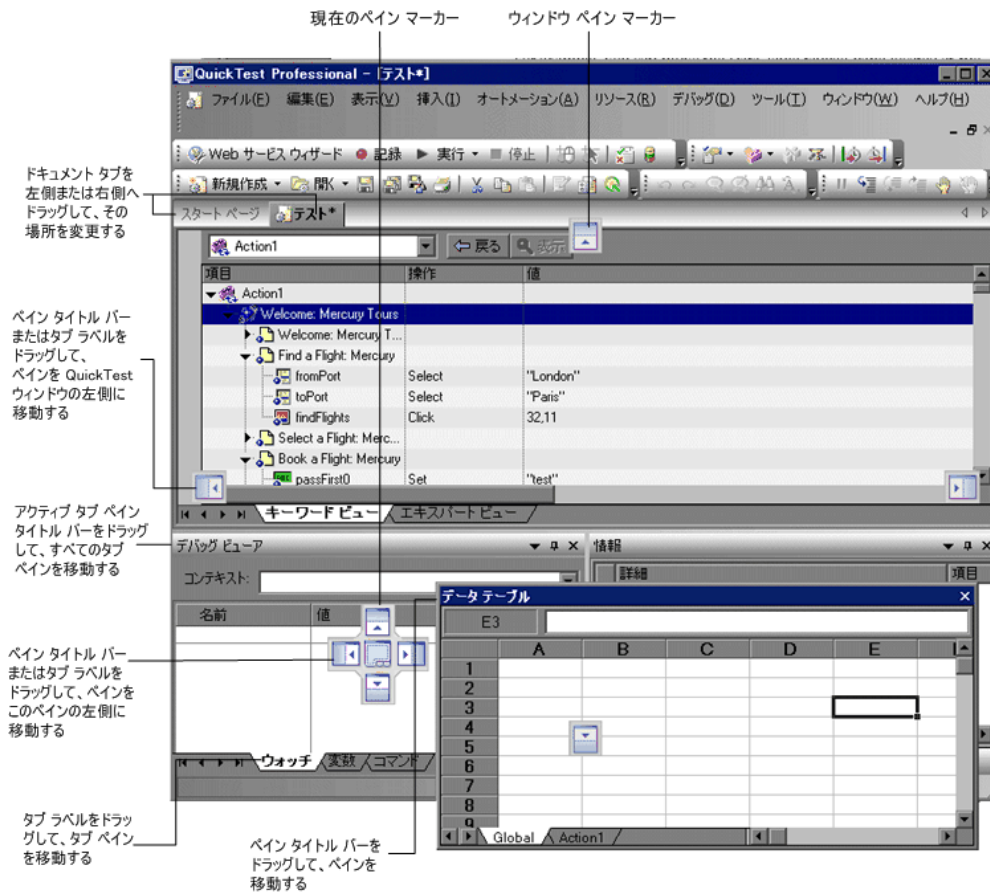
表示枠をドラッグしている間は、QuickTest ウィンドウ上にマーカが表示されます。いずれかのマーカの上にカーソルを置くと、そのマーカによって示される領域がグレー表示され、選択した位置に表示枠を移動した場合のウィンドウのレイアウトを確認できます。

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。

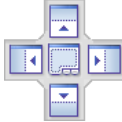




たとえば、左下にあるタブ形式の [データ テーブル] 表示枠を、新しい表示枠としてウィンドウの右上に移動できます。表示枠をドラッグすると、アクティブな表示枠および QuickTest ウィンドウの各辺にマークが表示されます。



ヒント：

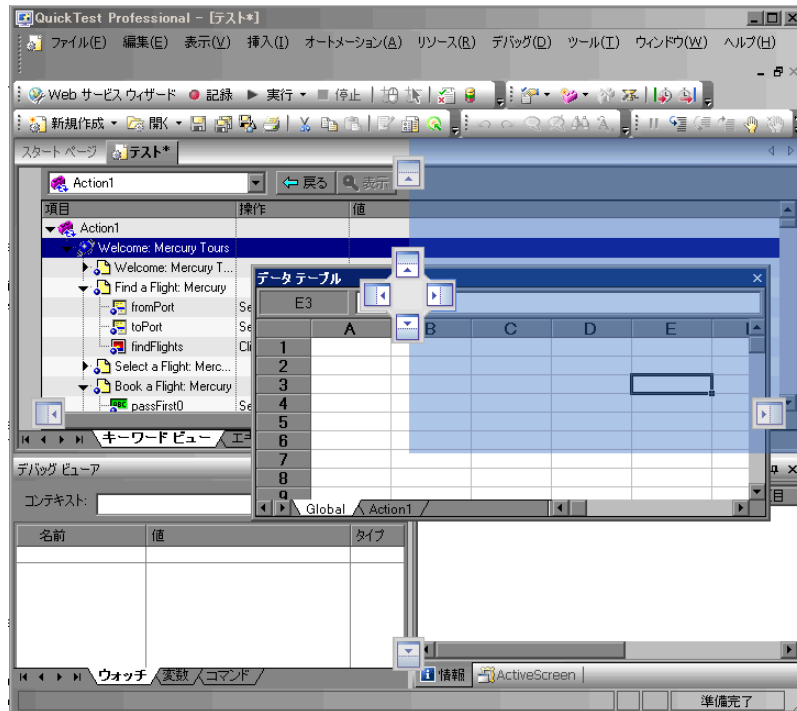
- ▶ タブ形式の表示枠を 1 つだけ移動するには、タブのラベルをドラッグします。タブ形式の表示枠をドラッグし始めると、そのタブが消え、タイトル・バーが表示されます。
- ▶ タブ形式の表示枠をすべて移動するには、アクティブなタブ形式の表示枠のタイトル・バーをドラッグします。

次のマーカが表示されます。

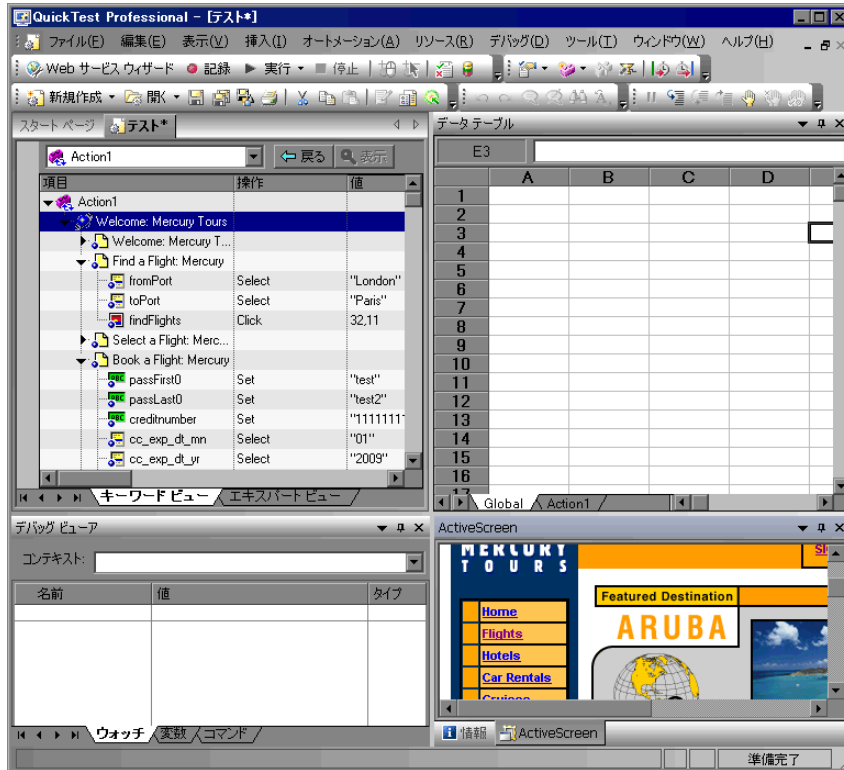
タイプ	マーカ	説明
現在表示枠 マーカ		次のことが可能です。 <ul style="list-style-type: none"> ▶ マウス・ボタンを放したときに選択されていた矢印マーカに応じて、移動した表示枠を、アクティブな表示枠内の上下左右いずれかの側、または中央に、新しい表示枠として配置します。 ▶ 中央のマーカが選択されているときにマウス・ボタンを放して、移動した表示枠を、アクティブなウィンドウ内に新しいタブ形式の表示枠として配置します。 注： 中央のマーカは、移動する表示枠を、(ドキュメント表示枠を除く)既存の表示枠内にドラッグしている場合にのみ表示されます。
ウィンドウ表示 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。



- 次の図のように、[データ テーブル] をドラッグし、アクティブな表示枠の右矢印マーカー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。[データ テーブル] 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、1129 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、マーク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なります。





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
	<p>[メニュー] ボタンでは次を選択できます。</p> <ul style="list-style-type: none"> ▶ [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。
	<p>[自動非表示] ボタン：表示枠を非表示にします。</p> <p>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。</p> <p>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。</p>
	<p>[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、</p> <p>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。</p>
	<p>[閉じる] ボタンは、表示枠を閉じます。</p> <p>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。</p> <p>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。</p>

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 44 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。これにより、ActiveScreen を表示せずに、テストを記録しながら、ステップが追加されるのを見ることができます。テストの実行時には、多くの場合 QuickTest を最小化して、テス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。テス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 新規または既存のテストを開きます。
 - 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、1071 ページ「ブレークポイントの設定」を参照してください。
 - b テストを実行します。

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

- ▶ 1133 ページ「カスタマイズ・モード・オプション」
- ▶ 1135 ページ「[ボタンの表示] ダイアログ・ボックス」
- ▶ 1136 ページ「[カスタマイズ] ダイアログ・ボックス - [コマンド] タブ」
- ▶ 1139 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」
- ▶ 1141 ページ「[カスタマイズ] ダイアログ・ボックス - [ツール] タブ」
- ▶ 1143 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」
- ▶ 1145 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。

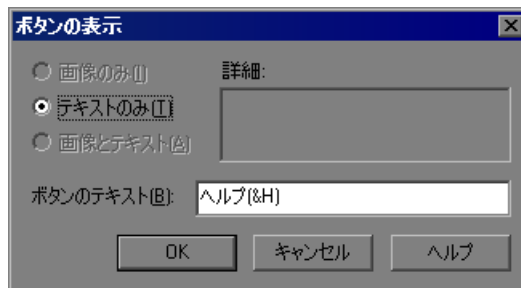
オプション	説明
[標準設定に戻す]	ボタンの標準設定に戻します。メニューに対しては、このオプションは無効です。
[ボタンの画像をコピーする]	ボタンの画像をクリップボードにコピーします。標準の画像がない項目に対しては、このオプションは無効です。
[削除]	<p>メニューまたはボタンを削除します。</p> <p>ボタンを復元するには、次の手順を実行します。</p> <ol style="list-style-type: none"> 1 通常モードのときにカスタマイズ・ツールバー・ボタン  をクリックします。 2 [ボタンの追加と削除] を選択します。 3 ボタンを復元するメニューを選択し、[ツールバーを元に戻す] を選択します。 <p>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。</p> <p>[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。</p> <p>注：メニュー・バーのカスタマイズは失われます。</p> <p>詳細については、1139 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。</p>
[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、1135 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	<p>次の手順のいずれかを実行します。</p> <ul style="list-style-type: none"> ▶ [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

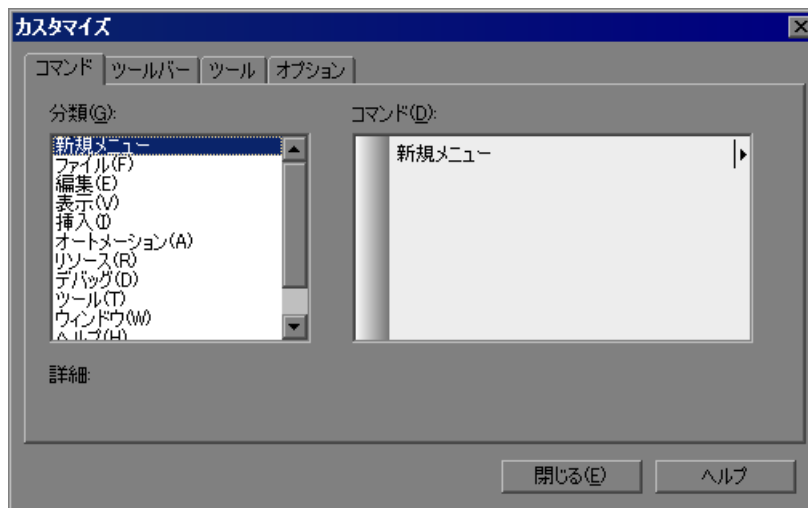
オプション	説明
画像のみ	ツールバーまたはメニューのボタンの画像が表示されます。標準の画像がない項目に対しては、このラジオ・ボタンは無効です。
テキストのみ	ツールバーまたはメニューのボタンのテキスト・ラベルが表示されます。
画像とテキスト	ツールバーまたはメニューのボタンあるいはメニューの画像とテキスト・ラベルが表示されます。標準の画像がない項目に対しては、このラジオ・ボタンは無効です。
詳細	ボタンの説明。
ボタンのテキスト	<p>ボタンまたはメニューのテキスト・ラベル。[テキストのみ] ラジオ・ボタンまたは [画像とテキスト] ラジオ・ボタンが選択されている場合に、ボタンのテキスト・ラベルを変更できます。</p> <p>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。</p>

[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	<p>次の手順のいずれかを実行します。</p> <ul style="list-style-type: none"> ▶ [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。

<p>重要情報</p>	<p>▶ 参照先</p> <ul style="list-style-type: none"> ▶ 1133 ページ「カスタマイズ・モード・オプション」 ▶ 1145 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」 <p>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。</p> <ol style="list-style-type: none"> 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。 <p>「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。</p>
<p>詳細情報</p>	<p>主要なタスク：1138 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。」</p>

[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 [テスト] は、[ファイル] カテゴリの個別のコマンド「新規作成：テスト」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。

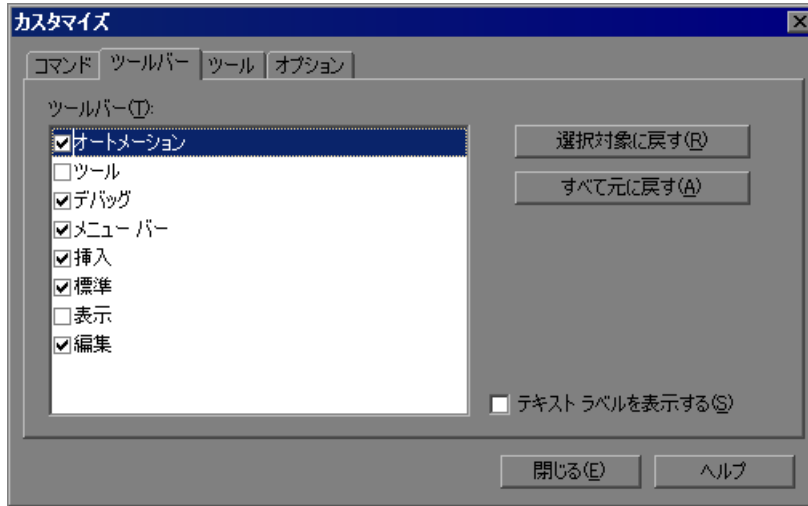
ツールバーまたはメニューにコマンドを追加するには、次の手順を実行します。

- 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、1135 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

<p>説明</p>	<p>次のことが可能です。</p> <ul style="list-style-type: none"> ▶ 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。
<p>アクセス方法</p>	<p>次の手順のいずれかを実行します。</p> <ul style="list-style-type: none"> ▶ [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。
<p>重要情報</p>	<ul style="list-style-type: none"> ▶ 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 <ul style="list-style-type: none"> ▶ 1133 ページ「カスタマイズ・モード・オプション」 ▶ 1145 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

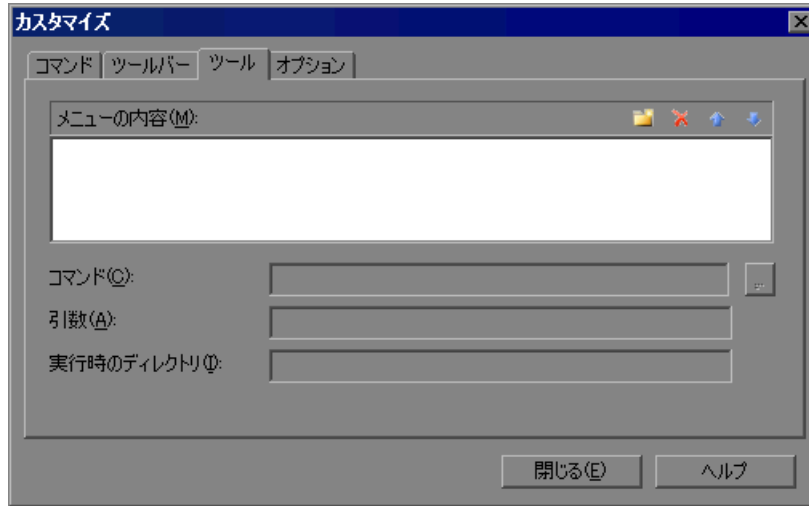
オプション	説明
ツールバー	QuickTest ウィンドウのツールバーのリスト。「メニュー・バー」も追加されています。ツールバーの表示と非表示を切り替えるには、チェック・ボックスを選択または選択解除します。
選択対象に戻す	<p>選択したツールバーまたはメニュー・バーの標準のレイアウトを復元します。</p> <p>ツールバーの標準のレイアウトを復元するには、次の手順も実行できます。</p> <ol style="list-style-type: none"> 1 カスタマイズ・ツールバー・ボタン  を右クリックします。メニュー・バーの場合、この方法は使用できません。 2 [ボタンの追加と削除] 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。 <p>[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。</p>

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	<p>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。</p> <p>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。</p> <p>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。</p>





[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	<p>次の手順のいずれかを実行します。</p> <ul style="list-style-type: none"> ▶ [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	<p>参照先</p> <ul style="list-style-type: none"> ▶ 1133 ページ「カスタマイズ・モード・オプション」 ▶ 1145 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。

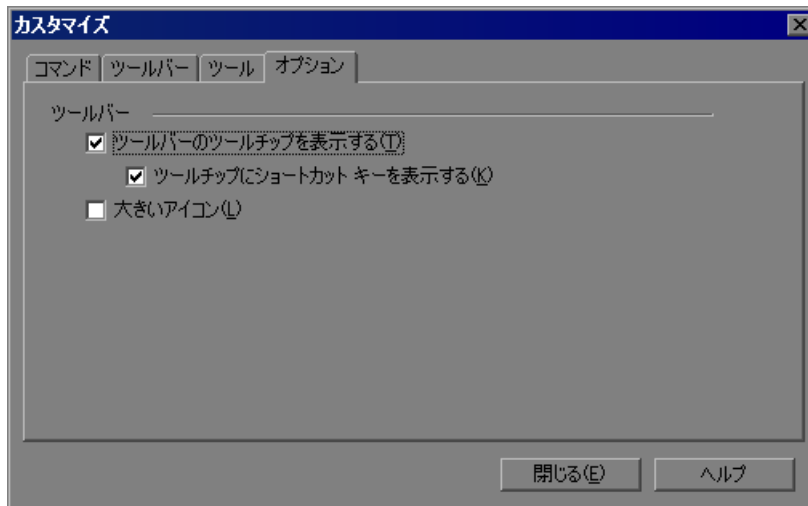
オプション	説明
コマンド	<p>[ツール] メニューに項目を追加するアプリケーション。参照ボタン  をクリックし、追加するアプリケーションに移動します。</p> <p>注: [コマンド] ボックスにはファイル名およびアプリケーションのパスのみ含まれるようにします。コマンド・ライン引数を追加する場合は、[引数] ボックスを使用します。</p> <p>ヒント: アプリケーションに関連付けられている、ファイル・システム内のドキュメントまたはそれ以外のファイルを指定できます (例: <code>c:\tmp¥a.txt</code>)。この場合、指定されたファイルが、関連付けられているアプリケーション (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。</p>
引数	<p>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。</p>
実行時のディレクトリ	<p>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。</p>

[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。
----	--

<p>アクセス方法</p>	<p>次の手順のいずれかを実行します。</p> <ul style="list-style-type: none"> ▶ [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。
<p>重要情報</p>	<p>参照先</p> <ul style="list-style-type: none"> ▶ 1133 ページ「カスタマイズ・モード・オプション」 ▶ 1145 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	<p>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。</p> <p>▶ [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。</p>
大きいアイコン	<p>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。</p>

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

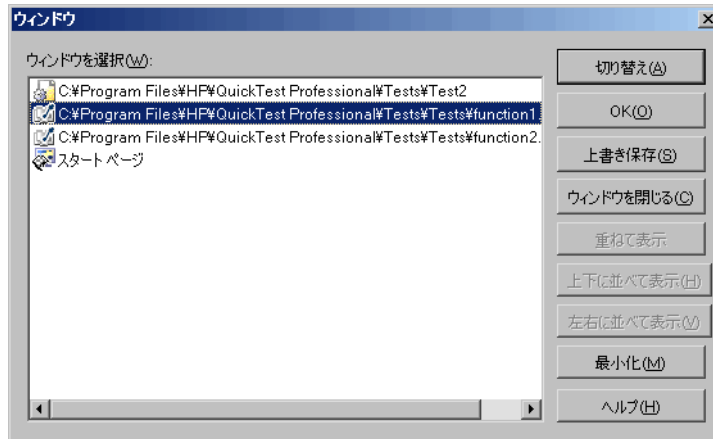
QuickTest では、1 つのテストを開いて作業できるのに加え、複数の関数ライブラリを同時に開いて操作できます。現在開いているテストに関連付けられているかどうかに関係なく、任意の関数ライブラリを開くことができます。

[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、開いているテストや、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 38 章


リソースの管理

QuickTest では、テストに関連付けられているリソースの管理を 1 つの表示枠で行うことができます。[リソース] 表示枠では、テストにおける関数ライブラリ、回復シナリオ、およびオブジェクト・リポジトリの関連付け、削除、開く操作、優先順位の変更、その他の管理が行えます。

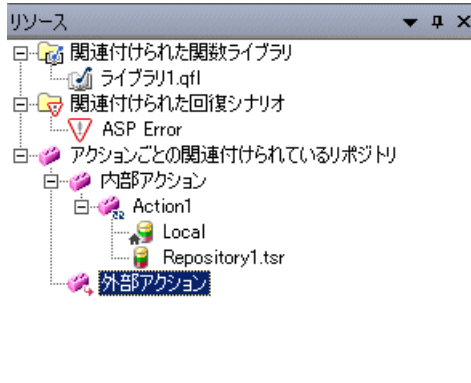
本章の内容

- ▶ [リソース] 表示枠 (1149 ページ)

[リソース] 表示枠

説明	テストに関連付けられているほとんどのリソースの追加、削除、管理、表示、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	テストおよびアクションは、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。 [リソース] 表示枠に、現在のテストのリソースが表示されます。関数ライブラリおよび回復シナリオは、リソースの種類に応じてグループ化されます。オブジェクト・リポジトリは、アクションに応じてグループ化されます。 [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	<p>テスト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。</p> <p>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】：【関数ライブラリを開く】ダイアログ・ボックスを開き、関数ライブラリをテストに関連付けることができます。</p> <p>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</p> <ul style="list-style-type: none"> ▶ 【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】：選択した関数ライブラリをテストから削除します。 ▶ 【上に移動】 または 【下に移動】：選択した関数ライブラリを、関連付けられている関数ライブラリの優先順位リストで上下に移動します。 <p>以下も参照してください：907 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」</p>

ノードの種類	説明
関連付けられている回復シナリオ	<p>テスト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。</p> <p>ルート・ノードのショートカット・メニュー・オプション</p> <p>〔回復シナリオの関連付け〕：〔回復シナリオの追加〕ダイアログ・ボックスを開きます。詳細については、1355 ページ「テストへの回復シナリオの追加」を参照してください。</p> <p>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</p> <ul style="list-style-type: none"> ▶ 〔回復シナリオのプロパティ〕：〔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。トリガ・イベントや回復操作などの回復シナリオのプロパティを読み取り専用で表示できます。また、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・ボックスを開くこともできます。詳細については、1352 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。 ▶ 〔リストから回復シナリオを削除〕：選択した回復シナリオをテストから削除します。 ▶ 〔上に移動〕または〔下に移動〕：選択した回復シナリオを、関連付けられている回復シナリオの優先順位リストで上下に移動します。 ▶ 〔回復シナリオの無効化〕または〔回復シナリオの有効化〕：選択した回復シナリオを無効または有効にします。 <p>以下も参照してください：1317 ページ「回復シナリオの定義と使用」</p>

ノードの種類	説明
アクションごとの 関連付けられているリポジトリ	<p>テストのすべてのアクション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。</p> <p>[リソース] 表示枠のこの部分のノードは、次の階層に従って編成されます。</p> <p>[アクションごとの関連付けられているリポジトリ] ノード</p> <p>[内部アクション] ノードおよび [外部アクション] ノード</p> <p>個別アクション</p> <p>個別アクションに関連付けられているローカル・オブジェクト・リポジトリおよび共有オブジェクト・リポジトリ（存在する場合）</p> <p>テストで呼び出されない場合でも、個別アクションには、テストに格納されているすべてのアクションが含まれます。</p> <p>アクション・ノードのショートカット・メニュー・オプション</p> <ul style="list-style-type: none"> ▶ アクションへのリポジトリの関連付け： [共有オブジェクトリポジトリを開く] ダイアログ・ボックスを開き、オブジェクト・リポジトリを選択したアクションに関連付けることができます。外部アクションに対しては、このオプションは無効になっています。 ▶ [アクションのプロパティ]： [アクションのプロパティ] ダイアログ・ボックスを開き、格納されたアクションのオプションを定義できます。アクションの名前を変更したり、アクションの説明を追加または変更したり、アクションを再利用可能または再利用不可能に設定することができます。外部アクションについては、データ・テーブル定義を設定できます。詳細については、445 ページ「アクションのプロパティの設定」を参照してください。 ▶ [アクションを削除]：テストからアクションを削除します。 <p>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</p> <ul style="list-style-type: none"> ▶ [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]：選択したオブジェクト・リポジトリをアクションから削除します。 ▶ [上に移動] または [下に移動]：選択したオブジェクト・リポジトリを関連付けられているリポジトリのリストで上下に移動すると、選択したオブジェクト・リポジトリの優先順位が変わります。 <p>以下も参照してください： 137 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」</p>

第 39 章

テス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テス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 (1155 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、テス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、テストにドラッグ・アンド・ドロップしたりできます。オブジェクトをアクションにドラッグ・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。関数をテストにドラッグ・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・オブジェクトをアクションにドラッグ・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。

関数をテストにドラッグ・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。その関数の引数をテス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。

- ▶ 185 ページ 「[オブジェクト リポジトリ] ウィンドウ」
- ▶ 230 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テストへのテスト・オブジェクトの追加」

[使用可能なキーワード] 表示枠には、テス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 40 章

QuickTest のタスクとコメントの管理

QuickTest では、テストのタスクを作成、管理できます。また、アクション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 (1159 ページ)
- ▶ [タスク] 表示枠 (1160 ページ)
- ▶ [タスク エディタ] ダイアログ・ボックス (1167 ページ)

タスクおよび TODO コメントを使った作業


QuickTest では、テスト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、現在開いているテストにリンクしているテスト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、アクション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。

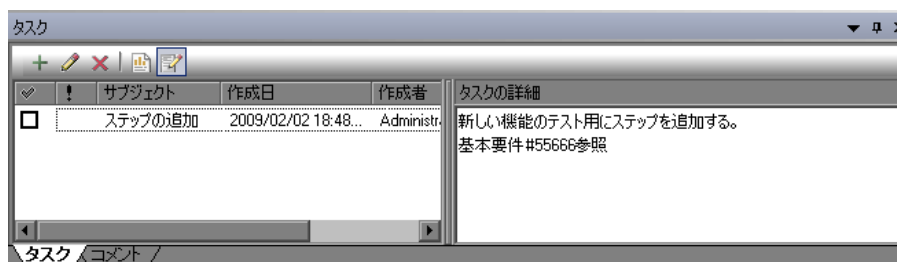
[タスク] 表示枠

<p>説明</p>	<p>テスト関連のタスクと TODO コメントを表示および管理できます。</p> <p>[タスク] 表示枠には次のタブがあります。</p> <ul style="list-style-type: none"> ▶ [タスク] タブ : テスト関連のタスクの作成および管理を行うことができます。詳細については、1161 ページ「[タスク] 表示枠 : [タスク] タブ」を参照してください。 ▶ [コメント] タブ : アクション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。詳細については、1164 ページ「[タスク] 表示枠 : [コメント] タブ」を参照してください。
<p>アクセス方法</p>	<ul style="list-style-type: none"> ▶ [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。 <p>注 : タスクが含まれているテストを開くと、[タスク] 表示枠が自動的に開きます。</p>
<p>詳細情報</p>	<p>概念的概要 : 1159 ページ「タスクおよび TODO コメントを使った作業」</p> <p>その他の関連トピック : 1167 ページ「[タスク エディタ] ダイアログ・ボックス」</p>

[タスク] 表示枠 : [タスク] タブ

説明	[タスク] タブには、現在開いているテスト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、テスト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	<p>概念的概要 : 1159 ページ「タスクおよび TODO コメントを使った作業」</p> <p>その他の関連トピック :</p> <ul style="list-style-type: none"> ▶ 1164 ページ「[タスク] 表示枠 : [コメント] タブ」 ▶ 1167 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、そのテストのために [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 (1167 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 (1167 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]	<p>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。</p> <p>取り得る値：</p> <p><input checked="" type="checkbox"/> タスク完了</p> <p><input type="checkbox"/> タスク未完了</p> <p>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。</p>
[優先度]	<p>タスクの重要度を示します。</p> <p>取り得る値：</p> <ul style="list-style-type: none"> ▶ 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。

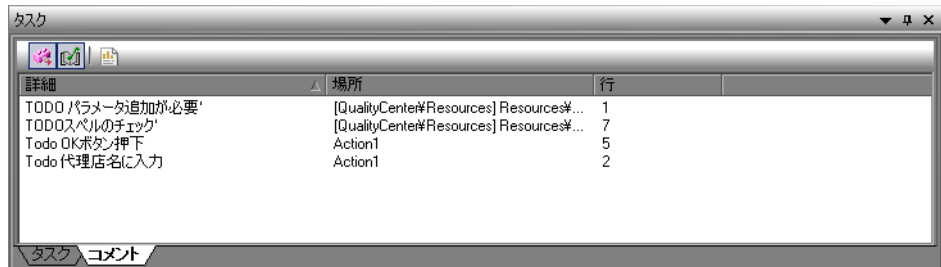
[タスク] タブのショートカット・メニュー・オプション

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	<p>[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。</p> <p>例 : to DO need to ask Sarah to add design steps</p> <p>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。</p> <p>次に関する TODO コメントを表示できます。</p> <ul style="list-style-type: none"> ▶ 現在のテストのローカル・アクション ▶ テストに関連付けられている外部アクション ▶ 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	<p>概念的概要 : 1159 ページ「タスクおよび TODO コメントを使った作業」</p> <p>その他の関連トピック :</p> <ul style="list-style-type: none"> ▶ 1161 ページ「[タスク] 表示枠 : [タスク] タブ」 ▶ 1167 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] タブにはテストのローカル・アクションのすべての TODO コメント・ステップが表示されます。また、テストで呼び出された外部アクション、および、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
	[外部アクション内のコメント]	外部アクションの 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
	[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（ローカル・アクションの 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。 <ul style="list-style-type: none"> ▶ 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アクションの名前、あるいは、関数ライブラリのパスが表示されます。
[行]	アクションまたは関数ライブラリの TODO コメントの行番号が表示されます。

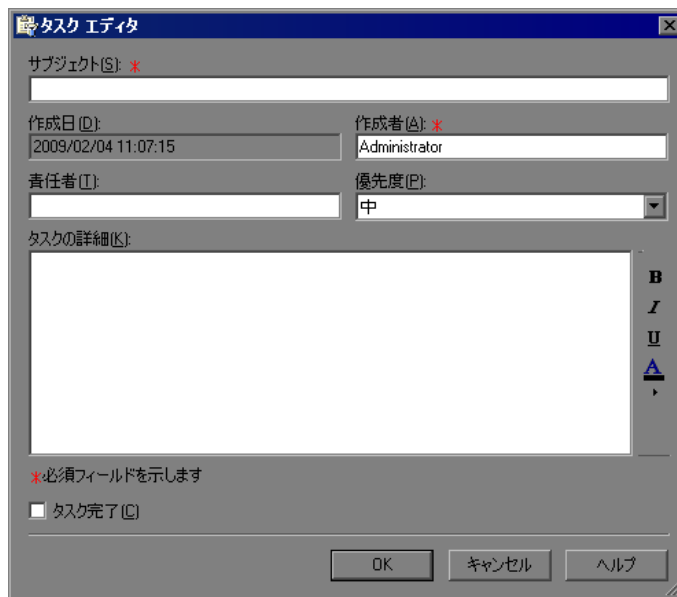
[コメント] タブのショートカット・メニュー・オプション

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]	アクション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、アクション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	<p>([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)</p> <p>[タスク] タブでは次のいずれかを実行します。</p> <ul style="list-style-type: none"> ▶ [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	<p>概念的概要 : 1159 ページ「タスクおよび TODO コメントを使った作業」</p> <p>その他の関連トピック : 1160 ページ「[タスク] 表示枠」</p>

[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： <input checked="" type="checkbox"/> タスク完了 <input type="checkbox"/> 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 41 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・リポジトリまたは欠落アクションの呼び出しなど）がテスト内にある場合や、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、またはテスト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（1169 ページ）
- ▶ 欠落している関数ライブラリの処理（1178 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（1180 ページ）
- ▶ 欠落している回復シナリオの処理（1181 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（1184 ページ）

欠落リソースの処理について






QuickTest では、テストを開くたびに、そのテスト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テストから削除したりできます。

[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。

項目	詳細
 Missing environment variables file: Env_var.xml	L:\QuickTest\Tests\Missing Resources\Libraries_Repos\Env_var.xml
 Missing Object Repository: Repository_1.tsr	L:\QuickTest\Tests\Missing Resources\Libraries_Repos\Repository_1.tsr
 Repository Parameters	Unmapped repository parameters
 Missing Recovery Scenario: RS_2	L:\QuickTest\Tests\Missing Resources\Libraries_Repos\RS2.qrs
 Missing Function Library: Common.bt	Common.bt
 Missing Action: ReturnHome	From test: L:\QuickTest\Tests\actionA

[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。

-  ▶ **[欠落アクション]** : 欠落しているアクションがテストに含まれている場合は、QuickTestによって、欠落アクションが含まれているテスト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、1172 ページ「欠落アクションの処理」を参照してください。
-  ▶ **[欠落している環境変数ファイル]** : テストによって外部ファイルからユーザ定義の環境変数が読み込まれようとしたときに、このファイルが見つからない場合は、QuickTestによって、欠落している XML ファイル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、1177 ページ「欠落している環境変数ファイルの処理」を参照してください。
-  ▶ **[欠落している関数ライブラリ]** : テスト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、1178 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]** : テスト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、1180 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]** : テスト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、1181 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]** : テスト内の少なくとも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、1184 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]** : テスト内のすべての欠落リソースが表示されます。
- ▶ **[アクション]** : 欠落アクションごとに、欠落アクションが含まれている各テストを探すためのパスを指定する1行が表示されます。
- ▶ **[環境変数ファイル]** : ユーザ定義の環境変数を格納するために使用する外部XMLファイルが表示されます。
- ▶ **[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。

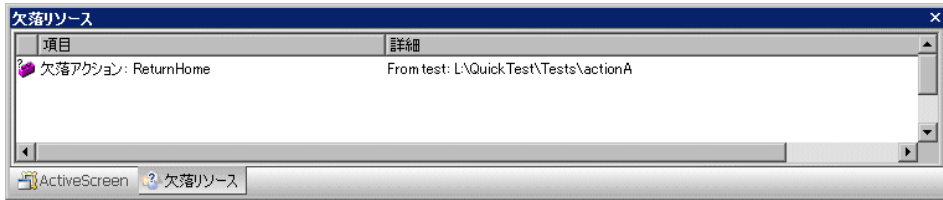
[欠落リソース] 表示枠は、選択したリソース・タイプに従ってフィルタ処理され、適用されたフィルタを示す次の表示が表示枠の下部に示されます。

✕ □ = 'オブジェクトリポジトリ'

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。

欠落アクションの処理

テストに含まれているアクションの呼び出しで、見つからないものがある場合は、それらのアクションが [欠落リソース] 表示枠に表示されます。



さらに、テストに含まれている特定のアクションに対する呼び出しがテスト・フローに含まれているが、そのアクションが見つからない場合は、QuickTest によってそのアクションが [欠落リソース] 表示枠に表示されます。たとえば、テスト作成時に新規の再利用可能なアクションの呼び出しを挿入したとします。その後、(464 ページで説明している) [アクションを削除] ダイアログ・ボックスの [選択したアクションの呼び出しを削除する] を選択して、そのアクションの呼び出しを削除したとします。この場合、アクションの呼び出しを削除しても、そのアクションはテストに参照されたままであるため、見つからない場合は [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、次の操作によって欠落アクションの問題を解決できます。

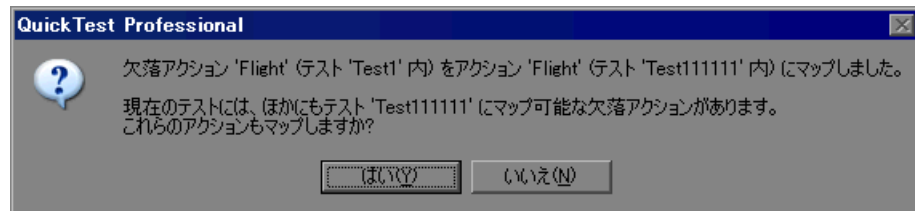
- ▶ 欠落アクションの特定
- ▶ 欠落アクションの削除

注：テストが読み取り専用形式で開かれている場合は、テスト内の欠落アクションの表示や割り当てを行うことはできません。

欠落アクションの特定

[欠落リソース] 表示枠では、テスト内の欠落しているアクションを探せます。

テストに複数の欠落アクションに対する呼び出しが含まれている場合、別のテストで欠落アクションを見つけると、次の例のように、QuickTest によって同じテスト内でほかの欠落アクションが見つかる場合があります。

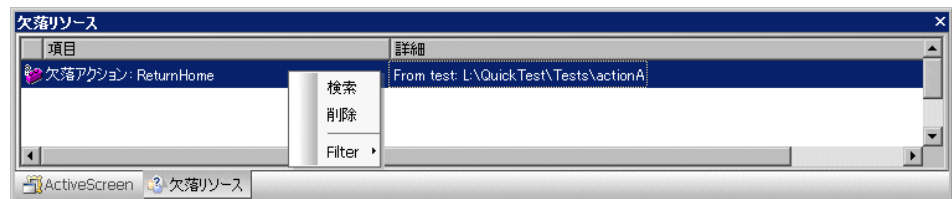


これは、たとえば、呼び出されるアクションが含まれているソース・テストの名前が変更されたり、別のフォルダに移動されたりした場合に起こることがあります。

これらのアクションの呼び出しを同時に見つけるように QuickTest に指定できます。また、欠落アクションの呼び出しを個別に処理することもできます。

欠落アクションを見つけるには、次の手順を実行します。

- 1 [欠落リソース] 表示枠で、検出するアクション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、検出するアクションをダブルクリックします。



[アクションの選択] ダイアログ・ボックスが開きます。



[アクションの選択] ダイアログ・ボックスが開くと、[テスト] ボックスに欠落アクションを含むテストの名前（QuickTest がソース・テストを識別できる場合）または「<現在のテスト>」が表示されます。

注：欠落アクションが別のテストから呼び出されたネストしたアクションである場合は、[選択] ボタンを使ってそのアクションを参照することはできません。その場合は、外部テストの中から欠落アクションを解決する必要があります。たとえば、(テスト A の) アクション AA が (テスト B の) アクション BB を呼び出し、アクション BB が (テスト C の) アクション CC を呼び出すとします。ここで、テスト A を開いてもアクション CC の呼び出しが見つからない場合は、テスト B を開いてアクション CC を見つけないと、欠落アクションを解決できません（テスト A の内部では解決できません）。



- 2 見つけ出したいアクションが含まれているテストを探すには、参照ボタンをクリックします。[アクション] ボックスには、選択したテストのすべての再利用可能なアクションが表示されます。

注：

- ▶ テストを選択すると、[**テスト**] ボックスの名前が [**コピー元のテスト**] に変わります。選択したテストに再利用可能なアクションが含まれている場合は、それらのアクションが [**アクション**] ボックスに表示されます。
 - ▶ [**テスト**] または [**コピー元のテスト**] ボックスには、Quality Center フォルダまたは相対パスを入力できます。相対パスを入力すると、[オプション] ダイアログ・ボックスの [フォルダ] 表示枠で指定されているフォルダの中で、QuickTest によってテストが検索されます。詳細については、1226 ページ「テストのフォルダ・オプションの設定」および 322 ページ「QuickTest での相対パスの使用」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 の絶対パスを指定してください。詳細については、1432 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。
-

- 3 [**アクション**] リストで、呼び出すアクションを選択します。アクションを選択すると、そのタイプ（再利用可能なアクション）および存在する場合は説明が表示されます。これによって、呼び出すアクションを識別できます。アクションの説明の詳細については、448 ページ「一般的なアクション・プロパティの設定」を参照してください。
 - 4 [**OK**] をクリックします。QuickTest によって、変更を反映してテストが更新され、アクションが [欠落リソース] 表示枠から削除されます。
-

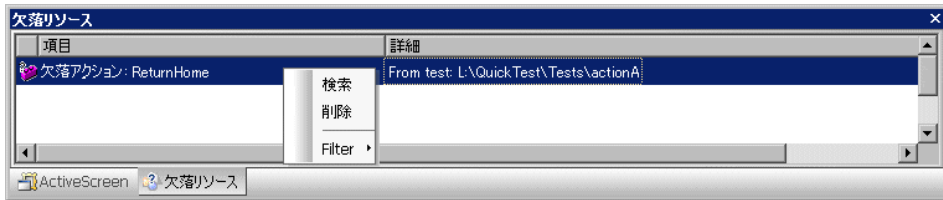
注： 同じテストの中で見つかる欠落アクションがほかにもテストに含まれている場合、メッセージ・ボックスが開き、それらのアクションも割り当てるかどうか確認を求められます。該当するアクションをすべて割り当てるには、[はい] をクリックします。指定したアクションのみを割り当てるには、[いいえ] をクリックします。

欠落アクションの削除

不要になった欠落アクションは、テストから削除できます。

欠落アクションを削除するには、次の手順を実行します。

削除するアクションを [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。

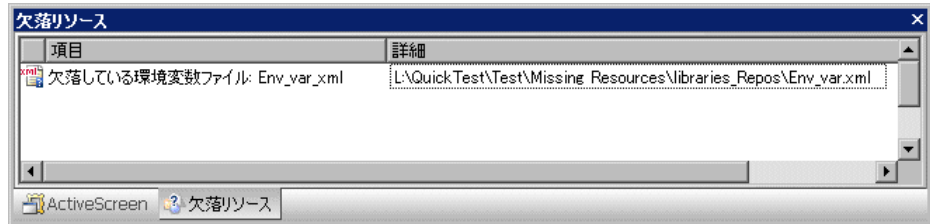


確認メッセージが表示されます。[OK] をクリックして欠落アクションを削除します。QuickTest によってアクションがテストおよび [欠落リソース] 表示枠から削除されます。

注：テストにほかにも同じテスト内に欠落アクションがある場合、パスが同じすべてのアクションを削除するかどうかを確認するメッセージ・ボックスが表示されます。同じパス内の欠落アクションをすべて削除するには、**[はい]** をクリックします。指定したアクションのみを削除するには、**[いいえ]** をクリックします。

欠落している環境変数ファイルの処理

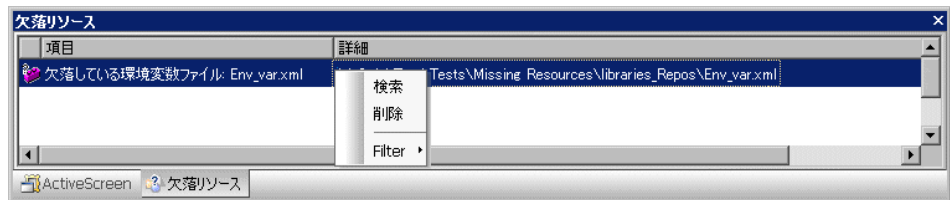
外部の環境変数ファイルを使用するテストを開くと、QuickTest によって、ファイルがアクセス可能であることが確認されます。外部の環境変数ファイルが見つからない場合は、テストを開いたときに、見つからない外部環境変数ファイルの名前とパスが [欠落リソース] 表示枠に表示されます。



[欠落リソース] 表示枠では、欠落している外部環境変数ファイル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。

欠落している外部環境変数ファイルを見つけるには、次の手順を実行します。

- 1 検出する欠落環境変数ファイル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、検出する欠落環境変数ファイルをダブルクリックします。

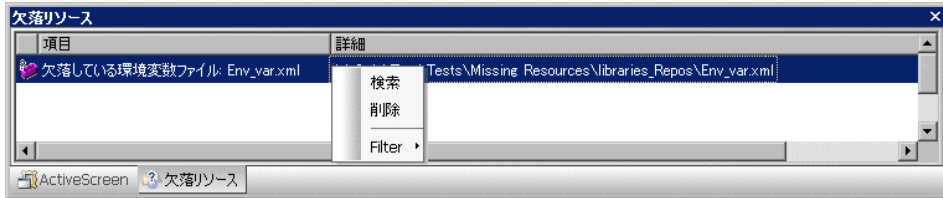


[環境変数ファイルの特定] ダイアログ・ボックスが開きます。

- 2 テストで使用する環境変数ファイルを参照して [開く] をクリックします。選択した環境変数ファイルがテストで使用され、欠落環境変数ファイルは [欠落リソース] 表示枠から削除されます。

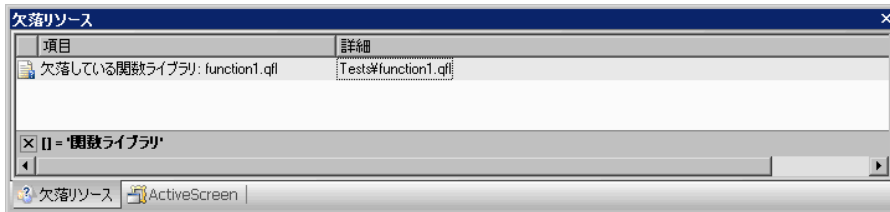
欠落環境変数ファイルを削除するには、次の手順を実行します。

削除する欠落環境変数ファイルを右クリックし、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]** をクリックして欠落環境変数を削除します。欠落環境変数ファイルが、テストと **[欠落リソース]** 表示枠から削除されます。



欠落している関数ライブラリの処理

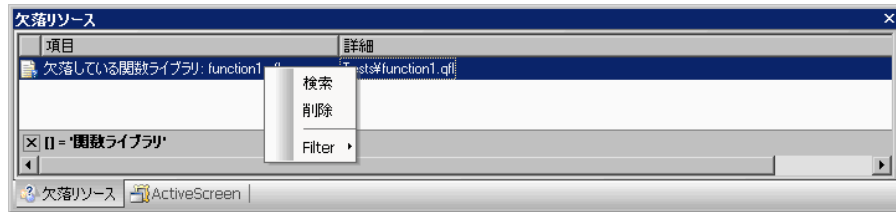
関数ライブラリが関連付けられているテストを開くと、QuickTest によって、指定したライブラリがアクセス可能であることが確認されます。関数ライブラリが見つからない場合は、テストを開いたときに、見つからない関数ライブラリの名前とパスが **[欠落リソース]** 表示枠に表示されます。



[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。

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。

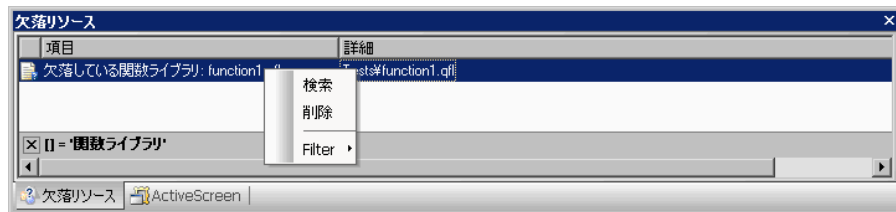


[関数ライブラリの検索] ダイアログ・ボックスが開きます。

- 2 テスト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テストに関連付けられ、[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。

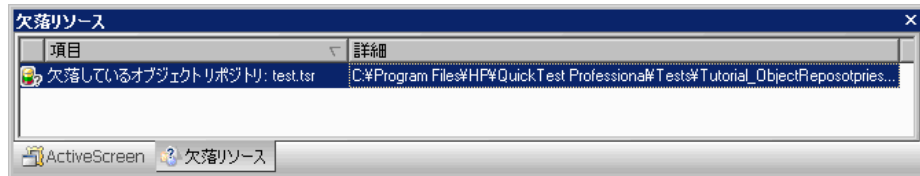
右クリックし、ショートカット・メニューから [削除] を選択します。確認メッセージが表示されます。[OK] をクリックして関数ライブラリを削除します。テストおよび [欠落リソース] 表示枠から欠落関数ライブラリが削除されます。



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。関数ライブラリがテストから削除されても、これらの関数に対する呼び出しはテス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・リポジトリをアクションに関連付けると、指定したリポジトリがアクセス可能かどうかを確認されます。また、テストを開くたびに、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、テストを開いたときに、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・リポジトリをテストに割り当てる必要があります。

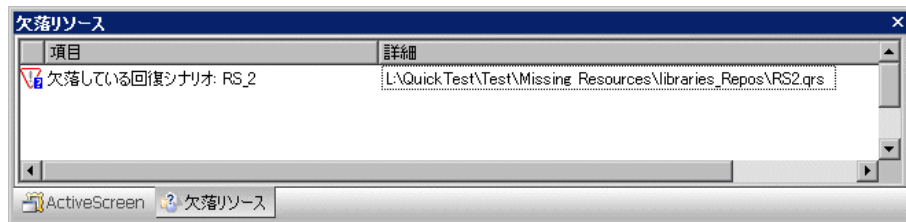
欠落しているオブジェクト・リポジトリが表示されている行を右クリックして [解決] を選択するか、または欠落しているオブジェクト・リポジトリが表示されている行をダブルクリックすると、[リポジトリの関連付け] ダイアログ・ボックスが開きます。[リポジトリの関連付け] ダイアログ・ボックスでは、1つ以上の共有オブジェクト・リポジトリをテスト内の1つ以上のアクションに関連付けることができます。また、選択したアクションまたはテスト内のすべてのアクションからオブジェクト・リポジトリの関連付けを削除することもできます。

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。

詳細については、201 ページ「共有オブジェクト・リポジトリの関連付けの管理」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テストを開くと、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、テストを開いたときに、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。



[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せん。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られてい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

テス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、またはテス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、233 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テストを開くと、[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。

欠落リソース	
項目	詳細
リポジトリ パラメータ	未割り当てのリポジトリ パラメータ

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。この値をデータ・テーブル・パラメータ、環境変数パラメータ、乱数パラメータ、テスト・パラメータ、またはアクション・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、テストまたは特定のアクション内の（値が割り当て済みおよび未割り当ての）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、204 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 42 章

データ・テーブルを使った作業

QuickTest では、データ・テーブルに格納されたデータを使って動作するステップを挿入し、実行できます。

本章の内容

- ▶ データ・テーブルを使った作業について (1185 ページ)
- ▶ グローバル・シートおよびアクション・シートを使った作業 (1187 ページ)
- ▶ データ・テーブルの保存 (1189 ページ)
- ▶ データ・テーブルの編集 (1190 ページ)
- ▶ Quality Center でのデータ・テーブル・ファイルの使用 (1200 ページ)
- ▶ データベースからのデータのインポート (1201 ページ)
- ▶ データ・テーブルでの数式の使用 (1204 ページ)
- ▶ データ・テーブルのスクリプト・メソッドの使用 (1208 ページ)

データ・テーブルを使った作業について

テストによって使用されるデータは「**設計時**」データ・テーブルに格納されます。これはステップを挿入、編集するときに [データ テーブル] 表示枠に表示されます。



[データ テーブル] 表示枠の表示と非表示を切り替えるには、[表示] > [データ テーブル] を選択するか、[データ テーブル] ツールバー・ボタンをクリックします。

データ・テーブルには、Microsoft Excel のスプレッドシートと同じ機能があり、セルでデータを格納、使用し、またセル内で数式を計算することもできます。

データ・テーブルは、QuickTest で提供されるものを使用できます。また、Microsoft Excel (.xls) ファイルを使用することもできます。データ・テーブルの場所の設定は、[テストの設定] ダイアログ・ボックスの [リソース] 表示枠 ([ファイル] > [設定] > [リソース] ノード) で行います。

データ・テーブル内の任意のセルでデータを操作するには、DataTable、DTSheet および DTPParameter ユーティリティ・オブジェクトを使用できます。これらのオブジェクトの詳細については、『HP QuickTest Professional Object Model Reference』（英語版）の **Utility** の項を参照してください。

注：データ・テーブルでは、複雑な式またはネストされた式の使用はサポートされていません。

テストにはデータ・テーブル・パラメータと出力値を挿入できます。テストでデータ・テーブル・パラメータまたは出力値を使用すると、指定したデータを使用して複数回実行する「**データ駆動**」テストまたはアクションを作成できます。実行の繰り返し、つまり**反復**のたびに、QuickTest によってデータ・テーブルの異なる値が使用されます。

実行セッション中、QuickTest によって**実行時**データ・テーブルが作成されます。これは、テストに関連するデータ・テーブルの実使用版です。実行セッション中、QuickTest によって [データ テーブル] 表示枠に実行時データが表示され、データ・テーブルに変更が加えられる様子を確認できます。

実行セッション終了時に、実行時データ・テーブルが閉じられ、[データ テーブル] 表示枠には格納されている設計時データ・テーブルが再度表示されます。実行セッション中に実行時データ・テーブルに入力されたデータは、テストとともに保存されません。実行時データ・テーブルの最終的なデータは、[テスト結果] ウィンドウの [**実行時データ テーブル**] に表示されます。実行時データ・テーブルの詳細については、1049 ページ「実行時データ・テーブルの表示」を参照してください。

ヒント：実行時データ・テーブルの結果データを保存しておく必要がある場合は、`DataTable.Export` ステートメントをテストの最後に挿入し、実行時データ・テーブルをファイルにエクスポートします。これによって、データ・テーブルを右クリックして表示されるメニューから **[ファイル] > [ファイルからインポート]** を選択して、データを設計時データ・テーブルにインポートできます。あるいは、テストの最初に `DataTable.Import` ステートメントを追加して、前回の実行セッションの最後にエクスポートされた実行時データ・テーブルをインポートできます。これら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。

グローバル・シートおよびアクション・シートを使った作業

テストを使って作業する際、データ・テーブルには、**[Global]** と **[Action]** の 2 種類のデータ・シートが表示されます。それぞれのシートにアクセスするには、データ・テーブルの下のそれぞれのタブをクリックします。

- ▶ データをテスト内のすべてのアクションで使用できるようにし、また多くのテスト反復を制御できるようにする場合は、**[Global]** タブに保存します。
- ▶ データ・テーブル・パラメータのデータを特定のアクションにのみ使用できるようにし、多くのアクション反復をするようにする場合は、データを **[Action]** のタブに保存します。

たとえば、サンプルの **Mercury Tours Web** サイトを対象としたテストを作成するとします。この場合、ログイン、フライト予約、ログアウトのそれぞれに対してアクションを 1 つずつ作成することになるでしょう。ユーザが 1 度サイトにログインし、5 名分のフライトを予約するテストを作成する必要があるとします。乗客に関するデータは 2 つ目のアクションにだけ関係するため、データはそのアクションに対応するアクション・タブに格納するようにします。

グローバル・シート

グローバル・シートには、テストの反復ごとにパラメータを置換するデータが格納されています。Arrivals というグローバル・パラメータを作成すると、グローバル・シートは次のようになります。

A3		Paris						
	A	B	C	D	E	F	G	
1	San Fransisco							
2	NewYork							
3	Paris							
4								
5								
6								
7								

アクション・シート

新しいアクションをテストに追加するたびに、新しい**アクション・シート**がデータ・テーブルに追加されます。アクション・シートのラベルには、対応するアクションの名前がそのまま自動的に付けられます。アクション・シートに格納されているデータは、対応するアクションのデータ・テーブル・パラメータにだけ関連しています。たとえば、テストに次のようなデータ・テーブルがある場合、**Purchase** アクション内でアクション・パラメータ・ステップを反復実行するとき、QuickTest によって [Purchase] シートに格納されているデータが使用されます。

B1								
	Departure	B	C	D	E	F	G	
1	NewYork							
2	Paris							
3	Los Angels							
4								
5								
6								
7								

グローバル・パラメータおよびアクション・パラメータの作成の詳細については、第 24 章「値のパラメータ化」を参照してください。

データ・テーブルの保存

データ・テーブルには、入力した値や数式だけでなく、テスト実行時にデータ・テーブル・パラメータを置き換えるために QuickTest によって使用される値が含まれます。テストを保存するとき、QuickTest によって、テストのデータ・テーブルが自動的に **.xls** ファイルとして保存されます。

テストを使って作業を行う場合、標準設定ではデータ・テーブルはテストと一緒に保存されます。データ・テーブルを別の場所に保存し、テスト実行時にこのデータ・テーブルを使用することができます。データ・テーブルの名前と場所は、[テストの設定] ダイアログ・ボックスの [リソース] 表示枠で指定します。

[テストの設定] ダイアログ・ボックスの詳細については、第 45 章「個別のテストのオプション設定」を参照してください。

指定した場所にデータ・テーブルを保存すると、次のような環境で便利です。

- ▶ 同じテストを異なる入力値のセットを使って実行する場合。たとえば、テストする言語ごとに、異なるデータ・テーブル・ファイルを用意してテストを実行することで、アプリケーションのローカライズ機能をテストできます。テストを実行するたびに異なる環境パラメータ・ファイルを使用することで、検査対象ユーザ・インタフェース文字列を言語ごとに変更することもできます。詳細については、第 24 章「値のパラメータ化」を参照してください。
- ▶ 異なるテストに関して、同じ入力情報が必要である場合。たとえば、使用するテストは異なるけれども使用するデータ・テーブル・ファイルが同じである、同じアプリケーションの Web バージョンと標準 Windows バージョンをテストできます。

注：外部ファイルをデータ・テーブルとして選択した場合、外部データ・テーブルのカラム名はテストのパラメータ名と一致し、外部データ・テーブルのシートはテストのアクション名と一致する必要があります。

データ・テーブルの編集

データ・テーブルの情報は、テーブル・セルに直接入力して編集できます。データ・テーブルは、Microsoft Excel スプレッドシートと同じように使います（数式をセルに挿入するなど）。また、Microsoft Excel、タブ区切りのテキスト・ファイル（.txt）、または ASCII 形式で保存されたデータをインポートすることもできます。

サポートされている Microsoft Excel のバージョンの詳細については、『**HP QuickTest Professional 最初にお読みください**』を参照してください。



[データ テーブル] 表示枠の表示と非表示を切り替えるには、[表示] > [データ テーブル] を選択するか、[データ テーブル] ツールバー・ボタンをクリックします。

	departure	arrival	C	D	E	F	G
1	Acapulco	NewYork					
2	NewYork	Paris					
3	London	Frankfurt					
4							
5							
6							
7							

テーブルの各行は、テストまたはアクションの特定の反復で、パラメータ化された引数に対して送信される値のセットを表します。アクションが実行される反復回数の定義は、[アクション呼び出しプロパティ] ダイアログ・ボックスの [実行] タブ ([編集] > [アクション] > [アクション呼び出しプロパティ] > [実行] タブ)で行います。テストが実行される反復回数の定義は、[設定] ダイアログ・ボックスの [実行] 表示枠 ([ファイル] > [設定] > [実行] 表示枠)で行います。

テーブルの各カラムは、パラメータ化された特定の引数に対する値のリストです。カラムのヘッダは、パラメータ名です。

また、データ・テーブル・パラメータとともに使用することを前提としていないカラム（カラムのヘッダにパラメータ名がないカラム）のセルにデータおよび数式を入力することもできます。

データ・テーブルを使用した作業に関するガイドライン

- ▶ データ・テーブルにデータを追加する場合、データの入力、上から下の行、左から右のカラムの順に行う必要があります。行またはカラムに何も入力せずに空けておくことはできません。たとえば、行 1 にデータが存在する場合、行 2 にデータを入力するまで、行 3 のセルにデータを入力することはできません。同様に、カラム A にデータが存在する場合、カラム B にデータを入力するまで、カラム C にデータを入力することはできません。
- ▶ データ・テーブルから返される値は常に文字列に変換されます。値を文字列以外に変換する場合は、CInt, CLng, CDbI など、VBScript の変換機能を使用できます。次に例を示します。

```
Window("Flight Reservation").WinComboBox("Fly From:").Select
  CInt(DataTable("ItemNumber", dtGlobalSheet))
```

- ▶ データ・テーブルのセルに内容を追加すると、行の下側のグリッド線がグレーから黒に変わります。[すべての行で実行する] オプションを使ってテストを実行すると（[ファイル] > [設定] > [実行] 表示枠、または [編集] [アクション] > [アクション呼び出しプロパティ] > [実行] タブで定義されます）、下側のグリッド線が黒の行ごとに反復を 1 回実行します。

[すべての行で実行する] オプションを選択したときに、特定の行に対して反復を実行したくない場合、データ・テーブルからその行全体を削除する必要があります。これを行うには、行を選択し、テーブルを右クリックして、データ・テーブルのショートカット・メニューから [編集] > [削除] を選択します（あるいは、CTRL キーを押しながら K キーを押します）。(削除すると、下側のグリッド線が黒からグレーに戻ります)。テーブルを右クリックして表示されるメニューから [編集] > [クリア] オプションを使用するか（または CTRL+X キー）、セルを選択してキーボードの **Delete** キーを押すと、データはセルから削除されますが、行は削除されず、黒の線も残ります。これは、行にデータが存在しなくても、QuickTest によってこの行が反復実行されることを意味します。

データ・テーブルの仕様

データ・テーブルを使った作業の主な制限事項は次のとおりです。

- ▶ **最大ワークシート・サイズ** : 65,536 行, 256 カラム
- ▶ **カラム幅** : 0 から 255 文字
- ▶ **テキストの長さ** : 16,383 文字
- ▶ **数式の長さ** : 1024 文字
- ▶ **数値の精度** : 15 桁
- ▶ **整数の最大値** : 9.999999999999999E307
- ▶ **整数の最小値** : 1E-307
- ▶ **負の数の最大値** : -1E-307
- ▶ **負の数の最小値** : -9.999999999999999E307
- ▶ **ワークブックごとの名前の最大数** : 使用できるメモリによって制限されます。
- ▶ **最長の名前** : 255
- ▶ **最短の数式文字列** : 255
- ▶ **テーブルの最大数** (ワークブック) : システム・リソース (ウィンドウとメモリ) によって制限されます。
- ▶ データ・テーブルでは, 色の使用と書式設定は行えません。
- ▶ データ・テーブルでは, 複雑な数式およびネストされた数式はサポートされていません。
- ▶ データ・テーブルでは, コンボ・ボックスやリスト・セル, 条件付き書式などの特別なセル書式はサポートされていません。

カラム名の変更

カラム・ヘッダのセルをダブルクリックすると、パラメータのカラム名を変更できます。[パラメータ名の変更] ダイアログ・ボックスで、新しいパラメータ名を入力します。このパラメータ名は、テスト内で一意でなければなりません。名前には、英字、数字、カンマ、アンダスコアを使用できます。ただし、最初の文字は、英字またはアンダスコアである必要があります。

注： テーブル内のカラム名を変更したら、テスト内の対応するパラメータに対して定義されている名前も変更する必要があります。

データ・テーブル・メニュー・コマンドの使用

次のデータ・テーブル・メニュー・コマンドを使用すると、データ・テーブルのデータを編集できます。データ・テーブル・メニューを開くには、セル、行のヘッダ、またはカラム・ヘッダを右クリックします。

次のメニューを使用できます。

- ▶ ファイル (1194 ページを参照してください)
- ▶ シート (1195 ページを参照してください)
- ▶ 編集 (1195 ページを参照してください)
- ▶ データ (1197 ページを参照してください)
- ▶ 書式 (1197 ページを参照してください)

【ファイル】メニュー

【ファイル】メニューでは、次のコマンドを使用できます。

【ファイル】の コマンド	説明
【ファイルから インポート】	<p>既存の Microsoft Excel ファイルまたはタブ区切りのファイルをデータ・テーブルにインポートします。このコマンドは、選択した Microsoft Excel ファイルのすべてのシートをインポートします。既存の Microsoft Excel ファイルからシートを 1 枚だけインポートするには、下記のように [シート] > [インポート] > [ファイルから] コマンドを使用します。</p> <p>注：</p> <ul style="list-style-type: none"> ▶ インポートするテーブル・ファイルは、テーブルのすべてのシートのデータをすべて置き換えます。また、各 Microsoft Excel シートの最初の行は、対応するデータ・テーブル・シートのカラム・ヘッダに置き換えられます。したがって、Microsoft Excel シートの最初の行がテストのパラメータ名に正しく一致しており、ファイルには現在のデータ・テーブルと少なくとも同数のシートが含まれていなければなりません。 ▶ コンボ・ボックスやリスト・セル、条件付き書式などの特別なセル書式が含まれる Microsoft Excel テーブルをインポートする場合、書式はインポートされず、セルは固定値とともにデータ・テーブルに表示されます。
【エクスポート】	<p>指定された Microsoft Excel (.xls) ファイルにテーブルをエクスポートします。</p>
【印刷】	<p>テーブル全体または選択したシートを印刷します。</p>

[シート] メニュー

[シート] メニューでは、次のコマンドを使用できます。

[シート] の コマンド	説明
[インポート] > [ファイルから]	タブ区切りのテキスト・ファイルまたは既存の Microsoft Excel ファイルからの 1 シートがテーブルにインポートされます。 注： インポートするシートは、テーブルで現在選択されているシートのデータをすべて置き換え、Excel シートの最初の行は、対応するデータ・テーブル・シートのカラム・ヘッダを置き換えます。したがって、Microsoft Excel シートの最初の行がテストのパラメータ名に正しく一致していなければなりません。
[インポート] > [データベース から]	指定されたデータベースから現在のシートにデータをインポートします。
[エクスポート]	指定された Microsoft Excel (.xls) ファイルに、データ・テーブルの現在のシートをエクスポートします。

[編集] メニュー

[編集] メニューでは、次のコマンドを使用できます。

[編集] の コマンド	説明
[切り取り]	テーブルの選択されている範囲を切り取り、クリップボードに格納します。
[コピー]	テーブルの選択されている範囲をコピーし、クリップボードに格納します。
[貼り付け]	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ます。
[値を貼り付け]	クリップボードに格納されている値を、現在テーブルで選択されている範囲に貼り付けます。値に適用されている書式はすべて無視されます。さらに、数式の計算結果だけが貼り付けられ、数式は無視されます。

【編集】の コマンド	説明
【クリア】	現在選択されている部分から書式または内容を消去します。書式のみ、内容（数式も含む）のみ、または書式と内容の両方を消去できます。
【挿入】	現在選択している位置に空のセルを挿入します。セルを挿入すると、その隣のセルは挿入したセルの分だけ位置がずれません。このオプションが使用できるのは、行のヘッダまたはカラム・ヘッダが選択されている場合のみです。
【削除】	現在選択されている行とカラムの全体を削除します。削除されたセルによってできたスペースを埋めるために、削除部分に隣接するセルがずらされます。このオプションが使用できるのは、行のヘッダまたはカラム・ヘッダが選択されている場合のみです。
【右方向へコピー】	選択された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します。
【縦方向にコピー】	選択された範囲内の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します。
【検索】	指定されたテキストを含むセルを検索します。テーブルの行ごとまたはカラムごとの検索、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だけの検索ができます。また、数式や値の検索も可能です。
【置換】	指定されたテキストを含むセルを見つけ、これを別のテキストに置き換えます。テーブルの行ごとまたはカラムごとの検索、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だけの検索ができます。また、数式や値の検索も可能です。見つかったテキストのすべてのインスタンスを置換することもできます。
【移動】	指定されたセルに移動します。移動先のセルがアクティブになります。セルのカラムや行の番号を入力する必要があります。

[データ] メニュー

[データ] メニューでは、次のコマンドを使用できます。

[データ] の コマンド	説明
[再計算]	テーブル内のすべての数式のセルを再計算します。
[並べ替え]	選択したセルを、行またはカラムおよびキーに従って、昇順または降順で並べ替えます。
[オートフィルの リスト]	[オートフィルのリスト] ダイアログ・ボックスが開きます (1198 ページを参照してください)。
[暗号化]	<p>選択したセルのテキストを暗号化します。暗号化されたデータは復号できません。</p> <p>また、パスワード・エンコーダを使って、任意のテキスト文字列を暗号化することもできます。これは、[エキスパートビュー] でメソッドの引数として暗号化した文字列を入力するのに便利です。詳細については、410 ページ「で使用するパスワードを暗号化できます。メソッド引数およびデータ・テーブル・セル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。</p>

[書式] メニュー

[書式] メニューでは、次のコマンドを使用できます。

[書式] の コマンド	説明
[一般]	一般書式に設定します。一般書式では、小数位は入力したとおりに表示されます。カンマは表示されません。
[通貨 (0)]	書式の設定をカンマ区切りで小数点を使わない通貨形式にします。QuickTest では、Windows の [地域のオプション] ダイアログ・ボックスで定義された通貨記号が使用されます。
[通貨 (2)]	通貨の書式に設定します。QuickTest では、Windows の [地域のオプション] ダイアログ・ボックスで定義された通貨記号が使用されます。
[固定]	固定の精度の書式に設定します。カンマ (桁区切り) を使用し、小数位は表示されません。

【書式】の コマンド	説明
【パーセント】	パーセントの書式に設定します。小数位は表示されません。数字を百分率で表示し、数字の後にパーセント記号 (%) が表示されます。
【分数】	1/2 など、分子 / 分母の形式の分数に書式を設定します。
【指数】	小数位 2 桁の指数表現の書式に設定します。
【日付 (ダイナミック)】	[月 / 日 / 年] の日付書式に設定します。
【午前 / 午後 時 : 分】	[午前 / 午後 時 : 分] の時間書式に設定します。
【ユーザ定義数】	ユーザ定義の数値の書式に設定します。このオプションにより、パーセンテージ、通貨、日付、時間などに関して、特別なユーザ定義書式を設定できます。

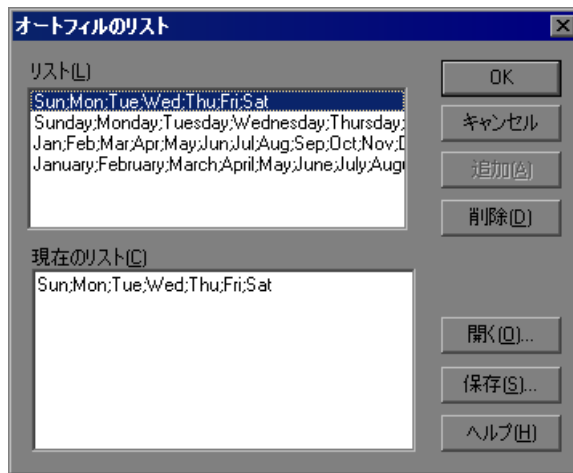
ショートカット・キーを使用してデータ・テーブル・メニュー・コマンドを実行することもできます。詳細については、47 ページ「QuickTest コマンドの実行」を参照してください。

【オートフィルのリスト】 ダイアログ・ボックス

説明	<p>オートフィルのリストの作成、編集、削除を行うことができます。オートフィルのリストには、月や曜日など、よく使われる一連のテキストが含まれています。</p> <p>オートフィルのリストを使用するには、次の手順を実行します。</p> <ol style="list-style-type: none"> 1 テーブルのセルに最初の項目を入力します。 2 カーソルをセルの右下の角からドラッグすると、オートフィルのリストに従い、QuickTest によって範囲内のセルが自動的に埋められます。
----	--

アクセス方法	データ・テーブルで右クリックし、 [オートフィルのリスト] を選択します。
詳細情報	<p>主要なタスク : 1185 ページ「データ・テーブルを使った作業について」</p> <p>その他の関連トピック : 1199 ページ「[オートフィルのリスト] ダイアログ・ボックスのオプション」</p>

[オートフィルのリスト] ダイアログ・ボックスの画像を次に示します。



[オートフィルのリスト] ダイアログ・ボックスのオプション

オプション	説明
[リスト]	プロジェクトで使用できるリスト。4つの標準設定のリストが含まれています。
[現在のリスト]	選択したリスト。このウィンドウは、新しいリストの作成に使用できます。新しいリストの各項目はセミコロンで区切ります。
[追加]	新しいリストを [リスト] ボックスに追加します。
[削除]	[リスト] ボックスからリストを削除します。

オプション	説明
[開く]	[ファイルを開く] 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、あらかじめ作成しておいたリストを参照できます。
[保存]	[名前を付けて保存] ダイアログ・ボックスが開きます。このダイアログ・ボックスを使って、新しいリストを保存できます。

Quality Center でのデータ・テーブル・ファイルの使用

Quality Center とデータ・テーブルを使って作業を行う場合は、[テストの設定] ダイアログ・ボックスの [リソース] 表示枠でデータ・テーブル・ファイルを指定する前に、Quality Center プロジェクトのテスト・リソース・モジュールにデータ・テーブル・ファイルを保存しておく必要があります。

新規または既存のデータ・テーブル・ファイルを Quality Center プロジェクトに追加できます。ファイル・システムの既存のデータ・テーブル・ファイルを Quality Center プロジェクトに追加すると、そのファイルのコピーが作成されます。したがって、プロジェクトにそのファイルを保存すると、Quality Center データ・テーブル・ファイルに加えた変更はファイル・システム上のデータ・テーブル・ファイルに影響せず、逆もまた同様です。

Quality Center でデータ・テーブル・ファイルを使用するには、次の手順を実行します。

- 1 新しいデータ・テーブル・ファイルを追加する場合は、**.xls** という拡張子の付いた新しい Microsoft Excel ファイルをファイル・システムに作成します。
- 2 Quality Center で新しいデータ・テーブル・リソースを作成し、前の手順で作成した **.xls** ファイルをプロジェクトのテスト・リソース・モジュールにアップロードします。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。
- 3 [テストの設定] ダイアログ・ボックスで、[リソース] ノードをクリックします。
- 4 [その他] を選択し、参照ボタンをクリックして、データ・テーブル・ファイルを検索します。

- 5 テストを作成します。テストを保存すると、QuickTest によって、データ・テーブル・ファイルが Quality Center プロジェクトに保存されます。

データベースからのデータのインポート

Microsoft Query からクエリを選択するか、SQL ステートメントを手作業で指定することによって、データベースからデータをインポートできます。

Microsoft Query は、Microsoft Office のカスタム・インストール・オプションでインストールできます。

注： Excel ファイルのインポート（[ファイル] > [ファイルからインポート]）とは異なり、データベースからデータをインポートするときに、データ・テーブルの既存のデータは置き換えられません。インポートするデータベースに既存のカラムと同じ名前のカラムがある場合、そのデータベースのカラムはカラム名の後に連番が付き、新しいカラムとして追加されます。たとえば、データ・テーブルに「departures」という名前のカラムがすでに存在する場合、同じ名前のデータベースのカラムは「departures1」という名前でそのデータ・テーブルに挿入されます。

データベースからデータをインポートするには、次の手順を実行します。

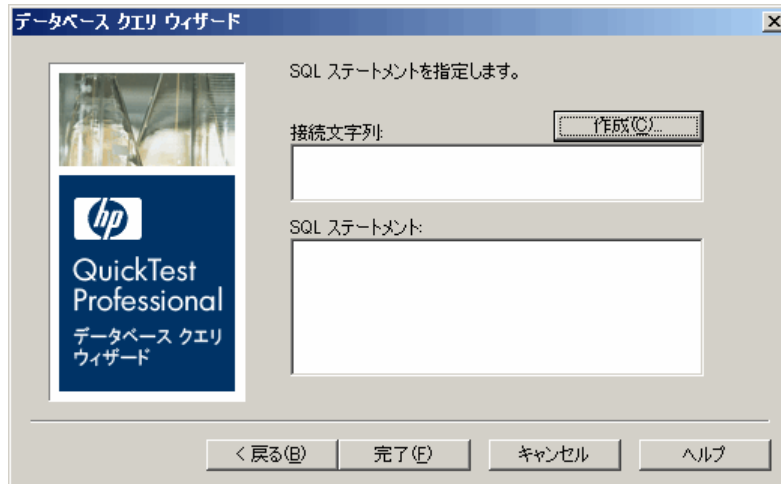
- 1 データのインポート先にするデータ・テーブル・シートを右クリックし、[シート] > [インポート] > [データベースから] を選択します。データベース・クエリ・ウィザードが起動します。



- 2 対象のデータベースを選択し、[次へ] をクリックします。次のオプションのいずれかを選択することができます。
 - ▶ **[Microsoft Query でクエリを作成する]** : Microsoft Query が起動し、新しいクエリを作成できるようになります。クエリの定義を完了すると、QuickTest に戻ります。このオプションは、コンピュータに Microsoft Query がインストール済みである場合にのみ使用できます。
 - ▶ **[SQL ステートメントを手作業で指定する]** : ウィザードに **[SQL ステートメントを指定します。]** 画面が表示され、接続文字列と SQL ステートメントを定義できる状態になります。詳細については、手順 3 を参照してください。
 - ▶ **[最大行数]** : このチェック・ボックスを選択して、インポートするデータベース行の最大数を入力します。最大で 32,000 行を指定できます。
 - ▶ **[Microsoft Query の使い方を表示する]** : [次へ] をクリックすると、Microsoft Query を開く前に、説明画面が表示されます（[Microsoft Query でクエリを作成する] を選択した場合にのみ使用できます）。

- 3 前の手順で [Microsoft Query でクエリを作成する] を選択した場合、Microsoft Query が起動します。データ・ソースを選択し、クエリを定義します。クエリの作成の詳細については、1204 ページ「Microsoft Query でのクエリの作成」を参照してください。

前の手順で [SQL ステートメントを手作業で指定する] を選択した場合、次の画面が表示されます。



接続文字列と SQL ステートメントを指定し、[完了] をクリックします。

- ▶ [接続文字列] : 接続文字列を入力するか、[作成] をクリックし、[データソースの選択] ダイアログ・ボックスを表示します。[データソースの選択] ダイアログ・ボックスで *.dsn ファイルを選択するか、新しい *.dsn ファイルを作成すると、接続文字列がこのボックスに挿入されます。
 - ▶ [SQL ステートメント] : SQL ステートメントを入力します。
- 4 数秒間でデータベース・クエリがキャプチャされ、QuickTest ウィンドウに戻ります。データベース・クエリの結果データは、データ・テーブルに表示されます。

Microsoft Query でのクエリの作成

Microsoft Query を使用して、データ・ソースを選択し、データ・ソースを対象とするクエリを定義することができます。サポートされている Microsoft Query のバージョンの詳細については、『HP QuickTest Professional 最初にお読みください』を参照してください。

Microsoft Query でデータ・ソースを選択し、クエリを定義するには、次の手順を実行します。

- 1 データベースからのデータのインポート処理中に Microsoft Query が起動したら、新しいデータ・ソースまたは既存のデータ・ソースを選択します。
- 2 クエリを定義します。
- 3 Query ウィザードの [完了] 画面で、[終了し、QuickTest へ戻る] を選択し、[完了] をクリックして Microsoft Query を終了します。

あるいは、[Microsoft Query でデータの表示またはクエリの編集を行う] をクリックし、[完了] をクリックします。データを表示または編集したら、[ファイル] > [終了し、QuickTest へ戻る] を選択し、Microsoft Query を閉じて QuickTest に戻ります。

Microsoft Query での作業の詳細については、Microsoft Query のドキュメントを参照してください。

データ・テーブルでの数式の使用

データ・テーブルでは、Microsoft Excel の数式を使用できます。したがって、実行セッション中に、実行コンテキストに応じたデータを作成できます。また、チェックポイントの一部として数式を使用し、実行中に作成された（動的に生成された）ページのオブジェクト、または Web ページ内やアプリケーションのその他の可変オブジェクトに、任意のコンテキストに応じた期待値が設定されているかどうかを検査することもできます。

データ・テーブルで数式を使用し、値を比較する場合（通常はチェックポイントで）、比較する値は同じタイプ（整数、文字列など）である必要があります。異なる関数を使用して、アプリケーションの異なる場所から値を抽出すると、値のタイプが異なる場合があります。これらの値は、画面上では同じものに見えるかもしれませんが、たとえば数値の 8.2 と文字列の「8.2」は等しくないため、比較は失敗します。

注：データ・テーブルでは、複雑な式またはネストされた式の使用はサポートされていません。

TEXT および VALUE 関数を使用して、値のタイプを別のタイプに変換できます。

- ▶ TEXT(value, format) は数値と同等のテキストを指定した形式で返します。たとえば、数式 =TEXT(8.2, "0.00") は "8.20" となります。
- ▶ VALUE(string) は文字列と同等の数値を返します。たとえば、=VALUE("\$8.20") は 8.20 となります。

ワークシートの関数の使用方法については、Microsoft Excel のドキュメントを参照してください。

パラメータ化データを作成する数式の使用

パラメータのカラムのセルに、固定値の代わりに数式を入力できます。

たとえば、今日以降の日付値を必要とする WebEdit オブジェクトに対する値をパラメータ化する必要があるとします。[Date] カラムのセルを日付の書式に設定し、最初の反復用に今日の日付を値に設定するために、Excel の数式 =NOW() を最初の行に入力します。

これによって、次のように残りの行に別の数式を使用して、上の行の日付に 1 日足した日付を入力できます。この数式を使用することで、テストをいつ実行しても、日付は常に有効になります。

	A2	=A1+1
	Date	B
1	1/3/2006	
2	1/4/2006	
3	1/5/2006	
4	1/6/2006	
5	1/7/2006	
6	1/8/2006	
7	1/9/2006	
8	1/10/2006	
9		

パラメータの使用方法の詳細については、第 24 章「値のパラメータ化」を参照してください。

チェックポイントでの数式の使用

チェックポイントで数式を使用し、実行時に作成された（動的に生成された）ページのオブジェクト、または Web ページ内やアプリケーションの別の可変オブジェクトに、任意のコンテキストに応じた期待値が設定されていることを確認できます。たとえば、ショッピング・カートの Web サイトに、合計金額が表示されるとします。この場合、表示された合計金額を対象とするテキスト・チェックポイントを作成してから、データ・テーブル数式を使用して、各反復で選択された商品のそれぞれの価格を合計し、サイトの合計金額として比較して正しく計算されているかどうかを検査できます。

チェックポイントでデータ・テーブル数式を使う場合、QuickTest はデータ・テーブルに 2 つのカラムを作成します。最初のカラムには、標準のチェックポイント数式が含まれています。2 番目のカラムには、検査対象の値が出力パラメータの形式で格納されています。数式の結果は Boolean で、TRUE または FALSE です。

	A1	
		=B1="337"
	Total_Price	Total_Price_out
1	TRUE	337
2		

テストの実行中にチェックポイントのカラムで FALSE の結果が出ると、テストが失敗します。

チェックポイントの追加が終了したら、最初のカラムで標準の数式を変更し、必要な検査を実行できます。

チェックポイントで数式を使うには、次の手順を実行します。

- 1 第 17 章「チェックポイントについて」の説明に従って、チェックポイントを作成するオブジェクトまたはテキストを選択し、[チェックポイントの挿入] ダイアログ・ボックスを開きます。
- 2 [値の設定] 領域で、[パラメータ] をクリックします。



- 3 **[パラメータ オプション]** ボタンをクリックします。[パラメータ オプション] ダイアログ・ボックスが表示されます。

パラメータ オプション

パラメータのタイプ
DataTable

名前(N): userName_name

データテーブル内の場所
 グローバルシート(G)
 現在のアクションシート (ローカル)(L)

詳細設定
 正規表現(E)
 データテーブル数式(E)

OK キャンセル ヘルプ

- 4 パラメータ・タイプとして **[Data Table]** を選択し、**[名前]** ボックス・リストからパラメータを選択するか、新しい名前を入力します。
- ▶ 既存のパラメータを使用するには、リストからパラメータを選択します。
 - ▶ 新しいパラメータを作成するには、標準設定のパラメータ名を使用するか、わかりやすいパラメータ名を入力します。
- 5 **[データ テーブル数式]** チェック・ボックスを選択して、**[OK]** をクリックし、[パラメータ オプション] ダイアログ・ボックスを閉じます。

注：[正規表現] が選択されている場合は、[データ テーブル数式] を選択できません。

- 6 第 17 章「チェックポイントについて」の説明に従って、チェックポイントを設定します。

- 7 [OK] をクリックします。テーブルに 2 つのカラムが追加され、テストにはチェックポイント・ステップが挿入されます。
- 8 最初の (式の) カラムにある値を強調表示して数式を確認し、必要に応じて数式を変更します。
- 9 複数の反復を実行する場合、テストまたはアクションの反復ごとに、数式のカラムの続く行に適切な数式を追加します。

ヒント: 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またはデータ・テーブル・パラメータ値として使用できます。詳細については、410 ページ「で使用するパスワードを暗号化できます。メソッド引数およびデータ・テーブル・セル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。データ・テーブル・メニューの [暗号化] オプションを使用して、データ・テーブル列の文字列を暗号化することもできます。詳細については、1197 ページ「[データ] メニュー」を参照してください。

データ・テーブルのスクリプト・メソッドの使用

QuickTest には、実行時データ・テーブルに関する情報を取得したり、実行時データ・テーブルのセルに値を設定したりするためのデータ・テーブル・メソッドがいくつか用意されています。これらのステートメントは [エキスパートビュー] の中で手作業で入力します。エキスパート・ビューでの作業の詳細については、第 29 章「エキスパート・ビューおよび関数ライブラリ・ウィンドウを使用した作業」を参照してください。

プログラミングの観点から見ると、データ・テーブルは、DataTable、Sheet (シート)、Parameter (カラム) の 3 種類のオブジェクトで構成されています。各オブジェクトには、値の取得や設定に使用できる、いくつかのメソッドとプロパティがあります。

データ・テーブル・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 43 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、テス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、1555 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

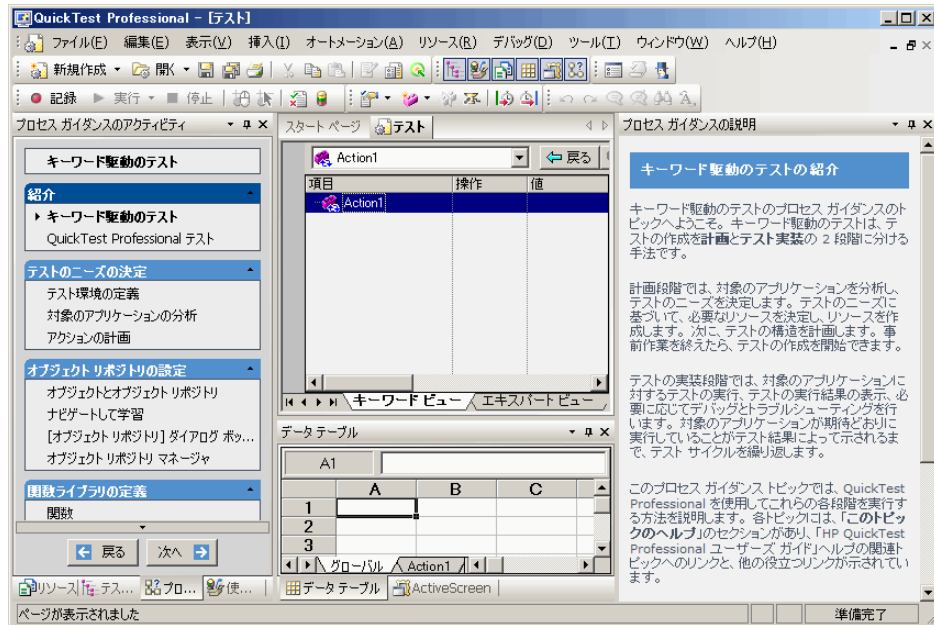
- ▶ [プロセス ガイダンス] 表示枠 (1210 ページ)
- ▶ プロセス・ガイダンスの開始 (1212 ページ)
- ▶ 利用できるプロセスのリストの管理 (1213 ページ)
- ▶ [プロセス ガイダンス管理] ダイアログ・ボックス (1214 ページ)

[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せん。

- ▶ 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 ガイドランス リスト] が空の場合は、[ファイル] > [プロセス ガイドランス管理] を選択し、[プロセス ガイドランス管理] 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、1214 ページ「[プロセス ガイドランス管理] ダイアログ・ボックス」を参照してください。

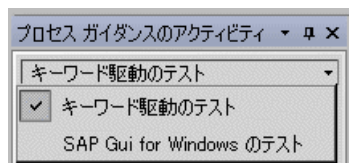
[オートメーション] メニュー・コマンド

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 ガイドランス 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 ガイドランス リスト] が空の場合は、[ファイル] > [プロセス ガイドランス管理] を選択し、[プロセス ガイドランス管理] 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 ガイドランス 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 ガイドランス 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・ガイドランス・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

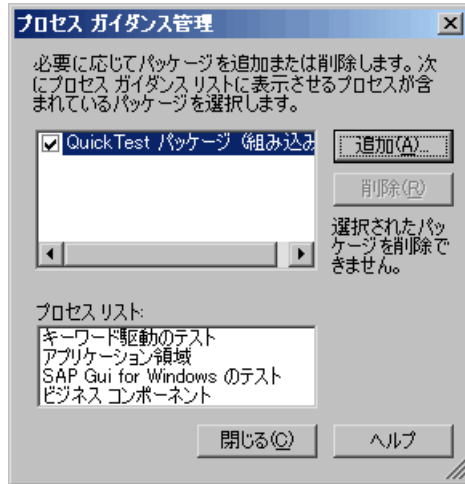
ページは、[プロセス ガイダンス管理] ダイアログ・ボックスに標準で表示されます。

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、1555 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	<ul style="list-style-type: none">▶ 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	<p>概念的概要： 1209 ページ「プロセス・ガイダンスを使った作業」</p> <p>主要なタスク：</p> <ul style="list-style-type: none">▶ 1215 ページ「パッケージの利用と除外」▶ 1216 ページ「プロセス・ガイダンス・パッケージの追加」 <p>その他の関連トピック： 1215 ページ「補足リファレンス」を参照してください。</p>

[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	1210 ページ 「[プロセス ガイダンス] 表示枠」
関連作業	1212 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 ガイドランス 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 ガイドランス 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 ガイドランス管理] を選択します。[プロセス ガイドランス管理] 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 ガイドランス リスト] に追加または削除されます。

プロセス・ガイドランス・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 ガイドランス 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 ガイドランス管理] 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 ガイドランス管理] を選択します。[プロセス ガイドランス管理] ダイアログ・ボックスが開きます。
- 2 [プロセス ガイドランス管理] 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・ガイドランス・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 IX 部

QuickTest の設定

第 44 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 とテス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 (1219 ページ)
- ▶ [オプション] ダイアログ・ボックスの使用 (1220 ページ)
- ▶ テストの一般オプションの設定 (1223 ページ)
- ▶ テストのフォルダ・オプションの設定 (1226 ページ)
- ▶ テストの実行オプションの設定 (1243 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、テス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 がテス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、すべてのテス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 (1220 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。

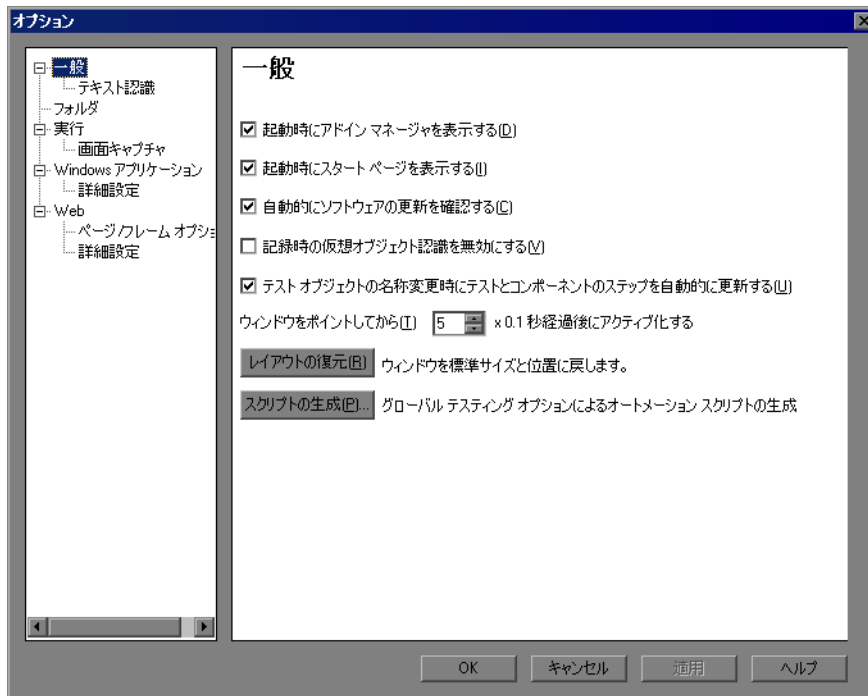
また、QuickTest で現在開いているテストだけで有効なテスト・オプションを設定することもできます。詳細については、第 45 章「個別のテストのオプション設定」を参照してください。

[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。

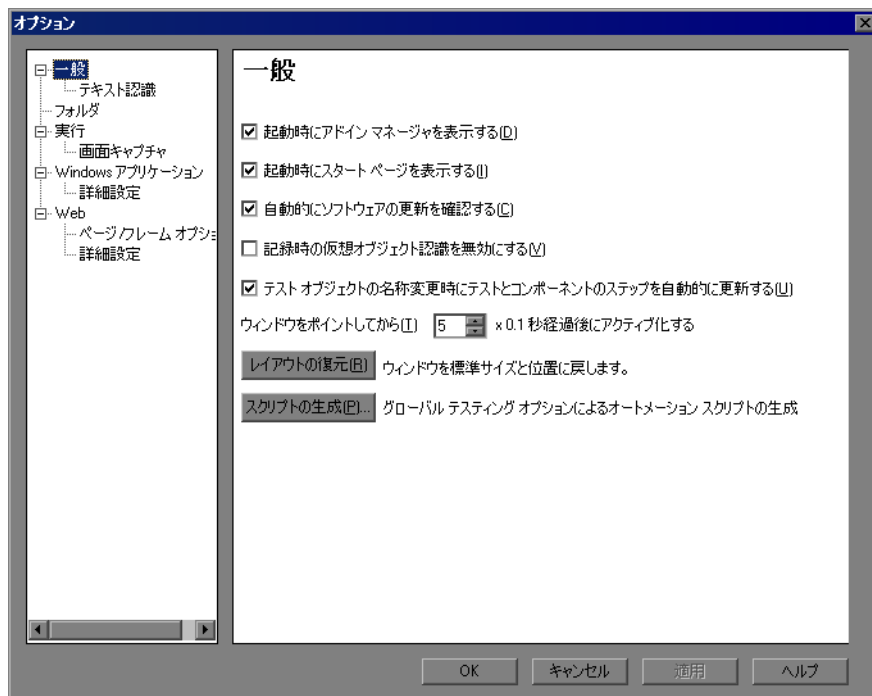
ノード	オプション
[一般]	<p>テストの一般設定のオプション。詳細については、1223 ページ「テストの一般オプションの設定」を参照してください。</p> <p>[一般] ノードには [テキスト認識] サブノードも含まれています。詳細については、1225 ページ「テキスト認識オプションの設定」を参照してください。</p>
[フォルダ]	<p>QuickTest が、ダイアログ・ボックスおよびステートメントの中で相対パスを使って指定されているテスト、アクション、またはファイルを検索するフォルダ (検索パス) を入力するオプション。詳細については、1226 ページ「テストのフォルダ・オプションの設定」を参照してください。</p> <p>注： 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 の絶対パスを指定してください。詳細については、1432 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</p>
[ActiveScreen]	<p>記録中に QuickTest が ActiveScreen に保存、表示する情報を設定するオプション。詳細については、1229 ページ「ActiveScreen オプションの設定」を参照してください。</p>

ノード	オプション
<p>[実行]</p>	<p>テストを実行するためのオプション。詳細については、1243 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。</p> <p>[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、1246 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。</p>
<p>[Windows アプリケーション]</p>	<p>QuickTest テストが 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。</p> <p>[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。</p>

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、750 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、16 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。詳細については、第 47 章「仮想オブジェクトの学習」を参照してください。
[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、171 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。
[記録後、「With」ステートメントを自動生成する]	QuickTest が記録時に自動的に With ステートメントを生成するよう指示します。詳細については、809 ページ「テストに対する「With」ステートメントの生成」を参照してください。
[内部で「With」ステートメントを生成するオブジェクト数]	With ステートメントを適用する対象となるオブジェクトが最低限連続する数を示します。この設定は、記録後に QuickTest が自動的に With ステートメントを生成する場合、および既存のアクションのために With ステートメントを生成するよう選択した場合に使用します。 標準設定 = 2 。詳細については、809 ページ「テストに対する「With」ステートメントの生成」を参照してください。

オプション	説明
[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、チェックポイント、ステップ・ジェネレータ、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
[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、1132 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、1387 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、750 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。

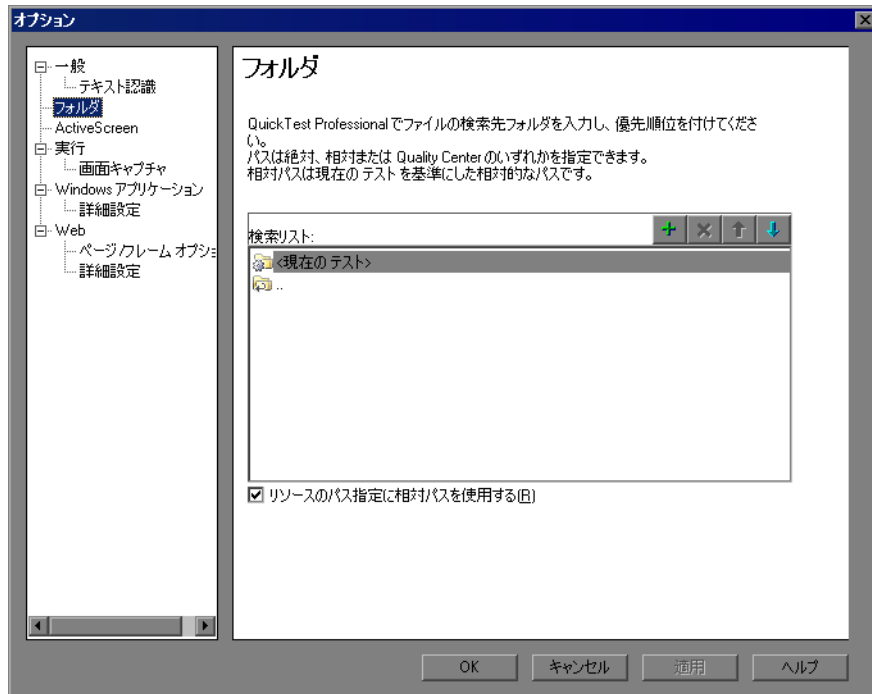
テストのフォルダ・オプションの設定

[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テスト、アクション、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。たとえば、すべてのテストが格納されているフォルダをフォルダ・リストに追加するとします。後にアクションのコピーをテストに挿入する際は、挿入するアクションを含むテストの名前を [アクションのコピーを挿入] ダイアログ・ボックスに入力するだけで済みます。QuickTest は [フォルダ] 表示枠に指定されたフォルダの中でテストのパスを検索します。





注：

- ▶ 標準設定では現在のテストが**検索リスト**に表示されます。これは削除できません。
 - ▶ 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 の絶対パスを指定してください。詳細については、1432 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。
 - ▶ 相対パスまたは絶対パスの詳細については、322 ページ「QuickTest での相対パスの使用」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、指定されたテスト、アクション、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。

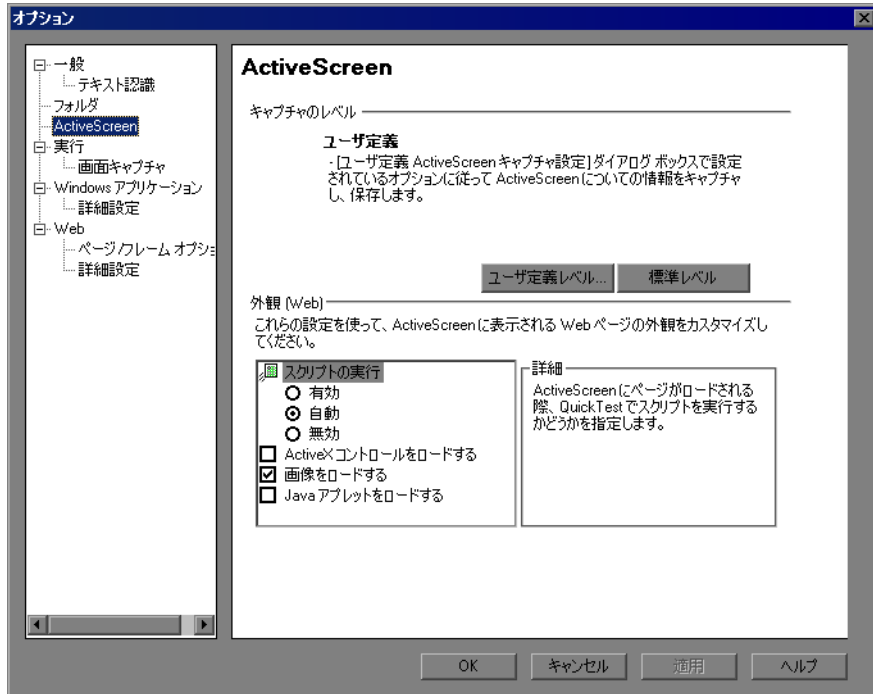
オプション	説明
[検索リスト]	QuickTest がテスト、アクション、またはファイルを検索するフォルダを示します。ここでフォルダを定義しておけば、ほかのダイアログ・ボックスや呼び出しステートメントの中でテスト、アクション、またはファイルのフルパスを指定する必要がなくなります。リスト内の検索パスの順序に応じて、QuickTest が特定のアクションやファイルを検索する順序が決まります。
	新しいフォルダを検索リストに追加します。 ヒント： <ul style="list-style-type: none"> ▶ Quality Center に接続されている状態で Quality Center 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、322 ページ「QuickTest での相対パスの使用」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のみ、リマインダが表示されます。

ヒント : テストの中で PathFinder.Locate ステートメントを使うと, [フォルダ] 表示枠で指定されたフォルダに基づく特定の相対パスに対応する, QuickTest で使われるフル・パスを取得できます。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。

ActiveScreen オプションの設定

説明	<p>テストの記録および実行時に QuickTest が ActiveScreen に保存および表示する情報を指定できます。</p> <p>ActiveScreen に保存される情報が多いほど, 記録後のテストを編集しやすくなります。しかし, ActiveScreen に保存される情報が多くなると, 記録に必要な時間とディスク容量も多くなります。Windows ベースのアドインは ActiveScreen データを保存するのに多くのディスク容量を必要とするため, Windows ベースのアドインではこのオプションが特に重要になります。</p>
アクセス方法	<p>[ツール] メニュー > [オプション] 項目 > [ActiveScreen] ノード</p>
重要情報	<p>MDI (Multiple Document Interface) アプリケーションの記録時には, ActiveScreen にアクティブでない子フレームの情報は保存されません。</p>
詳細情報	<p>概念的概要 : 380 ページ「ActiveScreen を使った作業」</p> <p>主要なタスク : 381 ページ「テストに保存する ActiveScreen 情報の増減」</p> <p>その他の関連トピック : 1233 ページ「補足リファレンス」</p>

[オプション] ダイアログ・ボックスの [ActiveScreen] 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 : [ActiveScreen] 表示枠のオプション

オプション	説明
[キャプチャのレベル]	<p>QuickTest が ActiveScreen にデータを格納するべきオブジェクトを指定します。スライダを使用して次のオプションから 1 つを選択します。</p> <ul style="list-style-type: none"> ▶ [完全] : ステップごとに、アプリケーションのアクティブなウィンドウ、ダイアログ・ボックス、Web ページのすべてのオブジェクトのすべてのプロパティを取得します。このレベルは、動的な変更後の Web ページを保存し、ActiveScreen ファイルを圧縮した形式で保存します。 ▶ [部分的] : (標準設定) アプリケーションに対して実行された最初のステップで、アプリケーションのアクティブなウィンドウ、ダイアログ・ボックス、Web ページのすべてのオブジェクトのすべてのプロパティ、および、同じウィンドウにおいて以降のステップで記録されるオブジェクトのすべてのプロパティを取得します。このレベルは、動的な変更後の Web ページを保存し、ActiveScreen ファイルを圧縮した形式で保存します。 ▶ [最小] : 各ステップの ActiveScreen に、記録されたオブジェクトとその親のみのプロパティを保存するよう設定します。このレベルは、(動的な変更の前の) 全 Web ページのオリジナルの HTML ソースを保存し、ActiveScreen ファイルを圧縮した形式で保存します。 ▶ [なし] : すべてのアプリケーションおよび Web ページを対象とする ActiveScreen ファイルのキャプチャを無効にします。
[ユーザ定義レベル]	<p>カスタムの ActiveScreen オプションを指定できるようにします。詳細については、1233 ページ「[ActiveScreen キャプチャのユーザ定義設定] ダイアログ・ボックス」を参照してください。</p>
[標準レベル]	<p>キャプチャ・レベルの設定を既定のレベル (部分的) に戻します。</p>

オプション	説明
[外観 (Web)]	<p>キャプチャした Web ページを ActiveScreen で表示する方法を変更できます。</p> <ul style="list-style-type: none"> ▶ [スクリプトの実行]：次のいずれかのオプションに従って、ActiveScreen に Web ページが読み込まれるときに QuickTest でスクリプトを実行するかどうかを指定します。 <ul style="list-style-type: none"> ▶ [有効]：ActiveScreen にページを読み込むたびにスクリプトを実行します。 ▶ [自動]：表示されるページに応じて、必要な場合にスクリプトを実行します。 ▶ [無効]：ActiveScreen にページを読み込むときに、スクリプトを実行しません。 <p>注：このオプションはページの読み込み時に実行されるスクリプトだけを対象としません。ActiveScreen の画面で操作を行なったときにスクリプトを動作させるものではありません。</p> <ul style="list-style-type: none"> ▶ [ActiveX コントロールをロードする]：各ステップにおいてページが実際にどのようにアプリケーションに表示されるかプレビューできるように、ブラウザ・ページから ActiveScreen に ActiveX コントロールを読み込むよう QuickTest を設定します。このオプションをクリアすると、すべての ActiveX コントロール・オブジェクトについて、ActiveScreen に標準の ActiveX 画像が表示されます。 ▶ [画像をロードする]：ブラウザ・ページから ActiveScreen 表示枠に画像をロードするように QuickTest を設定します。 ▶ [Java アプレットをロードする]：各ステップにおいてページが実際にどのようにアプリケーションに表示されるかプレビューできるように、ブラウザ・ページから ActiveScreen に Java アプレットを読み込むよう QuickTest を設定します。このオプションをクリアすると、すべての Java アプレット・オブジェクトについて、ActiveScreen に標準の Java 画像が表示されます。 <p>注：</p> <ul style="list-style-type: none"> ▶ QuickTest では、ActiveX コントロールおよび Java アプレットが ActiveScreen に表示専用モードで読み込まれます。読み込まれた ActiveX オブジェクトまたは Java オブジェクトを操作したり、それらから追加情報を取得したりできません。ActiveScreen でこれらの項目を操作するには、対応するアドインを読み込み、ActiveX または Java オブジェクトを対象に直接記録を行う必要があります。 ▶ ActiveScreen に読み込まれた ActiveX コントロールおよび Java アプレットの動作は、アプリケーションでの動作と異なる場合があります。読み込まれた個々のコントロールまたはアプレットの実装にもよりますが、そのことが原因で予期しない動作をする場合があります。

補足リファレンス

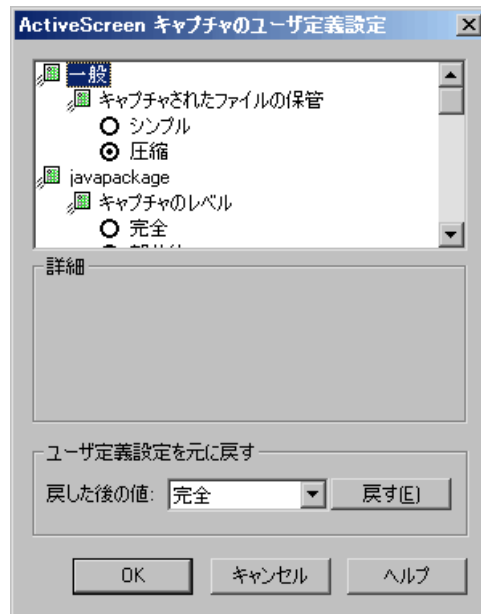
関連ユーザ・ インタフェース・トピック	1233 ページ「[ActiveScreen キャプチャのユーザ定義設定] ダイアログ・ボックス」
関連作業	<ul style="list-style-type: none"> ▶ 383 ページ「単一 ActiveScreen キャプチャの更新」 ▶ 1111 ページ「更新モード・オプションを使ったテストの更新」
その他の関連情報	385 ページ「ActiveScreen のパフォーマンスを向上させるためのヒント」

[ActiveScreen キャプチャのユーザ定義設定] ダイアログ・ボックス

説明	<p>QuickTest による ActiveScreen 情報のキャプチャと保存の方法をカスタマイズできます。</p> <p>ActiveScreen のユーザ設定を適用すると、[ActiveScreen] 表示枠でのキャプチャ・レベルの設定が、[ActiveScreen キャプチャのユーザ定義設定] ダイアログ・ボックスでのすべての設定で置き換えられます。</p>
アクセス方法	[ツール] メニュー > [オプション] ダイアログ・ボックス > [ActiveScreen] ノード > [ユーザ定義レベル] ボタン

<p>重要情報</p>	<p>[ActiveScreen キャプチャのユーザ設定] ダイアログ・ボックスの標準設定には, [オプション] ダイアログ・ボックスの [ActiveScreen] 表示枠で選択されたキャプチャ・レベルの設定は反映されません。特定の設定だけカスタマイズする場合は, [戻した後の値] オプションを使用してほかのすべての設定に既定のキャプチャ・レベルの設定が使用されるようにした後で, 必要に応じて個別の設定を変更します。</p>
<p>関連作業</p>	<ul style="list-style-type: none"> ▶ [保存] ダイアログ・ボックスの [ActiveScreen ファイルの保存] オプションを使用すれば, ActiveScreen の画面キャプチャを保存するかどうか指定できます。詳細については, 330 ページ「テストの保存」を参照してください。 ▶ ActiveScreen のキャプチャ設定の変更後, [更新モード] オプションを使用すれば, ActiveScreen に保存される情報の量を変更できます。詳細については, 1111 ページ「更新モード・オプションを使ったテストの更新」を参照してください。

[ActiveScreen キャプチャのユーザ定義設定] ダイアログ・ボックスの画像を次に示します。



注： [ActiveScreen キャプチャのユーザ設定] ダイアログ・ボックスには、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。

[ActiveScreen キャプチャのユーザ定義設定] ダイアログ・ボックスのオプション

オプション	説明
設定ボックス	<p>QuickTest による ActiveScreen 情報のキャプチャと保存の方法を決定する特定の設定オプションを選択できます。設定ボックスには、コンピュータにインストールされている 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。</p> <p>参照先</p> <ul style="list-style-type: none"> ▶ 1236 ページ 「[一般] オプション」 ▶ 1237 ページ 「[キャプチャのレベル] オプション」 ▶ 1242 ページ 「Web オプション」
[説明]	<p>設定ボックスで選択したオプションの説明が表示されます。</p>
[ユーザ定義設定を元に戻す]	<p>ユーザ定義設定を QuickTest においてあらかじめ定義されているいずれかのレベル ([完全], [部分的], [最小] または [なし]) に戻すには, [戻した後の値] リストからレベルを選択して [戻す] ボタンをクリックします。指定可能なキャプチャ・レベルの詳細については, 1237 ページ 「[キャプチャのレベル] オプション」を参照してください。</p>

[一般] オプション

[**キャプチャされたファイルの保管**] オプションでは、キャプチャされた ActiveScreen 情報を格納するために QuickTest が使用する圧縮のタイプを指定できます。

- ▶ [**圧縮なし**] : ActiveScreen のキャプチャ・データを標準の非圧縮ファイル形式 (.html, .png など) で保存するよう QuickTest を設定します。
- ▶ [**圧縮**] : ActiveScreen のキャプチャ・データを圧縮ファイル形式 (ZIP 形式) で保存するよう QuickTest を設定します。このオプションを使用するとディスク領域を節約できますが, ActiveScreen への画像の読み込み時間に影響が出る場合があります。これは標準設定のオプションです。

[キャプチャのレベル] オプション

[ActiveScreen キャプチャのユーザ設定] ダイアログ・ボックスでは、QuickTest による ActiveScreen 情報のキャプチャと保存の方法をカスタマイズできます。

[キャプチャのレベル] オプションは、Java アプレットおよびアプリケーション、SAP GUI for Windows アプリケーション、Oracle アプリケーション、Windows ベースのアプリケーション、ターミナル・エミュレータ・アプリケーションで使用できます。[ActiveScreen キャプチャのユーザ定義設定] ダイアログ・ボックスで使用できるオプションは、インストールされているアドインに応じて変わります。

[キャプチャのレベル] オプションでは、ActiveScreen 用にキャプチャするアプリケーションの各オブジェクトについて、どのプロパティをキャプチャするかを設定できます。

テスト要件に応じて、さまざまな ActiveScreen キャプチャ・レベルを選択できます。ただし、ActiveScreen 用にキャプチャされる情報が少ないほどパフォーマンスがよくなるという点を考慮に入れる必要があります。

たとえば、[完全] キャプチャ・レベル・オプションを選択した場合、記録後に ActiveScreen キャプチャに表示されるすべてのテスト・オブジェクトにチェックポイントを追加できますが、1つの操作を記録するのに多くの時間と多くのディスク容量が使用されることとなります。[部分的] を選択すれば、記録時間を短くし、使用するディスク容量も少なくすることができます。しかし、記録後に ActiveScreen で実行できる操作が制限される可能性があります。

次の項では、それぞれの環境で使用できるキャプチャ・レベルのオプションについて説明します。

Java アプリケーションおよびアプレット

Java アプリケーションおよび Java アプレットでは、次の [キャプチャのレベル] オプションを使用できます。

- ▶ [完全] : 各ステップの ActiveScreen に、アプリケーションあるいはアプレットの開いているウィンドウまたはダイアログ・ボックス内のすべてのオブジェクトのすべての認識プロパティを保存するよう QuickTest を設定します。

- ▶ **[部分的]** : (標準設定) ウィンドウで実行された最初のステップの ActiveScreen に、アプリケーションあるいはアプレットの開いているウィンドウまたはダイアログ・ボックスのすべてのオブジェクトのすべての認識プロパティを保存し、同じウィンドウにおいて以降のステップで記録されるオブジェクトのすべてのプロパティを保存するよう QuickTest を設定します。
- ▶ **[最小]** : 記録されたオブジェクトのすべての認識プロパティ、および、記録階層の親オブジェクトのすべての認識プロパティを保存するよう QuickTest を設定します。
- ▶ **[なし]** : Java アプリケーションまたは Java アプレットの ActiveScreen ファイルのキャプチャを無効にします。

Java アプリケーションおよび Java アプレットでは、**[完全]** または **[最小]** のキャプチャ・レベルを選択した場合、**[ActiveScreen キャプチャのユーザ定義設定]** ダイアログ・ボックスの次の設定も関係してきます。

[次のオブジェクトのキャプチャを無効にする] : 選択したテスト・オブジェクト・タイプのほかのオブジェクトに対して実行されたステップのデータが ActiveScreen にキャプチャされないようにします。このオブジェクトは、ActiveScreen で画像としてのみ見ることができます。

標準設定では、**JavaObject** と **JavaMenu** が選択されています (したがって、これらのオブジェクトでは認識プロパティはキャプチャされません)。

注 : 特定のテスト・オブジェクトを記録する場合は、**[次のオブジェクトのキャプチャを無効にする]** オプションが選択されていても、認識プロパティがキャプチャされます。

SAP GUI for Windows アプリケーション

SAP GUI for Windows アプリケーションでは、次の **[キャプチャのレベル]** オプションを使用できます。

- ▶ **[完全]** : 各ステップの ActiveScreen に、アプリケーションの開いているウィンドウまたはダイアログ・ボックス内のすべてのオブジェクトのプロパティ値を保存するよう QuickTest を設定します。

このオプションにより、すべてのステップの **ActiveScreen** で、ウィンドウまたはダイアログ・ボックスの任意のオブジェクトに対してチェックポイントを挿入したりほかの操作を実行したりできるようになります。ただし、記録時間が長くなり、多くのディスク容量が必要になる場合があります。

注：一部のコンテナ・オブジェクトの内部オブジェクト（テーブル・セルやツリー・ノードなど）のプロパティは、**ActiveScreen** にキャプチャされません。このようなオブジェクトの情報にアクセスするには、適切な **SAPGuiTable** メソッドまたは **SAPGuiTree** メソッドを使用します。詳細については、『**HP QuickTest Professional Object Model Reference**』（英語版）の **SAP GUI for Windows** の項を参照してください。

- ▶ **[部分的]**：（標準設定）各ステップの **ActiveScreen** に、記録されたオブジェクトとその親のプロパティを保存するよう **QuickTest** を設定します。

このオプションによって、記録時間が短くなり、必要となるディスク容量は比較的小さくなります。ただし、チェックポイントの挿入やほかの操作の対象にできるのは、記録されたオブジェクトとウィンドウまたはダイアログ・ボックスだけとなります。**ActiveScreen** に表示されるほかのオブジェクトに対して操作は実行できません。

- ▶ **[なし]**：**SAP GUI for Windows** アプリケーションの **ActiveScreen** ファイルのキャプチャを無効にします。

このオプションによって、非常に記録時間が短くなり、必要となるディスク容量は最小限になります。ただし、**ActiveScreen** で記録後のテスト編集（チェックポイントや出力値の挿入など）を実行することはできません。

注：

- ▶ ActiveScreen のオブジェクトのプロパティ値は、ステップがテストに追加された時点（情報が SAP サーバに送信されたとき）の値を反映しています。場合によっては、この値は、特定のステップが実行された時点でのプロパティ値とは異なることがあります。
 - ▶ ActiveScreen には、ステップがテストに追加されたときに SAP GUI for Windows アプリケーション・ウィンドウに表示されていた部分のみキャプチャされます。
-

Oracle アプリケーション

Oracle アプリケーションでは、次の **[キャプチャのレベル]** オプションを使用できます。

- ▶ **[完全]**：各ステップの ActiveScreen に、アプリケーションの開いているウィンドウまたはダイアログ・ボックス内のすべてのオブジェクトのすべての認識プロパティを保存するよう QuickTest を設定します。
- ▶ **[部分的]**：（標準設定）ウィンドウで実行された最初のステップの ActiveScreen に、アプリケーションの開いているウィンドウまたはダイアログ・ボックスのすべてのオブジェクトのすべての認識プロパティを保存し、同じウィンドウにおいて以降のステップで記録されるオブジェクトのすべての認識プロパティを保存するよう QuickTest を設定します。
- ▶ **[最小]**：記録されたオブジェクトのすべての認識プロパティ、および、記録階層の親オブジェクトのすべての認識プロパティを保存するよう QuickTest を設定します。
- ▶ **[なし]**：Oracle アプリケーションの ActiveScreen ファイルのキャプチャを無効にします。

Windows アプリケーション

Windows アプリケーションでは、次の **[キャプチャのレベル]** オプションを使用できます。

- ▶ **[完全]**：各ステップの ActiveScreen に、アプリケーションの開いているウィンドウまたはダイアログ・ボックス内のすべてのオブジェクトのすべてのプロパティを保存するよう QuickTest を設定します。

このオプションにより、すべてのステップの **ActiveScreen** で、ウィンドウまたはダイアログ・ボックスの任意のオブジェクトに対してチェックポイントを挿入したりほかの操作を実行したりできるようになります。

- ▶ **[部分的]**：(標準設定) アプリケーションに対して実行された最初のステップで、アプリケーションの開いているウィンドウおよびダイアログ・ボックスのすべてのオブジェクトのすべてのプロパティ、および、同じウィンドウにおいて以降のステップで記録されるオブジェクトのすべてのプロパティを保存するよう **QuickTest** を設定します。

このオプションにより、**ActiveScreen** に表示される任意のオブジェクトに対して、チェックポイントを挿入したり、ほかの操作を実行したりできるようになると同時に、記録時間とディスク容量を節約できます。このオプションの場合、以降のステップで **ActiveScreen** の情報が一部更新されないことがあります。

- ▶ **[最小]**：各ステップの **ActiveScreen** に、記録されたオブジェクトとその親のみのプロパティを保存するよう **QuickTest** を設定します。

このオプションによって、記録時間が短くなり、必要となるディスク容量は比較的小さくなります。ただし、チェックポイントの挿入やほかの操作の対象にできるのは、記録されたオブジェクトとウィンドウまたはダイアログ・ボックスだけとなります。**ActiveScreen** に表示されるほかのオブジェクトに対して操作は実行できません。

- ▶ **[なし]**：Windows アプリケーションの **ActiveScreen** ファイルのキャプチャを無効にします。

このオプションによって、非常に記録時間が短くなり、必要となるディスク容量は最小限になります。ただし、**ActiveScreen** で記録後のテスト編集を実行できません。

ターミナル・エミュレータ・アプリケーション

ターミナル・エミュレータで実行されるアプリケーションでは、次の **[キャプチャのレベル]** オプションを使用できます。

- ▶ **[完全]**：各ステップの **ActiveScreen** に、アプリケーションの開いているウィンドウまたはダイアログ・ボックス内のすべてのオブジェクトのすべてのプロパティを保存するよう **QuickTest** を設定します。

このオプションにより、すべてのステップの **ActiveScreen** で、ウィンドウまたはダイアログ・ボックスの任意のオブジェクトに対してチェックポイントを挿入したりほかの操作を実行したりできるようになります。

- ▶ **[なし]** : ターミナル・エミュレータ・アプリケーションの ActiveScreen ファイルのキャプチャを無効にします。

Web オプション

ActiveScreen 用に Web ページをキャプチャするかどうかを QuickTest に指定することができます。

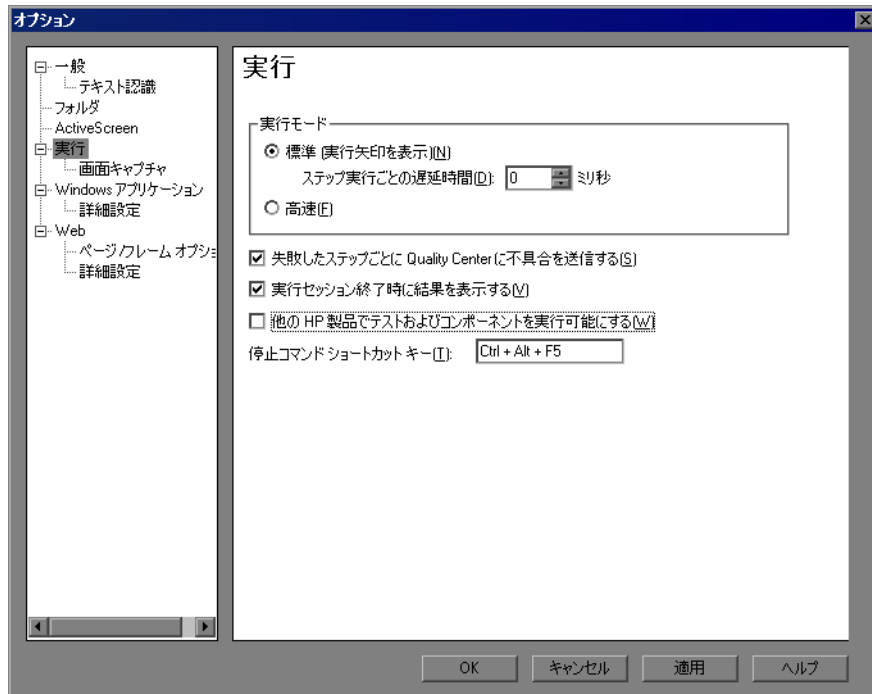
- ▶ **[ActiveScreen のキャプチャを無効にする]** : ActiveScreen でのすべてのステップの画面キャプチャを無効にします。

このオプションを選択しない場合、テストの編集を終えた後に **[名前を付けて保存]** を選択し、**[ActiveScreen ファイルを保存する]** チェック・ボックスをクリアすれば、ActiveScreen の情報を削除することもできます。詳細については、330 ページ「テストの保存」を参照してください。

- ▶ **[元の HTML ソースをキャプチャする]** : Web ページにおいてスクリプトが実行される前の最初の状態での HTML ソースをキャプチャします。このオプションの選択を解除すると、HTML ソースに動的な変更が（ページの読み込み時にスクリプトが自動的に実行されるなどして）加えられた後に Web ページの HTML ソースをキャプチャするよう QuickTest が設定されます。

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 がテス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、1246 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。

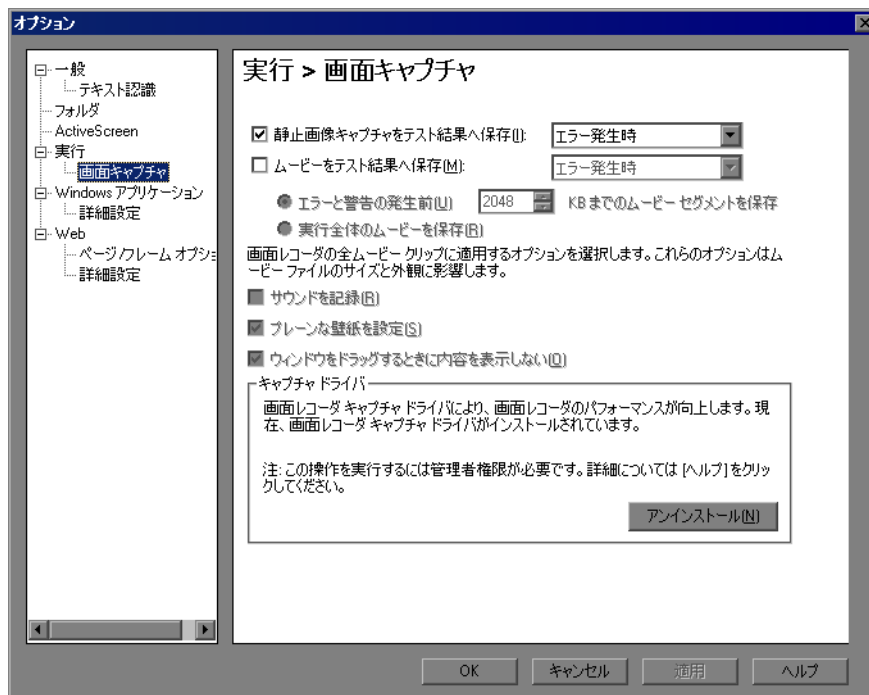
オプション	説明
<p>[実行モード]</p>	<p>QuickTest に対してテストの実行方法を指定します。</p> <p>▶ [標準 (実行矢印を表示)]：キーワード・ビューまたはエキスパート・ビューの左に実行矢印を表示しながらテストを実行し、実行した各ステップまたはステートメントがわかるようにします。テストに複数のアクションが含まれている場合、キーワード・ビューの [項目] カラムのツリーが展開されてステップが表示され、エキスパート・ビューには現在実行しているアクションのスクリプトが表示されます。</p> <p>[ステップ実行ごとの遅延時間]：連続する各ステップを実行するまでに QuickTest 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)。</p> <p>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。</p> <p>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。</p> <p>▶ [高速]：キーワード・ビューまたはエキスパート・ビューの左に実行矢印を表示せずにテストを実行します。項目ツリーの展開表示や、実行している各アクションのスクリプトの表示は行いません。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。</p> <p>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、テストは自動的に高速モードで実行されます。</p>
<p>[失敗したステップごとに Quality Center に不具合を送信する]</p>	<p>テストの失敗したステップごとに不具合を自動的に Quality Center に送信するように QuickTest を設定します。このオプションは、Quality Center プロジェクトに接続されている場合にのみ利用できます。詳細については、1011 ページ「Quality Center プロジェクトへの自動による不具合の送信」を参照してください。</p>

オプション	説明
[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。
[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]	<p>このコンピュータ上で、Quality Center や Test Batch Runner などのほかの HP 製品で QuickTest のテストを実行できるようになります。</p> <p>注：このオプションは WinRunner で QuickTest のテストを実行する場合には必要ありません。</p>
[停止コマンドショートカットキー]	<p>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。</p> <p>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。</p> <p>標準のキーの組み合わせは Ctrl+Alt+F5 です。</p> <p>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。</p> <ul style="list-style-type: none"> ▶ [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。

[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

<p>説明</p>	<p>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。</p>
<p>アクセス方法</p>	<ul style="list-style-type: none"> ▶ [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。
<p>重要情報</p>	<p>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。</p>
<p>詳細情報</p>	<p>その他の関連トピック : 1250 ページ「補足リファレンス」</p>

[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
<p>[静止画像キャプチャをテスト結果へ保存]</p>	<p>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。</p> <p>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。</p> <ul style="list-style-type: none"> ▶ [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。 <p>詳細については、989 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。</p> <p>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。</p> <ul style="list-style-type: none"> ▶ 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
<p>[ムービーをテスト結果へ保存]</p>	<p>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。</p> <p>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。</p> <ul style="list-style-type: none"> ▶ [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。 <p>詳細については、989 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。</p>
<p>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。</p>	
<p>[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]</p> <p>([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)</p>	<p>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。</p>
<p>[実行全体のムービーを保存する]</p> <p>([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)</p>	<p>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。</p>
<p>[サウンドを記録]</p>	<p>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。</p>

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	989 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 45 章

個別のテストのオプション設定

テストごとにテスト・オプションを設定することで、QuickTest による個々のテストの使用方法を制御できます。

本章の内容

- ▶ [テストの設定] ダイアログ・ボックスの使用 (1252 ページ)
- ▶ テストのためのプロパティの定義 (1255 ページ)
- ▶ テストのための実行設定の定義 (1259 ページ)
- ▶ テストのためのリソース設定の定義 (1263 ページ)
- ▶ テストのパラメータの定義 (1269 ページ)
- ▶ テストのための環境設定の定義 (1271 ページ)
- ▶ テストのための回復シナリオ設定の定義 (1279 ページ)
- ▶ テストのためのシステム監視の有効化 (1284 ページ)

[テストの設定] ダイアログ・ボックスの使用

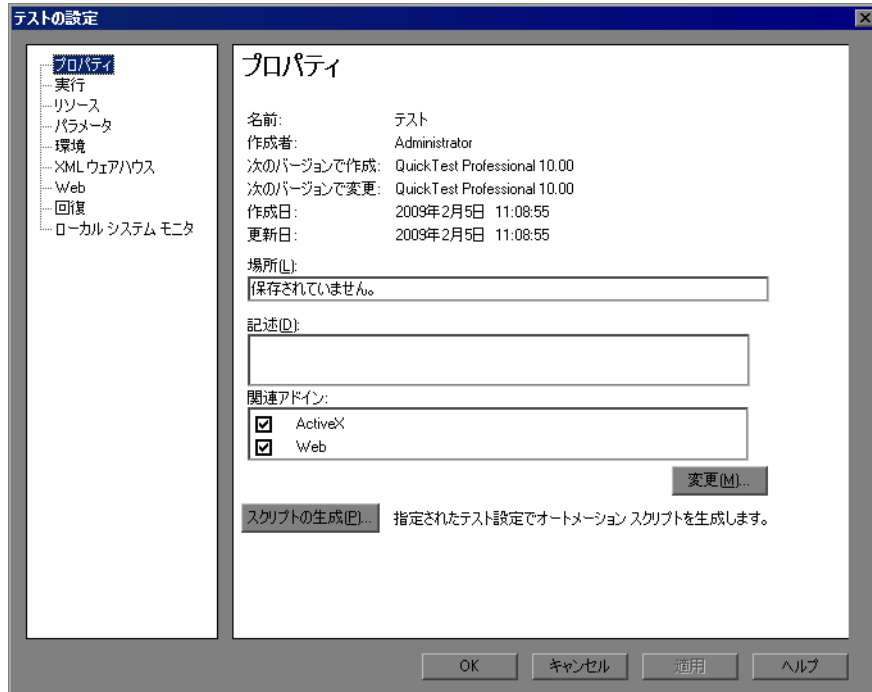
[テストの設定] ダイアログ・ボックスを使用して、QuickTest による特定のテストの使用方法に影響を与えるテスト・オプションを設定できます。たとえば、データ・テーブルの特定の行についてだけパラメータ化されたテストを実行するように、QuickTest に指示することができます。指定した個々のテスト・オプションは、テストを保存するときに保存されます。

注：すべてのテス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 44 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。

単独のテストのテスト・オプションを設定するには、次の手順を実行します。



- 1 [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。[テストの設定] 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、右側は設定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、設定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するには [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。

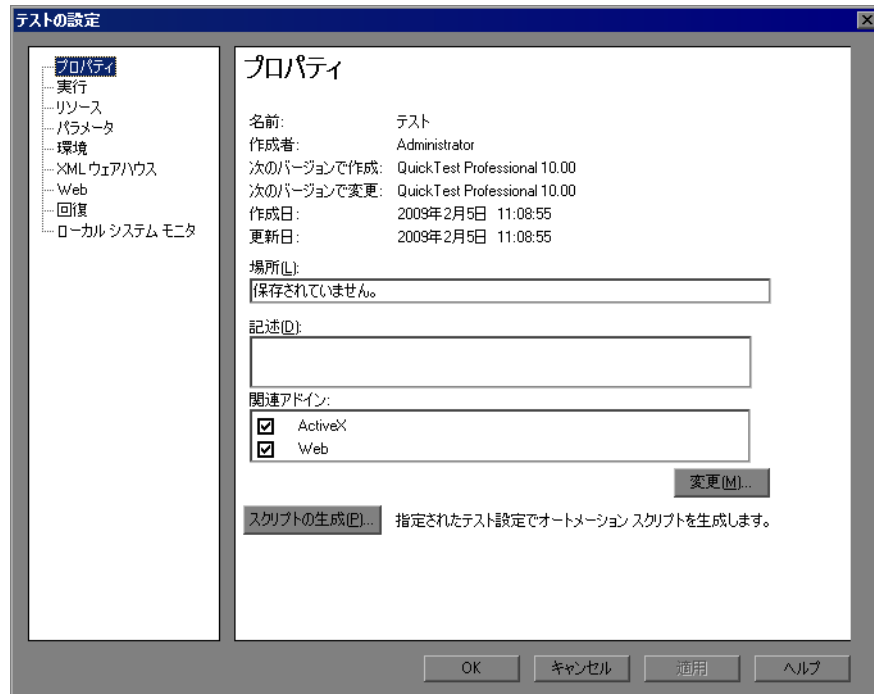
ナビゲーション・ツリーには次のノードが含まれています。

ノード	オプション
[プロパティ]	テストの説明や関連アドインなど、テストのプロパティを設定するためのオプション。詳細については、1255 ページ「テストのためのプロパティの定義」を参照してください。
[実行]	テストの実行セッションに関する設定を行うためのオプション。詳細については、1259 ページ「テストのための実行設定の定義」を参照してください。
[リソース]	関数ライブラリに格納されている VBScript 関数ライブラリなど、テストに関連付けるリソースを指定するオプション。詳細については、1263 ページ「テストのためのリソース設定の定義」を参照してください。
[パラメータ]	テス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、1269 ページ「テストのパラメータの定義」を参照してください。
[環境]	既存およびユーザ定義の環境変数の表示のほか、ユーザ定義環境変数の追加、変更、保存、およびアクティブな外部環境変数ファイルの選択のためのオプション。詳細については、1271 ページ「テストのための環境設定の定義」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 QuickTest が回復する方法を設定するオプション。詳細については、1279 ページ「テストのための回復シナリオ設定の定義」を参照してください。
[ローカル システム モニタ]	実行セッション中にシステム・カウンタを追跡するための設定を行い、有効にするオプション。詳細については、1284 ページ「テストのためのシステム監視の有効化」を参照してください。

[テストの設定] ダイアログ・ボックスにはこれらのノードのほかに、インストールまたは読み込まれている QuickTest アドインがあれば、それらに対応するノードが含まれることもあります。アドインの詳細については、『**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。

テストのためのプロパティの定義

[テストの設定] ダイアログ・ボックス ([ファイル] > [設定] > [プロパティ] ノード) の [プロパティ] 表示枠を使用して、テストに関連付けられているアドインなど、テストに関する一般情報の表示および定義が可能です。また、テスト設定のためのオートメーション・スクリプトを生成することもできます。



[テストの設定] ダイアログ・ボックスの [プロパティ] 表示枠には、次のオプションがあります。

オプション	説明
[名前]	テストの名前を示します。テストが Quality Center のバージョン・コントロール・プロジェクトに保存されている場合は、バージョン番号も表示されます。
[作成者]	テストを作成した人物の Windows ユーザ名を示します。

オプション	説明
[次のバージョンで作成]	テス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]	最後にテストに変更を加えた QuickTest のバージョンを示します。
[作成日]	テストが作成された日時を示します。
[更新日]	テストが最後に変更された日時を示します。
[場所]	テストのパスおよびファイル名を示します。
[記述]	テストの説明を指定できます。
[関連アドイン]	テストに関連付けられているアドインが表示されます。詳細については、1256 ページ「テストへのアドインの関連付け」を参照してください。
[変更]	テストに関連付けるアドインを選択できます。詳細については、1257 ページ「関連アドインの変更」を参照してください。
[スクリプトの生成]	現在のテスト設定を含んだオートメーション・スクリプトを生成します。詳細については、1387 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。

テストへのアドインの関連付け

アドインを選択して読み込むには、QuickTest の起動時に [アドイン マネージャ] ダイアログ・ボックスで対象アドインを選択します。必要なアドインが読み込まれている任意の環境で使用するテストを作成および編集できます。

新しいテストを作成すると、現在読み込まれているアドインが自動的にテストに関連付けられます。

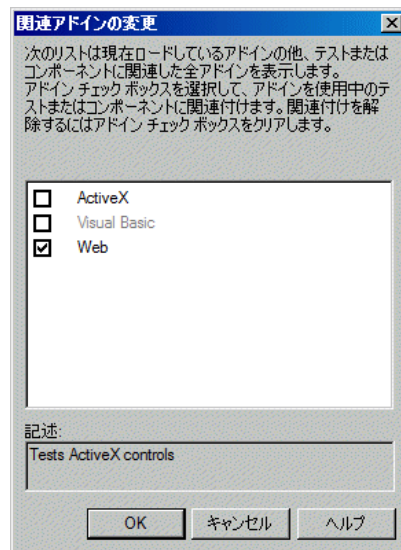
アドインをテストに関連付けると、そのテストを開くたびに、関連付けられているアドインが読み込まれているかどうか QuickTest によってチェックされます。

テストを開いたとき、関連付けられているアドインが現在読み込まれていないか、テストに関連付けられていないアドインが読み込まれている場合は、そのことが QuickTest によって通知されます。このチェック処理により、アドインを読み込んでいないことが原因で実行セッションが失敗するのを確実に防止できます。また、現在開いているテストで使用するアドインを、必須アドインとして関連アドイン・リストに追加できます。アドインの読み込みと使用方法の詳細については、『HP QuickTest Professional アドイン・ガイド』を参照してください。

Quality Center では、関連付けられているアドイン・リストを使用して、テストの実行または表示のために QuickTest を起動する際に読み込むアドインを決定します。Quality Center での作業の詳細については、第 51 章「Quality Center との統合」を参照してください。

関連アドインの変更

アドインをテストに関連付けたり、関連付けを解除したりするには、[関連アドインの変更] ダイアログ・ボックスを使います。



このダイアログ・ボックスには、テストに現在関連付けられているすべてのアドインが、現在 QuickTest に読み込まれているその他のアドインとともに一覧表示されます。テストに関連付けられているが現在読み込ま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、追加の環境またはコントロールをサポートするためにユーザまたはサードパーティがアドイン拡張を使用して開発したアドインを表す子ノードが含まれている場合があります。詳細については、関連するアドイン拡張の開発者ガイドを参照してください（拡張のセットアップに含まれています）。

テストに関連付けるアドインのチェック・ボックスを選択したり、テスト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たりできます。[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、その親アドインも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、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 が読み込まれ、テストに関連付けられている。
- ▶ ActiveX は読み込まれているが、テストに関連付けられていない。
- ▶ Visual Basic はテストに関連付けられているが、読み込まれていない。

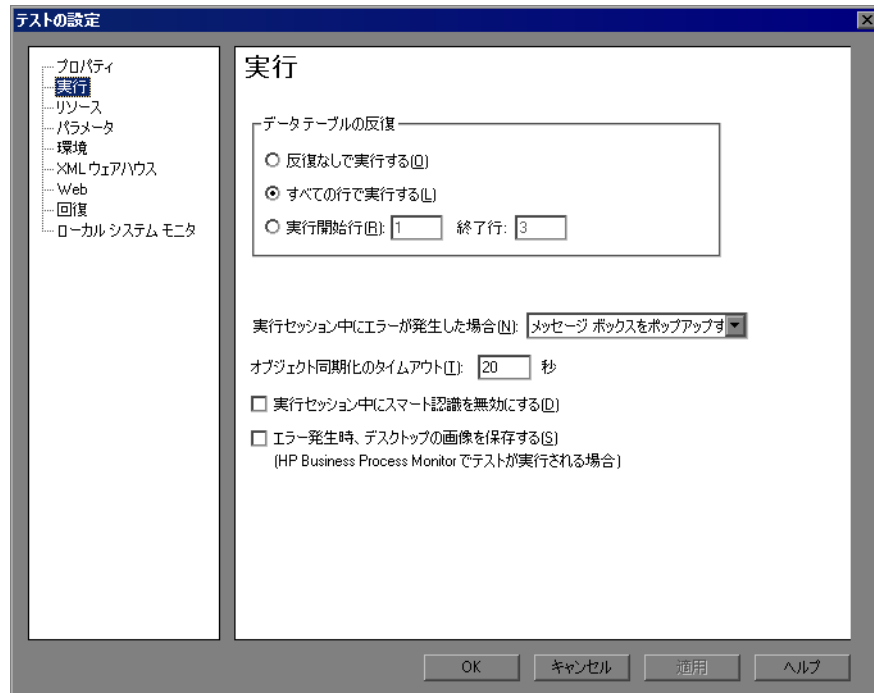
注：現在読み込まれていないアドインをテストに関連付けるには、QuickTest を再起動して、そのアドインを [アドイン マネージャ] から読み込みます。QuickTest の起動時に [アドイン マネージャ] ダイアログ・ボックスが表示されない場合には、QuickTest の次回起動時に表示されるように設定できます。それには、[オプション] ダイアログ・ボックスの [一般] 表示枠で、**[起動時にアドイン マネージャを表示する]** を選択します。[オプション] ダイアログ・ボックスの詳細については、第 44 章「グローバル・テスト・オプションの設定」を参照してください。アドイン・マネージャの詳細については、『**HP QuickTest Professional アドイン・ガイド**』の QuickTest アドインでの作業に関する項を参照してください。

オートメーション・スクリプトを使用してこのリストを取得し、リストに基づいてアドインを読み込みます。オートメーション・スクリプトでの作業の詳細については、『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]）を参照してください。

テストのための実行設定の定義

テストを実行すると、QuickTest によってアプリケーションでテスト・ステップが実行されます。

[テストの設定] ダイアログ・ボックス（[ファイル] > [設定] > [実行] ノード）の [実行] 表示枠を使って、実行セッション中にエラーが生じた場合に何をするかを選択し、オブジェクト同期タイムアウトを設定し、スマート認識メカニズムを無効にするかどうかを選択できます。



標準設定では、グローバル・パラメータを持つテストを実行した場合、QuickTest によって、指定したパラメータに従って、データ・テーブルの行ごとにテストが実行されます。詳細については、651 ページ「グローバルまたはアクション・データ・テーブル・パラメータの選択」を参照してください。

[実行] 表示枠を使用して、データ・テーブルの [グローバル] タブの特定の行に限定して、テストの反復を実行するように QuickTest に指示できます。

注： [テストの設定] ダイアログ・ボックスの [実行] 表示枠は、テスト全体に適用されます。テストの個々のアクションの実行プロパティを設定するには、選択したアクションの [アクション呼び出しプロパティ] ダイアログ・ボックスの [実行] タブを使用します。アクションの実行プロパティの詳細については、485 ページ「アクションの実行プロパティの設定」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
[データ テーブルの反復]	テストの反復の設定を行います。次のいずれかのオプションを選択します。 <ul style="list-style-type: none"> ▶ [反復なしで実行]：グローバル・データ・テーブルの最初の行だけを使って、テストを 1 回だけ実行します。 ▶ [すべての行で実行する]：グローバル・データ・テーブルのすべての行を使って、テストを反復実行します。 ▶ [実行開始行 X 終了行]：グローバル・データ・テーブルの指定した範囲にある行の値を使って、テストを反復実行します。
[実行セッション中にエラーが発生した場合]	実行セッション中に発生したエラーに対応する QuickTest の応答を指定します。詳細については、1262 ページ「エラーに対する応答の指定」を参照してください。

オプション	説明
[オブジェクト同期化のタイムアウト]	<p>QuickTest がテストのステップを実行する前に、オブジェクトの読み込みが終わるのを待機する最長の時間（秒単位）を設定します。</p> <p>注： Web オブジェクトで作業しているとき、QuickTest は [ブラウザナビゲーションのタイムアウト] に設定されている時間と [オブジェクト同期化のタイムアウト] に対して設定されている時間の合計の長さだけ待機します。[ブラウザナビゲーションのタイムアウト] オプショ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。</p>
[実行セッション中にスマート認識を無効にする]	<p>実行セッション中にスマート認識メカニズムを使わないように QuickTest を設定します。</p> <p>注： このオプションを選択すると、[オブジェクトのプロパティ] と [オブジェクトリポジトリ] ダイアログ・ボックスの中の [スマート認識を有効にする] チェック・ボックスが無効になります。ただし設定は保存されています。このオプションをクリアすると、[スマート認識を有効にする] チェック・ボックスは以前の設定（オンまたはオフ）に戻ります。</p>
[エラー発生時、デスクトップの画像を保存する (HP Business Process Monitor でテストが実行される場合)]	<p>このオプションは、HP Business Availability Center の Business Process Monitor コンポーネントによって実行されるテストにのみ適用されます。</p> <p>このオプションを選択すると、Business Process Monitor によって開始されたテストの実行セッションでエラーが生じたときにデスクトップの画面ショットを取得するよう QuickTest が指示されます。取得した画像は Business Availability Center に保存されます。実行結果は、Business Process Monitor によって Business Availability Center サーバに転送されます。</p>

エラーに対する応答の指定

標準の設定では、実行セッション中にエラーが生じると、エラーを説明するポップアップ・メッセージ・ボックスが QuickTest によって表示されます。実行セッションを続けるか終了するには、メッセージ・ボックスのボタンをクリックします。

「ポップアップ・メッセージ・ボックス」オプションを受け入れるか、**[実行セッション中にエラーが発生した場合]** ボックスのリストにある次のいずれかのオプションを選択することで、別の応答方法を指定できます。

- ▶ **次のアクション反復に進む**：エラーが発生したときに、QuickTest は次の反復に進みます。
- ▶ **実行を停止する**：エラーが発生したときに、QuickTest はテストを停止します。
- ▶ **次のステップに進む**：エラーが発生したときに、QuickTest はテストの次のステップに進みます。

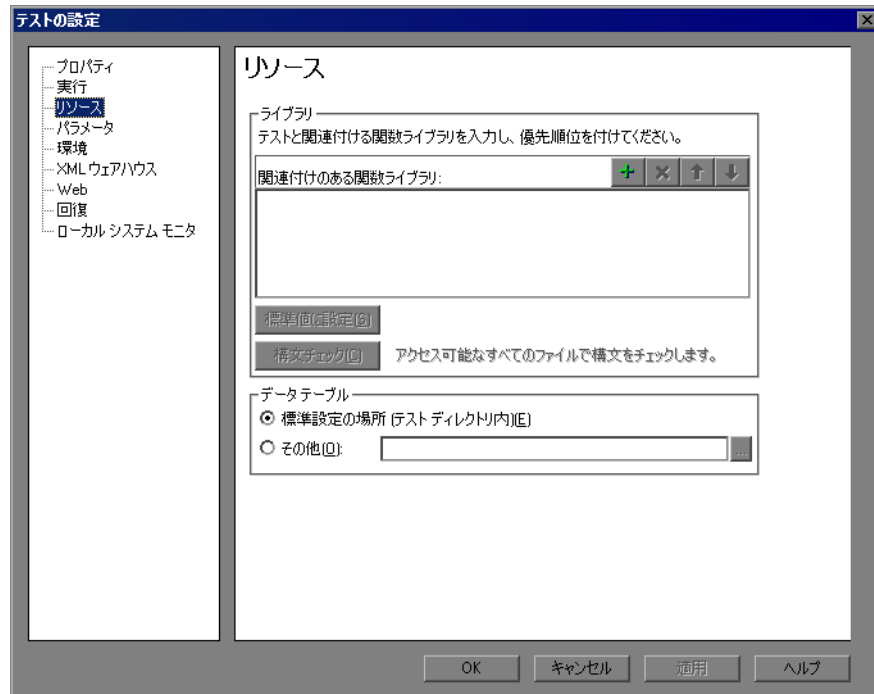
QuickTest では、まずテストに関連付けられている回復シナリオがあればそれが実行され、回復シナリオでエラーを解消できない場合にのみ、前述の手順で選択したオプションが実行されます。詳細については、1279 ページ「テストのための回復シナリオ設定の定義」を参照してください。

注：多数のテストで作業している場合は、QuickTest オートメーション・スクリプトを使用してテストごとに異なる値を設定する必要がある場合があります。このオプションを制御するオートメーション・スクリプトの行にアクセスするには、[テストの設定] ダイアログ・ボックスの [プロパティ] 表示枠にある **[スクリプトの生成]** ボタンを使用します。

詳細については、1387 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。

テストのためのリソース設定の定義

[テストの設定] ダイアログ・ボックス ([ファイル] > [設定] > [リソース] ノード) の [リソース] 表示枠を使って、VBScript 関数ライブラリやデータ・テーブル・ファイルなど、特定のファイルをテストに関連付けます。また、現在関連付けられている関数ライブラリの設定を、以降すべての新規テストの標準設定として設定できます。



注: オブジェクト・リポジトリは、テスト内の個々のアクションに関連付けられます。オブジェクト・リポジトリをアクションに関連付けるには、[アクションのプロパティ] ダイアログ・ボックス ([編集] > [アクション] > [アクションのプロパティ]) および [リポジトリの関連付け] ダイアログ・ボックス ([リソース] > [リポジトリの関連付け]) を使用します。

[テストの設定] ダイアログ・ボックスの [リソース] 表示枠には、次のオプション領域があります。

オプション領域	説明
[ライブラリ]	<p>テストに関連付けられている関数ライブラリが表示されます。ファイルの追加、削除、優先順位設定が可能です。また、以降の新規テストの標準関数ライブラリを設定できます。詳細については、1265 ページ「関連関数ライブラリの指定」を参照してください。</p>
[標準値に設定]	<p>関数ライブラリの現在のリストを、以降の新規テストに関連付ける標準設定のリストとして設定します。</p> <p>注：[標準値に設定] オプションは、テストが対象の場合にのみ使用できます。このオプションは、対象テストの設定が全テストの標準設定値と異なる場合に有効になります。</p> <p>注意：標準の関数ライブラリの移動または名前の変更が行われると、QuickTest はその関数ライブラリを認識できなくなります。新しいアクションまたはテストの作成時、この関数ライブラリは [欠落リソース] 表示枠に表示されます。欠落リソースの解決の詳細については、第 41 章「欠落リソースの処理」を参照してください。</p>
[構文チェック]	<p>関連付けられた関数ライブラリにテストの正常な実行を妨げる構文エラーが含まれているかどうかをチェックします。テストを確定する前に、[構文チェック] ボタンをクリックしてファイルの構文エラーをチェックしてください。構文エラーが見つかり、[情報] 表示枠が開き、構文エラーを含んでいるファイルのリストが表示されます。エラーがなければ、すべての関数ライブラリの構文が有効であることを知らせる情報ボックスが開きます。</p> <p>注：QuickTest は、アクセスできる関連付けられた関数ライブラリのみを検査します。たとえば、関連付けられた関数ライブラリが、現在接続されていない Quality Center プロジェクトに格納されている場合、その構文は検査されません。</p>

オプション領域	説明
[データ テーブル]	<p>テストで使用するデータ・テーブルの場所を指定します。</p> <ul style="list-style-type: none"> ▶ [標準設定の場所 (テスト ディレクトリ内)] を選択すると、テスト・フォルダの下にあるデータ・テーブルの標準の格納場所に保存されているデータを使用するよう、QuickTest が設定されます。 ▶ [その他] を選択すると、データ・テーブルが格納されている指定の場所に保存されているデータを使用するよう、QuickTest が設定されます。データ・テーブルには、任意の Microsoft Excel (.xls) ファイルを使用できます。 <p>データ・テーブルの詳細については、1185 ページ「データ・テーブルを使った作業について」を参照してください。</p> <p>注： Quality Center に格納されている Microsoft Excel ファイルをデータ・テーブルとして指定できます。詳細については、1200 ページ「Quality Center でのデータ・テーブル・ファイルの使用」を参照してください。</p>

関連関数ライブラリの指定

[リソース] 表示枠の [関連付けのある関数ライブラリ] 領域は、テストに関連付けられている関数ライブラリのリストを示します。それらのファイルの中から、テストで指定した VBScript 関数、サブルーチンなどが検索されます。

テストのステップから呼び出される関数またはサブルーチンが QuickTest によって検索される順序は、このリスト内の関数ライブラリの順序によって決まります。同じ名前の関数またはサブルーチンが 2 つある場合、QuickTest は最初に見つけた方を使用します。詳細については、920 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。

関連関数ライブラリは、絶対パスまたは相対パスを使って指定できます。相対パスを使って指定した場合、実行セッション中、QuickTest は現在のテストのディレクトリでファイルを検索し、次に [オプション] ダイアログ・ボックスの [フォルダ] 表示枠に表示されるフォルダで検索を行います。詳細については、1226 ページ「テストのフォルダ・オプションの設定」および 322 ページ「QuickTest での相対パスの使用」を参照してください。





注：

- ▶ テストで使用する関数ライブラリがファイル・システムに格納されていて、そのテストをほかのユーザまたは HP 製品がほかのコンピュータ上で実行できるようにする場合は、パスを相対パスとして指定することができます（パスを一度クリックするとパスが強調表示され、その状態で再度クリックすると編集モードになります）。以後はほかのユーザも、QuickTest で相対パスを参照するドライブ文字とフォルダを [オプション] ダイアログ・ボックス ([ツール] > [オプション] > [フォルダ] ノード) の [フォルダ] 表示枠で指定することにより、同じテストを実行できるようになります。詳細については、1226 ページ「テストのフォルダ・オプションの設定」および 322 ページ「QuickTest での相対パスの使用」を参照してください。

重要： 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、関数ライブラリを Quality Center テスト・リソース・モジュールに格納し、[フォルダ] 表示枠で 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、1432 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。


- ▶ [リソース] 表示枠を使用して、テストに関連付けられている関数ライブラリの追加、削除、優先順位設定を行うこともできます。詳細については、1149 ページ「[リソース] 表示枠」を参照してください。
-

次の関数ライブラリ制御ボタンを使用して、テストに関連付けられている関数ライブラリの追加、削除、優先順位設定が可能です。

オプション	説明
	<p>関数ライブラリをテストに関連付けます。関数ライブラリの絶対パスまたは相対パス、およびファイル名を入力するか、参照ボタンを使用して必要なファイルを指定します。関数ライブラリに構文エラーがある場合、構文エラーが原因で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。</p> <p>関数ライブラリは、ファイル・システムまたは Quality Center プロジェクト・フォルダに置くことができます。Quality Center に格納されている関数ライブラリの関連付けの詳細については、次に示す 1267 ページ「Quality Center プロジェクト・フォルダ内の関数ライブラリの関連付け」を参照してください。</p> <p>注： 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 の絶対パスを指定してください。詳細については、1432 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。</p>
	<p>リストから関連付けられている関数ライブラリを削除します。</p>
	<p>選択した関数ライブラリの優先順位を上げます。</p>
	<p>選択した関数ライブラリの優先順位を下げます。</p>

Quality Center プロジェクト・フォルダ内の関数ライブラリの関連付け

Quality Center に接続されている状態で  ボタンをクリックすると、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 に接続していないときに Quality Center プロジェクト・フォルダ内のファイルを追加するには、SHIFT キーを押しながら  ボタンをクリックします。QuickTest によって [QualityCenter] が追加され、パスが入力できるようになります。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。

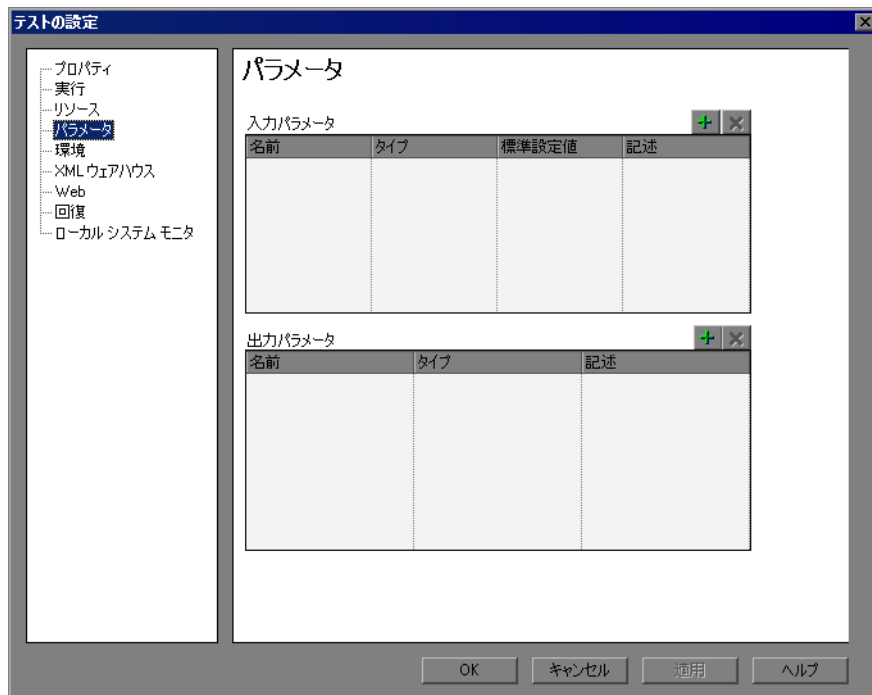
注：QuickTest でテストを実行する際には、対応する Quality Center プロジェクトに接続している場合にかぎり、Quality Center のプロジェクト・フォルダに格納されている関連付けられている関数ライブラリが使用されます。

Quality Center プロジェクトでの作業の詳細については、第 51 章「Quality Center との統合」を参照してください。

テストのパラメータの定義

[テストの設定] ダイアログ・ボックス ([ファイル] > [設定] > [パラメータ] ノード) の [パラメータ] 表示枠では、テストに値を渡す入力パラメータ、およびテストから外部のソースに値を渡す出力パラメータを定義します。また、既存のテスト・パラメータの変更や削除を行うこともできます。

テストのパラメータは、アクションのパラメータに似ています。アクション・パラメータの詳細については、477 ページ「アクション・パラメータの設定」を参照してください。





[パラメータ] 表示枠には、次のパラメータ・リストが表示されます。

- ▶ **[入力パラメータ]**：実行元または呼び出し元から渡される値をテストで受け取るためのパラメータを指定します。
- ▶ **[出力パラメータ]**：実行元または呼び出し元にテストから値を渡すための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、詳細を編集します。

テストの入力パラメータおよび出力パラメータを追加または削除するには、次のパラメータ制御ボタンを使用します。

オプション	説明
	<p>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（大文字と小文字は区別されます）を入力し、パラメータのタイプを選択してください。テス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。</p> <p>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」コラムで変更できます。詳細については、次に示す 1270 ページ「入力パラメータの値の定義」を参照してください。</p> <p>テストのパラメータは、アクション・パラメータと同様の方法で定義します。パラメータおよびパラメータ・タイプの定義方法の詳細については、477 ページ「アクション・パラメータの設定」を参照してください。</p>
	<p>選択したパラメータをテストから削除します。</p>

入力パラメータの値の定義

テス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、テストを呼び出したアプリケーション（QuickTest または Quality Center）から送信される値です。

ドキュメントのタイプ	呼び出し元	パラメータ値の指定場所
テスト	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、953 ページ「テスト全体の実行」を参照してください。
テスト	Quality Center	[テスト実行のプロパティ] ダイアログ・ボックス（Test Lab モジュール）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。

テストの実行時に、QuickTest または Quality Center から値が渡されない入力パラメータがあった場合、それらのパラメータについては標準設定値が使用されます。

[テストの設定] 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したり、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たりすることができます。

値のタイプ	QuickTest の標準設定値
文字列	空文字列
ブール	True
日付	現在の日付
数値	0
パスワード	空文字列
任意	空文字列

ステップにおけるテスト・パラメータの使用

テストのパラメータに直接アクセスできるのは、最上位アクションの入力パラメータをパラメータ化する場合および最上位レベルの出力パラメータの格納場所を指定する場合のみです。アクション内のステップでテスト・パラメータの値を使用するには、ステップを含んでいるアクションにテスト・パラメータを渡す必要があります。詳細については、477 ページ「アクション・パラメータの設定」を参照してください。

テストのための環境設定の定義

[テストの設定] ダイアログ・ボックス ([ファイル] > [設定] > [環境] ノード) の [環境] 表示枠には、既存の組み込み環境変数およびユーザ定義環境変数が表示されます。また、このタブでは、ユーザ内部定義環境変数の追加、変更、および削除が可能なほか、定義されている変数の外部 XML ファイルへの保存、および、ファイルからの変数の取得が可能です。

ユーザ定義変数を外部 XML ファイルにエクスポートすれば、エクスポートした環境変数ファイルをほかの任意のテストで使用できます。

環境変数および環境パラメータの詳細については、653 ページ「環境変数パラメータの使用」を参照してください。

[環境] 表示枠には、[**変数のタイプ**] に関して次のオプションがあります。

- ▶ **組み込み**：QuickTest Professional によって定義される組み込み環境変数およびそれらの現在の値が表示されます。
- ▶ **ユーザ定義**：内部および外部のユーザ定義環境変数およびそれらの現在の値が表示されます。

組み込み環境変数

[**組み込み**] を選択すると、[環境] 表示枠には QuickTest Professional によって定義される組み込み環境変数のリストが表示されます。



組み込み環境変数に関して、次の情報が表示されます。

- ▶ [**名前**]：各組み込み環境変数の名前。
- ▶ [**詳細**]：各組み込み環境変数の簡単な説明。

- ▶ **[現在の値]**：選択した環境変数の現在の値。

ユーザ定義環境変数

[**ユーザ定義**] を選択すると、[環境] 表示枠には、そのテストで利用できるユーザ定義環境変数のリストが表示されます。






注：外部環境変数ファイルの変数は、青い文字で表示されます。内部環境変数は、黒い文字で表示されます。

[環境] 表示枠は、ユーザ定義環境変数に関して次の情報を提供します。

- ▶ [名前] : 各ユーザ定義変数の名前。
- ▶ [値] : 各ユーザ定義変数に割り当てられた値。
- ▶ [タイプ] : 各ユーザ定義変数のタイプ。**内部**または**外部**のいずれかのタイプ。内部環境変数は、その変数が定義されているテスト内においてのみ使用できます。


[環境] 表示枠は、ユーザ定義環境変数に関して次のオプションを提供します。

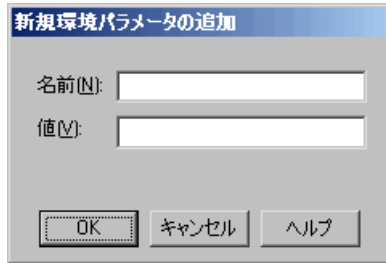
オプション	説明
	新規環境変数を定義してリストに追加できます。詳細については、次に示す「ユーザ定義環境変数の追加」を参照してください。
	選択した内部環境変数をリストから削除します。 注 : 環境変数の削除をいったん確定すると、[テストの設定] ダイアログ・ボックスで [キャンセル] をクリックしても、変数は元に戻せません。
	選択した内部環境変数の編集および選択した外部環境変数のプロパティの表示を可能にします。詳細については、1275 ページ「ユーザ定義環境変数の表示および変更」を参照してください。
[エクスポート]	ユーザ定義環境変数を外部 XML ファイルにエクスポートし、ほかのテストで使用できるようにします。エクスポートした環境変数ファイルは、任意のテストで使用できます。詳細については、1277 ページ「ユーザ定義環境変数のエクスポートと読み込み」を参照してください。
[変数と値を外部ファイルからロードする]	テストで使用するよう指定した XML ファイルから、保存されている変数を読み込みます。詳細については、1277 ページ「ユーザ定義環境変数のエクスポートと読み込み」を参照してください。

ユーザ定義環境変数の追加

ユーザ定義内部変数の追加は、[テストの設定] ダイアログ・ボックスの [環境] 表示枠を使用して行います。内部環境変数は、それが定義されているテストだけで使用できます。

ユーザ定義内部環境変数を追加するには、次の手順を実行します。

- 1 [環境] 表示枠の [**変数のタイプ**] ボックスで [**ユーザ定義**] を選択します。
- 2  [**新規作成**] ボタンをクリックします。[新規環境パラメータの追加] ダイアログ・ボックスが開きます。



- 3 変数の定義を次のように入力します。
 - ▶ [**名前**] : 変数の名前を入力します。
 - ▶ [**値**] : 変数の値を入力します。
- 4 [**OK**] をクリックして保存し、[新規環境パラメータの追加] ダイアログ・ボックスを閉じます。変数が、[テストの設定] ダイアログ・ボックスの [環境] 表示枠のリストに追加されます (黒色で表示されます)。

ユーザ定義環境変数の表示および変更

ユーザ定義内部変数の編集は、[テストの設定] ダイアログ・ボックスの [環境] 表示枠を使用して行います。外部ユーザ定義変数のプロパティを表示することも可能です。

内部変数および外部変数の値をコピーして、データ・テーブルなど、QuickTest のほかの場所で使用できます。

ユーザ定義内部環境変数を変更するには、次の手順を実行します。



- 1 [テストの設定] ダイアログ・ボックスの [環境] 表示枠で、変更する内部変数をダブルクリックするか、内部変数を選択して [環境変数の表示 / 編集] ボタンをクリックします。[環境パラメータの編集] ダイアログ・ボックスが開きます。

- 2 変数の値を変更するには、[値] ボックスに異なる値を入力します。
- 3 変数の値をクリップボードにコピーするには、値のテキストを選択し、右クリックして [コピー] を選択します。
- 4 [OK] をクリックして保存し、[環境パラメータの編集] ダイアログ・ボックスを閉じます。[テストの設定] ダイアログ・ボックスの [環境] 表示枠上で、変数の値が更新されます。

ユーザ定義の外部環境変数を表示するには、次の手順を実行します。



- 1 [テストの設定] ダイアログ・ボックスの [環境] 表示枠で、表示する外部変数をダブルクリックするか、外部変数を選択して [環境変数の表示 / 編集] ボタンをクリックします。[環境パラメータの表示] ダイアログ・ボックスに、選択した変数の詳細が表示されます。



変数の値が複合的な値の場合（値全体を [値] ボックスに表示できない場合）、値の内容を表示するには、[複雑な値の表示 / 編集] ボタンをクリックします。

- 2 変数の値をクリップボードにコピーするには、値のテキストを選択し、右クリックして [コピー] を選択します。
- 3 [閉じる] をクリックし、[環境パラメータの表示] ダイアログ・ボックスを閉じます。

ユーザ定義環境変数のエクスポートと読み込み

ユーザ定義環境変数は、外部 XML ファイルにエクスポートすることにより、ほかのテストで使用できます。エクスポートした環境変数は、ファイルから外部ユーザ定義変数として読み込み、任意のテストで使用できます。

ファイルをファイル・システムに保存した場合、値はテストを実行するたびに読み込まれます。ファイルを Quality Center プロジェクトに保存した場合、値はテストを最初に読み込むときに読み込まれます。テストの読み込み後に値が変更されても、新しい値は次にテストを読み込むまでは使用されません。

ユーザ定義環境変数をエクスポートするには、次の手順を実行します。

- 1 [テストの設定] ダイアログ・ボックスの [環境] 表示枠で、[エクスポート] ボタンをクリックします。ユーザ定義変数と値の現在のリストを XML ファイルにエクスポートできる [環境変数ファイルの保存] 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（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。

ヒント： ファイルをテストの添付ファイルとして Quality Center のテスト計画モジュールに保存する場合は、サイドバーの [Quality Center テスト計画] を選択し、テストを参照してダブルクリックし、[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、322 ページ「QuickTest での相対パスの使用」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、1432 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。

ユーザ定義外部環境変数ファイルから変数を読み込むには、次の手順を実行します。

- 1 [テストの設定] ダイアログ・ボックスのナビゲーション表示枠で、[環境] ノードをクリックします。
- 2 [環境] 表示枠で、[変数のタイプ] ボックスから [ユーザ定義] を選択します。ユーザ定義変数のオプションが表示されます。
- 3 [変数と値を外部ファイルからロードする] チェック・ボックスを選択します。
- 4 [ファイル] ボックスで、ファイル名を入力するか、参照ボタンをクリックしてユーザ定義外部変数ファイルを探します。

選択したファイルから読み込まれた環境変数は、[テストの設定] ダイアログ・ボックスの [環境] 表示枠上に青色で表示されます。

注：環境変数ファイルの相対パスを入力すると、[オプション] ダイアログ・ボックスの [フォルダ] 表示枠で指定されているフォルダの中で、QuickTest によってファイルが検索されます。詳細については、1226 ページ「テストのフォルダ・オプションの設定」および 322 ページ「QuickTest での相対パスの使用」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、1432 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。

組み込み環境変数とユーザ定義変数の詳細について、および、外部ユーザ定義環境変数ファイルの作成方法については、653 ページ「環境変数パラメータの使用」を参照してください。

テストのための回復シナリオ設定の定義

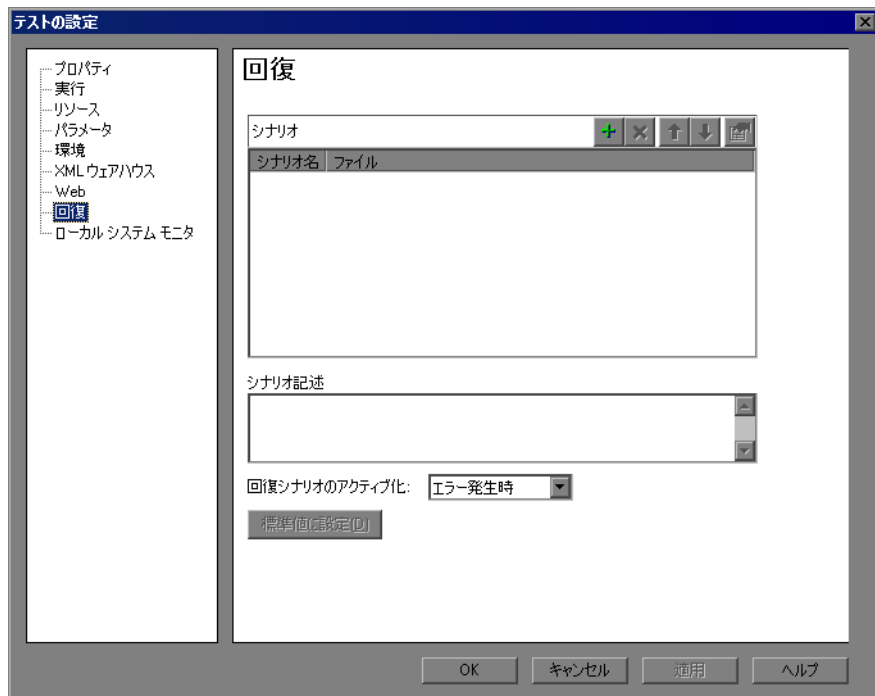
[テストの設定] ダイアログ・ボックス ([**ファイル**] > [**設定**] > [**回復**] ノード) の [回復] 表示枠には、現在のテストに関連付けられているすべての回復のシナリオのリストが表示されます。また、テストに関連付ける回復シナリオの追加、テストからのシナリオの削除、実行セッションに対する回復シナリオ適用順序の変更、および、各シナリオの表示専用サマリの表示が可能です。

対象のテストに対して、特定のシナリオの有効と無効を切り替えることも、回復メカニズム全体の有効と無効を切り替えることもできます。

テストに対する作業中には、現在のシナリオのリストを以降の新規テストすべてにおいて使用するよう指定できます。

[リソース] 表示枠で、テストへの回復シナリオの関連付け、およびテストに関連付けられている回復シナリオの削除、有効化、無効化、優先順位設定、プロパティ表示を行うこともできます。詳細については、1149 ページ「[リソース] 表示枠」を参照してください。

回復シナリオの詳細については、第 48 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。

オプション領域	説明
[シナリオ]	<p>テス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。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、次に示す第 45 章「関連回復シナリオの指定」を参照してください。</p>
[シナリオ記述]	<p>[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。</p>
[回復シナリオのアクティブ化]	<p>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 を設定します。</p> <ul style="list-style-type: none"> ▶ [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。 ▶ [エラー発生時]：回復メカニズムはエラーの戻り値を返すステップの後にのみ起動されます。 ▶ [なし]：回復メカニズムは無効にされます。 <p>注：[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。</p>
[標準値に設定]	<p>回復シナリオ・ファイルの現在のリストを、以降の新規テストに関連付ける標準設定のリストとして設定します。</p> <p>注：[標準値に設定] オプションは、テストが対象の場合にのみ使用できます。このオプションは、対象テストの設定が全テストの標準設定値と異なる場合に有効になります。</p> <p>注意：回復シナリオが含まれるファイルの移動または名前の変更が行われると、QuickTest はその関数ライブラリを認識できなくなります。新しいアクションまたはテストの作成時、この回復シナリオ・ファイルは [欠落リソース] 表示枠に表示されます。欠落リソースの解決の詳細については、第 41 章「欠落リソースの処理」を参照してください。</p>

注：テストで使用する回復ファイルがファイル・システムに格納されている場合、そのテストをほかのユーザまたは HP 製品がほかのコンピュータ上で実行できるようにするには、回復ファイルのパスを相対パスとして指定する必要があります（パスを一度クリックするとパスが強調表示され、その状態で再度クリックすると編集モードになります）。以後はほかのユーザも、QuickTest で相対パスを参照するドライブ文字とフォルダを [オプション] ダイアログ・ボックス ([ツール] > [オプション] > [フォルダ] ノード) の [フォルダ] 表示枠で指定することにより、同じテストを実行できるようになります。詳細については、1226 ページ「テストのフォルダ・オプションの設定」および 322 ページ「QuickTest での相対パスの使用」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、回復ファイルを Quality Center テスト・リソース・モジュールに格納し、[フォルダ] 表示枠で 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、1432 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。

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、現在のテストに対して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。たとえば、絶対ファイル・パスをファイルの相対ファイル・パスに変更したい場合があります。回復シナリオ・ファイルのパスを変更した場合は、テストを実行する前に、回復シナリオが新しいパスに存在することを確認する必要があります。

シナリオは、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、テスト内で正常に実行されないステップがあった場合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、そのテストから利用できなくなっていることを示します。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、メッセージがテスト結果に表示されます。

注： QuickTest にあらかじめ用意されている標準設定の回復シナリオは、QuickTest のインストール・フォルダにインストールされています。[回復] 表示枠内において標準設定の回復シナリオを指定するパスは、パスの中で環境変数 (**%ProductDir%**) を使用します。そうすることにより、当該回復シナリオに対応するテストを別のコンピュータや別の HP 製品で実行した場合にも、QuickTest でシナリオを発見できるようになります。これら標準設定の回復シナリオ・ファイルのパスを変更したり、この環境変数をほかの目的に使用したりしてはなりません。

次の回復シナリオ・ファイル制御ボタンを使用して、テス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、優先順位設定が可能です。

オプション	説明
	[回復シナリオの追加] ダイアログ・ボックスが開き、テストに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、1355 ページ「テストへの回復シナリオの追加」を参照してください。
	選択した回復シナリオをテストから削除します。
	選択したシナリオをリストの中で上に移動することで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することで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表示専用の形式で表示します。詳細については、1358 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。

テストのためのシステム監視の有効化

[テストの設定] ダイアログ・ボックス ([**ファイル**] > [**設定**] > [**ローカルシステム モニタ**] ノード) の [ローカルシステム モニタ] 表示枠を使って、実行セッション中にシステム・カウンタを追跡するための設定を行い、有効にすることができます。

ローカル・システム・モニタ追跡オプションにより、実行セッション中にアプリケーション・パフォーマンス・カウンタを追跡できるようになります。このカウンタでは、アプリケーションによって使用されるリソースを監視できません。

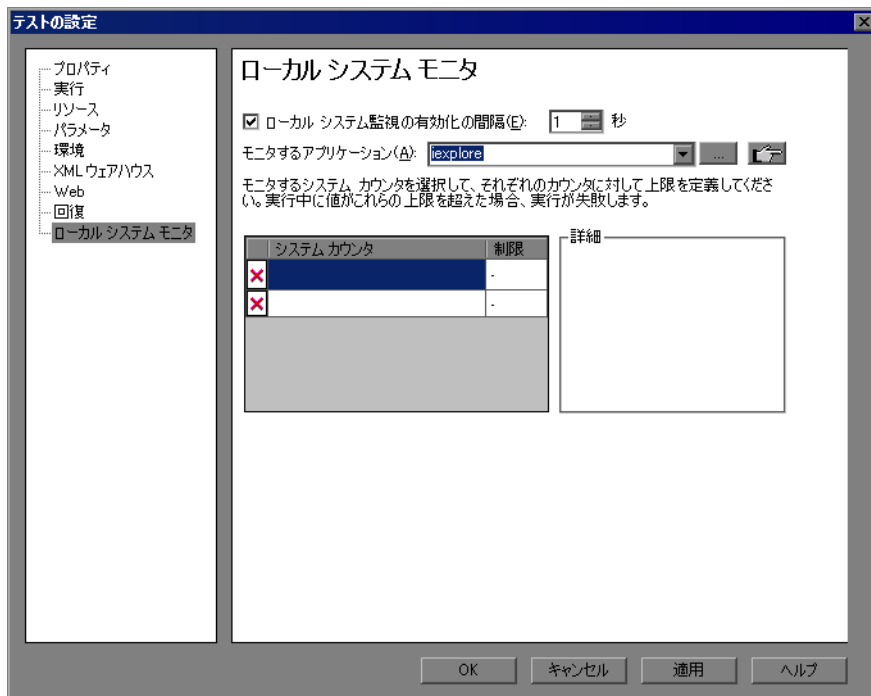
監視可能なシステム・カウンタは、パフォーマンス・コンソール ([**スタート**] > [**ファイル名を指定して実行**] を選択し、「Perfmon」と入力) からアクセスできるプロセス・カウンタです。パフォーマンス・コンソールからアクセスできるプロセス・カウンタの詳細については、パフォーマンス・コンソールのヘルプを参照してください。

カウンタに制限を定義することもできます。指定されたカウンタがその制限を超えた場合、テスト実行は失敗となります。システム・カウンタの結果は、[テスト結果] ウィンドウに表示されます。詳細については、1054 ページ「[システム モニタ] の結果表示」を参照してください。




[テストの設定] ダイアログ・ボックス : [ローカル システム モニタ] 表示枠


説明	システム監視を有効にし、実行セッション中に追跡するシステム・カウンタを定義できます。
アクセス方法	[ファイル] メニュー > [設定] 項目 > [ローカル システム モニタ] ノード
重要情報	<ul style="list-style-type: none"> ▶ テスト実行中にキャプチャされたローカル・システム・モニタ・データは、[テスト結果] ウィンドウに表示されます。詳細については、1054 ページ「[システム モニタ] の結果表示」を参照してください。 ▶ テスト時に同じ名前の複数のプロセスが実行されていて、そのプロセスのカウンタを監視した場合（たとえば、iexplorer.exe プロセスのカウンタの監視を選択し、実行セッション中にデスクトップに複数の Internet Explorer ブラウザが開いている場合）、カウンタは、テストの少なくとも 1 つのテスト・オブジェクトが含まれるアプリケーションからサンプリングされます。複数のアプリケーションがこの条件を満たす場合は、1 つのアプリケーションのみ監視されます。
詳細情報	概念的概要 : 1284 ページ「テストのためのシステム監視の有効化」。

[テストの設定] ダイアログ・ボックスの [ローカル システム モニタ] 表示枠の画像を次に示します。



[テストの設定] ダイアログ・ボックス : [ローカル システム モニタ] 表示枠のオプション

オプション	説明
<p>[ローカル システム監視の有効化の間隔: __ 秒]</p>	<p>このアプリケーションのシステム・カウンタがチェックされる頻度を秒単位で定義します。秒数を変更するには、上矢印または下矢印を使うか、エディット・ボックスに値を入力します。最小値は 1 秒です。</p>
<p>[モニタするアプリケーション]</p>	<p>システム・カウンタを監視するアプリケーションを定義します。次のいずれかの方法でアプリケーションを定義できます。</p> <ul style="list-style-type: none"> ▶ エディット・ボックスに、アプリケーションの実行ファイルの名前（ファイル拡張子なし）を入力します。 ▶ エディット・ボックスの下矢印をクリックし、QuickTest で以前実行されたアプリケーション、現在実行されているアプリケーション、[記録と実行環境設定] ダイアログ・ボックスの [Windows アプリケーション] タブに現在示されているアプリケーションのリストを表示します。 ▶ 参照ボタン  をクリックし、アプリケーションの実行ファイルを参照します。 ▶ アプリケーションが現在実行されていることを確認し、指差しアイコン  をクリックして、デスクトップ上のアプリケーションをポイントします。 <p>注： プロセスは、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起動プログラムとしてのみ使用される場合があります。指定した実行ファイルが、アプリケーションの機能を実際に提供する実行ファイルであることを確認してください。</p>
<p>[削除] ボタン </p>	<p>テストからシステム・カウンタの定義を削除するには、[削除] ボタンをクリックします。</p>

オプション	説明
<p>[システム カウンタ] カラム</p>	<p>選択したアプリケーションで追跡するシステム・カウンタを定義します。セルの中をクリックし、下矢印をクリックします。リストからカウンタを選択します。より多くのカウンタを表示するには、展開ボタン  をクリックします（表示されている場合）。</p> <p>監視可能なシステム・カウンタは、パフォーマンス・コンソール（[スタート] > [ファイル名を指定して実行]）を選択し、「Perfmon」と入力）からアクセスできるプロセス・カウンタです。パフォーマンス・コンソールからアクセスできるプロセス・カウンタの詳細については、パフォーマンス・コンソールのヘルプを参照してください</p>
<p>[制限] カラム</p>	<p>[システム カウンタ] カラムで選択したカウンタの上限を定義します。実行セッション中に、選択したカウンタがこの値を超えた場合、テストは失敗します。制限値は任意です。値を指定しなくてもカウンタは追跡され、[テスト結果] ウィンドウに結果が表示されます。</p>
<p>[説明]</p>	<p>[システム カウンタ] カラムで選択したカウンタの説明（パフォーマンス・コンソール・アプリケーションによって提供されたもの）が表示されます。</p>

第 46 章

実行セッション中にテスト・オプションを設定するための Setting オブジェクトの使用

Setting オブジェクトを使用して、実行セッション中にテスト・オプションを設定したり取得したりすることによって、QuickTest でのテストの実行方法を制御できます。

本章の内容

- ▶ 実行セッション中のテスト・オプションの設定について (1289 ページ)
- ▶ テスト・オプションの設定 (1290 ページ)
- ▶ テスト・オプションの取得 (1292 ページ)
- ▶ テスト実行の制御 (1292 ページ)
- ▶ テスト実行設定の追加と削除 (1293 ページ)

実行セッション中のテスト・オプションの設定について

QuickTest テスト・オプションは、テストの作業方法に影響します。たとえば、Web ページを読み込むときに、URL アドレスが見つからないと判断するまでに QuickTest で許容される最長時間を設定できます。

テスト・オプションの値を実行セッション中に設定および取得するには、エキスパート・ビューで **Setting** オブジェクトを使います。エキスパート・ビューでの作業の詳細については、第 29 章「エキスパート・ビューおよび関数ライブラリ・ウィンドウを使用した作業」を参照してください。

Setting オブジェクトを使ってテスト・オプションを取得および設定することで、QuickTest でのテストの実行方法を制御できます。

多くのテスト・オプションは、[オプション] ダイアログ・ボックス（グローバルなテスト・オプション）および [テストの設定] ダイアログ・ボックス（テスト固有の設定）を使っても設定できます。詳細については、第 44 章「グローバル・テスト・オプションの設定」および第 45 章「個別のテストのオプション設定」を参照してください。

本章では、テスト・スクリプト内から **Setting** オブジェクトを使って設定できる QuickTest のテスト・オプションの一部について説明します。**Setting** オブジェクトで使用可能なすべてのメソッドとプロパティの詳細については、『**HP QuickTest Professional Object Model Reference**』（英語版）の「**Utility**」の項を参照してください。

注：QuickTest オプションは、ほかの QuickTest 操作と同様、自動スクリプトを使って制御することもできます。詳細については、1387 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。

テスト・オプションの設定

Setting オブジェクトを使って、テスト・オプションの値をテスト・スクリプト内から設定できます。オプションを設定するには、次の構文を使います。

Setting (テスト・オプション) = 新変数

オプションの中にはグローバルなもの、現在のテストにのみ影響するものがあります。Setting オブジェクトを使用してテスト・オプションを設定すると、もう一度変更するか、または現在の QuickTest セッションが終了まで設定が有効です。Setting オブジェクトを使い、特定のテストの特定の部分の設定だけを変更することもできます。詳細については、1292 ページ「テスト実行の制御」を参照してください。

Setting オブジェクトを使用して設定できるテスト・オプションの一部は、[オプション] ダイアログ・ボックス（グローバル・オプション）または [テストの設定] ダイアログ・ボックス（テスト固有の設定）でも利用できます。

Setting オブジェクトを使用してこれらのオプションを設定すると、変更は関連するダイアログ・ボックスに反映されます。ほかのテスト設定は、関連するダイアログ・ボックスまたは Setting オブジェクトのうち、いずれかの方法を使用してのみアクセスできます。

例：[オプション] ダイアログ・ボックスに反映されるオプションを設定するための Setting オブジェクトの使用

Web Add-in を読み込んだ状態で、次のステートメントを実行します。

```
Setting("AutomaticLinkRun")=1
```

QuickTest により、テストの自動的に作成されたチェックポイントが無効になります。この設定は、現在の QuickTest セッション中に別の Setting ステートメントを使うか、[Web Advanced] 表示枠 ([ツール] > [オプション] > [Web] > [詳細設定] ノードを選択) の [テストまたはコンポーネントの実行時に自動チェックポイントを無視する] チェック・ボックスをクリアして設定を変更するまで有効です。

例：[テストの設定] ダイアログ・ボックスに反映されるオプションを設定するための Setting オブジェクトの使用

次のステートメントを実行します。

```
Setting("WebTimeOut")=50000
```

QuickTest によって、テストのステップを実行する前に Web ページの読み込みが終わるまで待機する時間の上限が、自動的に 50000 ミリ秒に変更されます。この設定は、現在の QuickTest セッション中に別の Setting ステートメントを使うか、[テストの設定] ダイアログ・ボックスの [Web] 表示枠で [ブラウザナビゲーションのタイムアウト] オプションを使って、設定を変更するまで有効です。

注： Setting オブジェクトを使って行った変更が [オプション] ダイアログ・ボックスと [テストの設定] ダイアログ・ボックスに反映されている場合でも、同じダイアログ・ボックスでほかの変更を手作業で行って [適用] または [OK] をクリックしなければ、これらの変更は QuickTest を閉じたときに保存されません。

テスト・オプションの取得

Setting オブジェクトを使って、テスト・オプションの現在の値を取得することもできます。

変数に値を格納するには、次の構文を使います。

新変数 = Setting (テスト・オプション)

メッセージ・ボックスに値を表示するには、次の構文を使います。

MsgBox (Setting (テスト・オプション))

次に例を示します。

```
LinkCheckSet = Setting("AutomaticLinkRun")
```

これは、AutomaticLinkRun 設定の現在の値を、ユーザ定義変数の LinkCheckSet に代入します。

設定を取得する対象にできるその他のテスト・オプションの例については、1290 ページ「テスト・オプションの設定」を参照してください。

テスト実行の制御

Setting オブジェクトの取得機能と設定機能を組み合わせることで、グローバル設定を変更せずに、実行セッションを制御できます。たとえば、DefaultTimeout テスト・オプションを、ある特定の Web ページ上にあるオブジェクトについてだけ 5 秒に変更するには、テスト・スクリプトの中でその Web ページを開く処理の後に、次のステートメントを挿入します。

```
' DefaultTimeout テスト・オプションの元の値を保存  
old_delay = Setting ("DefaultTimeout")
```

```
' DefaultTimeout テスト・オプションの一時値を設定  
Setting("DefaultTimeout")= 5000
```


Web ページの最後で DefaultTimeout テスト・オプションを元の値に戻すには、スクリプトの中で次ページにリンクする処理の直前に、次のステートメントを挿入します。

```
' DefaultTimeout テスト・オプションを元の値に戻す  
Setting("DefaultTimeout")=old_delay
```

テスト実行設定の追加と削除

グローバルな設定や固有の設定だけでなく、実行環境の設定も追加、変更、削除できます。これらの設定は、実行セッション中にのみ適用されます。

新しい実行環境の設定を追加するには、次の構文を使います。

Setting.Add "テスト・オプション", "値"

たとえば、現在のテスト実施者の名前を表示し、その名前をメッセージ・ボックスに表示する設定を作成できます。

```
Setting.Add "Tester Name", "Mark Train"  
MsgBox Setting("Tester Name")
```

ヒント : Setting.Add ステートメントを使用するときに、既存の設定オプションを追加しようとするとエラーが発生します。このエラーを防ぐには、先に **Setting.Exists** ステートメントを使う必要があります。すべての **Setting** 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。

すでに初期化されている実行環境設定を変更するには、標準の設定オプションを設定する場合と同じ構文を使います。

Setting (テスト・オプション) = 新しい値

次に例を示します。

```
Setting("Tester Name")="Alice Wonderlin"
```

ユーザ定義の実行環境の設定を削除するには、次の構文を使います。

Setting.Remove (テスト・オプション)

次に例を示します。

```
Setting.Remove ("Tester Name")
```

ヒント : `Setting.Remove` ステートメントを使用するときに、存在しない設定オプションを削除しようとするエラーが発生します。このエラーを防ぐには、先に `Setting.Exists` ステートメントを使う必要があります。すべての `Setting` 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 X 部

高度なテスト機能を使用した作業

第 47 章

仮想オブジェクトの学習

アプリケーションの任意の領域を「**仮想オブジェクト**」として定義することにより、QuickTest でこの領域がオブジェクトとして認識されるように設定できます。仮想オブジェクトを利用すると、QuickTest が通常ならば認識しないオブジェクトを対象とするテストの作成と実行が行えます。

コンピュータで定義されている仮想オブジェクトは、仮想オブジェクト・マネージャで管理できます。

本章の内容

- ▶ 仮想オブジェクトの学習について (1297 ページ)
- ▶ 仮想オブジェクトについて (1299 ページ)
- ▶ 仮想オブジェクト・マネージャについて (1300 ページ)
- ▶ 仮想オブジェクトの定義 (1302 ページ)
- ▶ 無効な仮想オブジェクト定義の削除と (1315 ページ)

仮想オブジェクトの学習について

アプリケーションには、標準のオブジェクトと同様に動作するにもかかわらず、QuickTest が認識できないオブジェクトが含まれることがあります。これらのオブジェクトを仮想オブジェクトとして定義し、ボタンやチェック・ボックスなどの標準クラスに割り当てることができます。これにより、実行セッション中、QuickTest によって、この仮想オブジェクトに対するユーザのアクションがエミュレートされます。テスト結果には、仮想オブジェクトは標準クラスのオブジェクトとして表示されます。

たとえば、ユーザがクリックするビットマップが含まれる Web ページをテストするとします。このビットマップには、それぞれ異なるリンク・ページを表示するいくつかのハイパーリンク領域があります。テストを作成するときに、Web サイトでは、クリックしたビットマップの座標が検出され、リンク・ページが表示されます。

実行セッション中に QuickTest で必要な座標をクリックできるようにするため、その座標が含まれるビットマップ領域の仮想オブジェクトを定義し、これをボタン・クラスに割り当てます。テストを実行すると、QuickTest によって、仮想オブジェクトとして定義した領域のビットマップがクリックされ、対応するリンク・ページが Web サイトに表示されます。

仮想オブジェクトの定義は、仮想オブジェクト・ウィザードで行います ([ツール] > [仮想オブジェクト] > [新規仮想オブジェクト])。このウィザードでは、仮想オブジェクトの割り当ての対象にする標準のオブジェクト・クラスを選択するよう指示されます。その後、十字形のポインタで、仮想オブジェクトの境界を指定します。次に、この仮想オブジェクトの親とする、テスト・オブジェクトを選択します。最後に、仮想オブジェクトの名前と **コレクション** を指定します。詳細については、1302 ページ「仮想オブジェクトの定義」を参照してください。

仮想オブジェクトのコレクションとは、仮想オブジェクト・マネージャに格納されている仮想オブジェクトの集合に名前を付けたものです。詳細については、1300 ページ「仮想オブジェクト・マネージャについて」を参照してください。

[仮想オブジェクト マネージャ] に表示される仮想オブジェクト・コレクションは、お使いのコンピュータに格納されますが、仮想オブジェクト・ステップを含むテストと一緒に保存されません。つまり、テスト・ステップの中で仮想オブジェクトを使用した場合、オブジェクトが実行セッション中に認識されるのは、適切な仮想オブジェクト定義を含むコンピュータで実行された場合のみとなります。仮想オブジェクト・コレクション定義を別のコンピュータにコピーするには < QuickTest インストール・フォルダ > \dat\VoTemplate フォルダの内容 (またはこのフォルダ内の個々の .vot コレクション・ファイル) を、コピー先コンピュータの同じフォルダにコピーします。

注：QuickTest では、アナログや低レベル記録用の仮想オブジェクトはサポートされていません。低レベルの記録の詳細については、1537 ページ「テストの作成」を参照してください。

仮想オブジェクトについて

QuickTest では、境界線に基づいて仮想オブジェクトが識別されます。オブジェクトの境界を指定することで、Web ページまたはアプリケーション・ウィンドウにおけるオブジェクトのサイズと位置を指定します。仮想オブジェクトの親としてテスト・オブジェクトを割り当てると、仮想オブジェクトの境界座標はその親オブジェクトを基準とする相対座標と認識されます。テストを記録するときに、QuickTest によって、親オブジェクトに含まれる仮想オブジェクトが認識され、実行セッション中にこのオブジェクトを識別できるようオブジェクト・リポジトリにテスト・オブジェクトとして追加されます。QuickTest では、仮想オブジェクトは、オブジェクト・リポジトリに手作業で追加するときにもテスト・オブジェクトとして認識されます。

仮想オブジェクト・マネージャから削除しなくても、仮想オブジェクトが認識されないようにすることも可能です。詳細については、1315 ページ「無効な仮想オブジェクト定義の削除と」を参照してください。

注：

- ▶ 実行セッション中は、アプリケーション・ウィンドウは記録時と同じサイズで同じ位置になければなりません。それ以外の場合、親オブジェクトに対する仮想オブジェクトとの相対座標が記録時と異なり、それが実行セッションの正常な実行に影響を与える可能性があります。
 - ▶ 仮想オブジェクトは、テストの記録時と実行時にのみ使用できます。仮想オブジェクトへのチェックポイントの挿入や、オブジェクト・スパイによる仮想オブジェクトのプロパティの表示を行うことはできません。
 - ▶ マークを付けた仮想オブジェクトを対象に **ActiveScreen** で操作を実行するには、そのオブジェクトをあらかじめ記録しておく必要があります。それにより、仮想オブジェクトのプロパティが、オブジェクト・リポジトリのテスト・オブジェクト記述に保存されます。まだ記録していない仮想オブジェクトを対象に **ActiveScreen** で操作を実行すると、仮想オブジェクトは標準オブジェクトとして扱われます。
-

仮想オブジェクト・マネージャについて

仮想オブジェクト・マネージャでは、コンピュータで定義されている仮想オブジェクト・コレクションの表示および管理が行えます。仮想オブジェクト・マネージャで、仮想オブジェクトとコレクションの定義と削除が行えます。

仮想オブジェクト・マネージャの使用の詳細については、[仮想オブジェクトマネージャ] ダイアログ・ボックスを参照してください。

[仮想オブジェクト マネージャ] ダイアログ・ボックス

説明	仮想オブジェクトとコレクションの定義および削除を行うことができます。
アクセス方法	[ツール] > [仮想オブジェクト] > [仮想オブジェクト マネージャ] を選択します。
詳細情報	概念的概要 : 1297 ページ「仮想オブジェクトの学習について」 その他の関連トピック : 1299 ページ「仮想オブジェクトについて」

[仮想オブジェクト マネージャ] ダイアログ・ボックスの画像を次に示します。



【仮想オブジェクト マネージャ】 ダイアログ・ボックスのオプション

オプション	説明
【利用可能な仮想オブジェクトのコレクション】 リスト	お使いのコンピュータで定義されている仮想オブジェクト・コレクションと、それぞれのコレクションに含まれる仮想オブジェクトが表示されます。コレクションの横の [+] と [-] 記号をクリックして、コレクションに定義されている仮想オブジェクトを表示したり非表示にしたりできます。
【新規作成】 ボタン	【仮想オブジェクト ウィザード】 を開きます。このウィザードは、新規あるいは既存のコレクションに新規仮想オブジェクトを定義する手順を示します。詳細については、1302 ページ「仮想オブジェクトの定義」を参照してください。
【削除】 ボタン	選択した仮想オブジェクトまたは仮想オブジェクト・コレクションを削除します。詳細については、1315 ページ「無効な仮想オブジェクト定義の削除と」を参照してください。

仮想オブジェクトの定義

仮想オブジェクト・ウィザードを使って、仮想オブジェクトを標準のオブジェクト・クラスに割り当てたり、仮想オブジェクトの境界と親を指定したり、名前を割り当てたりできます。また、コレクションに割り当てることによって、仮想オブジェクトを論理的なグループにまとめることができます。

注：仮想オブジェクトは、QuickTest によって Click または DbClick メソッドが記録されるオブジェクトに対してのみ定義できます。それ以外の仮想オブジェクトは無視されます。たとえば、WinList オブジェクトを仮想オブジェクトとして定義しようとする、Select 操作は記録されますが、仮想オブジェクトは無視されます。QuickTest では、アナログや低レベル記録用の仮想オブジェクトはサポートされていません。低レベルの記録の詳細については、1537 ページ「FAQ (よくある質問)」を参照してください。

仮想オブジェクト・ウィザードの画面の詳細については、次を参照してください。

- ▶ 1305 ページ「[[仮想オブジェクト ウィザード]:[ようこそ] 画面」
- ▶ 1307 ページ「[[仮想オブジェクト ウィザード]:[標準クラスへの割り当て] 画面」
- ▶ 1308 ページ「[[仮想オブジェクト ウィザード]:[仮想オブジェクトのマーク付け] 画面」
- ▶ 1310 ページ「[[仮想オブジェクト ウィザード]:[オブジェクトの構成] 画面」
- ▶ 1313 ページ「[[仮想オブジェクト ウィザード]:[仮想オブジェクトの保存] 画面」

仮想オブジェクトを定義するには、次の手順を実行します。

- 1 QuickTest を記録モード以外のモードで起動している状態で、アプリケーションを開いて、仮想オブジェクトとして定義する領域を含むオブジェクトを表示します。
- 2 QuickTest での作業：
 - a 次のいずれかの方法で [仮想オブジェクト ウィザード] を開きます。
 - ▶ [ツール] > [仮想オブジェクト] > [新規仮想オブジェクト] を選択します。
 - ▶ [仮想オブジェクト マネージャ] で [新規作成] をクリックします。
 - b [次へ] をクリックします。
- 3 [標準クラスへの割り当て] 画面での作業：
 - a 仮想オブジェクトの割り当て先の標準クラスを選択します。**list** クラスの場合、仮想オブジェクトの行数を指定します。**table** クラスの場合、行数とカラム数を選択します。

詳細については、1307 ページ「[[仮想オブジェクト ウィザード]:[標準クラスへの割り当て] 画面」を参照してください。
 - b [次へ] をクリックします。
- 4 [仮想オブジェクトのマーク付け] 画面での作業：
 - a [オブジェクトのマーク付け] をクリックします。QuickTest ウィンドウと仮想オブジェクト・ウィザードが最小化されます。

- b 十字形のポインタで、仮想オブジェクトの領域を指定します。マウスの左ボタンを押しながら矢印キーを使用すると、十字形ポインタで定義した領域を微調整できます。

詳細については、1308 ページ「[仮想オブジェクト ウィザード] : [仮想オブジェクトのマーク付け] 画面」を参照してください。

- c [次へ] をクリックします。

注：仮想オブジェクトをマーク付けするときは、アプリケーションのほかの仮想オブジェクトと重ならないようにします。仮想オブジェクトが別の仮想オブジェクトと重なっていると、その仮想オブジェクトに対するテストが正しく記録または実行されないことがあります。

5 [オブジェクトの構成] 画面での作業：

- a 仮想オブジェクトの親として割り当てるオブジェクトをオブジェクト・ツリーの中で選択します。

- b [オブジェクト認識のレベル] ボックスで、仮想オブジェクトの識別方法と割り当て方法を選択します。

詳細については、1310 ページ「[仮想オブジェクト ウィザード] : [オブジェクトの構成] 画面」を参照してください。

- c [次へ] をクリックします。

6 [仮想オブジェクトの保存] 画面での作業：

- a 仮想オブジェクトの名前とコレクションを指定します。詳細については、1313 ページ「[仮想オブジェクト ウィザード] : [仮想オブジェクトの保存] 画面」を参照してください。

- b 次の手順のいずれかを実行します。

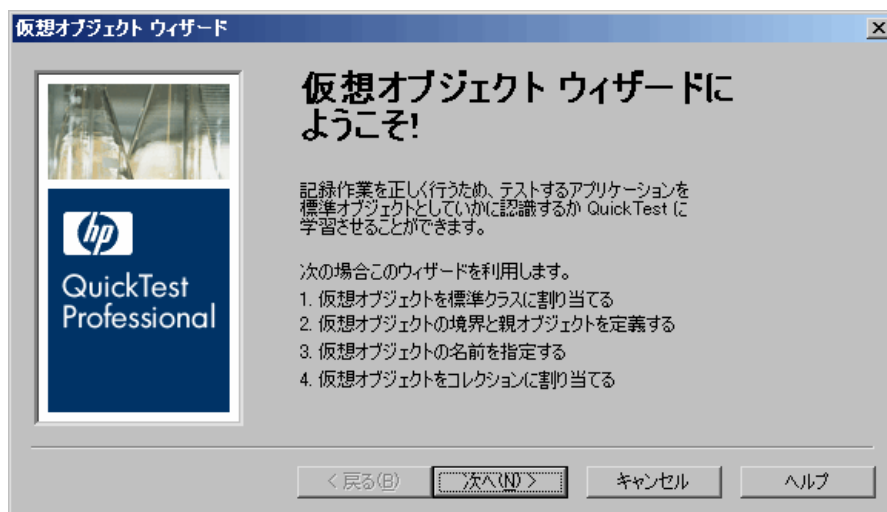
- ▶ 仮想オブジェクトを仮想オブジェクト・マネージャに追加してウィザードを閉じるには、[いいえ] を選択し、[完了] をクリックします。

- ▶ 仮想オブジェクトを仮想オブジェクト・マネージャに追加し、さらに別の仮想オブジェクトを定義するには、[はい] を選択し、[次へ] をクリックします。ウィザードが [標準クラスへの割り当て] 画面に戻り、次の仮想オブジェクトを定義できます。

[仮想オブジェクト ウィザード] : [ようこそ] 画面

説明	<p>次のようにして、ウィザードを使って仮想オブジェクトを定義する方法について説明します。</p> <ul style="list-style-type: none"> ▶ 標準クラスへの割り当て ▶ 境界の定義 ▶ 親オブジェクトの定義 ▶ 名前とコレクションの指定
アクセス方法	<ul style="list-style-type: none"> ▶ [ツール] > [仮想オブジェクト] > [新規仮想オブジェクト] を選択します。 ▶ [ツール] > [仮想オブジェクト] > [仮想オブジェクト マネージャ] を選択します。[仮想オブジェクト マネージャ] で [新規作成] をクリックします。
前の画面	この画面がウィザードの最初の画面です。
次の画面	1307 ページ「[仮想オブジェクト ウィザード] : [標準クラスへの割り当て] 画面」

[仮想オブジェクト ウィザード] : [ようこそ] 画面の画像を次に示します。



[ようこそ] 画面のオプション

オプション	説明
[次へ]:	[次へ] をクリックして, [標準クラスへの割り当て] 画面へ進みます。

補足リファレンス

関連作業	<ul style="list-style-type: none">▶ 1302 ページ「仮想オブジェクトの定義」▶ 1315 ページ「無効な仮想オブジェクト定義の削除と」
関連概念	<ul style="list-style-type: none">▶ 1299 ページ「仮想オブジェクトについて」▶ 1301 ページ「[仮想オブジェクトマネージャ] ダイアログ・ボックス」

[仮想オブジェクト ウィザード] : [標準クラスへの割り当て] 画面

説明	仮想オブジェクトの標準クラスを設定できます。
前の画面	1305 ページ「[仮想オブジェクト ウィザード] : [ようこそ] 画面」
次の画面	1313 ページ「[仮想オブジェクト ウィザード] : [仮想オブジェクトの保存] 画面」

[標準クラスへの割り当て] 画面の画像を次に示します。



[標準クラスへの割り当て] 画面のオプション

オプション	説明
[クラス]	<p>リストから標準オブジェクトのクラスを指定します。</p> <ul style="list-style-type: none"> ▶ list クラスの場合、仮想オブジェクトの行数を指定します。 ▶ table クラスの場合、行数とカラム数を選択します。

オプション	説明
[戻る]	[戻る] をクリックすると, [ようこそ] 画面に戻ります。
[次へ]	[次へ] をクリックすると, [仮想オブジェクトのマーク付け] 画面に進みます。

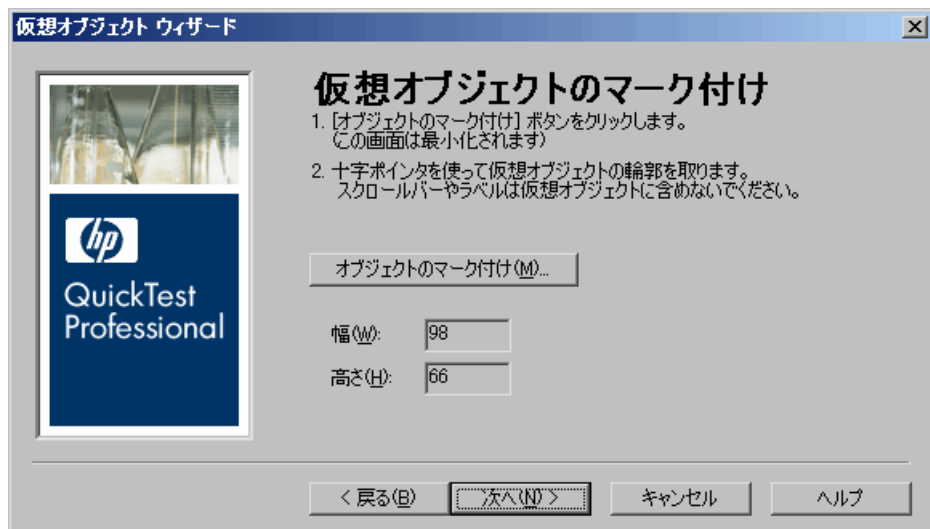
補足リファレンス

関連作業	<ul style="list-style-type: none"> ▶ 1302 ページ「仮想オブジェクトの定義」 ▶ 1315 ページ「無効な仮想オブジェクト定義の削除と」
関連概念	<ul style="list-style-type: none"> ▶ 1299 ページ「仮想オブジェクトについて」 ▶ 1301 ページ「[仮想オブジェクトマネージャ] ダイアログ・ボックス」

[仮想オブジェクト ウィザード] : [仮想オブジェクトのマーク付け] 画面

説明	仮想オブジェクトのサイズと場所を設定できます。
前の画面	1307 ページ「[仮想オブジェクト ウィザード] : [標準クラスへの割り当て] 画面」
次の画面	1310 ページ「[仮想オブジェクト ウィザード] : [オブジェクトの構成] 画面」

[仮想オブジェクトのマーク付け] 画面の画像を次に示します。



[仮想オブジェクトのマーク付け] 画面のオプション

オプション	説明
[オブジェクトのマーク付け]	<p>仮想オブジェクトの輪郭を囲みます。十字型ポインタを使用して、アプリケーションの仮想オブジェクトの輪郭を囲むことができます。</p> <p>実行セッション中に QuickTest が仮想オブジェクトを識別できなくなる可能性があるため、仮想オブジェクトをマーク付けするときは、ほかの仮想オブジェクトと重ならないようにしてください。</p> <p>注：テストの実行と記録を正しく行うには、アプリケーション・ウィンドウが仮想オブジェクトを定義したときと同じサイズで、同じ位置にある必要があります。</p>
[幅]	輪郭の幅をピクセルで示します。
[高さ]	輪郭の高さをピクセルで示します。

オプション	説明
[戻る]	[戻る] をクリックすると、[標準クラスへの割り当て] 画面に戻ります。
[次へ]	[次へ] をクリックすると、[オブジェクトの構成] 画面に進みます。

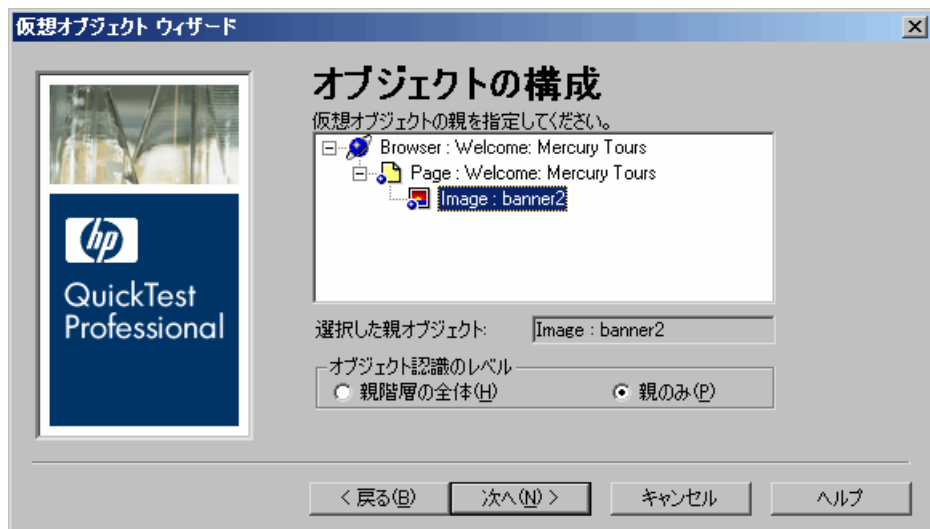
補足リファレンス

関連作業	<ul style="list-style-type: none"> ▶ 1302 ページ「仮想オブジェクトの定義」 ▶ 1315 ページ「無効な仮想オブジェクト定義の削除と」
関連概念	<ul style="list-style-type: none"> ▶ 1299 ページ「仮想オブジェクトについて」 ▶ 1301 ページ「[[仮想オブジェクトマネージャ] ダイアログ・ボックス」

[仮想オブジェクト ウィザード] : [オブジェクトの構成] 画面

説明	オブジェクトを仮想オブジェクトの親として設定できます。
前の画面	1308 ページ「[[仮想オブジェクト ウィザード] : [仮想オブジェクトのマーク付け] 画面」
次の画面	1313 ページ「[[仮想オブジェクト ウィザード] : [仮想オブジェクトの保存] 画面」

[オブジェクトの構成] 画面の画像を次に示します。



[オブジェクトの構成] 画面のオプション

オプション	説明
[仮想オブジェクトの親を指定してください。] 領域	ツリー内のオブジェクトを親オブジェクトとして選択できます。仮想オブジェクトの輪郭の座標は、親オブジェクトを基準とする相対座標です。
[選択した親オブジェクト] ボックス (読み取り専用)	親オブジェクトとして選択したオブジェクトの名前が表示されます。
[オブジェクト認識のレベル] 領域	下記の説明に従って、仮想オブジェクトの識別方法と割り当て方法を指定します。仮想オブジェクトの輪郭の座標は、選択した親オブジェクトを基準とする相対座標です。

オプション	説明
<p>[親階層の全体] ラジオ・ボタン</p>	<p>仮想オブジェクトの特定 1 回の出現を識別する場合は、このオプションを選択します。QuickTest では、親階層が完全に一致している場合のみ仮想オブジェクトが識別されます。</p> <p>たとえば、Browser("A").Page("B").Image("C") を使用して仮想オブジェクトが定義されている場合、階層が Browser("X").Page("B").Image("C") に変わるとこの仮想オブジェクトは識別されません。</p>
<p>[親のみ] ラジオ・ボタン</p>	<p>仮想オブジェクトのすべての出現を識別する場合は、このオプションを選択します。親階層全体ではなく、直接の親だけを基準にして仮想オブジェクトが識別されます。</p> <p>たとえば、Browser("A").Page("B").Image("C") を使用して仮想オブジェクトが定義されている場合、階層が、Browser("X").Page("Y").Image("C") であっても仮想オブジェクトが識別されます。</p>
<p>[戻る] ボタン</p>	<p>[戻る] をクリックすると、[仮想オブジェクトのマーク付け] 画面に戻ります。</p>
<p>[次へ] ボタン</p>	<p>[次へ] をクリックすると、[仮想オブジェクトの保存] 画面に進みます。</p>

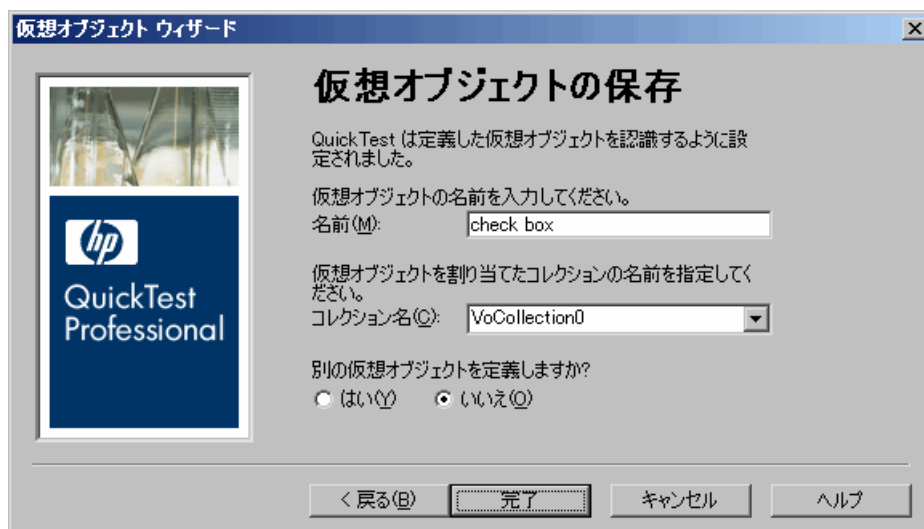
補足リファレンス

<p>関連作業</p>	<ul style="list-style-type: none"> ▶ 1302 ページ「仮想オブジェクトの定義」 ▶ 1315 ページ「無効な仮想オブジェクト定義の削除と」
<p>関連概念</p>	<ul style="list-style-type: none"> ▶ 1299 ページ「仮想オブジェクトについて」 ▶ 1301 ページ「[仮想オブジェクト マネージャ] ダイアログ・ボックス」

[仮想オブジェクト ウィザード] : [仮想オブジェクトの保存] 画面

説明	仮想オブジェクト用の名前とコレクションを設定できます。また、別の仮想オブジェクトの定義を開始することもできます。
前の画面	1310 ページ「[仮想オブジェクト ウィザード] : [オブジェクトの構成] 画面」
次の画面	この画面がウィザードの最後の画面です。

[仮想オブジェクトの保存] 画面の画像を次に示します。



[仮想オブジェクトの保存] 画面のオプション

オプション	説明
[名前] ボックス	仮想オブジェクト名を指定します。コレクションのリストから選択するか、[コレクション名] ボックスに新しい名前を入力して新しいコレクションを作成します。
[コレクション名] ボックス	コレクション名を指定します。既存の名前を選択するか、新しい名前を入力できます。

オプション	説明
<p>[別の仮想オブジェクトを定義しますか?] 領域</p>	<ul style="list-style-type: none"> ▶ ウィザードを使用して別の仮想オブジェクトを定義する場合は、[はい] を選択して [次へ] をクリックします。 ▶ ウィザードを閉じる場合は、[いいえ] を選択して [完了] をクリックします。
<p>[戻る] ボタン</p>	<p>[戻る] をクリックすると、[オブジェクトの構成] 画面に戻ります。</p>
<p>[次へ] ボタン</p>	<p>[別の仮想オブジェクトを定義しますか?] オプションで [はい] ラジオ・ボタンを選択した場合に、[次へ] ボタンが表示されます。</p> <p>[次へ] をクリックすると、仮想オブジェクトを保存し、[標準クラスへの割り当て] 画面に進んで、別の仮想オブジェクトの定義を開始します。</p>
<p>[完了] ボタン</p>	<p>[別の仮想オブジェクトを定義しますか?] オプションで [いいえ] ラジオ・ボタンを選択した場合に、[完了] ボタンが表示されます。</p> <p>[完了] をクリックして仮想オブジェクトを保存し、ウィザードを閉じます。</p>

補足リファレンス

<p>関連作業</p>	<ul style="list-style-type: none"> ▶ 1302 ページ「仮想オブジェクトの定義」 ▶ 1315 ページ「無効な仮想オブジェクト定義の削除と」
<p>関連概念</p>	<ul style="list-style-type: none"> ▶ 1299 ページ「仮想オブジェクトについて」 ▶ 1301 ページ「[仮想オブジェクトマネージャ] ダイアログ・ボックス」

無効な仮想オブジェクト定義の削除と

仮想オブジェクトをテストに含めないようにするには、記録された仮想オブジェクトをテストから削除するか、記録時に認識されないように設定します。

仮想オブジェクトを削除するには、次の手順を実行します。

- 1 [ツール] > [仮想オブジェクト] > [仮想オブジェクト マネージャ] を選択します。仮想オブジェクト・マネージャが起動します。詳細については、1301 ページ「[仮想オブジェクト マネージャ] ダイアログ・ボックス」を参照してください。
- 2 利用可能な仮想オブジェクトのコレクションのリストで、コレクションの横にあるプラス記号 (+) をクリックし、削除する仮想オブジェクトを表示します。仮想オブジェクトを選択し、[削除] をクリックします。
コレクション全体を削除するには、コレクションを選択し、[削除] をクリックします。
- 3 [閉じる] をクリックします。

ヒント：新しい仮想オブジェクトを定義するには、仮想オブジェクト・マネージャで [新規作成] をクリックし、仮想オブジェクト・ウィザードを起動します。

記録中に仮想オブジェクトが認識されないようにするには、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。
- 2 [一般] 表示枠で、[記録時の仮想オブジェクト認識を無効にする] チェック・ボックスをオンにします。
- 3 [OK] をクリックします。



注：記録中に QuickTest で仮想オブジェクトが認識されるようにするには、[オプション] ダイアログ・ボックスの [一般] 表示枠の **[記録時の仮想オブジェクト認識を無効にする]** チェック・ボックスがクリアされていることを確認します。詳細については、1223 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。

第 48 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 (1317 ページ)
- ▶ 回復シナリオを使用するタイミングの決定 (1319 ページ)
- ▶ 回復シナリオの定義 (1320 ページ)
- ▶ 回復シナリオ・ウィザードについて (1324 ページ)
- ▶ 回復シナリオの管理 (1351 ページ)
- ▶ テストへの回復シナリオの関連付け (1355 ページ)
- ▶ プログラムによる回復メカニズムの制御 (1361 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、テス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、テス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テス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、1319 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、テストを続行するよう 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 でテス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テス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、テス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、テス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、まずそのテストに回復シナリオを関連付ける必要があります。テス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。テス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、1355 ページ「テストへの回復シナリオの追加」を参照してください。

回復シナリオを定義した対象のテストを実行して、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。

テストに **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、1361 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: [テストの設定] 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、テス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、かつテス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、テス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、またはテス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、テストでは、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。テスト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、テストは数ステップ進む場合があります。

イベントがテス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またはオプション・ステップに追加追加して、テス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] ボタンを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 ステートメントを追加する、あるいはメッセージ・ボックスで [OK] をクリックするオプション・ステップを追加することで処理できます。

テス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テストで直接処理することをお勧めします。

オプション・ステップの詳細については、961 ページ「オプション・ステップの使用」を参照してください。If ステートメントなど、プログラミング・ステートメントの挿入の詳細については、第 28 章「プログラミング・ロジックを含むステップの追加」を参照してください。

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テストに関連付けて、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。

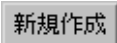

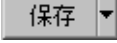
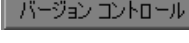





[回復シナリオ マネージャ] ダイアログ・ボックスについて


[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、1321 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、1349 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のみ、オプションを使用できます）。 詳細については、1462 ページ「Quality Center でのアセットのバージョン管理」および 1443 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、1324 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、1353 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、1352 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、1354 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、1354 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、1353 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、1351 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。

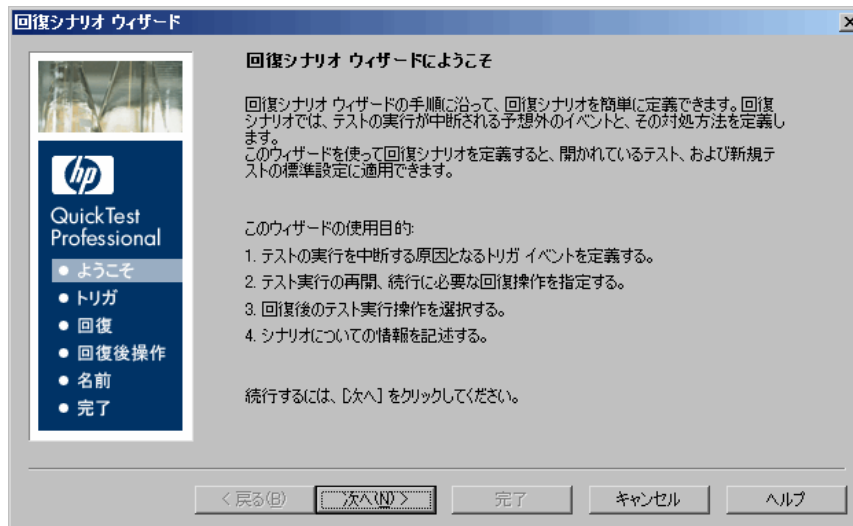
- ▶ 実行セッションを中断するトリガ・イベントの定義
- ▶ 継続に必要な回復操作の指定
- ▶ 回復後のテスト実行の操作の選択
- ▶ 回復シナリオの名前と説明の指定
- ▶ 回復シナリオを現在のテストに関連付けるか、すべての新規テストに関連付けるかどうかの指定



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 ウィザードによるこそ] 画面

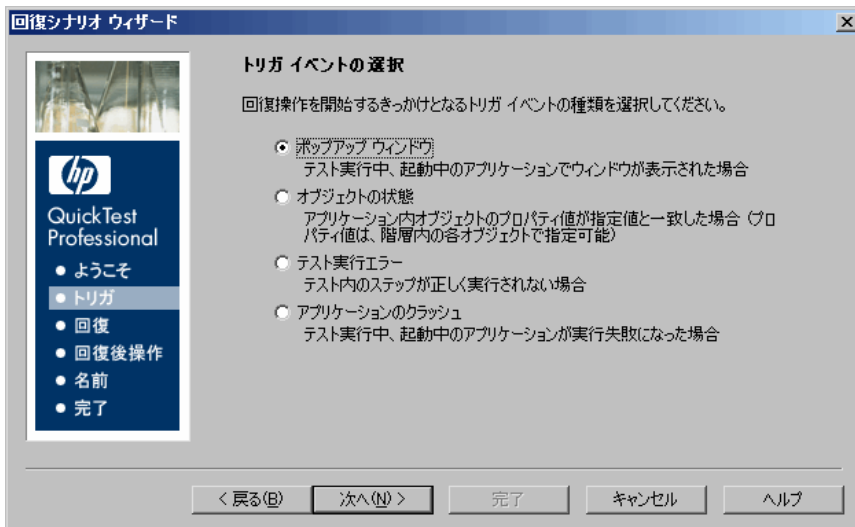
[回復シナリオ ウィザードによるこそ] 画面には、回復シナリオ・ウィザードのさまざまなオプションについての一般的な情報と、回復シナリオの定義に関連する段階の概要が表示されます。



[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 (1326 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 (1328 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 (1330 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 (1334 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。

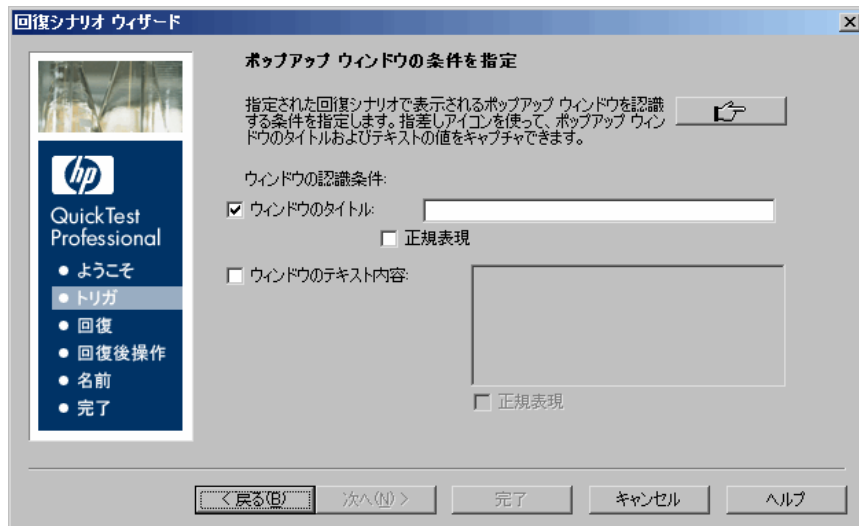
このオプションを選択し、**[次へ]** をクリックして、**[回復操作]** 画面に進みます (1337 ページを参照してください)。

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、テス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。テス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、テス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（1326 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、769 ページ「正規表現の概要と使い方」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、1329 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（1337 ページを参照してください）。

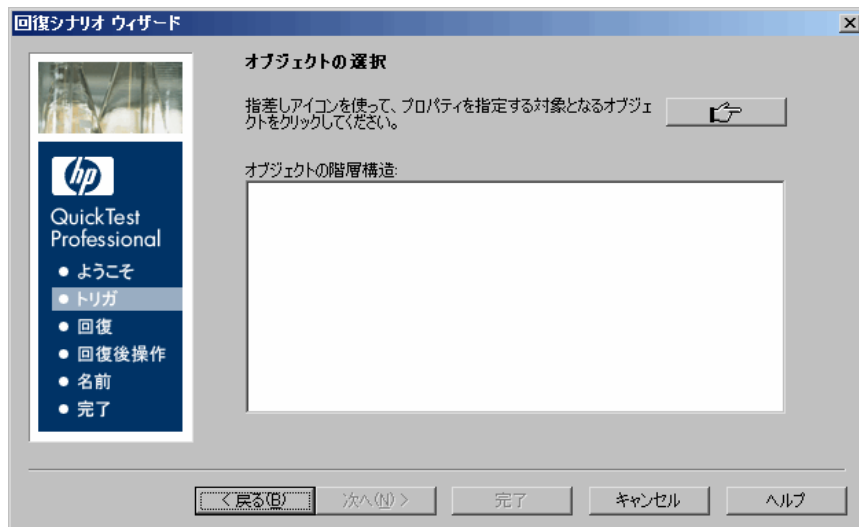
指差しマークの使い方のヒント

- ▶ 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。

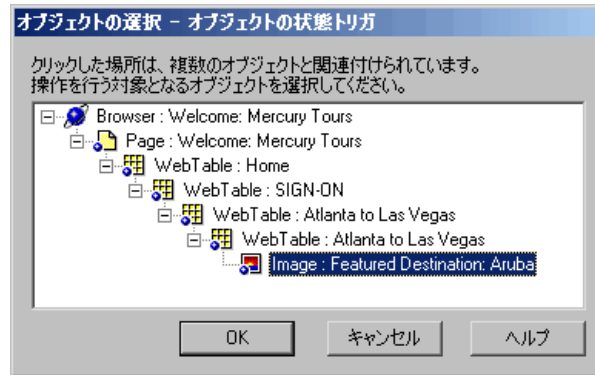
[オブジェクトの選択] 画面

[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（1326 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、1331 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（1333 ページを参照してください）。

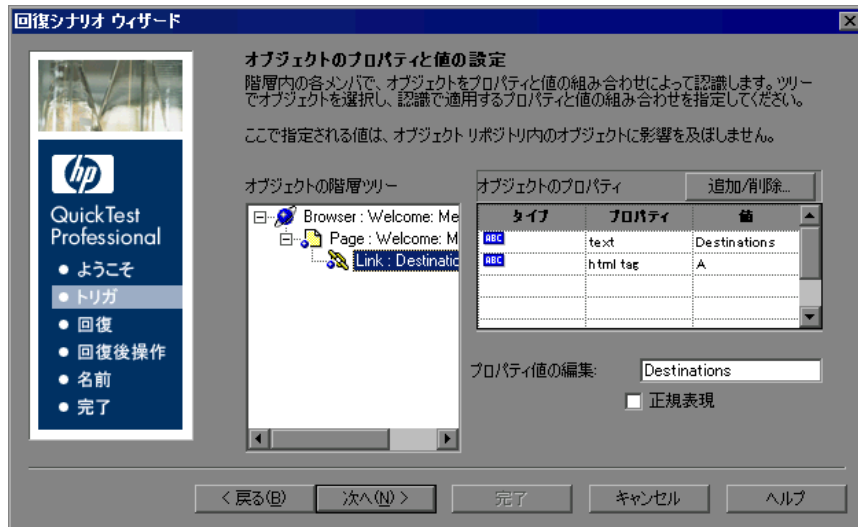
指差しマークの使い方のヒント

- ▶ 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（1330 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。



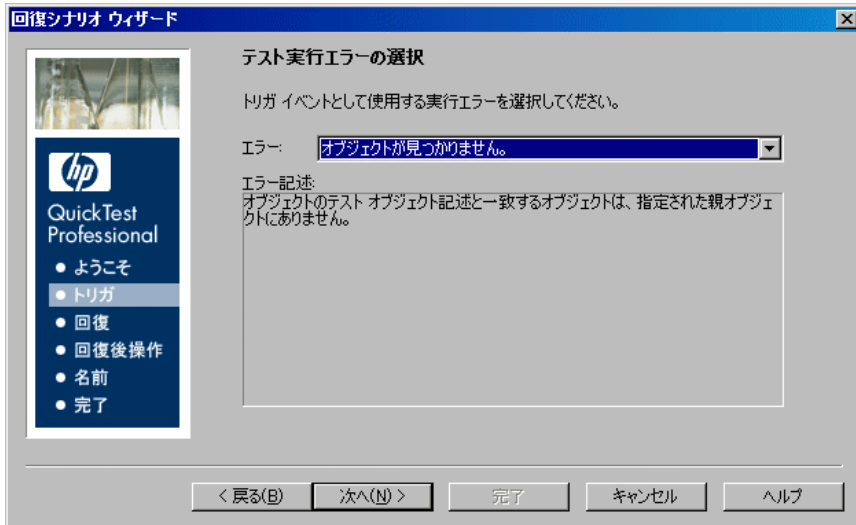
階層の各オブジェクトに対して、【プロパティ値の編集】ボックスで、オブジェクトの識別に使用するプロパティ値を変更できます。【追加 / 削除】ボタンをクリックし、検査対象のプロパティ値のリストからオブジェクト・プロパティを追加または削除することもできます。オブジェクトはクラスではなく、プロパティ値によってのみ識別されます。

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、769 ページ「正規表現の概要と使い方」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（1337 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 (1326 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。

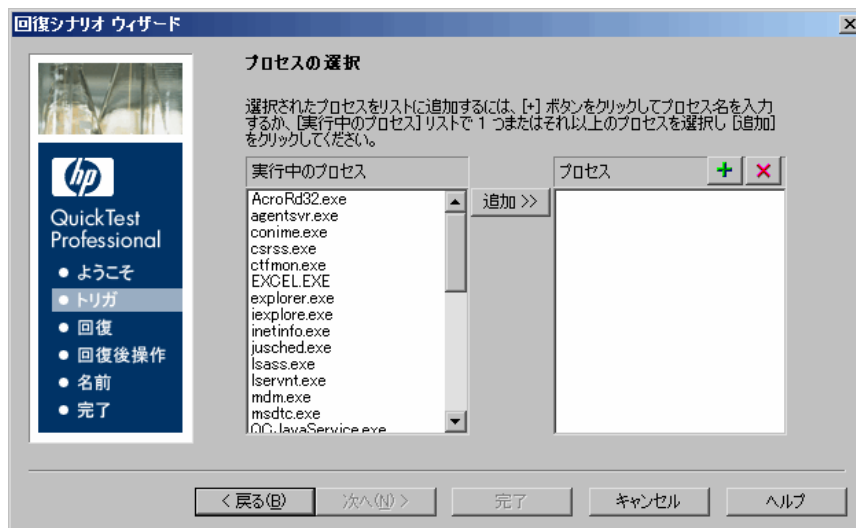
- ▶ [任意のエラー] : テスト・オブジェクト・メソッドによって返されるすべてのエラー・コード。
- ▶ [リストまたはメニュー内の項目が一意ではありません] : リスト, メニュー, またはツリーにある複数の項目に, メソッド引数で指定されている名前がある場合に発生します。
- ▶ [リストまたはメニュー内の項目が見つかりません] : メソッド引数で指定されているリスト, メニュー, またはツリー項目が QuickTest によって識別できない場合に発生します。この原因としては, その項目を現在利用できない, または名前が変更されていることが考えられます。
- ▶ [物理記述に対して複数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 1337 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（1326 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。



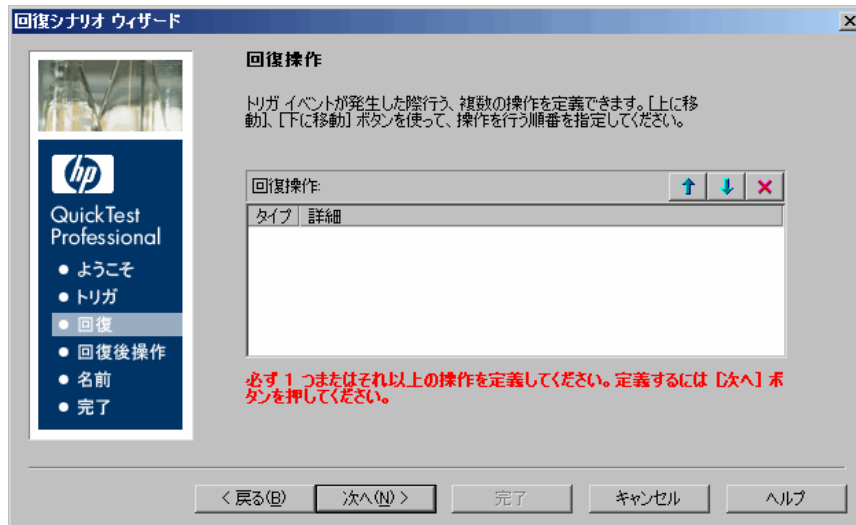
▶ [プロセス] リストからプロセスを削除するには, プロセスを選択して [プロセスの削除] ボタンをクリックします。

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 (1337 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（1338 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。

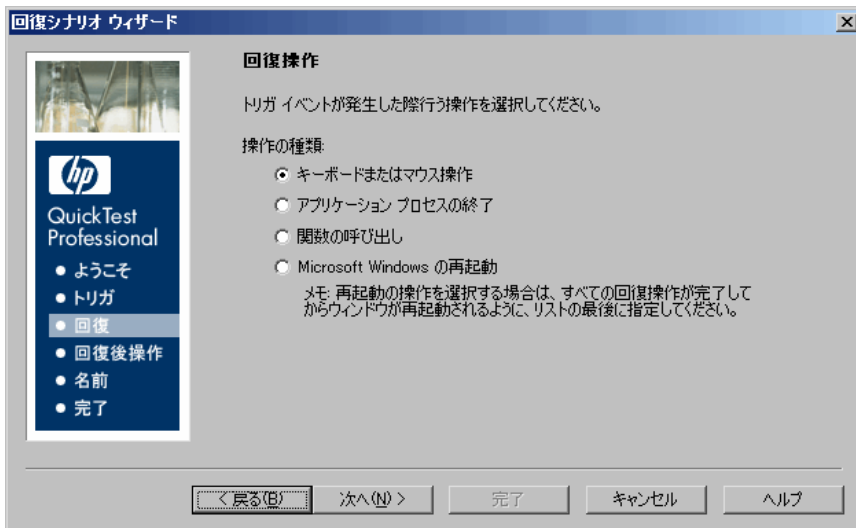
注：【Microsoft Windows の再起動】回復操作を定義する場合、常に最後の回復操作としてこの操作が挿入されるため、リストで位置は変更できません。

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 (1344 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。

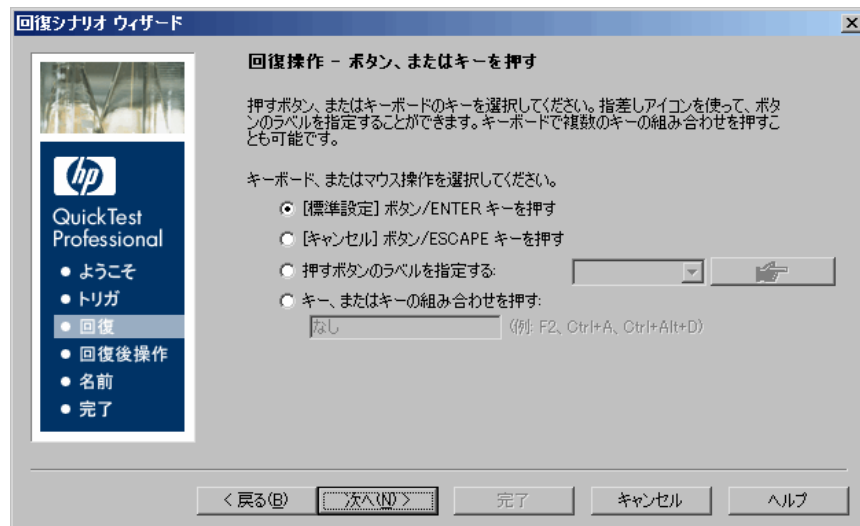
- ▶ **【キーボードまたはマウス操作】** : QuickTest によって、ウィンドウのボタンをクリックする操作、またはキーボードのキーを押す操作がシミュレートされます。このオプションを選択し、[次へ] をクリックして、[回復操作 - ボタン、またはキーを押す] 画面に進みます (1339 ページを参照してください)。
- ▶ **【アプリケーション プロセスの終了】** : QuickTest によって、指定のプロセスが閉じます。このオプションを選択し、[次へ] をクリックして、[回復操作 - プロセスの終了] 画面に進みます (1341 ページを参照してください)。

- ▶ **〔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 (1343 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 (1337 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テス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、テス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 (1338 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、1340 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。

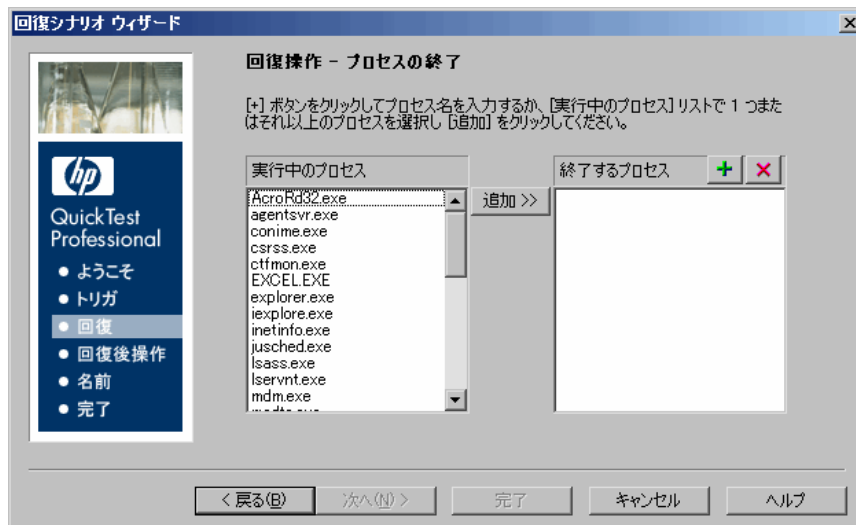
指差しマークの使い方のヒント

- ▶ 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（1338 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。



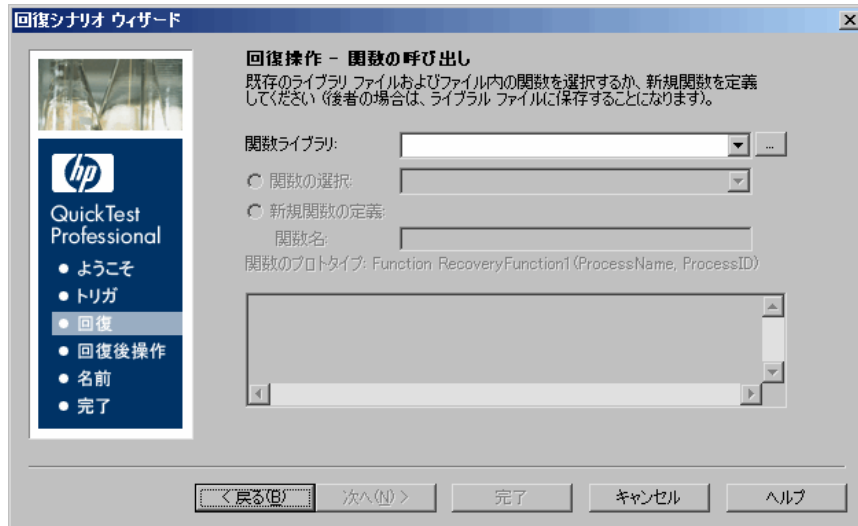
▶ [**終了するプロセス**] リストからプロセスを削除するには、プロセスを選択して [**プロセスの削除**] ボタンをクリックします。

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（1338 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： QuickTest では、選択した関数ライブラリが、テストと自動的に関連付けられます。したがって、〔テストの設定〕ダイアログ・ボックスの〔リソース〕表示枠で、それぞれ関数ライブラリとテストを関連付ける必要はありません。

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。
1326 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。

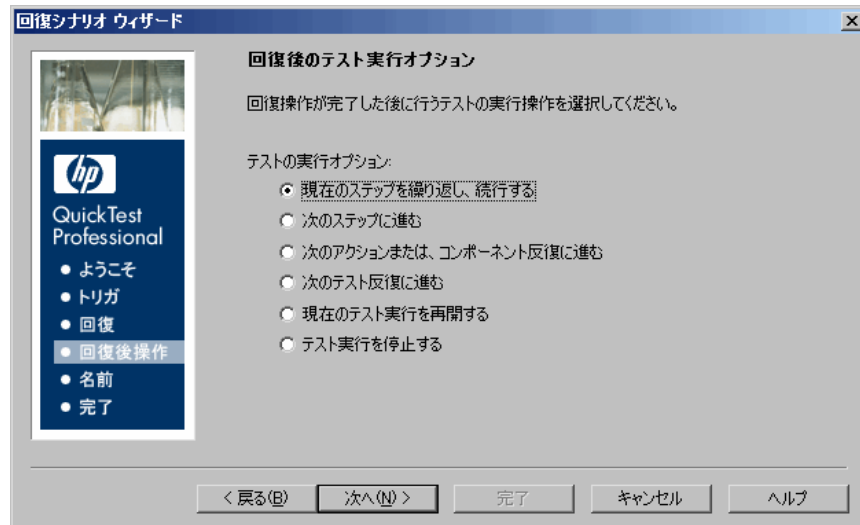
注 : 複数のシナリオで異なる関数ライブラリから同一の名前を持つ関数を使用すると、回復プロセスは失敗することがあります。この場合、実行セッション中に回復の失敗に関する情報が表示されます。

[**次へ**] をクリックします。[回復操作] 画面が再び開き (1337 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 (1337 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、1359 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可能性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】**

現在のアクションでのステップ実行を停止し、最初から次の QuickTest テストの反復を開始します（またはテストを追加的に反復する必要がない場合はテス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、テストを始めから再実行します。

▶ **【テストの実行を停止する】**

テストの実行を停止します。

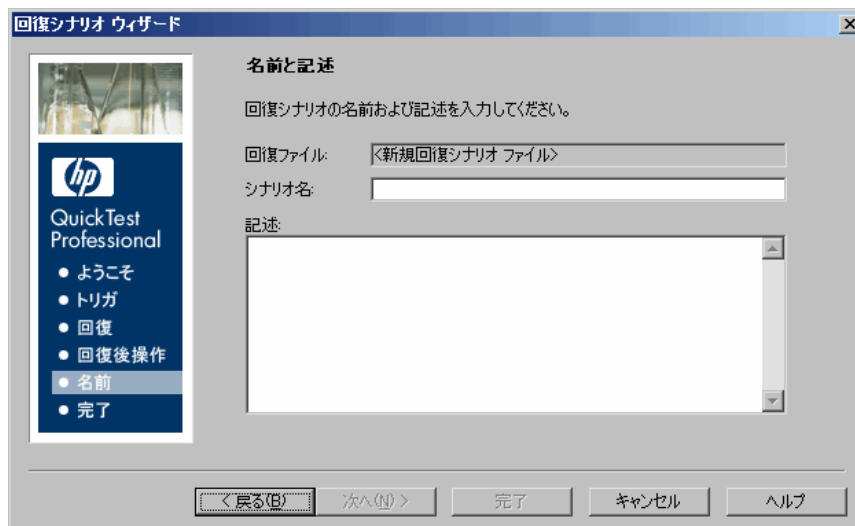
注：回復操作として **【Microsoft Windows の再起動】** を選択した場合、上記の最後 2 つのテスト実行のオプションのみから選択できます。

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（1346 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（1344 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。

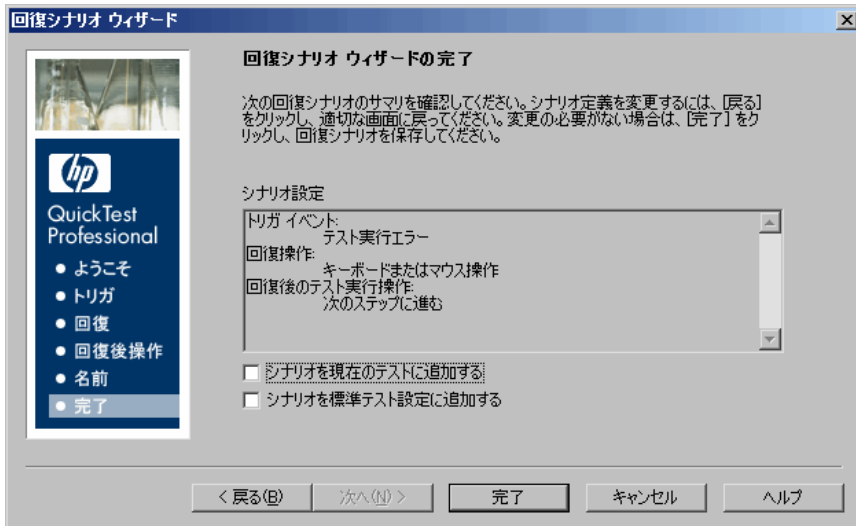


回復シナリオの名前とテキスト形式の説明を入力し、[次へ] をクリックして、[回復シナリオ ウィザードの完了] 画面に進みます (1348 ページを参照してください)。

【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（1346 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。回復シナリオを自動的に現在のテストに関連付けるかどうか、または新しいすべてのテストの標準設定に回復シナリオを追加するかどうかを指定できます。



【シナリオを現在のテストに追加する】チェック・ボックスを選択すると、この回復シナリオを現在のテストに関連付けることができます。【完了】をクリックすると、QuickTest によって、【テストの設定】ダイアログ・ボックスの【回復】表示枠にある【シナリオ】リストに回復シナリオが追加されます。

この回復シナリオを、新しく作るすべてのテストの標準設定のシナリオにするには、【シナリオを標準テスト設定に追加する】チェック・ボックスを選択します。次回テストを作成する際には、【テストの設定】ダイアログ・ボックスの【回復】表示枠の【シナリオ】リストに、このシナリオが表示されます。

注：シナリオは標準シナリオのリストから削除できます。詳細については、1279 ページ「テストのための回復シナリオ設定の定義」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、別の名前での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。

ヒント： ファイルをテストの添付ファイルとして Quality Center のテスト計画モジュールに保存する場合は、サイドバーの **[Quality Center テスト計画]** を選択し、テストを参照してダブルクリックし、**[保存]** をクリックします。

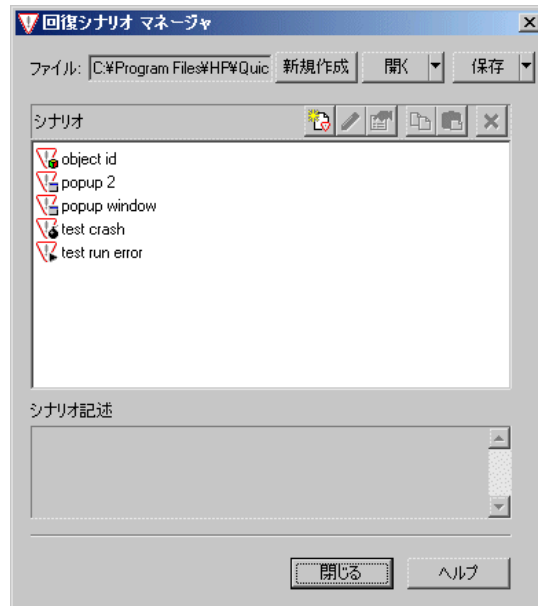
注： ファイル・システムまたは Quality Center 9.x のリソースへのパスを指定した場合は、そのパス、あるいはそのパスの一部が **[オプション]** ダイアログ・ボックスの **[フォルダ]** 表示枠 (**[ツール]** > **[オプション]** > **[フォルダ]** ノード) に存在するか検査されます。パスが存在する場合は、入力したパスの相対的な部分だけを使用してパスを定義するように求められます。パスが存在しない場合は、リソースのパスを **[フォルダ]** 表示枠に追加して、相対的なパスを定義するように求められます。詳細については、322 ページ「QuickTest で相対パスの使用」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、1432 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。
	テス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。

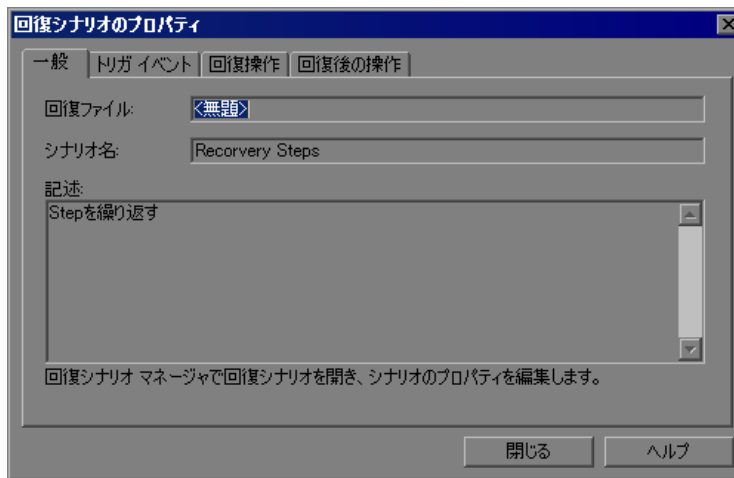
- ▶ 回復シナリオのプロパティの表示
- ▶ 回復シナリオの変更
- ▶ 回復シナリオの削除
- ▶ 回復シナリオ・ファイル間での回復シナリオのコピー

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、1320 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テス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。
-

テスト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テストを関連付け、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。テスト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。すべての新しいテスト用の標準シナリオとして、どの回復シナリオを使用するかも定義できます。

注： [リソース] 表示枠で、テストへの回復シナリオの関連付け、およびテストに関連付けられている回復シナリオの削除、有効化、無効化、優先順位設定、プロパティ表示を行うことができます。詳細については、1149 ページ「[リソース] 表示枠」を参照してください。

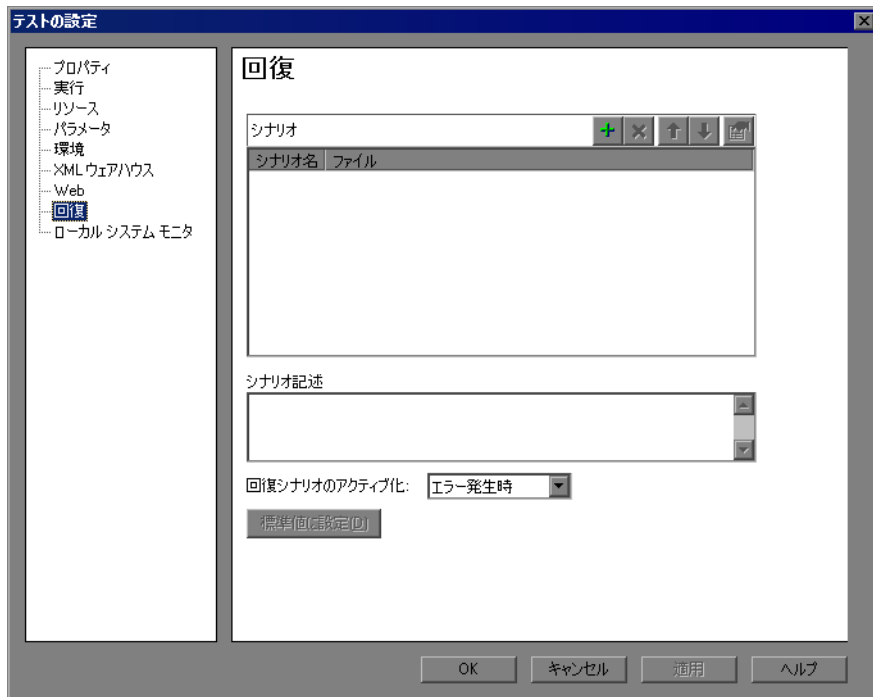
テストへの回復シナリオの追加

回復シナリオを作成した後は、トリガ・イベントが発生すると、実行セッション中に回復シナリオを実行するよう QuickTest に指示するために、1 つ以上のシナリオをテストと関連付けることができます。[テストの設定] ダイアログ・ボックスの [回復] 表示枠には、現在のテストに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。

ヒント: トリガ・イベントが発生すると、QuickTest によって、[回復] 表示枠に表示されている順序で適切な回復シナリオが確認されます。この順序は、1358 ページ「回復シナリオの優先順位の設定」で説明している手順で変更できます。

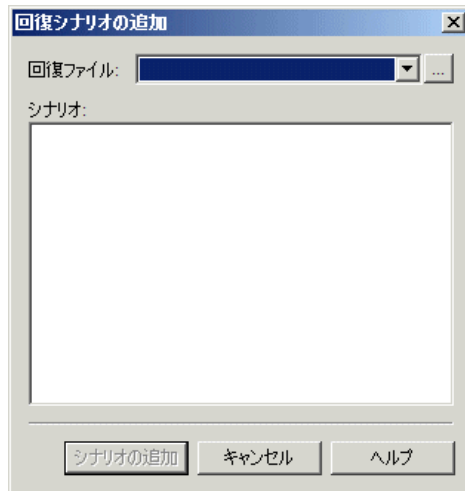
回復シナリオをテストに追加するには、次の手順を実行します。

- 1 [ファイル] > [設定] を選択します。[テストの設定] ダイアログ・ボックスが開きます。[回復] ノードを選択します。



- 2 [追加] ボタンをクリックします。[回復シナリオの追加] ダイアログ・ボック

スが開きます。



- 3 **[回復ファイル]** ボックスで、テストと関連付ける回復シナリオが含まれている回復ファイルを選択します。あるいは、参照ボタンをクリックして、選択する回復ファイルに移動します。**[シナリオ]** ボックスには、選択したファイルに保存されているシナリオの名前が表示されます。
- 4 **[シナリオ]** ボックスで、テストと関連付けるシナリオを選択し、**[シナリオの追加]** をクリックします。**[回復シナリオの追加]** ダイアログ・ボックスが閉じ、選択したシナリオが**[回復]** 表示枠の**[シナリオ]** リストに追加されます。

ヒント：パスを1回クリックして強調表示した後、再度パスをクリックして編集モードに入ることによって、回復シナリオ・ファイルのパスを編集できます。たとえば、絶対ファイル・パスをファイルの相対ファイル・パスに変更したい場合があります。回復シナリオ・ファイルのパスを変更する場合、テストを実行する前に、回復シナリオが新しいパスの位置に定義されていることを確認する必要があります。

回復シナリオのプロパティの表示

テスト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せん。

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、1353 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 **[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、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、1352 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、[テストの設定] 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 **[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。
- 2 **[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。



テストからの回復シナリオの削除

[テストの設定] ダイアログ・ボックスの [回復] 表示枠を使用して、特定のシナリオとテストの間の関連付けを削除できます。テスト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。

回復シナリオをテストから削除するには、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。選択したシナリオは、テスト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

[テストの設定] 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、テスト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。

- ▶ 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、テスト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、1361 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。

すべての新しいテストの標準回復シナリオの設定

[テストの設定] ダイアログ・ボックスの [回復] 表示枠で **[標準値に設定]** ボタンをクリックすると、回復シナリオの現在のリストを、すべての新しいテスト用の標準シナリオに設定できます。設定以降、現在の回復シナリオのリストに加える変更は現在のテストにのみ影響し、定義した標準設定のリストは変更されません。

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 49 章

QuickTest Script Editor を使った作業

QuickTest Script Editor は、複数のテスト・スクリプトおよび関数ライブラリを同時に開いて編集できるツールです。

本章の内容

- ▶ QuickTest Script Editor について (1364 ページ)
- ▶ [QuickTest Script Editor] ウィンドウについて (1366 ページ)
- ▶ QuickTest Script Editor ・ウィンドウのカスタマイズ (1367 ページ)
- ▶ [フロー] 表示枠について (1369 ページ)
- ▶ [リソース] 表示枠について (1372 ページ)
- ▶ 表示領域について (1375 ページ)
- ▶ テストを使った作業 (1377 ページ)
- ▶ 関数ライブラリを使った作業 (1381 ページ)

QuickTest Script Editor について

QuickTest Script Editor を使用すると、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。さらに、新しい関数ライブラリを作成することもできます。テストのスクリプトを変更することはできますが、新しいテストの作成、関数ライブラリの関連付け、関連付けられている関数ライブラリの削除、そして、既存のテスト名、テスト設定、パラメータ化、データ・テーブル値などの情報の変更はできません。

詳細については、次を参照してください。

- ▶ 1377 ページ「テストを使った作業」
- ▶ 1381 ページ「関数ライブラリを使った作業」

重要な考慮事項

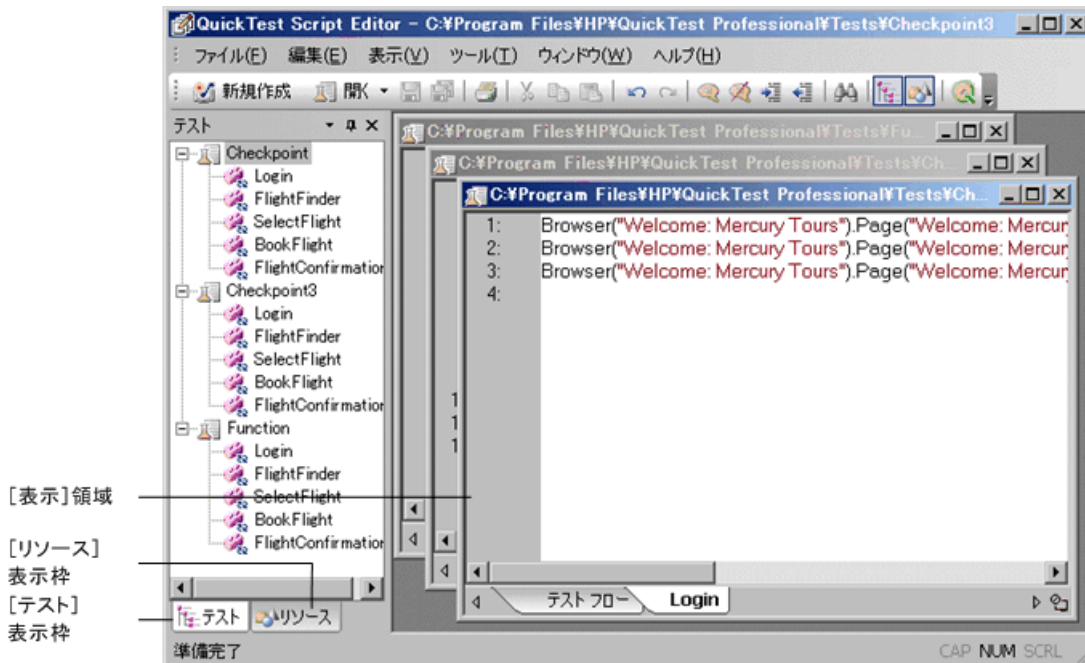
- ▶ QuickTest Script Editor では、QuickTest のテストおよび関数ライブラリでのみ作業できます。コンポーネントまたはスクリプト・コンポーネントの使用の詳細については、第 56 章「Business Process Testing を使用した作業」を参照してください。
- ▶ スクリプト・エディタでは、Quality Center に格納されているバージョン・コントロールが有効なファイルを変更することはできませんが、読み取り専用モードで開いて表示することはできます。テストおよび関数ライブラリが、Quality Center のバージョン・コントロールが有効なプロジェクトの一部である場合、このようなファイルを変更するには QuickTest を使用する必要があります。
- ▶ 以前のバージョンの QuickTest で作成されたテストは、読み取り専用モードで開きます。
 - ▶ ファイル・システムまたは Quality Center 9.x に格納されているテストは、QuickTest で開いて保存すれば現在のバージョンに更新できます。テストを保存した後は、以前のバージョンの QuickTest またはスクリプト・エディタでそのテストを開くことはできなくなります。

- ▶ テストが Quality Center 10.00 に格納されている場合、管理者は、QuickTest Professional Asset Upgrade Tool for Quality Center を使用してテストを現在のバージョンに更新する必要があります。このツールでは、プロジェクトのすべてのテストが同時にアップグレードされます。テストがアップグレードされた後は、以前のバージョンの QuickTest またはスクリプト・エディタでそのテストを開くことはできなくなります。バージョン・コントロールが有効なプロジェクトにテストが保存された場合、テストは読み取り専用モードのみ開きます。テストを変更するには、QuickTest で開く必要があります。
- ▶ QuickTest Script Editor によって、UTF-16 識別子が保存する各関数ライブラリ・ファイル（新規または既存）の開始位置に自動的に追加されます。

[QuickTest Script Editor] ウィンドウについて

QuickTest Script Editor を開くには、[スタート] > [プログラム] > [QuickTest Professional] > [Tools] > [QuickTest Script Editor] を選択します。

[QuickTest Script Editor] ウィンドウの例を次に示します



[QuickTest Script Editor] ウィンドウには、次の主要な要素があります。

- ▶ [フロー] 表示枠：開いているテストごとにアクションの呼び出しの流れを表示します。
- ▶ [リソース] 表示枠：開いているテスト、そのローカル・アクション、各テストに関連付けられている関数ライブラリ、および現在開いているすべての関数ライブラリのリストを表示します。
- ▶ [表示] 領域：開いているテストと関数ライブラリごとにウィンドウを表示します。

詳細については、次を参照してください。

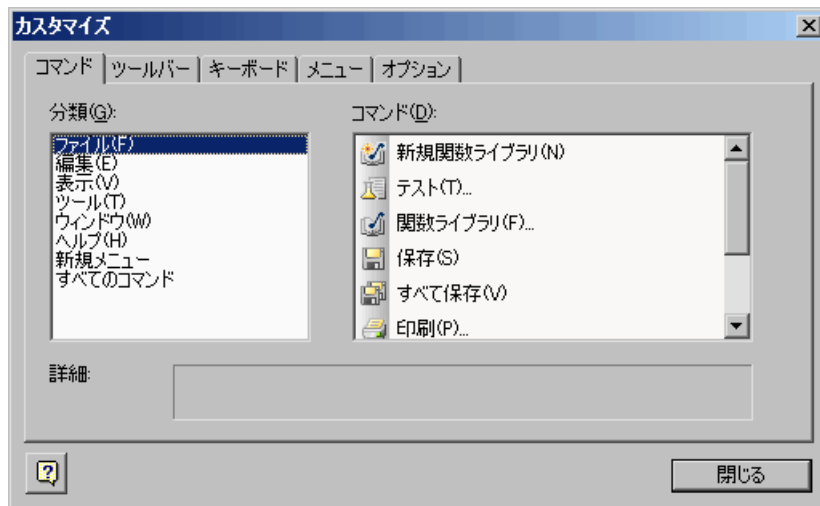
- ▶ 1367 ページ「QuickTest Script Editor・ウィンドウのカスタマイズ」
- ▶ 1369 ページ「[フロー] 表示枠について」
- ▶ 1372 ページ「[リソース] 表示枠について」
- ▶ 1375 ページ「表示領域について」

QuickTest Script Editor・ウィンドウのカスタマイズ

[カスタマイズ] ダイアログ・ボックスでは、スクリプト・エディタ・ツールバー、メニュー、およびその他の表示オプションを、ほかの多くの Windows アプリケーションに似た方法でカスタマイズできます。

[カスタマイズ] ダイアログ・ボックスを開くには、次の手順を実行します。

ツールバーまたはメニュー・バーを右クリックし、[カスタマイズ] を選択します。



タブをクリックし、要件に応じてスクリプト・エディタをカスタマイズします。

[コマンド] タブ

スクリプト・エディタのツールバーおよびメニューのボタンとコマンドを追加および移動できます。また、表示されているツールバーおよびメニューから、ボタンとコマンドを削除することもできます。

[ツールバー] タブ

スクリプト・エディタ・ウィンドウに表示する利用可能なツールバーを選択できます。ツールバー・ボタンに対してテキスト・ラベルを表示するかどうかを選択できます。また、ツールバー表示を標準設定にリセットすることもできます。

[キーボード] タブ

新しいキーボード・ショートカットをツールバーやメニュー・コマンドに割り当てたり、既存のショートカットを変更および削除したりできます。また、すべてのキーボード・ショートカットを標準設定にリセットすることもできます。

[メニュー] タブ

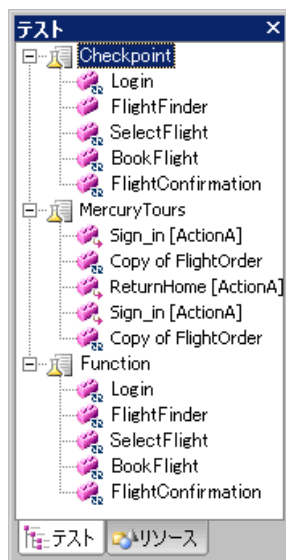
スクリプト・エディタのウィンドウに表示する利用可能なメニュー、およびショートカット・メニューに表示するコマンドを選択できます。メニューのアニメーション設定、および影を表示するかどうかを選択できます。また、表示されているメニューを標準設定にリセットすることもできます。

[オプション] タブ





ツールバー・ボタンのツールチップを表示するかどうか、ツールチップにショートカット・キーを表示するかどうか、およびツールバー・ボタンを大きいアイコンとして表示するか、小さいアイコンとして表示するかを選択できます。



[フロー] 表示枠について

[フロー] 表示枠には、現在開いているテストごとにテスト・フロー（アクション呼び出しのフロー）が表示されます。開いている各テストはツリー内にノードとして表示され、各ノードには、ローカルのアクション、再利用可能なアクション、および外部アクションに対する呼び出しなど、テストで呼び出されたすべてのアクションの階層が含まれます。各テストのアクション呼び出しは、表示領域にある関連するテスト・ウィンドウの [フロー] 表示枠にも表示できます。



[フロー] 表示枠には、次のアイコンが表示されます。

オプション	説明
	テスト
	ローカル・アクションの呼び出し
	外部アクションの呼び出し
	再利用可能なアクションの呼び出し


オプション	説明
	パスがテストとともに保存されていないアクションの呼び出し
	ループになっているアクションの呼び出し。つまり、テスト・フロー階層ですでに呼び出されたアクションの呼び出し。

[フロー] 表示枠では、次の操作を実行できます。

- ▶ **アクションのスクリプトを表示する。** アクションをダブルクリックするか、アクションを右クリックし、[表示] を選択します。表示される各アクションは、テスト・ウィンドウにタブとして表示されます。外部アクションを表示すると、呼び出されたアクションが含まれているテストは、[フロー] 表示枠および [リソース] 表示枠のツリーに追加され、選択したアクションは、表示領域の新しいテスト・ウィンドウに表示されます。
- ▶ **選択したアクションを呼び出すテスト・スクリプトの行を表示する。** アクションを右クリックして [アクションの呼び出しに移動] を選択します。アクション呼び出しのスクリプト行が、テスト・ウィンドウの関連するアクション・タブで強調表示されます。
- ▶ **テストまたはアクションのプロパティを表示する。** テストまたはアクションを右クリックして [プロパティ] を選択します。テストまたはアクションの名前およびそのパスが表示されます。QuickTest に相対パスとして定義されている場合は、パスが .¥ <アクションまたは関数ライブラリの名前> と表示されます。アクションが外部アクションの場合は、[外部] チェック・ボックスが選択されます。
- ▶ **テストを閉じる。** テストを右クリックして [閉じる] を選択します。変更内容を保存していない場合は、保存するように促されます。

テストに、存在しないアクションまたは見つからないアクションに対する呼び出しが含まれている場合も、そのアクションは [フロー] 表示枠のツリーに表示されます。そのアクションを表示しようとする、アクションを見つけることができないことを通知するエラー・メッセージが表示されます。

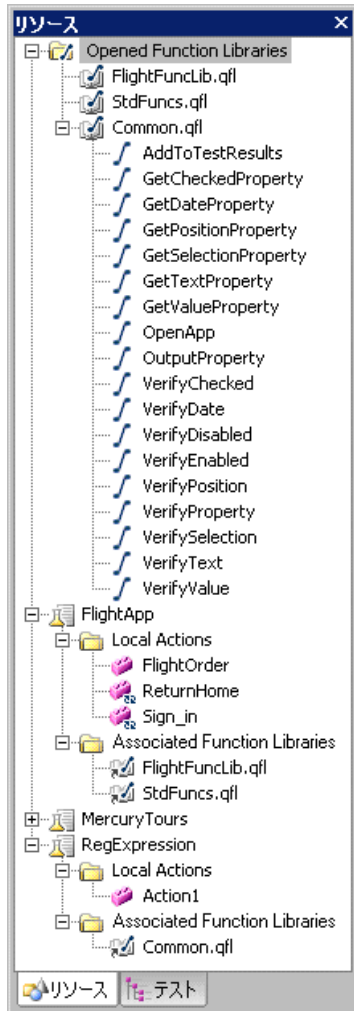
ヒント：

- ▶ [フロー] 表示枠のタイトル・バーで右クリックすると、利用できる表示オプションを表示して [フロー] 表示枠の表示方法を決定できます。たとえば、表示枠を自動的に非表示にしたり、固定したり、閉じたりできます。
- ▶  **[フロー ビューの切り替え]** ツールバー・ボタンをクリックすると、[フロー] 表示枠の表示と非表示を切り替えることができます。








詳細については、1377 ページ「テストを使った作業」を参照してください。

[リソース] 表示枠について

[リソース] 表示枠には、現在開いているテストとそのリソース（アクションおよび関連付けられている関数ライブラリ）がすべて表示されます。各テストはツリー内にノードとして表示され、各ノードには、テストに関連付けられているアクションおよび関数ライブラリが含まれています。現在開いている関数ライブラリおよびその関数はすべて、表示枠上部の個別のノードにも表示されます。




[リソース] 表示枠には、次のアイコンが表示されます。

オプション	説明
	開かれている関数ライブラリ
	関数ライブラリに定義されているパブリック関数
	関数ライブラリに定義されているプライベート関数
	テスト
	ローカル・アクション
	再利用可能なアクション
	テストに関連付けられている関数ライブラリへのリンク

[リソース] 表示枠では、次の操作を実行できます。

- ▶ **ローカル・アクションのスクリプトまたは関数ライブラリのコードを表示する。** アクションまたは関数ライブラリをダブルクリックするか、右クリックして **[表示]** を選択します。表示される各アクションは、テスト・ウィンドウにタブとして表示され、各関数ライブラリは個別のウィンドウに表示されます。
- ▶ **ローカル・アクションまたは関数ライブラリのプロパティを表示する。** アクションまたは関数ライブラリを右クリックして **[プロパティ]** を選択します。アクションまたは関数ライブラリの名前およびそのパスが表示されます。QuickTest に相対パスとして定義されている場合は、パスが **¥** アクションが再利用可能なアクションの場合は、**[再利用可能]** チェック・ボックスが選択されます。
- ▶ **関数ライブラリ内の関数の場所を表示する。** **[開かれている関数ライブラリ]** フォルダで関数を右クリックし、**[関数定義に移動]** を選択します。関数ライブラリ・ウィンドウで、関数定義の最初の行が強調表示されます。
- ▶ **関数ライブラリを閉じる。** **[開かれている関数ライブラリ]** フォルダで関数ライブラリを右クリックし、**[閉じる]** を選択します。変更内容を保存していない場合は、保存するように促されます。
- ▶ **テストを閉じる。** テストを右クリックして **[閉じる]** を選択します。変更内容を保存していない場合は、保存するように促されます。

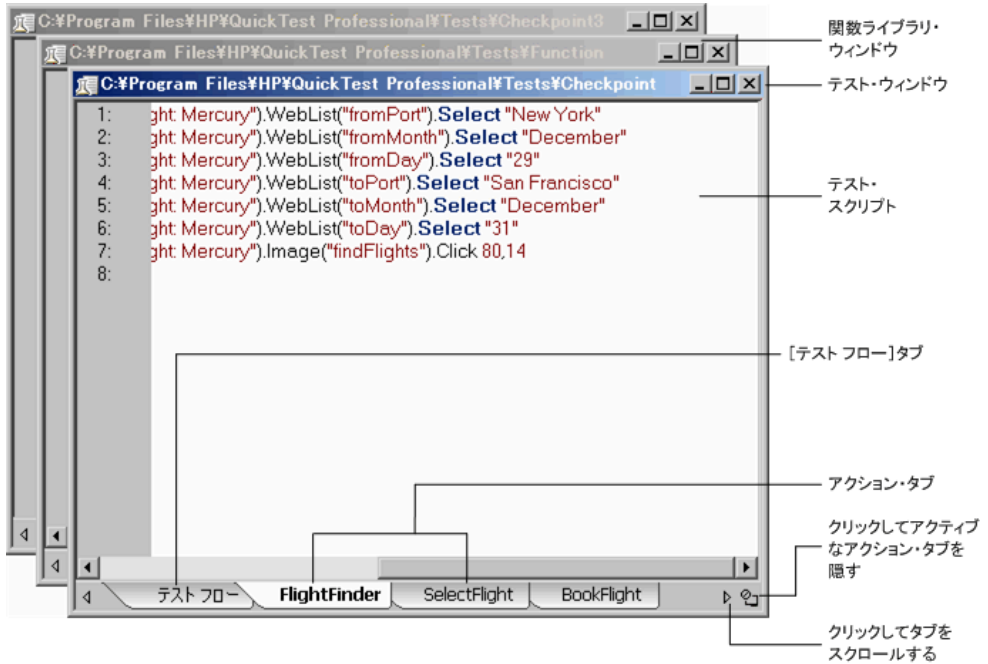
ヒント：

- ▶ [リソース] 表示枠のタイトル・バーで右クリックすると、利用できる表示オプションを表示して [リソース] 表示枠の表示方法を決定できます。たとえば、表示枠を自動的に非表示にしたり、固定したり、閉じたりできます。
 - ▶  **[リソース ビューの切り替え]** ツールバー・ボタンをクリックすると、[リソース] 表示枠の表示と非表示を切り替えることができます。
-

詳細については、1377 ページ「テストを使った作業」および 1381 ページ「関数ライブラリを使った作業」を参照してください。

表示領域について

表示領域には、開いているテストまたは関数ライブラリごとに個別のウィンドウが表示され、各テスト・ウィンドウには、そのテスト内で開いているローカル・アクションごとのタブが含まれています



ヒント: [ウィンドウ] メニューのオプションを使用すると、これらのウィンドウを表示領域に配置する方法を決定できます。

表示領域にアクションまたは関数ライブラリを表示するには、[フロー] または [リソース] 表示枠でアクションまたは関数ライブラリをダブルクリックするか、右クリックして **[表示]** を選択します。テスト・ウィンドウでは、タブは開いているローカル・アクションごとに表示されます。[フロー] または [リソース] 表示枠で外部アクションへの呼び出しをダブルクリックすると、呼び出されたアクションが含まれているテストは [フロー] 表示枠および [リソース] 表示枠のツリーに表示され、テストは呼び出されたアクションのタブとともに表示領域に新しいウィンドウとして表示されます（アクションが含まれているテストがすでに開いている場合、呼び出されたアクションのタブは、表示されていない場合はテスト・ウィンドウに追加されます）。

アクションが存在しない場合、または見つからない場合は、そのアクションを開こうとするとメッセージが表示されます。



開いているローカル・アクションで利用できるすべてのタブが、テスト・ウィンドウの下部に表示されるわけではありません。利用できるタブを切り替えるには、ウィンドウの下部にある矢印をクリックしてタブを切り替えます。



アクション・タブを選択してからウィンドウ右下の **[アクションを非表示にする]** ボタンをクリックすると、ウィンドウからアクションのタブを削除できます。アクションは非表示になるだけで終了していないため、加えた変更を保存するように通知されることはありません。

選択したアクションを呼び出すテスト・スクリプトの行を表示するには、[フロー] 表示枠のツリーでアクションを右クリックし、**[アクションの呼び出しに移動]** を選択します。アクション呼び出しのスクリプト行が、テスト・ウィンドウの関連するアクション・タブで強調表示されます。

関数ライブラリ内の関数の場所を表示するには、[リソース] 表示枠でツリーの最上位にある **[開かれている関数ライブラリ]** フォルダで関数を右クリックし、**[関数定義に移動]** を選択します。関数ライブラリ・ウィンドウで、関数定義の最初の行が強調表示されます。

[エディタ オプション] ダイアログ・ボックスを使用すると ([ツール] > [ビュー オプション])、テスト・スクリプトと関数ライブラリを、どのように QuickTest Script Editor に表示するかをカスタマイズできます。たとえば、行番号を表示するか、またはスクリプトの表示に使用するフォントと色を変更するかを選択できます。[エディタ オプション] ダイアログ・ボックスの詳細については、第 30 章「エキスパート・ビューおよび関数ライブラリ・ウィンドウのカスタマイズ」を参照してください。

詳細については、1377 ページ「テストを使った作業」および 1381 ページ「関数ライブラリを使った作業」を参照してください。

テストを使った作業

複数の既存のテストを開き、編集して保存できます。

また、テスト・スクリプトの表示方法をカスタマイズしたり、各テスト内のテキスト文字列を検索および置換したり、テストを印刷したりできます。詳細については、以下を参照してください。

- ▶ 897 ページ「エキスパート・ビューおよび関数ライブラリ・ウィンドウのカスタマイズ」
- ▶ 850 ページ「テキスト文字列の検索」
- ▶ 852 ページ「テキスト文字列の置換」
- ▶ 339 ページ「テストの印刷」

テストを開く

ファイル・システムのテスト、および Quality Center プロジェクトに保存されているテストを開くことができます。テストは必要な数だけ開くことができます。テストを開くとツリーに表示され、テスト・ウィンドウの [フロー] 表示枠に、テスト内の最上位レベルのアクションに対する呼び出しがすべて表示されます。

ヒント: 既存のテストを開くには、ファイル・システム (Windows Explorer) からスクリプト・エディタのウィンドウにドラッグします。最近使用したテストを開くには、[ファイル] メニューの**最近使用したファイル**のリストから選択します。

テストを開くには、次の手順を実行します。



- 1 (任意) 必要に応じて、[**Quality Center への接続**] ボタンをクリックして Quality Center に接続します。Quality Center への接続の詳細については、1399 ページ「Quality Center との接続と切断」を参照してください。
- 2 次のいずれかの方法でテストを開きます。
 - ▶ [**テストを開く**] ツールバー・ボタンをクリックするか、[**ファイル**] > [**開く**] > [**テスト**] を選択します。[テストを開く] ダイアログ・ボックスが開きます。サイドバーで、テストの場所を選択します (たとえば、**ファイル・システム**や **Quality Center テスト計画**など)。テストを参照して選択し、[**開く**] をクリックします。
 - ▶ [フロー] 表示枠で、開くテストをダブルクリックするか、テストを右クリックして [**表示**] を選択します。

テスト・スクリプトを表示するだけで変更しない場合は、ダイアログ・ボックス下部の [**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。

注：表示領域のアクティブ・ウィンドウに基づいて、[**開く**] ボタンは、[**テストを開く**] と [**関数ライブラリを開く**] の間で切り替えられます。[**関数ライブラリを開く**] ボタンを [テストを開く] に変更するには、ボタンの横にあるドロップダウン矢印をクリックして [**テスト**] を選択するか、表示領域でテスト・ウィンドウをクリックします。

表示領域に<テストのパス>ウィンドウが開き、テストのアクション呼び出しが表示されます。

テストおよびすべてのアクションは [フロー] 表示枠のツリーに表示され、ローカル・アクションおよび関数ライブラリは [リソース] 表示枠のツリーに表示されます。

注： 次の場合、テストは読み取り専用モードで開きます。

- ▶ 選択したテストは、現在、別のユーザが開いています。この場合は、テストがすでに開いていること、および開いているユーザが通知されます。
 - ▶ テストは、以前のバージョンの QuickTest で作成されています。詳細については、1364 ページ「重要な考慮事項」を参照してください。
 - ▶ テストを QuickTest で開き、その同じテストを同じコンピュータ上の QuickTest Script Editor で開こうとしています（逆の場合も同様）。
-

さらに、QuickTest Script Editor でテストを開くと、そのテストはロックされ、そのテストを閉じるまで、ほかのユーザは変更を加えることができません。

テストの編集

QuickTest Script Editor を使用すると、複数のテスト・スクリプトを同時に編集できます。テストを編集するには、情報の追加または変更、コピーと貼り付け、またはほかのテストおよび関数ライブラリから情報のドラッグ・アンド・ドロップを行います。

QuickTest Script Editor でテストを使って作業するときは、新しいテストを作成したり、既存のテストに新しい名前を付けて保存したりできません。テスト・スクリプトのみ変更できます。つまり、テストの設定、パラメータ化、データ・テーブル値などの情報は変更できません。

テスト・スクリプトまたは関数ライブラリを変更する場合は、正しい構文、形式、およびスペルを使用してテスト・スクリプトですべての変更を行うようにしてください（QuickTest Script Editor では行わないため）。

テストを編集するには、次の手順を実行します。



- 1 (任意) 必要に応じて、[**Quality Center への接続**] ボタンをクリックして Quality Center に接続します。Quality Center への接続の詳細については、1399 ページ「Quality Center との接続と切断」を参照してください。
- 2 編集するテスト、および必要に応じて情報のコピー元になるテストを開きます。また、必要になる可能性がある関数ライブラリも開くことができます。
- 3 必要に応じてテストを編集します。変更内容を保存するまで、アスタリスク (*) が編集したテスト・ウィンドウのタイトル・バーに表示されます。

ヒント：



- ▶ 選択したコメントに変換されたテキストをコメントに変換されていないテキストに変更したり、その反対の操作を行ったりするには、[コメントブロック] または [コメントブロック解除] ツールバー・ボタンを使用するか、または [編集] メニューのオプションを使用します。



- ▶ 選択したテキストをインデントまたはインデント解除するには、[インデント] または [インデント解除] ツールバー・ボタンを使用するか、または [編集] メニューのオプションを使用します。
-

テストの保存

アクティブなテストを保存することも、開いているすべてのテストと関数ライブラリを保存することもできます。

テストを保存するには、次の手順を実行します。



- ▶ [保存] ツールバー・ボタンをクリックするか、[ファイル] > [保存] を選択してアクティブなテストを保存します（アクティブなテストとは、現在表示領域でフォーカスが当てられているテスト・ウィンドウのことです）。



- ▶ [すべて保存] ツールバー・ボタンをクリックするか、[ファイル] > [すべて保存] を選択して開いているテストおよび関数ライブラリをすべて保存します。開いている関数ライブラリがこれまで保存されていなかった場合は、[関数ライブラリの保存] ダイアログ・ボックスが開きます。詳細については、1385 ページ「関数ライブラリの保存」を参照してください。

テストを閉じる

テストは、[フロー] 表示枠、[リソース] 表示枠、または表示領域から閉じることができます。

テストを閉じるには、次の手順を実行します。



[フロー] 表示枠または [リソース] 表示枠で、閉じるテストを右クリックして [閉じる] を選択するか、表示領域で閉じるテスト・ウィンドウ上部の [閉じる] ボタンをクリックします。テスト・ウィンドウが閉じられ、テストが [フロー] 表示枠および [リソース] 表示枠から削除されます。

注：保存していない変更がある場合は、テストを閉じる前にこれらの変更を保存するように促されます。

関数ライブラリを使った作業

関数ライブラリ・ファイルには、VBScript 関数、サブルーチン、クラス、モジュールなどを含めることが可能で、テストと関連付けて追加の機能を提供できます。QuickTest Script Editor を使用すると、複数の関数ライブラリを開いて編集したり、新しい関数ライブラリを作成したりすることができます。

関数ライブラリ・コードの表示方法をカスタマイズしたり、各関数ライブラリ内のテキスト文字列を検索および置換したり、関数ライブラリを印刷したりできます。詳細については、次を参照してください。

- ▶ 897 ページ「エキスパート・ビューおよび関数ライブラリ・ウィンドウのカスタマイズ」
- ▶ 850 ページ「テキスト文字列の検索」
- ▶ 852 ページ「テキスト文字列の置換」
- ▶ 918 ページ「関数ライブラリの印刷」

関数ライブラリを開く

関数ライブラリは、ファイル・システムおよび Quality Center プロジェクトの一部である関数ライブラリから開くことができます。関数ライブラリは必要な数だけ開くことができます。QuickTest Script Editor では、**.qfl**、**.vbs**、および **.txt** 関数ライブラリ・ファイルを使用できます。

関数ライブラリを開くと表示領域の関数ライブラリ・ウィンドウに表示され、関数ライブラリとその関数は、[リソース] 表示枠のツリーの最上位にある **[開かれている関数ライブラリ]** フォルダに表示されます。関数ライブラリが開いているテストに関連付けられている場合は、[リソース] 表示枠にあるツリーの **[関連付けられている関数ライブラリ]** フォルダにも関数ライブラリのリンクとしてテストの下に表示されます。

ヒント：既存の関数ライブラリを開くには、ファイル・システム (Windows Explorer) からスクリプト・エディタのウィンドウにドラッグします。最近使用した関数ライブラリを開くには、**[ファイル]** メニューの**最近使用したファイル**のリストから選択します。

関数ライブラリを開くには、次の手順を実行します。



- 1 (任意) 必要に応じて、**[Quality Center への接続]** ボタンをクリックして Quality Center に接続します。Quality Center への接続の詳細については、1399 ページ「Quality Center との接続と切断」を参照してください。
- 2 次のいずれかの方法で関数ライブラリを開きます。
 - ▶ [リソース] 表示枠で、開く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。
 - ▶ **[関数ライブラリを開く]** ツールバー・ボタンをクリックし、**[ファイル]** > **[開く]** > **[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。サイドバーで、関数ライブラリの場所 (たとえば、**ファイル・システム**や **Quality Center テスト計画**など) を選択し、関数ライブラリを参照して選択します。**[開く]** をクリックします。

注：表示領域のアクティブ・ウィンドウに基づいて、[開く] ボタンは、[テストを開く] と [関数ライブラリを開く] の間で切り替えられます。[テストを開く] ボタンを [関数ライブラリを開く] に変更するには、ボタンの横にある矢印をクリックして [関数ライブラリ] を選択するか、表示領域で関数ライブラリのウィンドウをクリックします。

<関数ライブラリのパス> ウィンドウが開き、関数ライブラリが、[リソース] 表示枠のツリーの最上位にある [開かれている関数ライブラリ] フォルダに表示されます。

ファイル・システムから別のユーザが開いている関数ライブラリを開くと、そのユーザによって変更が行われたかどうか通知され、加えられた変更を許可するか拒否するかを選択できます。

Quality Center に保存されている関数ライブラリを開いた場合、ファイルはロックされます。ファイルを閉じるまで、ほかのユーザがそのファイルを変更することはできません。

注：次の場合、関数ライブラリは読み取り専用モードで開きます。

- ▶ 選択した関数ライブラリは、現在、別のユーザが開いています。この場合は、関数ライブラリがすでに開いていること、および開いているユーザが通知されます。
 - ▶ 関数ライブラリは、以前のバージョンの QuickTest で作成されています。詳細については、1364 ページ「重要な考慮事項」を参照してください。
 - ▶ 関数ライブラリを QuickTest で開き、その同じテストを同じコンピュータ上の QuickTest Script Editor で開こうとしています（逆の場合も同様）。
-

関数ライブラリの作成

スクリプト・エディタでは、新しい関数ライブラリを作成できます。この関数ライブラリは、QuickTest を使用してテストに関連付けることができます。

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 [関数ライブラリの新規作成] ツールバー・ボタンをクリックするか、[ファイル] > [新規関数ライブラリ] を選択します。表示領域に関数ライブラリ・ウィンドウが開きます。標準では、関数ライブラリの名前は **Library <番号>** です。
- 2 関数ライブラリに必要なコードを入力します。
- 3 1385 ページ「関数ライブラリの保存」の説明に従って、関数ライブラリを保存します。

関数ライブラリの編集

複数の関数ライブラリの関数コードを編集できます。関数ライブラリを編集するには、情報の追加または変更、コピーと貼り付け、またはほかの関数ライブラリおよびテストから情報のドラッグ・アンド・ドロップを行います。読み取り専用モードで開いた関数ライブラリは編集できません。

関数ライブラリを編集するには、次の手順を実行します。

- 1 必要に応じて、編集する関数ライブラリ、および情報のコピー元の関数ライブラリを開きます。また、必要になる可能性があるテストも開くことができます。
- 2 必要に応じて関数ライブラリを編集します。変更内容を保存するまで、アスタリスク (*) が編集した関数ライブラリ・ウィンドウのタイトル・バーに表示されます。

ヒント:

- ▶ 選択したコメントに変換されたテキストをコメントに変換されていないテキストに変更したり、その反対の操作を行ったりするには、[**コメント ブロック**] または [**コメント ブロック解除**] ツールバー・ボタンを使用するか、または [**編集**] メニューのオプションを使用します。



- ▶ 選択したテキストをインデントまたはインデント解除するには、[**インデント**] または [**インデント解除**] ツールバー・ボタンを使用するか、または [**編集**] メニューのオプションを使用します。
-

関数ライブラリの保存

アクティブな関数ライブラリを保存したり、関数ライブラリの名前を変更して別の場所に保存したり、開いているすべての関数ライブラリおよびテストを保存したりできます。関数ライブラリは、ファイル・システムまたは Quality Center に保存できます。

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 (任意) Quality Center サーバ上のバージョン・コントロールが有効になっていないプロジェクトに接続します。詳細については、1399 ページ「Quality Center との接続と切断」を参照してください。



- 2 [**保存**] ツールバー・ボタンをクリックするか、[**ファイル**] > [**保存**] を選択してアクティブな関数ライブラリを保存します。アクティブな関数ライブラリとは、現在表示領域でフォーカスが当てられている関数ライブラリのウィンドウのことです。

- ▶ [**ファイル**] > [**名前を付けて保存**] を選択してアクティブな関数ライブラリの名前を変更するか、新しい場所に保存します。



- ▶ [**すべて保存**] ツールバー・ボタンをクリックするか、[**ファイル**] > [**すべて保存**] を選択して開いているテストおよび関数ライブラリをすべて保存します。

[関数ライブラリの保存] ダイアログ・ボックスが開きます。

- 3 サイドバーで、テストを保存する場所（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。

注：バージョン・コントロールが有効になっている Quality Center プロジェクトに接続している場合は、関数ライブラリを Quality Center に保存することはできません。したがって、サイドバーには **「ファイル システム」** ボタンのみ表示されます。

- 4 **「ファイル名」** ボックスに、関数ライブラリの名前とファイル拡張子を入力します。QuickTest Script Editor では、**.qfl**、**.vbs**、および **.txt** 関数ライブラリ・ファイルを使用できます。

ファイル名の先頭にスペースを使用することはできません。また、次の文字を含めはなりません。¥/:*?"<>|%'

関数ライブラリを Quality Center に保存する場合、ファイル・パスに 2 つの連続したセミコロン (;;) を含めることはできません。

- 5 **「保存」** をクリックします。指定された場所に関数ライブラリが保存されます。

関数ライブラリを閉じる

関数ライブラリは、**「リソース」** 表示枠または表示領域から閉じることができます。

関数ライブラリを閉じるには、次の手順を実行します。



「リソース」 表示枠で、閉じる関数ライブラリ (**「開かれている関数ライブラリ」** フォルダ内) を右クリックして **「閉じる」** を選択するか、表示領域で閉じる関数ライブラリのウィンドウ上部にある **「閉じる」** ボタンをクリックします。関数ライブラリのウィンドウが閉じ、関数ライブラリが **「リソース」** 表示枠の **「開かれている関数ライブラリ」** フォルダから削除されます。

注：保存していない変更がある場合は、関数ライブラリを閉じる前にこれらの変更を保存するように促されます。

第 50 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テス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、複数のテス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 (1388 ページ)
- ▶ QuickTest オートメーション・スクリプトを使用する条件 (1389 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 (1390 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 (1392 ページ)
- ▶ オートメーション・スクリプトの生成 (1393 ページ)
- ▶ QuickTest オートメーション・リファレンスの使用方法 (1394 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、複数のテス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、テス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、テストを開く操作、[オプション]、[テストの設定]、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、テストの実行、およびテス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テス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る、定期的に行う必要がある 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。
- ▶ **テスト** テス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テストを開き、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、すべてのテス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。
 - ▶ **既存のテス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、既存のテス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。既存のテス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。

- ▶ **アクションとアクション・パラメータの変更**：アクション・スクリプトの内容全体を取得し、新しいアクションの呼び出しなどの必要なステップを追加できます。また、アクションのアクション・パラメータのセットを取得し、アクション・パラメータの値を追加、削除、変更することもできます。
- ▶ **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。

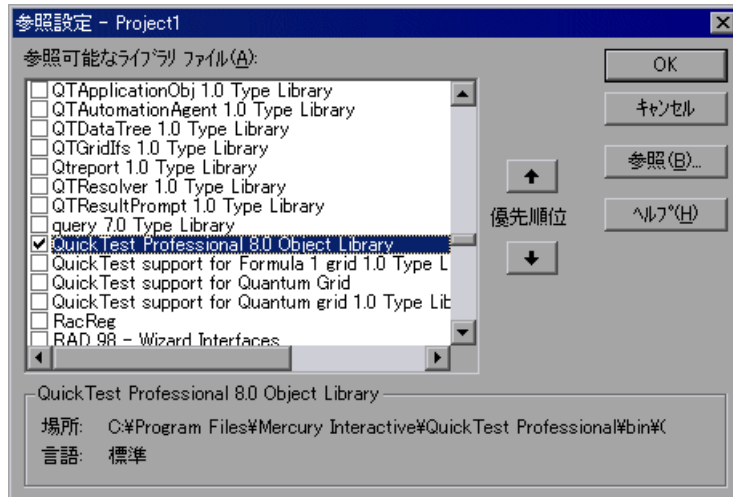
オートメーション・スクリプトの作成

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かあります。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** (テスト・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、テストの起動と保存、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、テス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いければ、**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、1394 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

[テストの設定] ダイアログ・ボックスの [プロパティ] 表示枠、[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], [オブジェクトの認識], [テストの設定] 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 4 章「オブジェクト認識の設定」, 第 44 章「グローバル・テスト・オプションの設定」, および第 45 章「個別のテストのオプション設定」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 51 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、多くのテス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 (1398 ページ)
- ▶ Quality Center との接続と切断 (1399 ページ)
- ▶ QuickTest の Quality Center との統合 (1404 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、テストを作成し、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テストに関連付けることができます。すべてのテストまたは1つのテストに対して、外部ファイルに関連付けることができます。たとえば、新規テストのための標準リポジトリ・モードとして、共有オブジェクト・リポジトリ・モードを設定するとします。この場合、Quality Center に格納されている特定のオブジェクト・リポジトリを使用するように QuickTest に指定できます。

すべてのテストで外部ファイルを使用するように指定する方法の詳細については、第 44 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。1つのテストで外部ファイルを使用するように指定する方法の詳細については、第 45 章「個別のテストのオプション設定」を参照してください。

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、1009 ページ「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、1404 ページ「Quality Center Connectivity Add-in」を参照してください。

QuickTest の Quality Center への接続

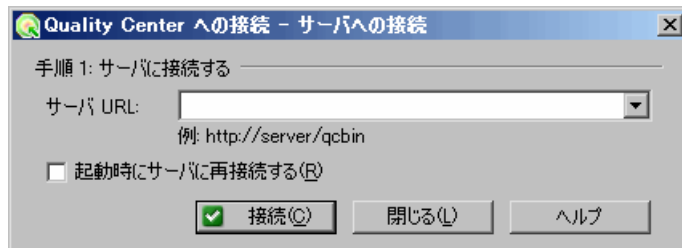
接続プロセスには 2 つの段階があります。まず、QuickTest をローカルまたはリモートの Quality Center サーバに接続します。このサーバによって、QuickTest と Quality Center プロジェクト間の接続が処理されます。

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。



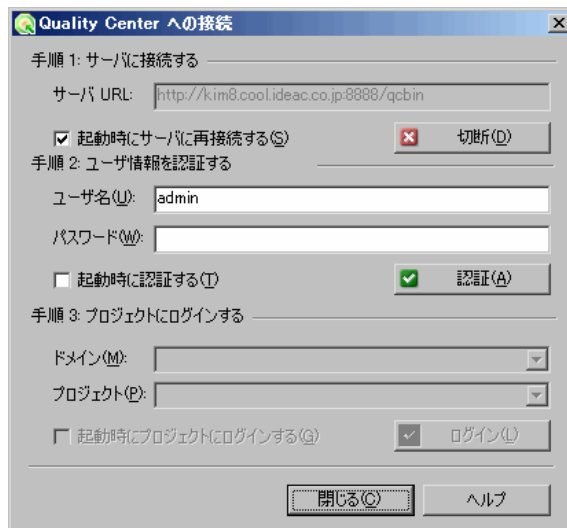
- 1 [ファイル] > [Quality Center への接続] を選択するか、[Quality Center への接続] ツールバー・ボタンをクリックします。[Quality Center への接続 - サーバへの接続] ダイアログ・ボックスが表示されます。



- 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

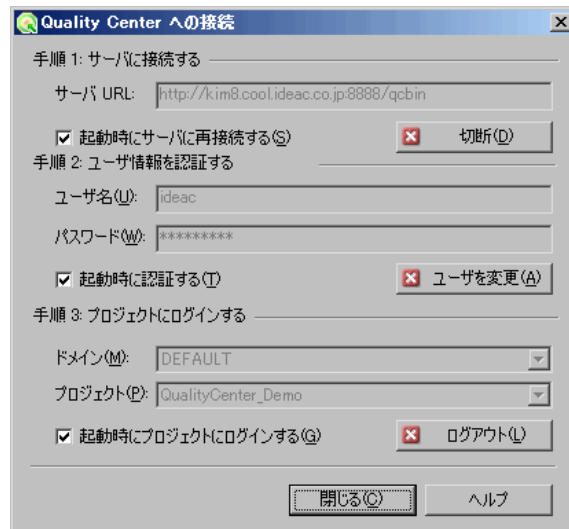
QuickTest は Quality Center プロジェクト、または Quality Center サーバから、いつでも切断できます。最初にプロジェクトへの接続を解除せずに、Quality Center サーバへの QuickTest の接続を解除すると、QuickTest から当該プロジェクト・データベースへの接続は自動的に解除されます。

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、75 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 テストおよびリソース・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、1429 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、テストおよび関数ライブラリで **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。

Quality Center プロジェクトへのテストの保存

QuickTest を Quality Center プロジェクトに接続すると、QuickTest で新しいテストを作成し、そのテストをプロジェクトに直接保存することができます。テストを保存するには、テストにわかりやすい名前を付け、テスト計画ツリーの対応するサブジェクトに関連付けます。これにより、各サブジェクトに作成したテストを容易に追跡し、テストの計画と作成の進行状況を素早く確認することができます。

ヒント：Quality Center プロジェクトに格納されているテストの個別のポータブル・コピーを後で保存できます。たとえば、Quality Center へのアクセス許可がないため、別のサイトを巡回中にテストを開くか実行する必要があるかもしれません。詳細については、332 ページ「テストの携帯用コピーの作成」を参照してください。

テストを Quality Center 10.00 プロジェクトに保存するには、次の手順を実行します。

- 1 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、1400 ページ「QuickTest の Quality Center への接続」を参照してください。



- 2 QuickTest で **[保存]** をクリックするか、**[ファイル]** > **[保存]** を選択して、テストを保存します。**[テストの保存]** ダイアログ・ボックスが開きます。
- 3 サイドバーで **[Quality Center テスト]** をクリックし、該当のサブジェクト・フォルダを探して選択します。
- 4 **[ファイル名]** ボックスにテストの名前を入力します。テストを識別しやすい、わかりやすい名前を付けます。テストの名前を付けるには、次のガイドラインを考慮します。
 - ▶ (パスも含めて) 220 文字までです。
 - ▶ 先頭および最後にスペースを使うことはできません。
 - ▶ 次の文字は使用できません。 ¥/:*?"<>|%'
 - ▶ セミコロンを 2 つ連続して使用できません (;;)。

- 5 テストと一緒に ActiveScreen ファイルを保存する場合、**[ActiveScreen ファイルの保存]** が選択されていることを確認します。このチェック・ボックスをクリアすると、ActiveScreen ファイルが削除されるため、ActiveScreen オプションを使ったテストの編集ができなくなります。詳細については、330 ページ「テストの保存」を参照してください。
- 6 **[保存]** をクリックし、テストを保存してダイアログ・ボックスを閉じます。QuickTest がテストを保存している間は、ステータス・バーのテキストが変わります。

次回 Quality Center 10.00 起動時に、新しいテストが Quality Center のテスト計画ツリーに表示されます。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。

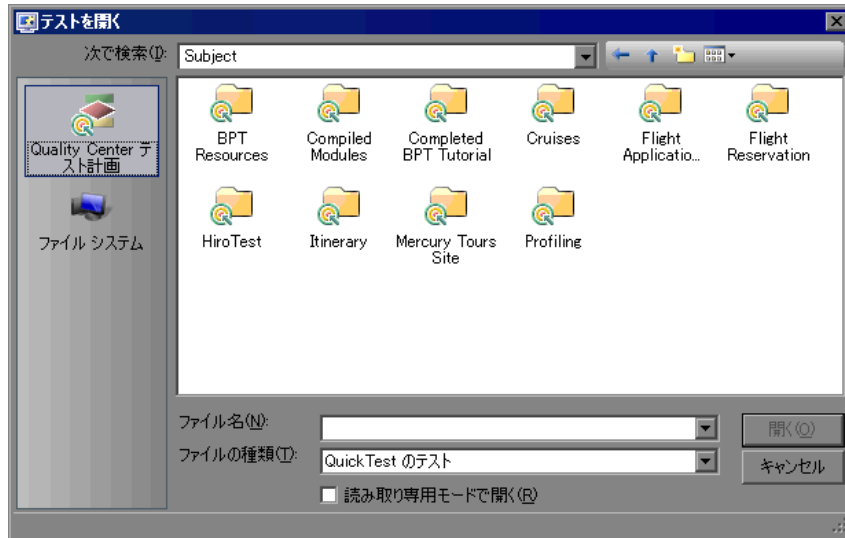
Quality Center プロジェクトからテストを開く

QuickTest を Quality Center プロジェクトに接続すると、Quality Center プロジェクトに含まれている QuickTest テストを開くことができます。ファイル・システム内の実際の格納場所ではなく、テスト計画ツリーの中でテストを探します。**[ファイル]** メニューの最近使用したテストのリストからテストを開くこともできます。

注：以前のバージョンを使用して作成したテストが Quality Center に格納されている場合、このテストを開くと読み取り専用モードになります。このテストを編集するには、QuickTest Professional Asset Upgrade Tool for Quality Center を使用して現在のバージョンにアップグレードする必要があります。

Quality Center 10.00 プロジェクトからテストを開くには、次の手順を実行します。

- 1 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、1400 ページ「QuickTest の Quality Center への接続」を参照してください。
- 2 QuickTest で、**[開く]** をクリックするか、**[ファイル] > [開く] > [テスト]** を選択し、テストを開きます。**[テストを開く]** ダイアログ・ボックスが開きます。



- 3 サイドバーで **[Quality Center テスト計画]** ボタンをクリックし、必要なテストを探して選択します。テストが **[ファイル名]** ボックスに表示されます。

注：テストが、バージョン・コントロールがサポートされている Quality Center プロジェクトに格納されている場合、**[表示]** の下向き矢印をクリックして **[詳細]** を選択すると、テストのバージョン・コントロール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テストの名前です。
- ▶ **[変更者]** カラムに表示されているのは、テストを作成、または最後に更新した Quality Center ユーザです。
- ▶ **[チェックアウト先]** カラムに表示されているのは、このテストを現在チェックアウトしている Quality Center ユーザです。テストがチェックインしていれば、このカラムは空白です。

4 読み取り専用モードでテストを開くには、**[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。

5 **[開く]** をクリックしてテストを開きます。または、バージョン・コントロールが有効なプロジェクトにチェックインされているテストを開いて変更する場合は、**[開く]** の下向き矢印をクリックして **[開いてチェックアウトする]** を選択します。

QuickTest がテストをダウンロードして開くのに応じて、実行される操作がステータス・バーに表示されます。

テストを開くと、QuickTest タイトル・バーに、「**[Quality Center]**」という文字列と、サブジェクトのフルパスおよびテスト名が表示されます。次に例を示します。

[Quality Center] Subject¥System¥qa_test1

次の場合、テストは読み取り専用モードで開きます。

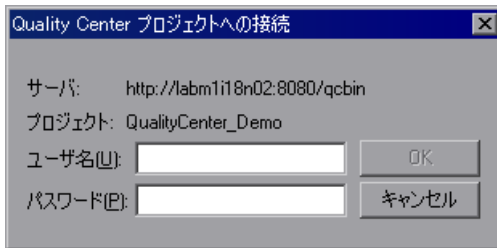
- ▶ **[読み取り専用モードで開く]** を選択した場合
- ▶ 現在（バージョン・コントロール・サポートのプロジェクトの）Quality Center バージョン・コントロール・データベースにチェックインしているテストを開いた場合

- ▶ 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
- ▶ 以前のバージョンの Quality Center のテストを開き、テストをまだ最新の形式に更新していない場合

詳細については、1410 ページ「バージョン・コントロール・サポート付きの Quality Center プロジェクトからテストを開く」を参照してください。

最近使用したテストのリストからテストを開く

[ファイル] メニューの最新ファイル・リストから Quality Center テストを開くこともできます。Quality Center プロジェクトのテス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、テス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テストを開いたユーザの名前が表示されます。



また、前回異なる Quality Center ユーザ名を使用して編集したテス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することもできます。

バージョン・コントロール・サポート付きの Quality Center プロジェクトからテストを開く

[開く] ツールバー・ボタンをクリックするか、[ファイル] > [開く] > [テスト] を選択してバージョン・コントロールがサポートされている Quality Center プロジェクトからテストを開き、[詳細] ビュー（[テストを開く] ダイアログ・ボックスで [表示] の下向き矢印をクリックして [詳細] を選択）を表示すると、テストがチェックアウトされていれば [チェックアウト先] カラムにテストをチェックアウトした Quality Center ユーザの名前が指定されます。

バージョン・コントロール・サポートの Quality Center プロジェクトからテストを開くと、テストは現在のバージョン・コントロール・ステータスに応じて、読み書きモードまたは読み取り専用モードで開きます。

次の表に、バージョン・コントロール・ステータスと各ステータスのオープン・モードを示します。

説明	オープン・モード
チェックイン ：現在テストがバージョン・コントロール・データベースにチェックインしている場合、ダイアログ・ボックスには何も示されません。	読み取り専用
ユーザがチェックアウト ：現在ユーザがテストをチェックアウトしている場合、ダイアログ・ボックスにユーザの名前が表示されます。	読み取り / 書き込み
ほかのユーザがチェックアウト ：現在ほかのユーザがテストをチェックアウトしている場合、ダイアログ・ボックスにそのユーザの名前が表示されます。	読み取り専用

バージョン・コントロールの Quality Center プロジェクトに格納されているテストを使った作業の詳細については、1462 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。

テンプレート・テストを使用した作業

テンプレート・テストは、Quality Center で作成されるすべての QuickTest テストの土台となります。テンプレート・テストは、標準のテスト設定を含む QuickTest テストです。たとえば、テンプレート・テストでは、QuickTest アドイン、関連付けられている関数ライブラリ、テストに関連付けられている回復シナリオを指定します。これらのテスト設定は、QuickTest の [テストの設定] ダイアログ・ボックス ([**ファイル**] > [**設定**]) で変更できます。

標準のテスト設定以外に、テンプレート・テストには Quality Center で作成されるすべての新規の QuickTest テストに追加したい、任意のコメントやステップを含めることもできます。たとえば、テンプレート・テストに関連付けられているアドインの種類をユーザに伝えるコメントを追加したり、すべてのテストの開始時に特定の Web ページまたはアプリケーションを開くステップを追加したりできます。テンプレート・テストに追加したステップやコメントは、そのテンプレート・テストを土台として Quality Center で作成されるすべての新規のテストに追加されます。

Quality Center 用の QuickTest Professional Add-in をインストールするとき、Quality Center クライアントごとに標準設定のテンプレート・テストがインストールされます。この標準設定のテンプレート・テストに変更を加えたり、さまざまなテスト設定が関連付けられたカスタマイズ・テンプレート・テストを作成したりできます。

テンプレート・テストはすべて (Quality Center クライアントに置かれる標準設定のテンプレート・テストを除き) Quality Center プロジェクトに保存されます。各ユーザのローカル・コンピュータにコピーする必要はありません。これにより、ユーザは必要な場合に各自の標準設定のローカル・テンプレート・テストをカスタマイズできる一方で、グローバルに管理されたテンプレート・テストにもアクセスできます。詳細については、1413 ページ「新しいテンプレート・テストを使用した作業」を参照してください。

特定のテンプレート・テストを土台とするテストが Quality Center から実行されると、テスト内の定義に従って、関連付けられているアドインが QuickTest によって自動的にロードされ、必要な設定が適用されます。

標準設定のテンプレート・テストを使用した作業

Quality Center 用の QuickTest Add-in をインストールすると、サポートされているすべての QuickTest バージョンに対応した標準設定のテンプレート・テストが、コンピュータの **< Quality Center 用 QuickTest Add-in フォルダ >** **¥bin¥Templates** フォルダ (C:¥Program Files¥HP¥QuickTest Add-in for Quality Center¥bin¥Templates¥Template10 など) にインストールされます。

Quality Center ユーザが Quality Center で QuickTest テストを新規に作成すると、ユーザが別のテンプレート・テストを選択しない限り、インストールされている QuickTest バージョンに対応した標準設定のテンプレート・テストが自動的にテストに関連付けられます。詳細については、1416 ページ「Quality Center での QuickTest テストの作成」を参照してください。

Quality Center 用の QuickTest Add-in とともに標準設定でインストールされるテンプレート・テストには、変更を加えることができます。標準設定のテンプレート・テストはローカルにインストールされるため、テンプレート・テストに加えた変更は自分のコンピュータで (Quality Center クライアントを使用し) て作成したテストにのみ適用されます。

あるいは、次項の説明に従って、テンプレート・テストを新規に作成することもできます。

標準設定のテンプレート・テストを Quality Center 内の新しいテストに適用する方法の詳細については、1416 ページ「Quality Center での QuickTest テストの作成」を参照してください。

新しいテンプレート・テストを使用した作業

新しいテンプレート・テストを作成すると、それらは Quality Center プロジェクトに格納され、その Quality Center プロジェクトで作成される新しい QuickTest テストの土台としてすべての Quality Center ユーザが使用できるようになります。

テンプレート・テストを複数作成し、それぞれを特定のテスト目的に使用することができます。たとえば、ActiveX コントロールを使用する Web アプリケーションをテストする QuickTest テスト向けのテンプレート・テストを作成し、それとは別に、標準の Windows アプリケーションをテストする QuickTest テスト用のテンプレート・テストを作成できます。この場合、前者のテンプレート・テストには ActiveX Add-in と Web Add-in を関連付けます。後者のテンプレート・テストには、QuickTest のアドインを一切関連付けずに、テストの対象となる Windows アプリケーションを指定します。このほか、必要に応じてそれぞれのテンプレート・テストのテスト設定に変更を加えることもできます。

作成した各テンプレート・テストは、`ActiveX_Web_Addins_Template` や `Std_Windows_Template_Test` のように、用途を明確に示すわかりやすい名前を付けて保存できます。そのようにすることで、ユーザは Quality Center で QuickTest テストを作成するときに適切なテンプレート・テストを選択できます。

注：特定の QuickTest アドインの関連付を行うテンプレート・テストを定義するときは、テストを最終的に実行する QuickTest コンピュータに、そのアドインが実際にインストールされていることを確かめておきます。アドインがインストールされていないと、テストを実行したときに、QuickTest が必要なアドインをロードすることができず、テストが失敗することがあります。Quality Center からの QuickTest テストの実行の詳細については、Quality Center のドキュメントを参照してください。

テンプレート・テストの新規作成

テンプレート・テストを作成するには、まず、QuickTest で必要なテスト設定を持つ空のテストを作成します。次に、Quality Center プロジェクトの**テスト計画**モジュールで QuickTest テストを参照し、それを「**テンプレート・テスト**」として指定します。

QuickTest でテストを保存するときは、用途を明確に示すわかりやすい名前を付けます。たとえば、テンプレート・テストを、ActiveX Add-in と Web Add-in を新しいテストに関連付ける目的に使用する場合、`ActiveX_Web_Addins_Template` という名前を付けることができます。

ヒント：Quality Center のテスト計画ツリー（テスト計画モジュール）で、テンプレート・テスト用の特別なフォルダを作成することもできます。このようにすれば、ほかのユーザが Quality Center で QuickTest テストを新規に作成するときに、関連するテンプレート・テストをすぐに見つけることができます。

テンプレート・テストを作成するには、次の手順を実行します。

1 QuickTest での作業 :

- a 必要なアドインをロードした状態で QuickTest を開きます。QuickTest 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**』の QuickTest アドインのロードの項を参照してください。
- b [テストの設定] ダイアログ・ボックス ([**ファイル**] > [**設定**]) で必要な設定を定義します。詳細については、1252 ページ「[テストの設定] ダイアログ・ボックスの使用」を参照してください。
- c このテンプレート・テストを土台とするすべてのテストにコメントやステップを追加したい場合は、それらを追加します。
- d [**保存**] ボタンをクリックするか、[**ファイル**] > [**保存**] を選択してテストを保存します。[Quality Center にテストを保存] ダイアログ・ボックスが表示されます 用途を明確に示すわかりやすい名前を付けて、テストを Quality Center プロジェクトに保存します。詳細については、1406 ページ「Quality Center プロジェクトへのテストの保存」を参照してください。



2 Quality Center での作業 :

- a Quality Center でプロジェクトを開き、サイドバーの [**テスト計画**] ボタンをクリックしてテスト計画モジュールを開き、手順 d で保存したテストを参照します。
- b テストを右クリックし、[**テンプレート テストとしてマーク**] を選択します。テストがテンプレート・テストに変換されます。



- 3 必要に応じて、手順 1 から 2 を繰り返して追加のテンプレート・テストを作成します。

Quality Center での QuickTest テストの作成

Quality Center では、テスト計画モジュールで QuickTest テストを作成します。QuickTest テストを作成するときは、テンプレート・テストをそのテストに適用します。テンプレート・テストは、定義済みのテスト設定を新しい QuickTest テストに適用します。たとえば、テンプレート・テストでは、テストに関連付ける QuickTest アドイン、関数ライブラリ、回復シナリオなどを指定できます。また、必要に応じて、コメントやステップ（ステートメント）を含めることもできます。QuickTest クライアントに格納されている標準設定のテンプレート・テストを洗濯することも、Quality Center プロジェクトに保存されているテンプレート・テストを選択することもできます。

Quality Center プロジェクトにテンプレート・テストをまったく保存していない場合や、[テンプレート] ボックスで [くなし>] を選択した場合（1406 ページに示した [テストの新規作成] ダイアログ・ボックスにあります）、Quality Center では、Quality Center クライアントに「**Quality Center 用 QuickTest Add-in**」と一緒にインストールされたテンプレート・テストに定義されている設定が使用されます。詳細については、1413 ページ「標準設定のテンプレート・テストを使用した作業」を参照してください。そうではなく、少なくとも 1 つのテンプレート・テストが Quality Center プロジェクトに保存されている場合は、QuickTest テストの新規作成時にはそのテンプレート・テストを選択できます。詳細については、1413 ページ「新しいテンプレート・テストを使用した作業」を参照してください。

注：Quality Center で QuickTest テストを作成するときは、テストに関連付ける QuickTest アドインを指定するテンプレート・テストを選択する必要があります。アドインを指定するテンプレートを選択しないと、必要な QuickTest アドインが実行セッション中にロードされません。

新しい QuickTest テストでは、選択したテンプレート・テストに定義されているすべての設定が使用されます。テストが Quality Center から実行されたときは、[テストの設定] ダイアログ・ボックスに指定されている設定が QuickTest によって使用され、必要な QuickTest アドインが自動的にロードされます。

注： 次の手順は、Quality Center でテンプレート・テストを使用してテストを作成する方法を示します。この手順は、使用する Quality Center のバージョンによって異なる場合があります。Quality Center でテストを新規に作成するための最新の手順については、『Mercury Quality Center ユーザーズ・ガイド』を参照してください。

Quality Center でテンプレート・テストを使用してテストを作成するには、次の手順を実行します。

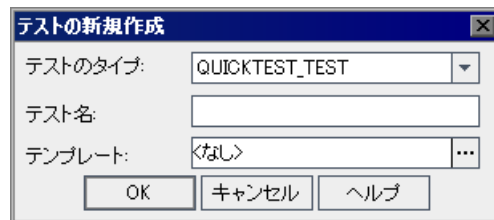


1 Quality Center で、サイドバーの [テスト計画] ボタンをクリックして、テスト計画モジュールを開きます。

2 テスト計画ツリーで、フォルダを選択します。



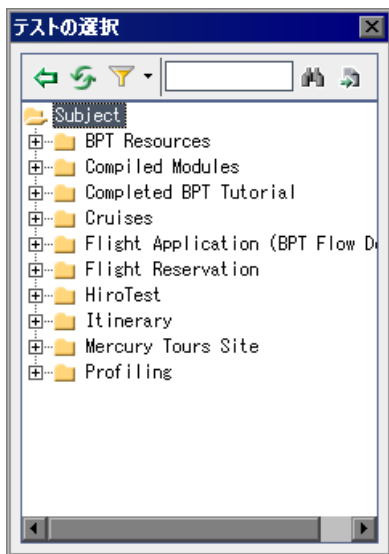
3 [テストの新規作成] ボタンをクリックするか、[テスト] > [テストの新規作成] を選択します。[テストの新規作成] ダイアログ・ボックスが開きます。



注： Quality Center 用の QuickTest Professional Add-in がコンピュータにインストールされている場合にのみ、[テンプレート] ボックスが表示されます。[テンプレート] ボックスが表示されない場合、Quality Center 用の QuickTest Professional Add-in を QuickTest Professional DVD からインストールするか、[その他の Quality Center アドイン] ページ ([Quality Center オプション] ウィンドウ、または [ヘルプ] > [アドイン ページ] から開きます) からインストールする必要があります。

4 [テストのタイプ] リストから、[QUICKTEST_TEST] を選択します。

- 5 [テスト名] ボックスに、英字 (Roman)、数字、アンダスコア (必要ならば) を使用してテストの名前を入力します。テストの名前は (パスも含めて) 220 文字までです。セミコロンを 2 つ連続して使用できません (;)。名前の先頭および最後にスペースを使うことはできません。また、次の文字は使用できません。¥/:*?"<>|%'
- 6 [テンプレート] ボックスの参照ボタンをクリックします。[テストの選択] ダイアログ・ボックスが開きます。
- 7 テンプレート・テストが格納されているフォルダを展開します。



- 8 新しいテストの土台とするテンプレート・テストを選択し、[追加] ボタンをクリックします。[テストの選択] ダイアログ・ボックスが閉じ、選択したテンプレート・テストが [テンプレート] ボックス ([テストの新規作成] ダイアログ・ボックス内) に表示されます。
- 9 [テストの新規作成] ダイアログ・ボックスで、[OK] をクリックします。テンプレート・テストに定義されているテスト設定を持つ、新しいテストが作成されます。
- 10 新しいテストが、テスト計画ツリー内の選択したサブジェクト・フォルダの下に表示されます。

注： [必須フィールド] ダイアログ・ボックスが表示された場合は、必要な値を設定して [OK] をクリックします。詳細については、『HP Quality Center Administrator Guide』（英語版）を参照してください。

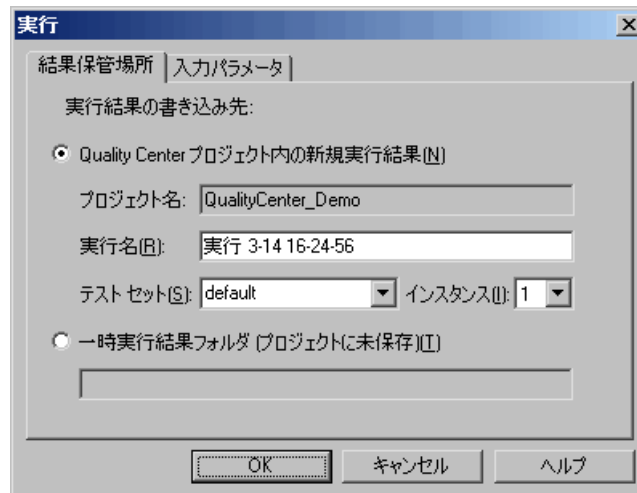
- 11 テストの作成を続けます。Quality Center でのテストの作成の詳細については、『Mercury Quality Center ユーザーズ・ガイド』を参照してください。

Quality Center プロジェクトに格納されているテストの QuickTest からの実行

QuickTest では、Quality Center プロジェクトからテストを実行し、実行結果をプロジェクトに保存できます。実行結果を保存するには、結果を格納する実行セッションとテスト・セットの名前を指定します。

テストの実行結果を Quality Center プロジェクトに保存するには、次の手順を実行します。

- 1 QuickTest で、[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・ボックスが表示されます。



- 2 [プロジェクト名] ボックスには現在接続している Quality Center プロジェクトが表示されます。

テストの実行結果を Quality Center プロジェクトに保存するには、標準設定の**実行名**を受け入れるか、別の名前をボックスに入力します。

- 3 標準設定の**テスト・セット**を受け入れるか、別のテスト・セットを選択します。
- 4 テスト・セットにテストのインスタンスが複数ある場合には、結果を保存するテストのインスタンスを [インスタンス] ボックスに指定します。

注: 「**テスト・セット**」とは、特定のテスト目標を達成するために選択されたテストで構成されたグループです。たとえば、アプリケーションのユーザ・インタフェースや、負荷をかけた状態のアプリケーションのパフォーマンスをテストするテスト・セットを作成できます。テスト・セットの定義は、Quality Center のテスト実行モードで行います。詳細については、Quality Center のドキュメントを参照してください。

テストを実行し、前のテスト実行の結果を上書きするには、[**一時実行結果フォルダ (プロジェクトに未保存)**] オプションを選択します。

注: QuickTest では、すべてのテストの一時的なテスト実行結果が **%TMP%\TempResults** に格納されます。[**一時実行結果フォルダ (プロジェクトに未保存)**] 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。

- 5 [OK] をクリックします。[実行] ダイアログ・ボックスが閉じ、QuickTest によってテストの実行が開始されます。QuickTest によるテストの実行が進行すると同時に、キーワード・ビューの各ステップが強調表示されます。

テストの実行が停止すると、[オプション] ダイアログ・ボックスの [実行] 表示枠で [**テスト実行終了時に結果を表示する**] チェック・ボックスをクリアしていないかぎり、[テスト結果] ウィンドウが表示されます。[オプション] ダイアログ・ボックスの詳細については、第 44 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。

テストの実行が停止すると、ステータス・バーに「アップロードしています」と表示されます。アップロード処理が完了すると、[テスト結果] ウィンドウが表示されます。

注：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。詳細については、1009 ページ「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 テストの実行に関する設定

Quality Center データベースに格納されている QuickTest テストの実行は、QuickTest、またはコンピュータにインストールされている Quality Center クライアント、あるいはリモートの Quality Center クライアントを介して行います。Quality Center は、QuickTest テストを実行するときに関連アドイン一覧を使用して、テストに適したアドインを QuickTest コンピュータにロードします。詳細については、1257 ページ「関連アドインの変更」および 1412 ページ「テンプレート・テストを使用した作業」を参照してください。

注：

- ▶ QuickTest コンピュータがログオフまたはロックされている場合、Quality Center から QuickTest テストを実行することはできません。
 - ▶ 標準設定では、Quality Center がテスト・セットのテストを実行するために QuickTest をアクティブにすると、QuickTest は非表示モードで起動して実行されます。これはパフォーマンスの向上に役立ちます。この設定は、QuickTest リモート・エージェントで変更できます。また、QuickTest が Quality Center テストを非表示モードで実行していることを示すツールチップ・ウィンドウを表示するよう、リモート・エージェントに指示することもできます。詳細については、1423 ページ「QuickTest リモート・エージェントの設定」を参照してください。
-

Quality Center テストを QuickTest コンピュータで実行するとき、失敗したステップに関する不具合を報告するよう指示できます。また、[QuickTest テスト結果] ウィンドウから Quality Center に手作業で不具合を送信できます。詳細については、1009 ページ「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。

QuickTest テストを実行するようにリモートの Quality Center クライアントに指示する前に、QuickTest アプリケーションを使用する権限を Quality Center に与える必要があります。QuickTest リモート・エージェントの設定の表示や修正もできます。


Quality Center に対する QuickTest コンピュータでのテストの実行許可

セキュリティ上の理由により、QuickTest アプリケーションへのリモート・アクセスは許可されていません。Quality Center（またはほかのリモート・アクセスのクライアント）に対して QuickTest テストを開いて実行する許可を与えるには、[オプション] ダイアログ・ボックスで **[ほかの HP 製品でテストまたはコンポーネントを実行可能にする]** オプションを選択する必要があります。

注： QuickTest テストを Quality Center からリモート実行する必要がある、QuickTest が Windows XP Service Pack 2、Windows 2003 Server、または Windows Vista にインストールされている場合は、先に DCOM のアクセス許可を変更してファイアウォール・ポートを開く必要があります。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。

また、QuickTest テストを Quality Center からリモートで実行する必要がある、QuickTest が Windows Vista にインストールされている場合、最初に Quality Center に接続する前に、**Windows のコントロール・パネル**の **[ユーザー アカウント]** を使用して、ユーザー・アカウント・コントロール (UAC) を無効にする必要があります。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。

リモートの Quality Center クライアントが QuickTest コンピュータでテストを実行可能にするには、次の手順を実行します。

- 1 QuickTest を開きます。
- 2  [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。
- 3 [実行] ノードをクリックします。
- 4 [他の HP 製品でテストまたは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。

このオプションの詳細については、1243 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。

ヒント：Quality Center から QuickTest テストにフル・アクセスするために、Quality Center の QuickTest Add-in が Quality Center クライアント・コンピュータにインストールされている必要があります。これにより、テストの表示やテスト結果ビューアでの実行結果の表示ができます。このアドインの詳細については、QuickTest Professional Add-in 画面（Quality Center のメイン画面からアクセスできます）を参照してください。

QuickTest リモート・エージェントの設定

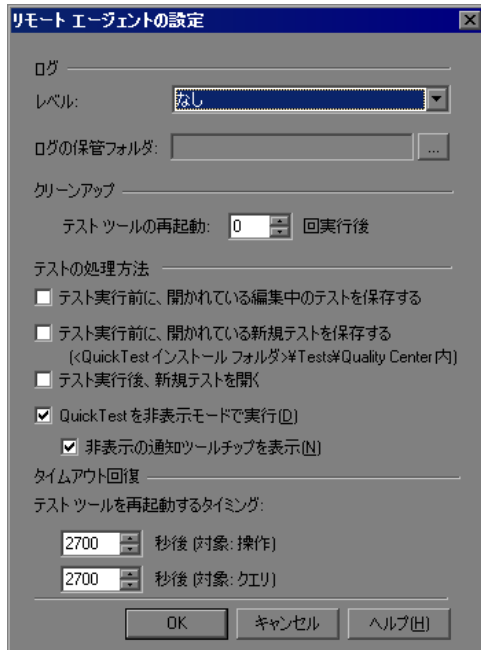
Quality Center から QuickTest テストまたはビジネス・プロセス・テストを実行すると、Quality Center コンピュータで QuickTest リモート・エージェントが開きます。テストが Quality Center などのリモートのアプリケーションによって実行された場合の QuickTest の動作は、QuickTest リモート・エージェントによって決まります。

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。

5:11 PM

- 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、1425 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 テストまたはビジネス・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
<p>【レベル】</p>	<p>Quality Center が QuickTest テストまたはビジネス・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。</p> <p>【なし】 (標準設定) : ログは作成されません。</p> <p>【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。</p> <p>【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。</p> <p>【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。</p>
<p>【ログの保管フォルダ】</p>	<p>ログ・ファイルが格納されているフォルダのパスです。</p> <p>【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。</p>
<p>【テスト ツールの再起動 X 回実行後】</p>	<p>QuickTest テストの場合、Quality Center がテスト実行を指定されている回数完了させたら、QuickTest を再起動します。QuickTest は再起動すると、テスト・セットの中の次のテストを実行します。</p> <p>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。</p> <p>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。</p>

オプション	説明
<p>[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]</p>	<p>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）テストまたはが開かれている場合、このオプションにより、開いているテストまたは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。</p> <p>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。</p>
<p>[テスト実行前に、開かれている新規テストを保存する]</p>	<p>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）テストが開かれている場合、テストにはシーケンシャルのテスト名がつけられ、< QuickTest のインストール・フォルダ> ¥Tests¥Quality Center に保存されます。</p>
<p>[テスト実行後、新規テストを開く]</p>	<p>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。</p>

オプション	説明
<p>[QuickTest を非表示モードで実行]</p>	<p>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。</p> <p>[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。</p> <p>注：</p> <ul style="list-style-type: none"> ▶ 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
<p>[テスト ツールを再起動するタイミング]</p>	<p>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。</p> <p>操作：開く、実行などの QuickTest 操作です。</p> <p>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)。</p> <p>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。</p> <p>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。</p>

第 52 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 テストを 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（1430 ページ）
- ▶ リソースおよび依存関係モデルについて（1431 ページ）
- ▶ アセットの依存関係を使用する利点（1433 ページ）
- ▶ Quality Center でのリソースおよび依存関係モデルの使用（1435 ページ）

リソースおよび依存関係モデルの用語

用語	説明
<p>アセット</p>	<p>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。</p> <ul style="list-style-type: none"> ▶ テスト ▶ アクション ▶ 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 ▶ データ・テーブル・ファイル ▶ 環境変数ファイル <p>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。</p>
<p>リソース</p>	<p>テストで使用されるすべてのアセット。たとえば、テス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、テス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。</p> <ul style="list-style-type: none"> ▶ 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 ▶ データ・テーブル・ファイル ▶ 環境変数ファイル

用語	説明
依存関係	<p>リソースまたは外部アクションと特定のテスト間のリンクされた関係。関連付けられているリソース・ファイルおよびアクションは、そのリソースを使用するかアクションを呼び出す各テストにリンクされています。</p> <p>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。</p> <ul style="list-style-type: none"> ▶ テスト計画モジュール：テスト ▶ 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、データ・テーブル・ファイル、環境変数ファイル <p>注：テスト計画モジュールの Unattached フォルダに格納されているテストは、テス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。</p>

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。テストはテスト計画モジュールに、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・ファイルをテストに関連付けると、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)。
- ▶ テストおよびリソース・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、1461 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せん。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、1443 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、1433 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

テストに相対パスを使用して関連付けられているリソース・ファイルおよびアクションは、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、1433 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・リポジトリをテストに関連付ける際に、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 (1437 ページ「[依存関係] タブ」を参照してください)。

- ▶ QuickTest の [アクションのプロパティ] ダイアログ・ボックスの [使用者] タブ (457 ページ「このアクションを使用するテストとアクションのリストを表示」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 (1433 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 (1472 ページ「ベースラインの履歴の表示」を参照してください)。
- ▶ ベースラインからテストを開くか実行するときに、関連付けられている外部アクションまたはリソース・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、テス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。1472 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テストを表示する場合、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 (1444 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・ファイルをテストに関連付けると、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) および QuickTest ([アクションのプロパティ] ダイアログ・ボックス) で表示できます。

- ▶ **アセットのリンクが保たれます。** テストを移動するか、テストまたはアクションの名前を変更すると、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なります。

- ▶ **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テストを一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、テス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、1461 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（再利用可能なアクションまたは関連付けられたリソース・ファイル）を削除すると、アセットがほかのテストで使用されている（または現在のテストで複数回使用されている）場合、警告メッセージが表示されます。メッセージには、必要に応じてテスト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、またはこのアクションへの呼び出しが含まれているテストを一覧表示する [詳細] セクションが含まれています。これにより、テストおよびアクション呼び出しを管理し、不注意でテストが失敗しないようにできます。

- ▶ **アクションの呼び出しを含むテストを確認できます。** アクションにフォーカスを当て、[アクションのプロパティ] ダイアログ・ボックスの [使用者] タブを開いて ([テストフロー] 表示枠でアクションを右クリックし、[アクションのプロパティ] を選択)、特定のアクションの呼び出しを含むテストの一覧を表示できます。詳細については、457 ページ「このアクションを使用するテストとアクションのリストを表示」を参照してください。

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 テストはテスト計画モジュールに保存されます。テス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。

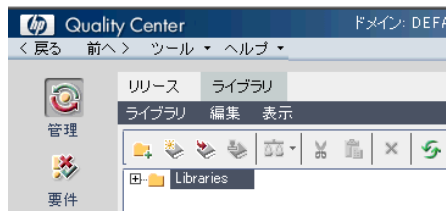
モジュールのは、本項では、QuickTest テストに関係するタブの概要を説明します。これらのタブの使用法の詳細については、『Mercury Quality Center ユーザーズ・ガイド』の関連する項を参照してください。

[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、1472 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。



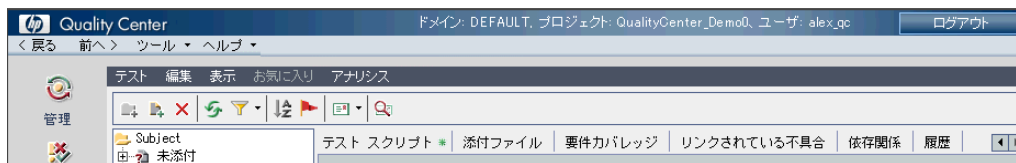
〔履歴〕 タブ

〔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。

- ▶ 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、1470 ページ「アセットのバージョンの履歴の表示」および 1472 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、テス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。

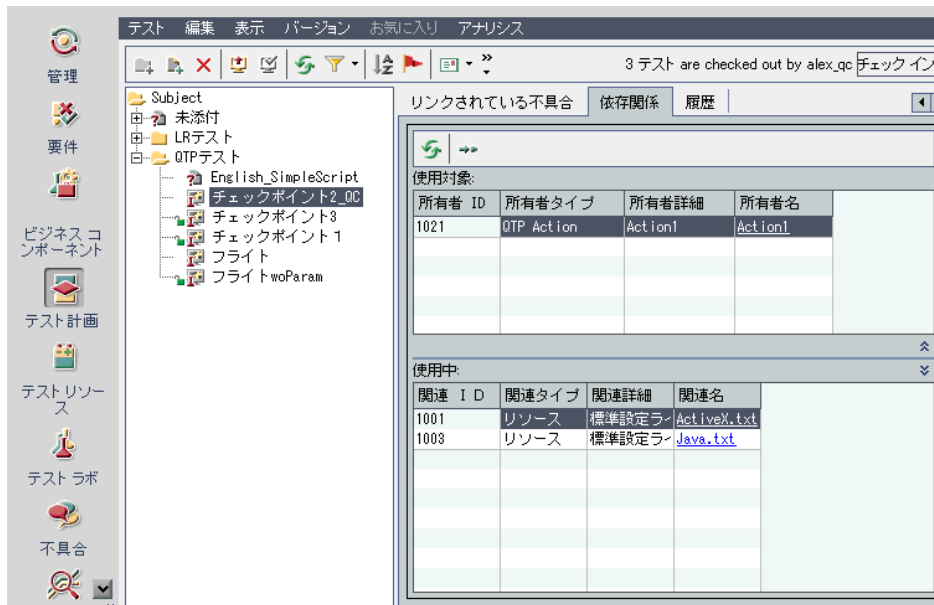
たとえば、共有オブジェクト・リポジトリのオブジェクトを変更するとします。テスト・リソース・モジュール内の共有オブジェクト・リポジトリに移動して、オブジェクトが関連付けられているテストの一覧を表示できます。これにより、このリソース・ファイルを使用しているアセットを特定したり、予定されている変更が依存アセットに与える可能性がある影響を分析したりできます。

Quality Center では、テスト計画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。

- ▶ テスト計画モジュール
- ▶ テスト・リソース・モジュール

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テスト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]	<p>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。</p> <p>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テストに移動できます（リンクからテスト計画モジュールに移動します）。</p>
[所有者タイプ]	<p>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。</p> <ul style="list-style-type: none"> ▶ 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。 ▶ テスト：テスト計画モジュール内の QuickTest テスト。 ▶ QTP アクション：テスト計画モジュール内のテストの一部であるアクション。

カラム	説明
[所有者名]	<p>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。</p> <p>QuickTest 関連の所有者名には次のものがあります。</p> <ul style="list-style-type: none"> ▶ Main Test Flow : テスト計画モジュールで現在選択されているテストの最上位アクションから呼び出されるテスト・コンテナを示す。Main Test Flow が表示される場合、[所有者タイプ] はテストです。 ▶ アクション <#> : テスト計画モジュールで現在選択されているテストのアクションから呼び出されるアクションの内部名を示す。アクション <#> は、アクションが作成されたときの順番番号を参照します。[所有者タイプ] が QTP アクション の場合、アクション <#> が表示されます。注 : テスト内でアクションの名前が変更されても、アクション <#> が表示されます。 ▶ アセットがテストではない場合はアセットの実際の名前。
[所有者詳細]	<p>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。</p> <ul style="list-style-type: none"> ▶ [所有者タイプ] が QTP アクション の場合、QuickTest に表示されているアクションの名前と、(存在する場合は) 詳細が表示されます。 ▶ [所有者タイプ] が テスト の場合、このセルは空白です。

ヒント : QuickTest では、アクションの [使用者] 情報は [アクションのプロパティ] ダイアログ・ボックスで表示できます。詳細については、457 ページ「このアクションを使用するテストとアクションのリストを表示」を参照してください。

また QuickTest では、依存アセットを削除しようとするとき [使用者] 情報が表示され、アセットを削除する前に変更が関連アセットに与える影響を確認できるようにします。

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、テストを見ているとします。テストで呼び出されるすべての外部アクション、テス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、テス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	<p>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。</p> <ul style="list-style-type: none"> ▶ 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。 ▶ テスト：テスト計画モジュール内の QuickTest テスト。 ▶ QTP アクション：テスト計画モジュール内のテストの一部であるアクション。

カラム	説明
<p>関連名</p>	<p>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。</p> <p>QuickTest 関連の名前には次のものがあります。</p> <ul style="list-style-type: none"> ▶ アクション <#> : テスト計画モジュールのテストのアクションから呼び出されるアクションの内部名を示す。アクション <#> は、アクションが作成されたときの順番番号を参照します。[関連タイプ] が QTP アクション の場合、アクション <#> が表示されます。 注 : テスト内でアクションの名前が変更されても、アクション <#> が表示されます。 ▶ アセットがテストではない場合はアセットの実際の名前。
<p>関連詳細</p>	<p>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。</p> <p>[関連タイプ] が QTP アクション の場合、QuickTest に表示されているアクションの名前と、(存在する場合は) 詳細が表示されます。</p>

第 53 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 (1444 ページ)
- ▶ QuickTest アセット比較ツール (1446 ページ)
- ▶ QuickTest アセット・ビューア (1456 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 (291 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、データ・テーブルやローカル・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。テス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、テス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。テス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）。

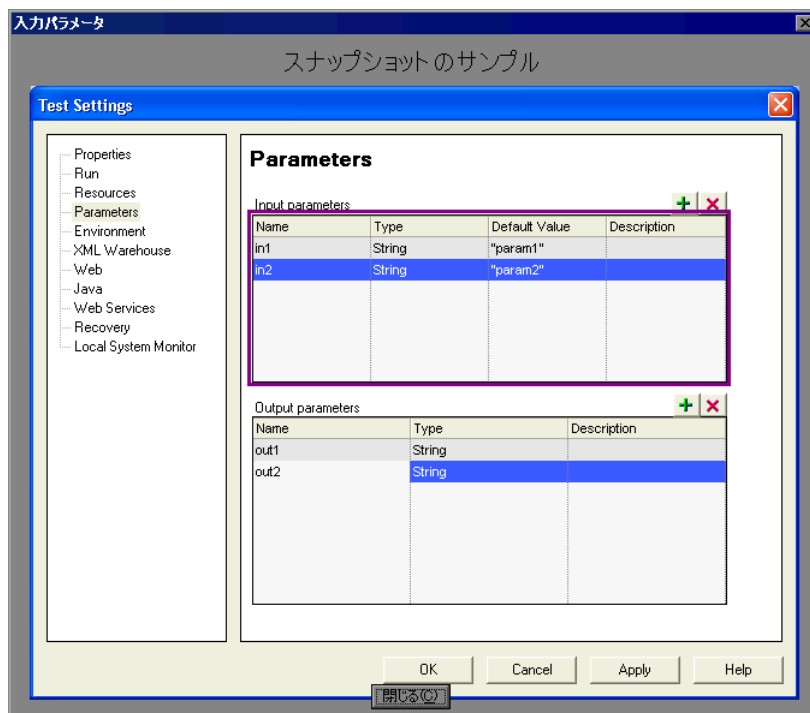
ドリルダウンを行うには、次のいずれかを行います。

- ▶ 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、1454 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、テストの比較を表示し、[実行セッション中にスマート認識を無効にする] 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 [テストの設定] ダイアログ・ボックスの [実行] 表示枠にあることが示されます。ダイアログ・ボックスのタイトル・バーに選択した要素が一覧表示され、オプションの周りに紫色の四角形が表示されます。



詳細については、1454 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、1455 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。

テストの設定 [4] [1] [7]

詳細については、1446 ページ「QuickTest アセット比較ツール」と 1456 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、テスト、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、関連データ・テーブルや共有オブジェクト・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、1432 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。またはコマンド・ライン・インタープリタ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テス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、222 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、1322 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 [テスト計画] ボタンをクリックし、テスト計画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、データ・テーブル、回復シナリオなど、テス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] 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、1462 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。

コマンド・ライン・インタプリタ (cmd.exe)

- 1 コマンド・ライン・インタプリタを開きます。
- 2 次の構文を使用してコマンドを入力します。

"<アセット比較ツール実行可能ファイルのパス>" P1: "<ファイル・パス 1>" P2: "<ファイル・パス 2>"

ここで **P1** は最初のアセットへのファイル・システム・パス、**P2** は 2 番目のアセットへのファイル・システム・パスです。

注：それぞれの引数の後ろにスペースが挿入されていることを確認します。このオプションは大文字と小文字を区別しません。また任意の順番で入力できます。

例：

```
"C:¥Program Files¥HP¥QuickTest Professional¥Bin¥QTPDiffApplication.exe" P1: "C:¥Program Files¥HP¥QuickTest Professional¥Tests¥Test1" P2: "C:¥Program Files¥HP¥QuickTest Professional¥Tests¥Test2"
```

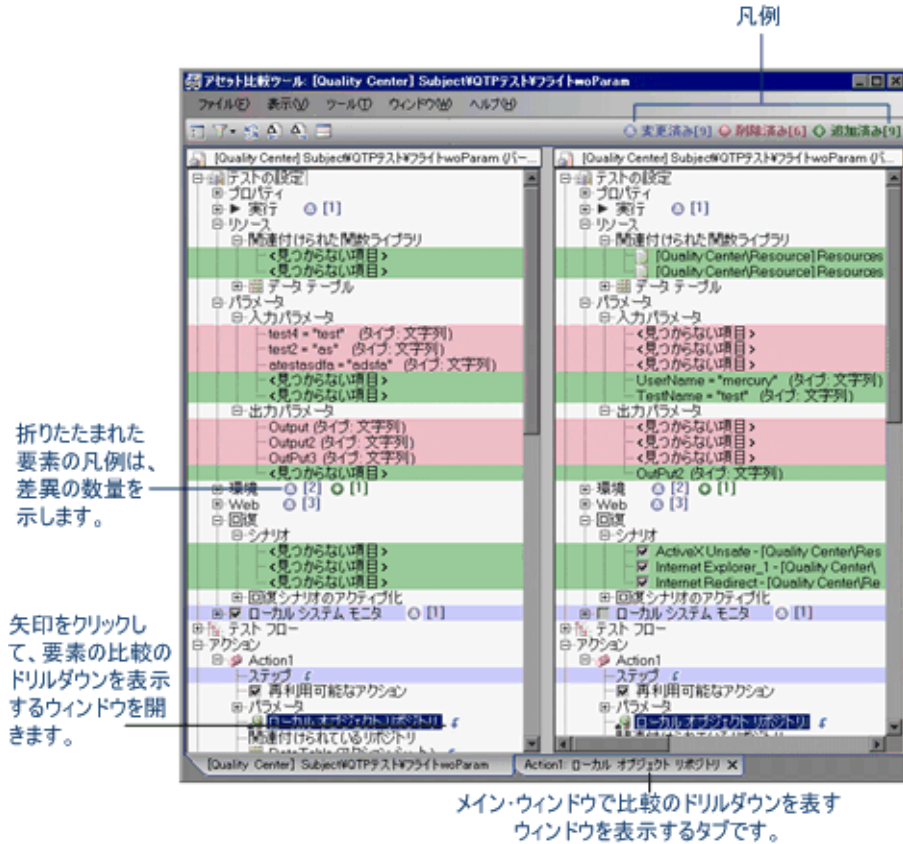
アセット比較ツールのコマンドとオプションについて

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、1444 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、1446 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、1456 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。
	[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。

	コマンド	ショートカット・キー	説明
	[表示] > [更新]		<p>選択したアセットのバージョンを新たに比較します。</p> <p>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。</p>
	[ツール] > [色設定]		<p>[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。</p> <p>1455 ページ「[色設定] ダイアログ・ボックス」を参照してください。</p>
	[ツール] > [フィルタ]		<p>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。</p> <ul style="list-style-type: none"> ▶ 変更済み  ▶ 削除済み  ▶ 追加済み  ▶ 同じ <p>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。</p> <p>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。</p>




	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]		<p>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。</p> <p>注：別のウィンドウを開いて、共有オブジェクト・リポジトリまたはデータ・テーブルなど、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</p> <p>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。</p>
	[ウィンドウ] > [水平に表示] または [ウィンドウ] > [垂直に表示]		<p>[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。</p> <p>[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。</p>
	[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。

 変更済み[2]  削除済み[1]  追加済み[5]


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	<p>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）。</p> <p>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。</p> <p>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。1432 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。</p>
[スナップショットのサンプルの表示]	CTRL+Q	<p>[テストの設定] 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。</p>


[色設定] ダイアログ・ボックス

説明	<p>[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。</p> <p>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。</p>
アクセス方法	<p>[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。</p> <ul style="list-style-type: none"> ▶ [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]	<p>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。</p> <ul style="list-style-type: none"> ▶ 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。
[戻る]	<p>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。</p>

QuickTest アセット・ビューア

QuickTest アセット・ビューアにより、テスト、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、関連データ・テーブルや共有オブジェクト・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テス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、222 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、1322 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 [テスト計画] ボタンをクリックし、テスト計画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、データ・テーブル、回復シナリオなど、テス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。

コマンド・ライン・インタプリタ (cmd.exe)

注：アセット比較ツールの実行可能ファイルのパスを使用して、アセット・ビューアを開きます。

- 1 コマンド・ライン・インタプリタを開きます。
- 2 次の構文を使用してコマンドを入力します。

"<アセット比較ツール実行可能ファイルのパス>" P1: "<ファイル・パス 1>"

ここで **P1** は、最初のアセットへのファイル・システム・パスです。

注：それぞれの引数の後ろにスペースが挿入されていることを確認します。このオプションは大文字と小文字を区別しません。また任意の順番で入力できます。

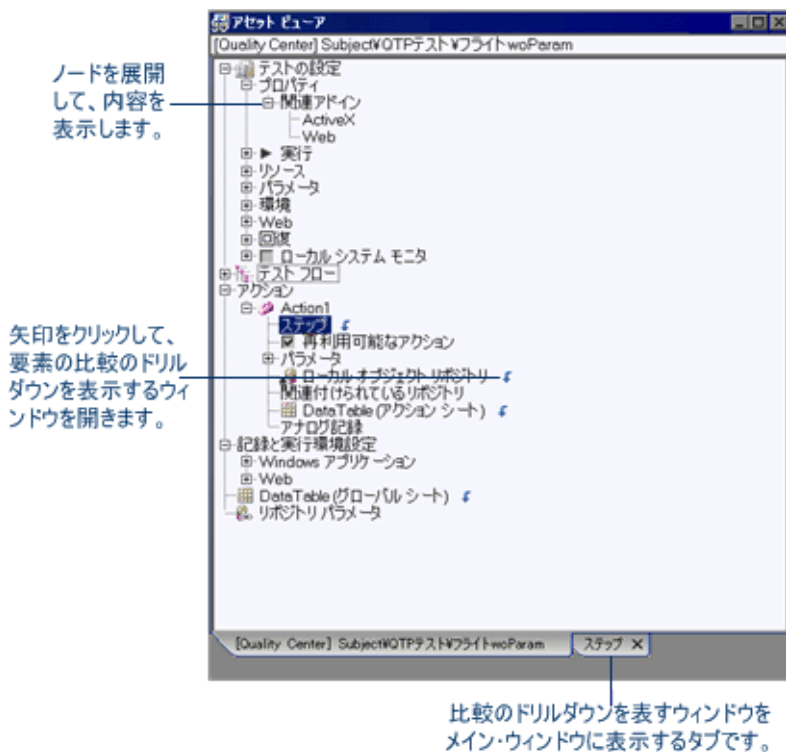
例：

```
"C:¥Program Files¥HP¥QuickTest Professional¥Bin¥QTPDiffApplication.exe" P1: "C:¥Program Files¥HP¥QuickTest Professional¥Tests¥Test1"
```

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	<p>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）。</p> <p>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。</p> <p>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。1432 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。</p>
[スナップショットのサンプルの表示]	CTRL+Q	[テストの設定] 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 54 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 に適用されません。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 (1462 ページ)
- ▶ アセットのバージョンの履歴の表示 (1470 ページ)
- ▶ ベースラインの履歴の表示 (1472 ページ)
- ▶ バージョンの履歴とベースラインの履歴 (1476 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、テスト、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	<ul style="list-style-type: none"> ▶ [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	<p>[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。</p>
ほかの Quality Center ユーザへチェックアウト	<p>[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。</p>

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からです。

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テス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テス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、テス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る場合は、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、1443 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、1472 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、1465 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、1469 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、1465 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、1462 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、1472 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、1468 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、1468 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、1469 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、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、1470 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
テス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] > [テスト] 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。

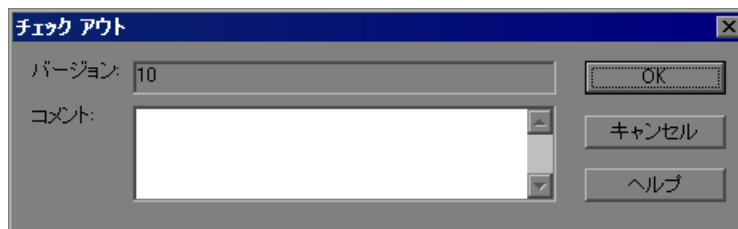
[Open < Asset type >] ダイアログ・ボックスが開きます。

- 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、1468 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、1469 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、いくつかの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、1469 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。

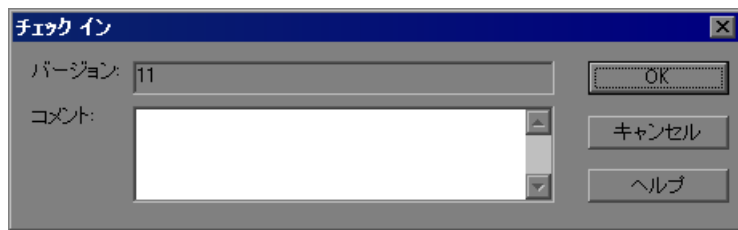
アセットをバージョン・コントロール・データベースにチェックインして戻すと、Quality Center はチェックアウト・ディレクトリからアセットのコピーを削除し、Quality Center プロジェクトのほかのユーザがそのバージョンのアセットを利用できるように、データベース内のアセットのロックを解除します。

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、1470 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。

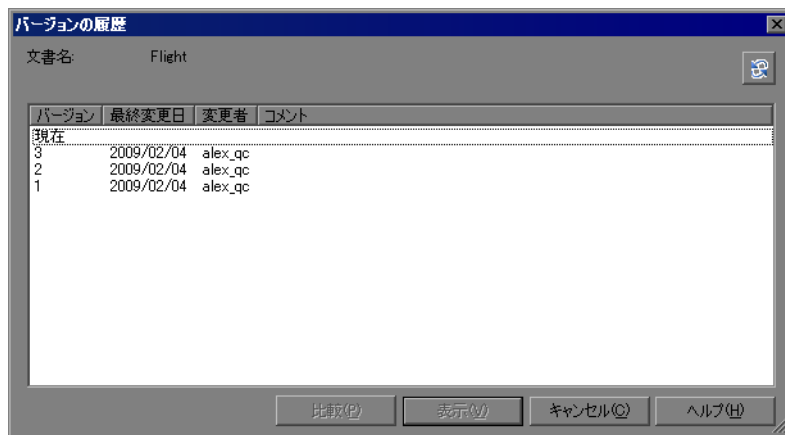
アセットのバージョンの履歴の表示

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

<p>説明</p>	<p>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。</p> <p>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。</p> <p>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。</p>
<p>アクセス方法</p>	<ul style="list-style-type: none"> ▶ 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。
<p>詳細</p>	<p>概念の概要：1462 ページ「Quality Center でのアセットのバージョン管理」</p> <p>関連ユーザ・インタフェースのトピック：1473 ページ「[ベースラインの履歴] ダイアログ・ボックス」</p>

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	<p>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。</p> <p>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、1446 ページ「QuickTest アセット比較ツール」を参照してください。</p>
	[表示] ボタン	<p>現在のアセットの選択したバージョンを表示できます。</p> <p>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、1456 ページ「QuickTest アセット・ビューア」を参照してください。</p>

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。

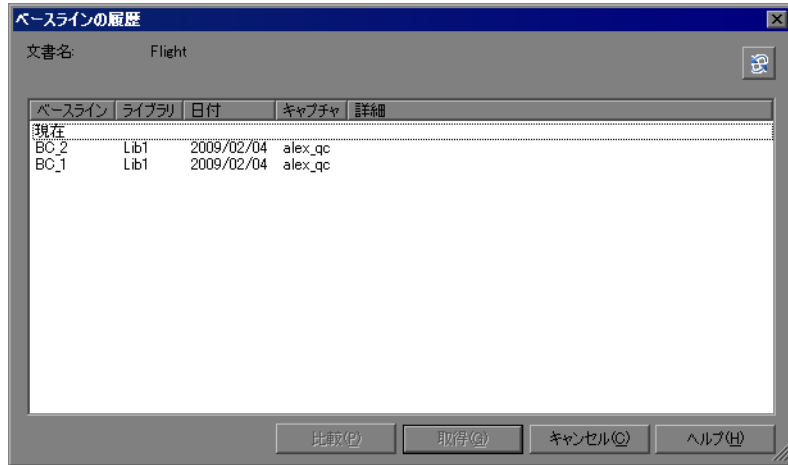
QuickTest では、ベースライン内に保存されたアセットを表示、比較できます。これにより、プロジェクトの全工程の中で、特定の段階のアセットの内容を確認できます。

ベースラインからテス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	<ul style="list-style-type: none"> ▶ 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	<p>概念の概要：1472 ページ「ベースラインの履歴の表示」</p> <p>関連ユーザ・インタフェースのトピック：1470 ページ「[バージョンの履歴] ダイアログ・ボックス」</p>

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	<p>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。</p> <p>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、1446 ページ「QuickTest アセット比較ツール」を参照してください。</p>
	[取得] ボタン	<p>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。</p> <p>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。</p> <p>[取得] をクリックすると、QuickTest により次の操作が行われます。</p> <ul style="list-style-type: none"> ▶ 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外部アクションおよびリソース・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：外部アクションまたはリソース・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アクションまたはリソース・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 55 章

Quality Center 9.x でのバージョン・コントロールを使った作業

本章では、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つ HP Quality Center の使用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 9.x 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 9.x バージ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

- ▶ バージョン・コントロール・サポート付きの Quality Center 9.x プロジェクトからテストを開く (1478 ページ)
- ▶ QuickTest でのテストのバージョン管理 (1478 ページ)

バージョン・コントロール・サポート付きの Quality Center 9.x プロジェクトからテストを開く

[開く] ツールバー・ボタンをクリックするか、[ファイル] > [開く] > [テスト] を選択してバージョン・コントロールがサポートされている Quality Center プロジェクトからテストを開くと、[テストを開く] ダイアログ・ボックスには、選択したサブジェクトの各テストのバージョン・コントロール・ステータスを示すアイコンが表示されます。

バージョン・コントロール・サポートの Quality Center プロジェクトからテストを開くと、テストは現在のバージョン・コントロール・ステータスに応じて、読み書きモードまたは読み取り専用モードで開きます。

- ▶ テストが現在バージョン・コントロール・データベースにチェックインされているか、別のユーザにチェックアウトされている場合、テストは読み取り専用モードで開きます。
- ▶ テストがユーザ自身にチェックアウトされている場合、テストは読み書きモードで開きます。

QuickTest でのテス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 9.x 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、自動化されたテスト・スクリプトの更新や変更ができます。これにより、各テストに加えた変更の追跡、テストのバージョン間の変更の確認、テストの旧バージョンの復元が容易になります。

バージョン・コントロール・サポート付きのプロジェクトにテストを保存すると、バージョン・コントロール・データベースにテストが追加されます。バージョン・コントロール・データベースにテストをチェックイン、チェックアウトすることで、テストのバージョンが管理されます。

最新バージョンのテストは、Quality Center のテスト・リポジトリにあり、Quality Center によってすべてのテストの実行に使用されます。

注：

- ▶ バージョン・コントロールがサポートされている Quality Center プロジェクトを利用するには、バージョン・コントロールのソフトウェアと Quality Center Version Control Add-in をインストールする必要があります。詳細については、Quality Center のドキュメントを参照してください。
 - ▶ [バージョン] メニュー・オプション群は、バージョン・コントロールがサポートされている Quality Center プロジェクトのデータベースに接続して Quality Center テストを開いている場合にだけ使用できます。
-

バージョン・コントロール・データベースへのテストの追加

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] を使って新規のテストを保存すると、QuickTest によって自動的にテス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 1.1.1 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。

QuickTest ステータス・バーにはそのつど各操作が表示されます。ただし、既存のテストに変更を保存しても、そのテストをチェックインすることにはなりません。テストを保存して閉じても、チェックインを選択するまではテストはチェックアウトしたままです。詳細については、1481 ページ「バージョン・コントロール・データベースへのテストのチェックイン」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テストのチェックアウト

[ファイル] > [開く] > [テスト] を選択して、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されているか別のユーザにチェックアウトされているテストを開くと、テストは読み取り専用モードで開きます。

チェックインしたテストを確認できます。また、テストを実行して結果を表示することもできます。

テストを変更するには、テストをチェックアウトする必要があります。テストをチェックアウトすると、テストは 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（初めてテス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、テストに加えた変更を 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。

テストを保存して閉じることはできますが、Quality Center データベースに戻されるまでテストはロックされています。テストを解放するには、テス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。テストのチェックインの詳細については、1481 ページ「バージョン・コントロール・データベースへのテストのチェックイン」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、1486 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。

標準では、チェックアウト・オプションはテストの最新バージョンにアクセスします。以前のバージョンのテストもチェックアウトできます。詳細については、1483 ページ「[バージョンの履歴] ダイアログ・ボックスの使用方法」を参照してください。

最新バージョンのテスト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テストを開きます。詳細については、1478 ページ「バージョン・コントロール・サポート付きの Quality Center 9.x プロジェクトからテストを開く」を参照してください。

注：開いたテス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テストを開くと、[**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テストを開くと、[**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テスト・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。読み取り専用のテストが閉じて、書き込み可能テストとして再び自動的に開きます。
- 5 必要に応じてテストを表示または編集します。

注：テス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、テス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。テストのチェックインの詳細については、1481 ページ「バージョン・コントロール・データベースへのテストのチェックイン」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、1486 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・データベースへのテストのチェックイン

あるテス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、テストのバージョン 1.2.3 をチェックアウトし、いくつか変更を加えてテストを保存するとします。バージョン 1.2.4 (または別の番号を割り当てる) としてテストをバージョン・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 1.2.3 を引き続き実行できます。

テストの変更が終了し、Quality Center ユーザが新しいバージョンを使用できるようになったら、バージョン・コントロール・データベースにテス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、1486 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。

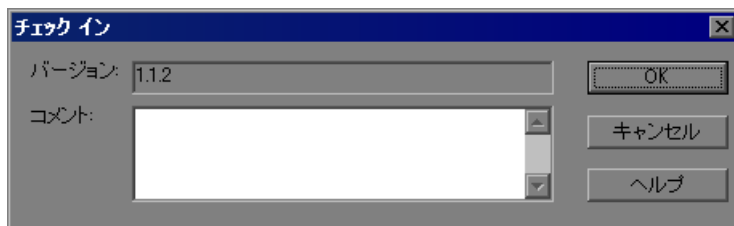
テストをバージョン・コントロール・データベースにチェックインして戻すと、Quality Center はチェックアウト・ディレクトリからテストのコピーを削除し、Quality Center プロジェクトのほかのユーザがそのバージョンのテストを利用できるように、データベース内のテストのロックを解除します。

現在開いているテストをチェックインするには、次の手順を実行します。

- 1 現在開いているテス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、1483 ページ「テストのバージョン情報の表示」を参照してください。

注：開いているテス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テストを開くと、**[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。



テス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、テストをチェックインします。テストが閉じて、読み取り専用のテストとして再び自動的に開きます。

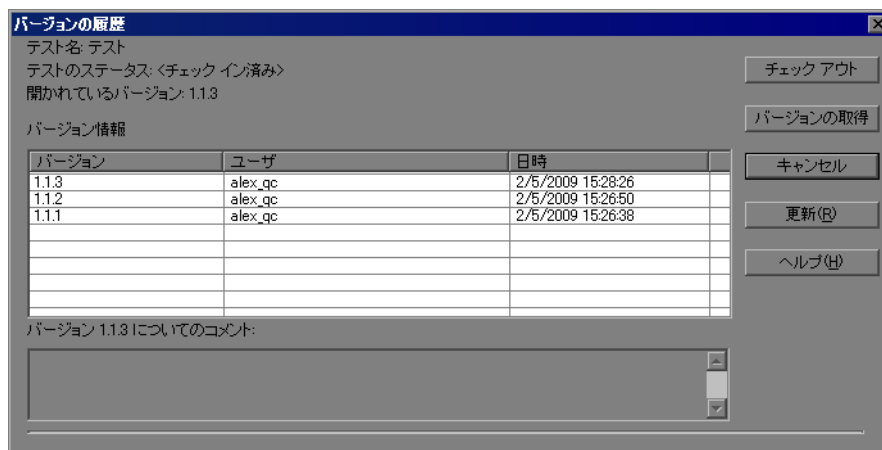
[バージョンの履歴] ダイアログ・ボックスの使用方法

[バージョンの履歴] ダイアログ・ボックスを使って、現在開いているテストに関するバージョン情報の表示や、以前のバージョンのテストの表示および取得が行えます。

テストのバージョン情報の表示

Quality Center バージョン・コントロール・データベースに格納されている任意の開いているテストのバージョン情報を、現在のステータスに関係なく表示できます。

テストの [バージョンの履歴] ダイアログ・ボックスを開くには、テストを開き [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] を選択します。



[バージョンの履歴] ダイアログ・ボックスには次の情報が表示されます。

[**テスト名**] : 現在開いているテストの名前です。

[**テストのステータス**] : テストのステータスです。次のいずれかになります。

- ▶ [**チェックイン**] : このテストは、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されています。テストは現在読み取り専用形式で開いています。テストをチェックアウトして編集できます。
- ▶ [**チェックアウト**] : このテストは、現在ユーザ自身がチェックアウトしています。テストは現在読み書き可能な形式で開いています。

- ▶ **[<ほかのユーザ>によるチェックアウト]**：このテストは、現在ほかのユーザによってチェックアウトされています。テストは現在読み取り専用形式で開いています。示されているユーザがテストをチェックインするまで、テストのチェックアウトや編集はできません。

[開いているテストのバージョン]：現在、お使いの QuickTest コンピュータで開いているテストのバージョンです。

[バージョン詳細]：テストのバージョンの詳細です。

- ▶ **[バージョン]**：テストのバージョンの一覧です。
- ▶ **[ユーザ]**：各バージョンのチェックインをしたユーザです。
- ▶ **[日付]**：各バージョンがチェックインされた日です。**[時刻]**：各バージョンがチェックインされた時刻です。

[コメント]：選択したテスト・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。

以前のバージョンのテストを対象とした作業

以前のバージョンのテストは、読み取り専用モードで表示するか、チェックアウトして最新バージョンとしてチェックインできます。

以前のバージョンのテストを表示するには、次の手順を実行します。

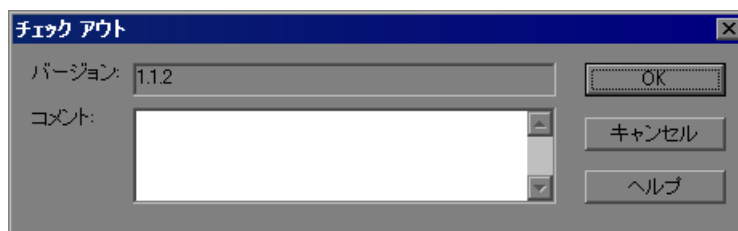
- 1 Quality Center テストを開きます。最新バージョンのテストが開きます。詳細については、1478 ページ「バージョン・コントロール・サポート付きの Quality Center 9.x プロジェクトからテストを開く」を参照してください。
- 2 **[ファイル]** > **[Quality Center バージョン コントロール]** > **[バージョンの履歴]** を選択します。**[バージョンの履歴]** ダイアログ・ボックスが表示されます。
- 3 表示するテストのバージョンを **[バージョン詳細]** リストから選択します。
- 4 **[バージョンの取得]** ボタンをクリックします。テストがチェックアウトされていないため読み取り専用モードで開く旨を知らせるメッセージが表示されます。
- 5 **[OK]** をクリックし、QuickTest メッセージを閉じます。選択したバージョンが読み取り専用モードで開きます。

ヒント：

- ▶ 現在 QuickTest で開いているバージョン番号を確認するには、[バージョンの履歴] ダイアログ・ボックスの [開いているテストのバージョン] の値を調べます。
- ▶ [バージョンの取得] オプションを使用して読み取り専用モードで以前のバージョンを開いたら、[ファイル] > [Quality Center バージョン コントロール] > [チェックアウト] を選択して、開いているテストをチェックアウトできます。これは、[バージョンの履歴] ダイアログ・ボックスで [チェックアウト] ボタンを使用するのと同等の機能です。

以前のバージョンのテストをチェックアウトするには、次の手順を実行します。

- 1 Quality Center テストを開きます。最新バージョンのテストが開きます。詳細については、1478 ページ「バージョン・コントロール・サポート付きの Quality Center 9.x プロジェクトからテストを開く」を参照してください。
- 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 表示するテストのバージョンを [バージョン詳細] リストから選択します。
- 4 [チェックアウト] ボタンをクリックします。確認メッセージが開きます。
- 5 以前のバージョンのテストをチェックアウトすることを確認します。[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テスト・バージョンが表示されます。



- 6 [コメント] ボックスに変更の詳細を入力します。

- 7 [OK] をクリックします。開いていたテストが閉じ、選択したバージョンが書き込み可能テストとして開きます。
- 8 必要に応じてテストを表示または編集します。
- 9 最新バージョンとしてテストを Quality Center データベースにチェックインするには、[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。変更したテストを Quality Center にアップロードしない場合は、[ファイル] > [Quality Center バージョン コントロール] > [チェックアウト作業を元に戻す] を選択します。

テストのチェックインの詳細については、1481 ページ「バージョン・コントロール・データベースへのテストのチェックイン」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、1486 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。

チェックアウト操作の取り消し

テストをチェックアウトした後に、変更を加えたテストを 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テス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] > [チェックアウト作業を元に戻す] 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テストが閉じ、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。

第 XII 部

その他の HP 製品を使用した作業

第 56 章

Business Process Testing を使用した作業

Business Process Testing のサポートが組み込まれた Quality Center プロジェクトに接続すると、QuickTest で Quality Center ビジネス・プロセス・テストで使用されるコンポーネントのステップの作成や実装を行えます。

本章の内容

- ▶ Business Process Testing での作業について（1489 ページ）
- ▶ Business Process Testing での役割について（1490 ページ）
- ▶ Business Process Testing のテスト方法について（1494 ページ）

Business Process Testing での作業について

Business Process Testing では、各分野のエキスパートがテストの実施および自動テスト環境の向上を目的として、キーワード駆動方式のテストを作成できます。

Business Process Testing は QuickTest を Quality Center と組み合わせて使用し、専用の Business Process Testing ライセンスを購入することによって有効にできます。QuickTest 内から Business Process Testing を使って作業するためには、Business Process Testing のサポートが組み込まれた Quality Center プロジェクトに接続する必要があります。

本項では Business Process Testing モデルの概要を説明します。詳細については、『**HP Business Process Testing ユーザーズ・ガイド**』および『**HP QuickTest Professional for Business Process Testing ユーザーズ・ガイド**』を参照してください。

Business Process Testing での役割について

Business Process Testing モデルはロールベースのモデルで、(Quality Center で作業する) 技術者でない各分野のエキスパートが、(QuickTest Professional で作業する) オートメーション・エンジニアと効率的に協力できるようにします。各分野のエキスパートは、ビジネス・プロセス、ビジネス・コンポーネント、およびビジネス・プロセス・テストを定義しドキュメント化します。一方、自動化エンジニアは、共有オブジェクト・リポジトリ、関数ライブラリ、および回復シナリオなどの必要なリソースや設定を定義します。両者が連携することで、各分野のエキスパートにプログラミングの知識がなくても、ビジネス・プロセス・テストを作成し、データ駆動し、ドキュメント化し、実行できます。

注：役割の構造と、組織における役割ごとに実行されるタスクは、組織で採用されている方法論によって、ここに記述されているものとは異なる場合があります。これらの役割は柔軟性があり、Business Process Testing を使用する人の能力と時間に応じて役割は異なります。たとえば、各分野のエキスパートのタスクおよびオートメーション・エンジニアのタスクを同じ人が実行する場合があります。特定の組織でどのようなロール（役割）を定義する必要があるか、またどの種類のユーザが Business Process Testing のどの作業（タスク）を実行できるか（ユーザが正しい権限を持っていることが前提となります）について、製品固有の規則や制限はありません。

Business Process Testing モデルには、次のユーザの役割の分類があります。

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。これにより、各分野のエキスパートはテストが必要な運用シナリオやビジネス・プロセスを決定したり、複数のビジネス・プロセスに共通の重要なビジネス・アクティビティを特定したりできます。

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・モジュールを使用して、アプリケーションで実行される特定のタスクと、こうしたタスクの前後のアプリケーションの条件または状態を記述するビジネス・コンポーネントを作成します。そして、ビジネス・プロセスを構成する各ビジネス・コンポーネントの個々のステップを、手動ステップまたは非自動ステップの形式で定義します。

デザイン段階で、各分野のエキスパートは、オートメーション・エンジニアと作業を行い、コンポーネントの自動化に必要なリソースおよび設定を特定し、オートメーション・エンジニアがそれらを準備できるようにします。リソースと設定が準備できたら、各分野のエキスパートがそれらをキーワード駆動型のコンポーネントに変換して、手動のステップを自動化します。このプロセスでは、各コンポーネントのアプリケーション領域を選択する必要があります。アプリケーション領域には、必要なすべてのリソース・ファイルおよび設定が含まれます。これらは、テスト対象アプリケーションの特定の領域に固有のリソース・ファイルおよび設定です。各コンポーネントをアプリケーション領域に関連付けることで、コンポーネントはこれらのリソースおよび設定にアクセスできるようになります。

各分野のエキスパートは Quality Center のテスト計画モジュールを使用してビジネス・コンポーネントを組み合わせて、連続するコンポーネントのフローで構成されるビジネス・プロセス・テストを作成します。たとえば、ほとんどのアプリケーションでは、ユーザはアプリケーションの何らかの機能にアクセスする前にログインする必要があります。各分野のエキスパートは、ログインの手順を行う 1 つのビジネス・コンポーネントを作成できます。このコンポーネント・プロシージャは、さまざまなビジネス・プロセス・テストで使用できます。その結果、保守、更新、テスト管理がより容易かつコスト効率の高いものになります。

各分野のエキスパートは、ビジネス・プロセス・テストに使用される値を設定し、それらをテスト・セットで実行して、結果を検討します。また、各ビジネス・コンポーネントのテスト・ステップの保守も担当します。

コンポーネントを定義する間、各分野のエキスパートはオートメーション・エンジニアと協力を続けます。たとえば、コンポーネントに新しい操作（機能）を要求したり、コンポーネントに対して計画されている今後の変更を検討したりすることが考えられます。

オートメーション・エンジニア：オートメーション・エンジニアは、QuickTest Professional のような自動テスト・ツールの使用に関するエキスパートです。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連携して、さまざまなビジネス・プロセス・テストに必要なリソースを特定します。

その後、オートメーション・エンジニアは、個別のコンポーネントに関連付けられている機能をテストするために必要なリソースおよび設定を準備し、それらを Quality Center プロジェクト内のアプリケーション領域に格納します。Quality Center プロジェクトは、特定のアプリケーションを対象にしたビジネス

ス・プロセス・テストの作成および実行を担当する各分野のエキスパートが使用するプロジェクトと同じものです。

各アプリケーション領域は、コンポーネントに必要なすべてのリソースおよび設定を保存する 1 つのエンティティとしての役割を果たします。これは、アプリケーションの特定部分のテストに関連付けられているすべての要素の保守を集中して行う場所となります。アプリケーション領域には一般に、1 つ以上の共有オブジェクト・リポジトリ、コンポーネントに対して使用できるキーワードのリスト、自動化関数（操作）を含む関数ライブラリ、失敗したステップのための回復シナリオ、そのほかコンポーネントが正しく動作するために必要なほかのリソースおよび設定が含まれています。コンポーネントは、アプリケーション領域内のリソースおよび設定に結び付いています。したがって、アプリケーション領域に変更が加えられると、関連するすべてのコンポーネントが自動的に更新されます。

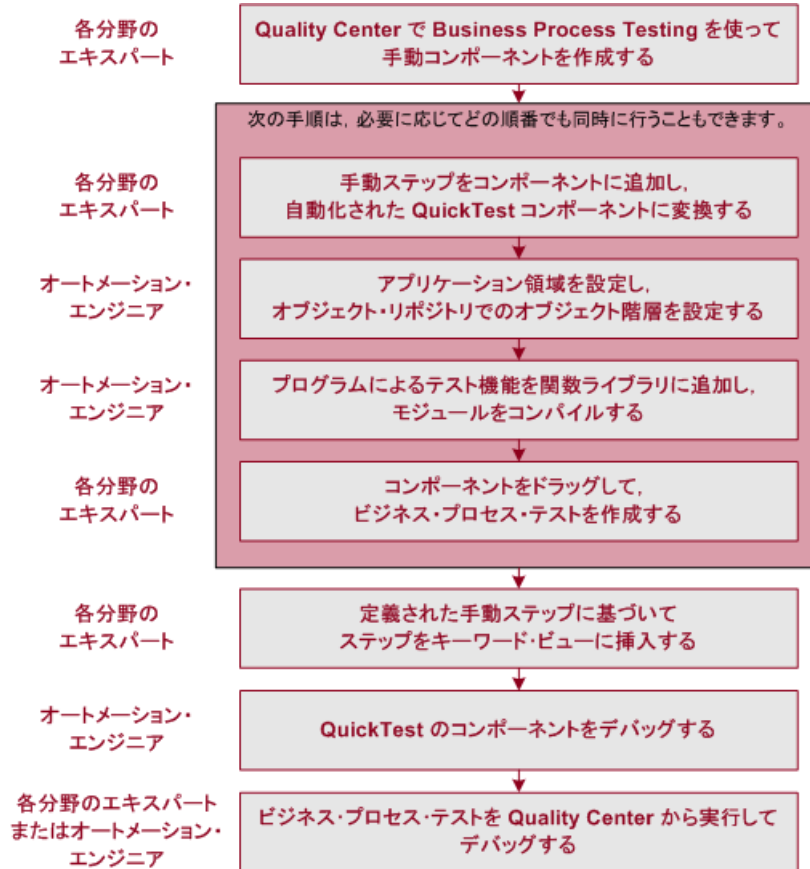
オートメーション・エンジニアは QuickTest の仕組みおよび機能を使用して、QuickTest からこれらのリソースを作成します。たとえば、QuickTest では、オートメーション・エンジニアは、アプリケーションの開発が完了する前でも、各種の共有オブジェクト・リポジトリを作成し、テスト対象アプリケーション内のさまざまなオブジェクトを表すテスト・オブジェクトを含めることができます。その後オートメーション・エンジニアは、必要に応じてリポジトリ・パラメータなどを追加できます。オートメーション・エンジニアはオブジェクト・リポジトリ・マネージャを使用してさまざまなオブジェクト・リポジトリを管理したり、Object Repository Merge Tool を使用してリポジトリを結合したりできます。また、オートメーション・エンジニアは、QuickTest を使用して、特定のタスクの実行に必要なステップをカプセル化するためにプログラミング・ロジックを使用する関数を含む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。

オートメーション・エンジニアによって作成されたリソースを使用して、各分野のエキスパートはコンポーネント・ステップを自動化し、コンポーネントおよびビジネス・プロセス・テストの作成と保守ができます。

自動化エンジニアは、必要に応じて QuickTest の中でコンポーネントを作成、デバッグ、および変更することもできます。

Business Process Testing のワークフローについて

次の例は QuickTest を使用した一般的な Business Process Testing のワークフローです。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。



Business Process Testing のテスト方法について

各分野のエキスパートが作成する各シナリオは、**ビジネス・プロセス・テスト**です。1つのビジネス・プロセス・テストは、連続する**コンポーネント**のフローで構成されています。各コンポーネントによって特定のタスクが実行されます。また、コンポーネントから後続のコンポーネントにデータが引き渡されます。

コンポーネントについて

コンポーネントは、保守が簡単で、特定のタスクを実行する再利用可能なスクリプトであり、効果的なビジネス・プロセス・テスト構造を作成できる基礎的な要素です。コンポーネントは、細分化されたビジネス・プロセスの一部です。たとえば、ほとんどのアプリケーションのユーザは、何かを行う前にログインする必要があります。各分野のエキスパートは、アプリケーションへログインする手続きを行う1つのコンポーネントを作成できます。各コンポーネントは別のビジネス・プロセスのテストで再利用できます。その結果、保守、更新、テスト管理がより容易になります。

コンポーネントはステップで構成されています。たとえば、ログイン・コンポーネントの最初のステップではアプリケーションを開きます。第2ステップはユーザ名の入力です。第3ステップはパスワードの入力、そして第4ステップは**[ログイン]** ボタンのクリックです。

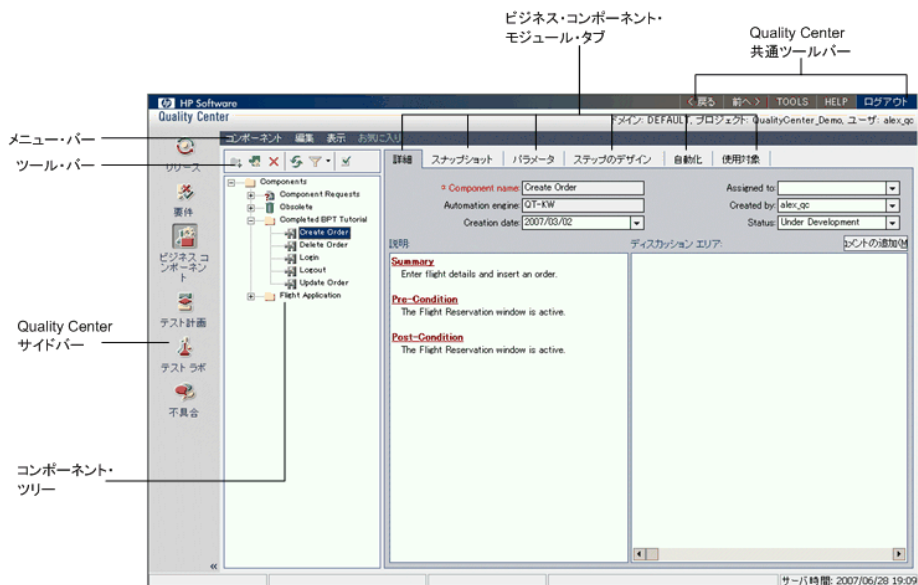
また、コンポーネントには、チェックポイント・ステップや出力値も追加できます。

- ▶ 「**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判断できます。コンポーネント・ステップでは、標準チェックポイントとビットマップ・チェックポイントを実行できます。詳細については、499 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**」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。詳細については、677 ページ「値の出力」を参照してください。

サポートされている任意の環境を対象としたステップの追加，特定の項目のパラメータ化，および特定のタスクの実行に必要なステップをカプセル化する関数（操作）の組み込みによるコンポーネントの拡張など，QuickTest の中でコンポーネントの作成および編集ができます。Quality Center では，各分野のエキスパートはコンポーネントを作成し，それらを組み合わせて，アプリケーションが想定どおりに動作することをチェックするために使用するビジネス・プロセス・テストを組み立てます。

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，定義できます。



ビジネス・コンポーネント・モジュールには、次のタブが含まれます。

- ▶ **[詳細]**：コンポーネントの目的またはゴールのサマリ、およびコンポーネントの実行前後のアプリケーションの状態（前提条件および後提条件）を指定します。現在選択されているビジネス・コンポーネントの詳細と実装要件を指定できます。
- ▶ **[スナップショット]**：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画像をキャプチャし、現在選択されているビジネス・コンポーネントに添付できます。
- ▶ **[パラメータ]**：ビジネス・コンポーネントの入力および出力のコンポーネント・パラメータと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]**：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]**：自動コンポーネントの表示またはアクセスができます。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しているアセットが一覧表示されます。
- ▶ **[履歴] タブ**：コンポーネントに加えられた変更のログが表示されます。

QuickTest Professional でのコンポーネントの実装

一般に、コンポーネントは各分野のエキスパートが Quality Center の中で作成しますが、QuickTest の中でも作成およびデバッグができます。

QuickTest 内では、ステップを手動で追加することでコンポーネントを作成します（オブジェクト・リポジトリにデータが設定されており、必要な操作が利用可能な場合）。また、サポートされているすべての環境のステップを記録することによって、コンポーネントを作成することもできます。特定の項目のパラメータ化が可能です。また、コンポーネント固有のオプションの表示や設定も行えます。

QuickTest では、**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**の 2 種類のコンポーネントの作成および変更が可能です。ビジネス・コンポーネントは、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。スクリプト・コンポーネントは、プログラム・ロジックを含むことのできる自動化されたコンポーネントです。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。

たとえば、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 の中でスクリプト・コンポーネントの作成、表示、変更、デバッグが行えます。スクリプト・コンポーネントは複雑なため、QuickTest 内でのみ編集可能です。

Quality Center では、各分野のエキスパートは QuickTest 内で作成されたコンポーネントを開けます。その後各分野のエキスパートはビジネス・コンポーネントを表示および編集できますが、スクリプト・コンポーネントについては詳細のみを表示できます。

Quality Center テスト計画モジュールでのビジネス・プロセスの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でビジネス・プロセスのテストまたはフローを作成します。ビジネス・プロセスのテストまたはフローを作成するには、各分野のエキスパートは該当のコンポーネントを選択（ドラッグ・アンド・ドロップ）し、それらの実行設定を設定します。

各コンポーネントは、異なるビジネス・プロセス・テストまたはフローで別々に使用できます。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。

ビジネス・プロセス・テストの実行と結果の分析

QuickTest で [実行] オプションや [デバッグ] オプションを使用して、個々のコンポーネントの実行やデバッグができます。

Quality Center のテスト計画モジュールからテストを実行して、ビジネス・プロセスのテストのデバッグが行えます。このモジュールから実行する場合、デバッグ・モードで実行するコンポーネントを選択できます。(この処理は、コンポーネントの開始時に実行を一時停止します。

ビジネス・プロセス・テストがデバッグされ、通常のテスト実行の準備が整ったら、各分野のエキスパートは、ほかのテストを Quality Center で実行する場合と同様の方法で、テストのラボ・モジュールからテストを実行します。テストを実行する前に、各分野のエキスパートは、テストのラボ・モジュールのグリッドの [反復] カラムを使用して、実行時のパラメータ値と反復を定義できます。

注：Quality Center からビジネス・プロセス・テストを実行する場合、テスト実行は QuickTest リモート・エージェントでの設定の影響も受ける可能性があります。QuickTest リモート・エージェントの詳細については、1423 ページ「QuickTest リモート・エージェントの設定」を参照してください。

テストのラボ・モジュールから、ビジネス・プロセス・テスト全体の実行結果を確認できます。結果には、各パラメータの値や、QuickTest によって報告された個々のステップの結果が含まれています。

[**レポートの表示**] のリンクをクリックして、QuickTest レポート全体を開くことができます。この階層化されたレポートには、ビジネス・プロセス・テスト実行での、それぞれの反復やコンポーネントがすべて含まれています。

コンポーネントとテストの違いについて

すでに QuickTest を使用したアクション・ベースのテストの作成に慣れている場合は、その手順とコンポーネントの作成や編集の手順が非常に似ていることに気付くでしょう。ただし、コンポーネント・モデルの設計や目的により、コンポーネントの作成、編集、実行の点で多少の違いがあります。次のガイドラインでは、これらの違いの概要を説明します。

- ▶ コンポーネントは単独のエンティティです。複数のアクションを含めたり、別のアクションや別のコンポーネントを呼び出すことはできません。
- ▶ コンポーネントを使って作業をすると、外部リソース・ファイルはすべて現在接続している 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ます。
- ▶ キーワード・ビューでのコンポーネントのノードの名前は、保存されているコンポーネントと同じです。ノード名は変更できません。
- ▶ ビジネス・コンポーネントは、エキスパート・ビューではなくキーワード・ビューで作成します。
- ▶ コンポーネントに直接ではなく、コンポーネントのアプリケーション領域を介してリソースを追加します。

第 57 章

WinRunner を使用した作業

QuickTest では、WinRunner のテストを実行したり、コンパイル済みモジュール内の TSL 関数またはユーザ定義関数を呼び出したりすることもできます。

本章の内容

- ▶ WinRunner を使用した作業について（1501 ページ）
- ▶ WinRunner テストの呼び出し（1502 ページ）
- ▶ WinRunner 関数の呼び出し（1505 ページ）

WinRunner を使用した作業について

コンピュータに WinRunner がインストールされている場合には、QuickTest のテストに WinRunner のテストまたは関数への呼び出しを含めることができます。

WinRunner テストまたは関数への呼び出しを作成したら、呼び出しステートメントをエキスパート・ビューまたはキーワード・ビューで編集して、そのステートメントの引数値を変更できます。

QuickTest を、WinRunner のテストまたはコンパイル済みモジュールが含まれている Quality Center プロジェクトに接続すると、Quality Center プロジェクトに格納されている WinRunner テストまたは関数を呼び出すことができます。

WinRunner テストの呼び出し

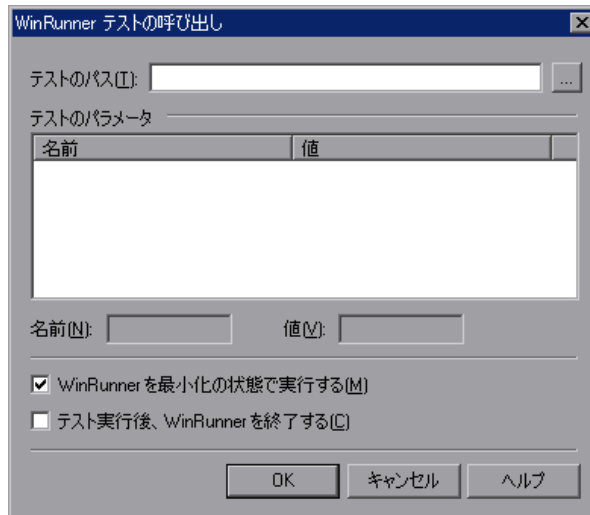
テストを実行するために、QuickTest から WinRunner に接続すると、WinRunner が起動されテストが開かれ、実行されます。WinRunner テスト実行に関する情報は、QuickTest の [テスト結果] ウィンドウに表示されます。

WinRunner テストへの呼び出しを挿入するには、[WinRunner テストの呼び出し] ダイアログ・ボックスを使用するか、エキスパート・ビューの中で **TSLTest.RunTestEx** ステートメントを入力します。

注： QuickTest テストへの呼び出しを含む WinRunner テストを呼び出すことはできません。

[WinRunner テストの呼び出し] ダイアログ・ボックスを使用して WinRunner テストへの呼び出しを挿入するには、次の手順を実行します。

- 1 [挿入] > [WinRunner の呼び出し] > [テスト] を選択します。[WinRunner テストの呼び出し] ダイアログ・ボックスが表示されます。



- 2 [テストのパス] ボックスに WinRunner テストのパスを入力するか、参照ボタンを使って WinRunner テストを指定します。

- 3 [テストのパラメータ] ボックスには、WinRunner テストに必要なすべてのテスト・パラメータのリストが表示されます。パラメータの値を入力するには、次の手順を実行します。
- ▶ **[テストのパラメータ]** リスト内のパラメータを選択して強調表示にします。選択した値がリストの下の **[名前]** ボックスに表示されます。
 - ▶ **[値]** ボックスに新しい値を入力します。

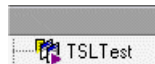
注： QuickTest ランダム・パラメータまたは環境変数パラメータのパラメータ値や、QuickTest データ・テーブルのパラメータ値を、WinRunner テストのパラメータとして使用することもできます。それにはパラメータの情報を TSLTest.RunTestEx ステートメントに手作業で入力しても行えます。詳細については、1504 ページ「WinRunner テストへの QuickTest でパラメータ化された値の引き渡し」を参照してください。

- 4 WinRunner ウィンドウがテストの実行中に表示されないようにするには、**[WinRunner を最小化の状態で行う]** を選択します
- 5 WinRunner テストを呼び出したステップが完了したときに WinRunner アプリケーションを終了させるには、**[テスト終了後、WinRunner を実行する]** を選択します
- 6 **[OK]** をクリックして、ダイアログ・ボックスを閉じます。

WinRunner のテスト・パラメータの詳細については、『**HP WinRunner ユーザーズ・ガイド**』を参照してください。

QuickTest では、WinRunner テストへの呼び出しが次のように表示されます。

- ▶ キーワード・ビューには、WinRunner の **RunTestEx** ステップが表示されます。次に例を示します。

	操作	値
 TSLTest	RunTestEx	"C:¥WinRunner¥Tests¥basic flight",True,0,MyValue

- ▶ エキスパート・ビューには、VBScript で書かれた TSLTest.RunTestEx ステートメントが表示されます。次に例を示します。

TSLTest.RunTestEx "C:¥WinRunner¥Tests¥basic_flight",TRUE, 0, "MyValue"

RunTestEx メソッドの構文は、次のとおりです。

TSLTest.RunTestEx テストのパス, 最小化実行, アプリケーション終了 [, パラメータ]

注 : QuickTest 6.0 で作成したテストは, RunTest メソッドを使用して WinRunner テストを呼び出している場合があります。このメソッドは, 構文が少し異なります。このメソッドでもテストは正常に実行されます。ただし, RunTestEx メソッド (および対応する引数の構文) を使用するようにテストを更新することをお勧めします。これら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。

テストの実行後, 結果を表示できます。詳細については, 1505 ページ「結果の表示」を参照してください。

RunTestEx メソッドの詳細とその使用例については, 『HP QuickTest Professional Object Model Reference』 (英語版) を参照してください。

WinRunner テストへの QuickTest でパラメータ化された値の引き渡し

WinRunner テストに要求されるパラメータに固定の値を設定する代わりに, QuickTest データ・テーブル, ランダム・パラメータまたは環境パラメータに定義されているパラメータ値を WinRunner に引き渡せます。これらのパラメータ値を指定するには, TSLTest.RunTestEx ステートメントのパラメータ引数として, 適切なステートメントを入力します。

たとえば, Windows ベースのフライト予約アプリケーションを対象に WinRunner テストを実行するとき, そのテストにフライトの乗客数と座席のクラスをパラメータ化するステートメントが組み込まれていたとします。その場合, 最初のパラメータの値を QuickTest ランダム・パラメータ (1 から 100 までの数をランダムに生成) から, また座席のクラスの値を QuickTest データ・テーブルの Class というカラムから WinRunner テストに渡します。その場合, QuickTest の TSLTest.RunTestEx ステートメントは, たとえば次のようになります。

```
TSLTest.RunTestEx "D:¥test1", TRUE, FALSE, RandomNumber(1, 100),  
DataTable("Class", dtGlobalSheet)
```

RandomNumber メソッド、Environment メソッド、および DataTable メソッドの構文および使用方法の詳細については、『**HP QuickTest Professional Object Model Reference**』（英語版）の「Utility」のセクションを参照してください。

結果の表示

WinRunner テストへの呼び出しを実行すると、通常は WinRunner の結果に含まれるイベントごとに、QuickTest の結果にノードが含まれます。WinRunner ステップに対応するノードを選択すると、右の表示枠に WinRunner テストのサマリと、選択したステップの詳細が表示されます。

注：呼び出された WinRunner テストの結果は、WinRunner テストの results フォルダから見ることもできます。Quality Center に格納されている WinRunner テストの場合、Quality Center から WinRunner テストの結果を見ることができます。

詳細については、1013 ページ「テスト結果での WinRunner テスト・ステップの表示」を参照してください。

WinRunner テストの設計と実行の詳細については、WinRunner のドキュメントを参照してください。

WinRunner 関数の呼び出し

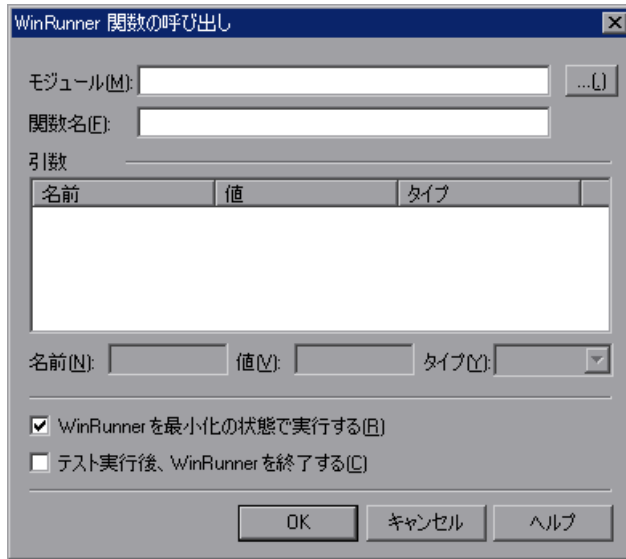
関数を呼び出すために、QuickTest から WinRunner に接続すると、WinRunner が起動され、コンパイル済みモジュールが読み込まれてから、関数が呼び出されます。これは、QuickTest で WinRunner のユーザ定義関数を使用する場合に便利です。

QuickTest から WinRunner 関数を呼び出すには、その関数と、その関数が含まれているコンパイル済みモジュールを指定します。

注：WinRunner 関数から返される値を QuickTest テストの中で取得することはできません。ただし、返された値は、結果の中で見ることができます。

WinRunner のコンパイル済みモジュールのユーザ定義関数を呼び出すには、次の手順を実行します。

- 1 [挿入] > [WinRunner の呼び出し] > [関数] を選択します。[WinRunner 関数の呼び出し] ダイアログ・ボックスが表示されます。



- 2 [モジュール] ボックスに、その関数を含んでいるコンパイル済みモジュールのパスを入力するか、参照ボタンを使って指定します。

WinRunner の TSL 関数を呼び出すには、任意のコンパイル済みモジュールのパスを入力します。

- 3 [関数名] ボックスに、指定したコンパイル済みモジュールに定義されている関数の名前を入力するか、任意の WinRunner TSL 関数を入力します。
- 4 [引数] ボックスの中をクリックします。コンピュータ上で WinRunner が開いている場合、[引数] ボックスには、選択した関数に対して定義されている引数の名前が表示されます。WinRunner が開いていなければ、[引数] ボックスには関数に対する最大 15 個までの引数を表す p1 から p15 までが表示されます。

5 次のように、**in** または **inout** 引数の値を入力します。

- ▶ **[引数]** ボックスで引数を選択して強調表示にします。引数の名前が **[名前]** ボックスに表示されます。
- ▶ 「in」または「inout」の引数タイプを選択した場合、**[値]** ボックスに新しい値を入力します。
- ▶ **[タイプ]** ボックスで、正しい引数のタイプ (**in/out/inout**) を選択します。

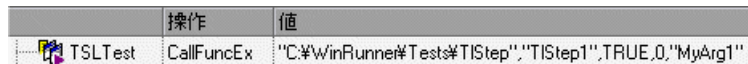
注： QuickTest ランダム・パラメータまたは環境パラメータのパラメータ値や、QuickTest データ・テーブルのパラメータ値を、関数の **in** 引数または **inout** 引数として使用することもできます。それには引数の情報を **TSLTest.CallFuncEx** ステートメントに手作業で入力します。詳細については、1508 ページ「WinRunner 関数への QuickTest パラメータの引き渡し」を参照してください。

関数パラメータの詳細については、『**HP WinRunner ユーザーズ・ガイド**』を参照してください。

- 6 関数が実行されている間、WinRunner ウィンドウが表示されないようにするには、**[WinRunner を最小化の状態で行う]** を選択します
- 7 WinRunner 関数を呼び出したステップが完了したときに WinRunner アプリケーションを終了させるには、**[テスト実行後、WinRunner を終了する]** を選択します
- 8 **[OK]** をクリックして、ダイアログ・ボックスを閉じます。

QuickTest では、TSL 関数への呼び出しが次のように表示されます。

- ▶ キーワード・ビューには、WinRunner の **CallFuncEx** ステップが表示されます。例を次に示します。

	操作	値
	CallFuncEx	"C:¥WinRunner¥Tests¥TIStep","TIStep1",TRUE,0,"MyArg1"

- ▶ エキスパート・ビューには、VBScript で書かれた **TSLTest.CallFuncEx** ステートメントが表示されます。次に例を示します。

CallFuncEx "C:¥WinRunner¥Tests¥TIStep","TISep1",TRUE, 0, "MyArg1"

CallFuncEx メソッドの構文は、次のとおりです。

TSLTest.CallFuncEx モジュールのパス, 関数, 最小化実行, アプリケーション終了 [, 引数]

注 : QuickTest 6.0 で作成したテストは、CallFunc メソッドを使用して WinRunner テストを呼び出している場合があります。このメソッドは、構文が少し異なります。このメソッドでもテストは正常に実行されます。ただし、**CallFuncEx** メソッド（および対応する引数の構文）を使用するようにテストを更新することをお勧めします。これら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。

テストの実行後、結果を表示できます。詳細については、1509 ページ「結果の表示」を参照してください。

WinRunner 関数、関数引数、および WinRunner のコンパイル済みモジュールの詳細については、『**HP WinRunner ユーザーズ・ガイド**』および『**HP WinRunner TSL リファレンス・ガイド**』を参照してください。

WinRunner 関数への QuickTest パラメータの引き渡し

WinRunner 関数の in 引数や inout 引数に固定した値を設定する代わりに、QuickTest ランダム・パラメータや環境パラメータで定義されたパラメータ値、または QuickTest データ・テーブルで定義されたパラメータ値を WinRunner に引き渡すように、QuickTest に指示できます。これらのパラメータを指定するには、TSLTest.CallFuncEx ステートメントの **パラメータ** 引数として、適切なステートメントを入力します。

たとえば WinRunner を使って、アプリケーションを起動し、そのアプリケーションにユーザ名とパスワードを入力するユーザ定義関数を作成するとします。

そのためには、ユーザ名とパスワードの値を QuickTest データ・テーブルの **FlightUserName** 列と **FlightPwd** 列から WinRunner に引き渡すよう、QuickTest に指示します。その場合、QuickTest の TSLTest.CallFuncEx ステートメントは、たとえば次のようになります。

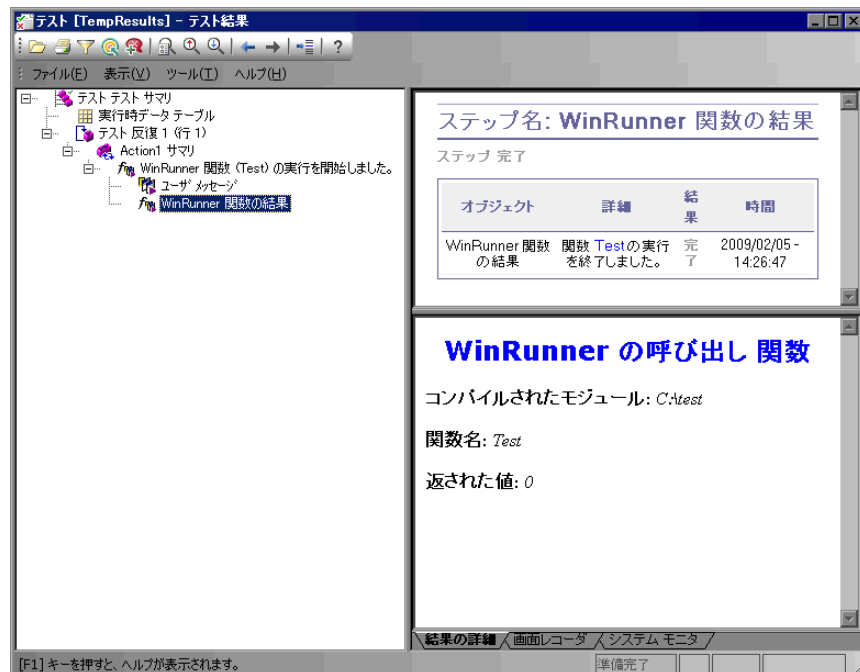

```
TSLTest.CallFuncEx "D:\flightfuncs", "run_flight", TRUE, FALSE,
DataTable("FlightUserName", dtGlobalSheet), DataTable("FlightPwd",
dtGlobalSheet)
```

RandomNumber メソッド, Environment メソッド, および DataTable メソッドの構文および使用方法の詳細については、『**HP QuickTest Professional Object Model Reference**』（英語版）の「**Utility**」のセクションを参照してください。

結果の表示

QuickTest から WinRunner 関数を実行した後、関数呼び出しの結果を表示することができます。QuickTest の [テスト結果] ウィンドウに、WinRunner 関数の開始と WinRunner 関数の結果が表示されます。呼び出した関数に `report_msg` または `tl_step` などのイベントが含まれている場合、それらのイベントの結果に関する情報も含まれます。

結果ツリーの [WinRunner 関数の結果] を選択して強調表示にすると、関数の戻り値と関数への呼び出しに関する追加情報が表示されます。



WinRunner 関数およびコンパイル済みモジュールを使用した作業の詳細については、WinRunner のドキュメントを参照してください。

第 58 章

HP のパフォーマンス・テストおよび Business Availability Center 製品を使用した作業

QuickTest を使用して、アプリケーションの機能をテストする一連のテストを作成して実行できたら、今度はアプリケーションがどれくらいの負荷を処理できるかを検証したり、アプリケーションが実行する様子を監視したりできます。

HP パフォーマンス・テスト製品（LoadRunner および Performance Center）は、一定の負荷または過負荷状態でのアプリケーションのパフォーマンスをテストします。負荷を生成するために、これらのパフォーマンス・テスト製品によって数百人の仮想ユーザが実行されます。これらの仮想ユーザは、一貫性のある再現可能かつ測定可能な負荷を提供し、現実のユーザとまったく同じようにアプリケーションを操作します。

HP Business Availability Center は、エンド・ユーザ体験をリアルタイムで監視できます。Business Process Monitor は、監視対象アプリケーションを対象に仮想ユーザを実行して典型的な操作を実行します。

すでに QuickTest でテストを作成して、それがユーザのアクションをうまく表現することがわかっている場合は、その QuickTest テストをパフォーマンス・テストおよびアプリケーション管理の基盤として使うことができます。サイレント・テスト・ランナーを使用して、QuickTest テストが LoadRunner、Performance Center、および Business Process Monitor から正常に実行されることを事前に確認できます。

本章の内容

- ▶ HP のパフォーマンス・テストおよび Business Availability Center 製品を使用した作業について（1512 ページ）
- ▶ QuickTest のパフォーマンス・テストおよび Business Availability Center の使用（1513 ページ）

- ▶ パフォーマンス・テスト製品または Business Process Monitor で使用する QuickTest テストの設計 (1514 ページ)
- ▶ パフォーマンス・テストまたは Business Process Monitor でのテストの挿入と実行 (1516 ページ)
- ▶ トランザクションの測定 (1518 ページ)
- ▶ サイレント・テスト・ランナーの使用 (1523 ページ)

HP のパフォーマンス・テストおよび Business Availability Center 製品を使用した作業について

QuickTest では、アプリケーション全般の機能を検査し、アプリケーションのすべての要素があらゆる状況で期待どおりに動作することを確認する複雑なテストを作成できます。

HP のパフォーマンス・テスト製品および HP Business Availability Center 製品で使用されている実行メカニズムは同じです。つまり、LoadRunner、Performance Center、および Business Process Monitor のすべてに互換性のあるテストを作成することができ、QuickTest で設計され、デバッグされたテストやテストのセグメントを利用できるということです。

たとえば、QuickTest テストをパフォーマンス・テストの特定のポイントに追加し、それらのポイントでの追加の負荷によってアプリケーションの機能が影響を受けていないことを確認できます。また、Business Process Monitor を対象として QuickTest テストを実行し、エンド・ユーザ体験をシミュレートし、アプリケーションが正しく適切なタイミングで実行されることを確認することもできます。

また QuickTest は、LoadRunner、Performance Center、および Business Process Monitor と統合専用設計されているいくつかの機能を提供します。しかし、これらの製品は、標準的なユーザ操作を同時に行う多数のユーザを表す仮想ユーザを使用して、テストを実行するように設計されているので、QuickTest とこれらの製品を統合すると、QuickTest のいくつかの機能が利用できない場合があります。

LoadRunner, Performance Center, Business Process Monitor のいずれかまたはすべてと、QuickTest とで、1 つのテストを使う場合は、テストを設計するとき、各製品がサポートするオプションの違いを考慮しなければなりません。詳細については、1514 ページ「パフォーマンス・テスト製品または Business Process Monitor で使用する QuickTest テストの設計」および 1516 ページ「パフォーマンス・テストまたは Business Process Monitor でのテストの挿入と実行」を参照してください。

QuickTest のパフォーマンス・テストおよび Business Availability Center の使用

Services オブジェクトと関連メソッドを使用して、パフォーマンス・テストおよび Business Availability Center に特に関連のあるステートメントを挿入できます。これらには、AddWastedTime, EndDistributedTransaction, EndTransaction, GetEnvironmentAttribute, LogMessage, Rendezvous, SetTransaction, SetTransactionStatus, StartDistributedTransaction, StartTransaction, ThinkTime, および UserDataPoint があります。これらのメソッドの詳細については、『**HP QuickTest Professional Object Model Reference**』（英語版）の「**Services**」の項、および HP パフォーマンス・テストまたは Business Availability Center のドキュメントを参照してください。



また、[挿入] > [トランザクションの開始] および [挿入] > [トランザクションの終了] メニュー・オプションまたはツールバー・ボタンを使用して、StartTransaction および EndTransaction ステートメントを挿入できます。これらのオプションの詳細については、1518 ページ「トランザクションの測定」を参照してください。

注： LoadRunner, Performance Center, および Business Process Monitor は、トランザクションに含まれたデータだけを使用し、トランザクション外のテストのデータは無視します。

パフォーマンス・テスト製品または Business Process Monitor で使用する QuickTest テストの設計

LoadRunner, Performance Center, または Business Process Monitor で使用する QuickTest テストは、シンプルに保ち、対象の操作を限定し、外部アクションの使用や外部ファイルへの参照は避けるべきです。また、アクションの反復を操作するときは、対応関係にある **StartTransaction** ステートメントおよび **EndTransaction** ステートメントが同じアクション内に含まれている必要があります。

パフォーマンス・テスト向けのテストの設計

パフォーマンス・テスト製品で使用するテストを設計するときは、次のガイドラインに従ってください。

- ▶ 外部アクションや、外部データ・テーブル・ファイル、環境変数ファイル、共有オブジェクト・リポジトリ、関数ライブラリなどの外部リソース（Quality Center に保存されているリソースを含む）への参照を含めないようにします。これは、LoadRunner または Performance Center が外部のアクションやリソースにアクセスできない場合があるためです（ただし、リソースがネットワーク上で見つかる場合、QuickTest はそのリソースを使用します）。
- ▶ パフォーマンス・テストで有益な情報を提供するためには、すべての QuickTest テストに少なくとも 1 つのトランザクションを含める必要があります。
- ▶ テスト内の最後の（1 つまたは複数の）ステップで、テスト対象のアプリケーションを、実行中のすべての子プロセスも含めて必ず終了するようにします。これにより、テストの次の反復で再びアプリケーションを開くことができます。

Business Process Monitor 向けのテストの設計

Business Process Monitor で使用するテストを設計するときは、次のガイドラインに従ってください。

- ▶ Business Process Monitor で有益な情報を提供するためには、すべての QuickTest テストに少なくとも 1 つのトランザクションを含める必要があります。
- ▶ 2 つの異なる Business Process Monitor プロファイルで分散トランザクションを測定する場合、**StartDistributedTransaction** ステートメントを使用するプロファイルを、関連付けられた **EndDistributedTransaction** ステートメントを使用するプロファイルより前に実行する必要があります。

- ▶ 分散トランザクションを測定する場合、テストを 1 つの **Business Process Monitor** のインスタンスに関連付けることを確認します。**Business Process Monitor** によってすべてのインスタンスの終了トランザクション名が検索され、複数のインスタンスに含まれている場合に不正な分散トランザクションを終了する場合があります。
- ▶ 分散トランザクションを 2 つの **Business Process Monitor** のプロファイルで測定する場合、指定するタイムアウト値の大きさが十分であり、**StartDistributedTransaction** ステップを含むプロファイルおよび **EndDistributedTransaction** ステップを含むプロファイルの前に実行するすべてのプロファイルが、指定したタイムアウト値より短い時間で実行し終えることを確認します。
- ▶ **Business Process Monitor** では、**Quality Center** に保存されているリソース（共有オブジェクト・リポジトリ、関数ライブラリ、外部データ・テーブル、外部アクションなど）を含む外部リソースへのアクセスを必要とする **QuickTest Professional** テストの実行はサポートされていません。外部リソースを必要とするテストは、**Business Process Monitor** 上での実行に失敗する場合があります（ただし、リソースがネットワーク上で見つかる場合、**QuickTest** はそのリソースを使用します）。
- ▶ テスト内の最後の（1 つまたは複数の）ステップで、テスト対象のアプリケーションを、実行中のすべての子プロセスも含めて必ず終了するようにします。このクリーンアップ・ステップにより、次のテスト実行で再びアプリケーションを開くことができますようになります。

パフォーマンス・テストまたは Business Process Monitor でのテストの挿入と実行

パフォーマンス・テストまたは Business Process Monitor で QuickTest テストを挿入し実行するときは、次のガイドラインに従ってください。

注：サイレント・テスト・ランナーを使用すれば、パフォーマンス・テストまたは Business Process Monitor からのテストの実行方法をシミュレートできます。詳細については、1523 ページ「サイレント・テスト・ランナーの使用」を参照してください。

Performance Center および LoadRunner でのテストの挿入と実行

- ▶ 1 台のコンピュータで同時に実行できる GUI Vuser は 1 つまでです (GUI Vuser とは QuickTest テストを実行する Vuser のことです)。
同じアプリケーションに対して複数の GUI Vuser を実行するには、GUI 仮想ユーザごとにターミナル・サーバ・セッションを開きます。詳細については、HP パフォーマンス・テストのドキュメントを参照してください。
- ▶ QuickTest テストを LoadRunner シナリオに挿入するには、Controller の [テストを開く] ダイアログ・ボックスでテスト・フォルダまで移動し、[**ファイルの種類**] ボックスで [**QuickTest テスト**] を選択します (LoadRunner 9.1 以前のバージョンの場合は [**Astra テスト**] を選択)。これで、フォルダ内の QuickTest テストが表示されます。
- ▶ Performance Center で QuickTest テストを使用するには、圧縮した QuickTest テストを作成し、[Performance Center User Site Vuser Scripts Page] にアップロードします。
- ▶ Performance Center または LoadRunner から QuickTest テストを実行する前に、QuickTest コンピュータ上の QuickTest が終了していることを確認します。
- ▶ QuickTest を使用して作成したテスト (スクリプト) では、トランザクション・ブレイクダウンはサポートされていません。

- ▶ 次のコンピュータ上では QuickTest を実行できません。
 - ▶ ログオフまたはロックされたコンピュータ。これらの場合は、ターミナル・サーバでの QuickTest の実行を検討してください。
 - ▶ すでに QuickTest テストが実行されているコンピュータ。テストの実行が完了していることを確かめてから、別の QuickTest テストを開始してください。
- ▶ LoadRunner または Performance Center の [実行環境設定] ダイアログ・ボックスの設定は、QuickTest テストには関係しません。
- ▶ パフォーマンス・テストを実行しているときは、**ResultDir** QuickTest 環境変数を使用することはできません。

LoadRunner または Performance Center での作業の詳細については、HP パフォーマンス・テストのドキュメントを参照してください。

Business Process Monitor でのテストの挿入と実行

- ▶ Business Process Monitor で QuickTest テストを実行する前に、使用しているバージョンの Business Process Monitor でどのバージョンの QuickTest がサポートされているのかを確認してください。詳細については、Business Process Monitor のドキュメントを参照してください。
- ▶ Business Process Monitor で QuickTest テストを実行するには、Business Process Monitor コンピュータに QuickTest がインストールされ閉じている必要があります。
- ▶ Business Process Monitor で一度に実行できる QuickTest テストは 1 つだけです。前の QuickTest 実行セッションが完了していることを確かめてから、別の QuickTest テストを開始してください。
- ▶ QuickTest を使用して作成したテストでは、トランザクション・ブレイクダウンはサポートされていません。
- ▶ QuickTest テストは、Business Process Monitor にアップロードする前に圧縮する必要があります。

QuickTest テストを Business Availability Center にアップロードした後に、テストのローカル・コピーに変更を加えた場合は、変更を加えたテストを Business Process Monitor で実行できるようにするために、ZIP 圧縮したテストを再度アップロードします。

- ▶ ログオフされたコンピュータ、ロックされたコンピュータ、または QuickTest を非対話型サービスとして実行しているコンピュータでは、QuickTest はテストを実行できません。
- ▶ Business Process Monitor でテストを実行しているときは、ResultDir QuickTest 環境変数を使用することはできません。

Business Availability Center での作業の詳細については、Business Availability Center のドキュメントを参照してください。

トランザクションの測定

トランザクションを定義することで、テストの特定セクションの実行にかかる時間を測定できます。トランザクションは、測定対象のアプリケーション内のプロセスを表しています。テストには、LoadRunner、Performance Center、または Business Process Monitor で使用できるトランザクションを含める必要があります。これらの製品は、トランザクションに含まれたデータだけを使用し、トランザクション以外のテストのデータは無視します。

トランザクション**開始**ステートメントとトランザクション**終了**ステートメントでテストの該当セクションを囲むことで、テスト内のトランザクションを定義できます。たとえば、飛行機の座席を予約するのにかかる時間や、クライアントのターミナルに確認メッセージが表示されるまでにかかる時間を測定するトランザクションを定義できます。

テストの実行中、StartTransaction ステップは、時間測定の開始を示します。時間測定は、EndTransaction ステップに到達するまで継続されます。テストレポートに、トランザクションの実行に要した時間が表示されます。

注：トランザクションの開始時にすでに同じ名前のトランザクションが開いていると、先行していたトランザクションは**失敗**ステータスで終了し、その後新しいトランザクションが開始されます。

トランザクション内で使用するステートメントの詳細については、『**HP QuickTest Professional Object Model Reference**』（英語版）を参照してください。

テストに追加できるトランザクションの数には上限がありません。トランザクション内にトランザクションを挿入することも可能です。

次に、トランザクションを含むサンプル・テストの一部を、キーワード・ビューに表示されるとおりに示します。

トランザクション開始	Services	StartTransaction	"ReserveSeat"	"ReserveSeat" トランザクションを開始する。
	Find a Flight: Mercury			
	fromPort	Select	"Frankfurt"	"fromPort" list から "Frankfurt" メニュー項目を選択する。
	fromMonth	Select	"Dec"	"fromMonth" list から "Dec" メニュー項目を選択する。
	fromDay	Select	"29"	"fromDay" list から "29" メニュー項目を選択する。
	toPort	Select	"Acapulco"	"toPort" list から "Acapulco" メニュー項目を選択する。
	toMonth	Select	"Dec"	"toMonth" list から "Dec" メニュー項目を選択する。
	toDay	Select	"31"	"toDay" list から "31" メニュー項目を選択する。
	servClass	Select	"Business"	"servClass" radio button group の中で ["Business"] ラジオ ボタンを...
	findFlights	Click	37.5	"findFlights" image をクリックする。
	Select a Flight: Mercury			
	reserveFlights	Click	63.12	"reserveFlights" image をクリックする。
トランザクション終了	Services	EndTransaction	"ReserveSeat"	"ReserveSeat" トランザクションを終了する。


エキスパート・ビューではテストのサンプル部分は次のように表示されます。

```

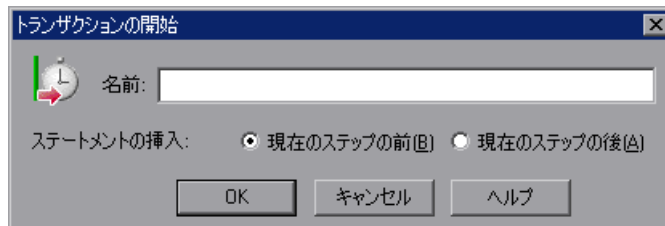
Services.StartTransaction "ReserveSeat"
Browser("Welcome: Mercury Tours").Page("Find a Flight: Mercury").
  WebList("fromPort").Select "London"
Browser("Welcome: Mercury Tours").Page("Find a Flight: Mercury").
  WebList("toPort").Select "Frankfurt"
Browser("Welcome: Mercury Tours").Page("Find a Flight: Mercury").
  WebList("toDay").Select "12"
Browser("Welcome: Mercury Tours").Page("Find a Flight: Mercury").
  WebRadioGroup("servClass").Select "Business"
Browser("Welcome: Mercury Tours").Page("Find a Flight: Mercury").
  WebList("airline").Select "Blue Skies Airlines"
Browser("Welcome: Mercury Tours").Page("Find a Flight: Mercury").
  Image("findFlights").Click 65,12
Browser("Welcome: Mercury Tours").Page("Select a Flight: Mercury").
  WebRadioGroup("outFlight").Select "Blue Skies Airlines"
Browser("Welcome: Mercury Tours").Page("Select a Flight: Mercury").
  WebRadioGroup("inFlight").Select "Blue Skies Airlines"
Browser("Welcome: Mercury Tours").Page("Select a Flight: Mercury").
  Image("reserveFlights").Click 46,8
Services.EndTransaction "ReserveSeat"
    
```

ステップ・ジェネレータまたはエキスパート・ビューを使用して、さまざまなトランザクション関連ステートメントを挿入できます。詳細については、『**HP QuickTest Professional Object Model Reference**』（英語版）の「**Services**」の項を参照してください。「トランザクション開始」ステップおよび「トランザクション終了」ステップは、QuickTest ウィンドウ内のオプションを使用して入力することもできます。

【トランザクションの開始】 ダイアログ・ボックス

説明	トランザクションの時間測定の開始を知らせるステップを挿入できます。
アクセス方法	<ul style="list-style-type: none"> ▶ [挿入] > [トランザクションの開始] メニュー・コマンドを選択します。 ▶ [トランザクションの開始] ツールバー・ボタン  をクリックします。
詳細	<p>概念の概要 : 1518 ページ「トランザクションの測定」</p> <p>その他の関連トピック : 1521 ページ「[トランザクションの終了] ダイアログ・ボックス」</p>


次は [トランザクションの開始] ダイアログ・ボックスの画像です。



[トランザクションの開始] ダイアログ・ボックスのオプション

オプション	説明
[名前]	測定するトランザクションの名前です。 注： トランザクション名にスペースを含めることはできません。
[ステートメントの挿入]	選択したステップに対して、 StartTransaction ステップを挿入する場所を示します。[現在のステップの前] または [現在のステップの後] を選択します。

[トランザクションの終了] ダイアログ・ボックス

説明	トランザクションの時間測定の終了を知らせるステップを挿入できます。
アクセス方法	<ul style="list-style-type: none"> ▶ [挿入] > [トランザクションの終了] メニュー・コマンドを選択します。 ▶ [トランザクションの終了] ツールバー・ボタン  をクリックします。
重要	実行セッション中にエラーが発生した場合でもトランザクション内のすべてのステップを実行するよう QuickTest を設定したい場合があります。これには、[テストの設定] ダイアログ・ボックス ([ファイル] > [設定] > [実行] ノード) の [実行] 表示枠で、[実行セッション中にエラーが発生した場合] リストから [次のステップに進む] を選択します。これらの問題に対処するために、回復シナリオまたはほかのエラー処理ステップを作成することもできます。詳細については、第 48 章「回復シナリオの定義と使用」を参照してください。
詳細	<p>概念の概要： 1518 ページ「トランザクションの測定」</p> <p>その他の関連トピック： 1520 ページ「[トランザクションの開始] ダイアログ・ボックス」</p>

次は [トランザクションの終了] ダイアログ・ボックスの画像です。



[トランザクションの終了] ダイアログ・ボックスのオプション

オプション	説明
[名前]	終了するトランザクションの名前です。 リストには、現在のアクションで選択したステップより前に開始されるすべてのトランザクション名が含まれています。
[ステートメントの挿入]	選択したステップに対して、 EndTransaction ステップを挿入する場所を示します。[現在のステップの前] または [現在のステップの後] を選択します。

サイレント・テスト・ランナーの使用

サイレント・テスト・ランナーを使用して、LoadRunner、Performance Center、および Business Availability Center からの QuickTest テストの実行方法をシミュレートできます。サイレント・テスト・ランナーを使用してテストを実行すると、QuickTest のユーザ・インタフェースを開かずにサイレント・テスト・ランナーが起動し、LoadRunner、Performance Center、または Business Availability Center から実行したときと同じ速度でテストが実行されます。テスト実行の最後では、テスト実行とトランザクション回数に関する情報を表示できます。詳細については、1526 ページ「サイレント実行のテスト実行情報の表示」を参照してください。

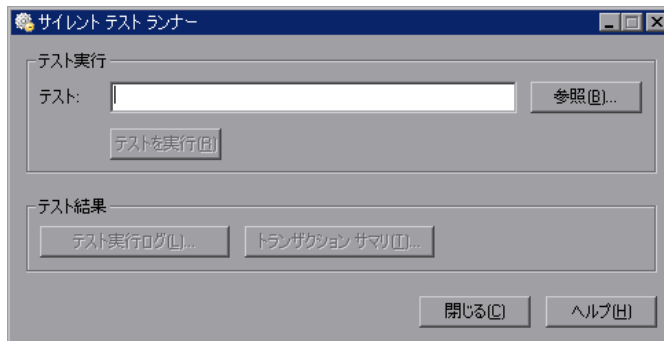
また、サイレント・テスト・ランナーを使用して、QuickTest テストが LoadRunner、Performance Center、および Business Availability Center と互換性があることを確かめることもできます。サイレント・テスト・ランナーを使用してテストを実行するときにこれらの製品でサポートされていない機能を使用している場合、テストは失敗します。サポートされていない機能の詳細については、1514 ページ「パフォーマンス・テスト製品または Business Process Monitor で使用する QuickTest テストの設計」および 1516 ページ「パフォーマンス・テストまたは Business Process Monitor でのテストの挿入と実行」を参照してください。

[サイレント テスト ランナー] ダイアログ・ボックス

説明	LoadRunner および Business Availability Center からの QuickTest テストの実行方法をシミュレートできます。また QuickTest テストが LoadRunner および Business Availability Center と互換性があることを確かめることもできます。
アクセス方法	[スタート] > [プログラム] > [QuickTest Professional] > [Tools] > [Silent Test Runner] メニュー・コマンドを選択します。

<p>重要</p>	<ul style="list-style-type: none"> ▶ QuickTest がすでに開いている場合や、別のテストが現在実行中の場合は、サイレント・テスト・ランナーを実行できません。QuickTest を閉じ、そのプロセスが終了するまで待ってから、サイレント・テスト・ランナーを使用してテストを実行する必要があります。 ▶ 起動できるサイレント・テスト・ランナーのインスタンスは 1 つだけで、実行を指定できるテストも 1 つだけです。 ▶ サイレント・テスト・ランナーでテストを実行しているときは、ResultDir QuickTest 環境変数は使用できません。
<p>詳細</p>	<p>概念の概要 : 1523 ページ「サイレント・テスト・ランナーの使用」</p> <p>追加関連トピック : 1526 ページ「サイレント実行のテスト実行情報の表示」</p>

次は [サイレント テスト ランナー] ダイアログ・ボックスの画像です。



[サイレントテストランナー] ダイアログ・ボックスのオプション

オプション	説明
[テスト]	<p>実行するテストのファイル・システムのフル・パスです。</p> <p>注：ネットワーク・パスを指定するには、ネットワーク・ドライブを割り当てる必要があります。</p>
[テストを実行]	<p>テストを実行できます</p> <p>(テスト・パスが [テスト] ボックスで指定されている場合にのみ有効です)。</p> <p>このボタンをクリックすると、QuickTest ユーザ・インタフェースを開かずにテストが実行されます。テストの実行中は、「テストを実行しています ...」というテキストが [テストを実行] ボタンの横に表示されます。</p> <p>テストの実行が完了すると、テキスト「テストを実行しています ...」が「テスト実行が完了しました」に変わります。サイレント・テスト・ランナーがテストを実行できなかった場合は、「テストを実行できませんでした」というテキストが表示されます。</p> <p>注：テストの実行を開始した後に、サイレント・テスト・ランナーからテストの実行を停止することはできません。サイレント・テスト・ランナーを閉じて、テストの実行は継続されます。実行を終了するには、mdrv.exe プロセスを手作業で終了します。</p>
[テスト実行ログ]	<p>選択したテストの最新の実行ログを表示できます。サイレント・テスト・ランナーでテストを実行するたびに、前のログ・ファイルが現在の実行結果で上書きされます</p> <p>(選択したテストがサイレント・テスト・ランナーで少なくとも 1 回実行された場合にのみ有効です)。</p> <p>詳細については、1526 ページ「テスト実行ログの表示」を参照してください。</p>
[トランザクションサマリ]	<p>テスト内のトランザクションのサマリを表示できます</p> <p>(選択したテストにトランザクションが少なくとも 1 つ含まれていて、選択したテストがサイレント・テスト・ランナーで少なくとも 1 回実行された場合にのみ有効です)。</p> <p>詳細については、1526 ページ「トランザクション・サマリの表示」を参照してください。</p>

サイレント実行のテスト実行情報の表示

サイレント・テスト・ランナーでは、テストの実行情報がログ・ファイルに記録されます。テストごとにテスト実行ログが生成され、トランザクションのあるテストの場合は追加のトランザクション・サマリが生成されます。

テスト実行ログの表示

テスト実行ログは、**< QuickTest Professional > ¥Tests¥ <テスト名>** フォルダに **output.txt** として保存されます。ログ・ファイルはサイレント・テスト・ランナーによるテスト実行のたびに保存され、テストを再実行すると上書きされます。ログ・ファイルを開くには、**[テスト実行ログ]** をクリックします。

ログ・ファイルにはテストの実行に関する情報が表示されます。たとえば、個々の反復、アクション呼び出し、ステップ・トランザクション、失敗したステップなどに関する情報が表示されます。各行には、メッセージまたはエラー ID が表示されます。ログ・ファイル内のメッセージおよびエラー・コードの詳細については、Performance Center または Business Availability Center のドキュメントを参照してください。

トランザクション・サマリの表示

トランザクション・サマリは、**< QuickTest Professional > ¥Tests¥ <テスト名>** フォルダに **transactions.txt** として保存されます。トランザクション・サマリはトランザクションを含むテストごとに保存され、テストを再実行すると上書きされます。トランザクション・サマリを開くには、**[トランザクションサマリ]** をクリックします。トランザクション・サマリには、テスト内の各トランザクションに対応した行が表示されます。各トランザクションについて、ステータスと、総継続時間および浪費時間（秒単位）が表示されます。サイレント・テスト・ランナーでのトランザクション測定は、テストを LoadRunner, Performance Center, または Business Availability Center から実行した場合とまったく同じになります。

注：注：

- ▶ トランザクション・サマリは、**EndTransaction** ステートメントで終わるトランザクションが含まれているテストについてのみ生成されます。トランザクションが開始されたものの、テストの失敗のために終了しなかった場合は、そのトランザクションはトランザクション・サマリに記録されません。
 - ▶ 分散トランザクション（あるテストで開始し、別のテストで終了するトランザクション）は、トランザクション・サマリには報告されず、テスト実行ログに記録されます。
 - ▶ トランザクション・サマリに記録されているトランザクション情報はすべて、テスト実行ログにも記録されます。
-

第 XIII 部

付録

付録 A

各アドイン用にサポートされているチェックポイントおよび出力値

本章の表に、各アドイン用に QuickTest Professional でサポートしているチェックポイントおよび出力値のカテゴリを示します。

個別のアドインにおけるチェックポイントおよび出力値の使用の詳細については、該当するアドインの項を参照してください。

本章の内容

- ▶ サポートされているチェックポイント（1532 ページ）
- ▶ サポートされている出力値（1534 ページ）

サポートされているチェックポイント

表の凡例

- ▶ S : サポート
- ▶ NS : 未サポート
- ▶ NA : 不適用

追加情報については、1533 ページ「脚注」を参照してください。

	アクセシビリティ	ビットマップ	データベース	画像	ページ	標準	テーブル	テキスト	テキスト領域	XHTML (アプリケーションから)	XHTML (リソースから)
ActiveX	NS	S	NA	NS	NA	S	S	S	S	NA	NS
Delphi	NS	S	NA	NS	NA	S	S	S	S	NA	S
Java	NA	S	NA	NA	NA	S	S	S ⁶	S	NA	NS
.NET Web Forms⁵	S	S	NA	NA	NA	S	S	S	NS	S	S
.NET Windows Forms	NA	S	NA	NA	NA	S	S	N	S	N	N
Oracle	NA	S	NA	NA	NA	S	S	NS	NS	NA	NA
PeopleSoft	S	S	NA	S	S	S	S	S ³	NS	S	S
PowerBuilder⁴	NS	S	NA	NS	NA	S	S	S	S	NA	NS
SAP Web	S	S	NA	S	S	S	S	S	NS	S	S
SAP Windows	S ⁷	S	NA	S ⁷	S ⁷	S	S	S ⁷	NS	S ⁷	NA
Siebel	S	S	NA	S	S	S	S	S	NS	S	S
標準 Windows	NS	S	NA	NS	NA	S	S	S	S	NA	NS
Stingray	NA	S	NA	NA	NA	S	S	S	S	NA	NS

	アクセシビリティ	ビットマップ	データベース	画像	ページ	標準	テーブル	テキスト	テキスト領域	XML (アプリケーションから)	XML (リソースから)
ターミナル・エミュレータ	NA	S	NA	NA	NA	S	NA	NA	NA	NA	NA
Visual Age	NA	S	NA	NA	NA	S	S	S	S	NA	NS
Visual Basic	NS	S	NA	NS	NA	S	S	S	S	NA	NS
Web ²	S	S	NA	S	S	S	S	S ³	NS	S	NS
Web サービス	NA	NA	NA	NA	NA	S	NA	NA	NA	S	NS
WPF	NA	S	NA	NA	NA	S	NA	S	S	NA	NA

脚注

¹ 標準およびビットマップ・チェックポイントはビジネス・コンポーネントでのみサポートされています。

² コンポーネントで Web オブジェクトのチェックポイントを作成するときは、ビットマップ・チェックポイントと標準チェックポイントのみを使用できます。

³ チェックポイントはページ、フレーム、および ViewLink オブジェクトのみでサポートされています。

⁴ チェックポイントを PowerBuilder DataWindow コントロールに挿入すると、QuickTest によってテーブルとして処理され、[テーブル チェックポイントのプロパティ] ダイアログ・ボックスが開きます (コンポーネントでは未サポート)。

⁵ NET Web Forms に関して、WbfTreeView, WbfToolbar, および WbfTabStrip オブジェクトのテキスト・チェックポイントはサポートされていません。

⁶Java オブジェクトのテキスト・チェックポイント・メカニズムは、標準設定では無効です。このチェックポイントは [Advanced Java Options] ダイアログ・ボックスで有効にできます (テスト専用)。

⁷これは、QuickTest が Web インフラストラクチャを使用して HTML 要素を記録する場合にのみサポートされていますが、SAPGui スクリプティング・インタフェース ([オプション] ダイアログ・ボックスの [SAP] 表示枠で選択) を使用して記録する場合はサポートされていません。

サポートされている出力値

表の凡例

- ▶ S : サポート
- ▶ NS : 未サポート
- ▶ NA : 不適用

追加情報については、1535 ページ「脚注」を参照してください。

	アクセスビリティ	ビットマップ	データベース	ページ	標準	テーブル	テキスト	テキスト領域	XML (アプリケーションから)	XML (リソースから)
ActiveX	NS	NA	NA	NA	S	S	S	S	NA	S
Delphi	NS	NA	NA	NA	S	NA	S	S	NA	S
Java	NA	NA	NA	NA	S	NA	S ⁶	NA	NA	NA
.NET Web Forms	NA	NA	NA	S	S	S	S	NA	NA	NA
.NET Windows Forms	NA	NA	NA	NA	S	S	NA	NA	NA	NA
Oracle	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
PeopleSoft	NA	NA	NA	S	S	S	S ³	NS	S	S

	アクセシビリティ	ビットマップ	データベース	ページ	標準	テーブル	テキスト	テキスト領域	XML (アプリケーションから)	XML (リソースから)
PowerBuilder⁴	NA	NA	NA	NA	S	NA	S	S	NA	S
SAP Web	NA	NA	NA	S	S	S	S	NS	S	S
SAP Windows	NA	NA	NA	S ⁶	S	S	S ⁶	NS	S ⁶	S
Siebel	NA	NA	NA	S	S	S	S	NS	S	S
標準 Windows	NA	NA	NA	NA	S	NA	S	S	NA	S
Stingray	NA	NA	NA	NA	S	NA	S	S	NA	S
ターミナル・エミュレータ	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Visual Age	NA	NA	NA	NA	NA	S	S	S	NA	NA
Visual Basic	NA	NA	NA	NA	S	NA	S	S	NA	S
Web²	NA	NA	NA	S	S	S	S ³	NS	S	NA
Web サービス	NA	NA	NA	NA	NA	NA	NA	NA	NA	S
WPF	NA	NA	NA	NA	S	NA	S	S	NA	NA

脚注

¹ 標準およびビットマップ出力値はビジネス・コンポーネントでのみサポートされています。

² コンポーネントで Web オブジェクトの出力値を作成するときは、標準出力値のみを使用できます。

³ 出力値はページ、フレーム、および ViewLink オブジェクトのみでサポートされています。

⁴ 出力値ステップを PowerBuilder DataWindow コントロールに挿入すると、QuickTest によってテーブルとして処理され、[テーブル出力値のプロパティ] ダイアログ・ボックスが開きます（コンポーネントでは未サポート）。

⁵ Java オブジェクトのテキスト出力メカニズムは、標準設定では無効です。このチェックポイントは [Advanced Java Options] ダイアログ・ボックスで有効にできます（テスト専用）。

⁶ これは、QuickTest が Web インフラストラクチャを使用して HTML 要素を記録する場合にのみサポートされていますが、SAPGui スクリプティング・インタフェース（[オプション] ダイアログ・ボックスの [SAP] 表示枠で選択）を使用して記録する場合はサポートされていません。

付録 B

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

- ▶ テストの作成（1537 ページ）
- ▶ エキスパート・ビューでのプログラミング（1539 ページ）
- ▶ 動的なコンテンツを使った作業（1541 ページ）
- ▶ Web に関する高度な問題（1543 ページ）
- ▶ 標準 Windows 環境（1546 ページ）
- ▶ テストの保守（1547 ページ）
- ▶ ローカライズされたアプリケーションのテスト（1549 ページ）

テストの作成

- ▶ **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、133 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テストを作成できます。
- ▶ テスト・オブジェクトと同じように振る舞うオブジェクトに「**仮想オブジェクト**」を定義して、通常の記録モードで記録できます。仮想オブジェクトの定義の詳細については、第 47 章「仮想オブジェクトの学習」を参照してください。
- ▶ **低レベル記録**あるいは**アナログ・モード**で、座標に基づいてクリックとキーボード入力を記録できます。低レベル記録とアナログ記録の詳細については、372 ページ「記録モードの選択」を参照してください。
- ▶ **アプリケーションをテストから起動するにはどうすればよいですか。**

次のように、**SystemUtil** ステップをテストに追加することによって、テストの中からアプリケーションを起動できます。

```
SystemUtil.Run "D:¥My Music¥Breathe.mp3","", "D:¥My Music¥Details", "open"
```

また、Windows ベース・アプリケーションの場合は、[記録と実行環境設定] ダイアログ・ボックスの [Windows アプリケーション] タブで、QuickTest が開くアプリケーションを対象として記録と実行を行うように QuickTest を設定していることを確認してください。

- ▶ **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)。記録したテストは、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。

エキスパート・ビューでのプログラミング

▶ 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。

関数は個々のアクション内で定義できます。または、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、任意のテストからそれらを読み出すことができます。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することもできます。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 31 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。

関数を再利用可能なアクションとして格納するのではなく関数ライブラリに格納することによって、QuickTest のパフォーマンスの向上を促進できます。

▶ 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 関数を使用すると、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テス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。

次の例は、InputBox 関数を使用してパスワードの入力をユーザに求めます。

```
Browser("Mercury Tours").Page("Mercury Tours").WebEdit("username").Set
"administrator"
Passwd = InputBox ("Enter password", "User Input")
Browser("Mercury Tours").Page("Mercury Tours").WebEdit("password").Set
Passwd
```

▶ Microsoft Access データベースのデータをテストで使用したいと考えています。どうすればよいですか。

エキスパート・ビューでは、ADO および ODBC を使用してデータベースにアクセスできます。次の例は、データベースの "Authors" テーブル内で、ある著者によって書かれた本を検索するテストです。

```
Dim MyDB
Dim MyEng
Set MyEng = CreateObject("DAO.DBEngine.35")
Dim Td
Dim rs
```

```
' 使用するデータベースを指定します。
Set MyDB = MyEng.OpenDatabase("BIBLIO.MDB")
```

```
' 最初の 10 人の著者の名前を読み取って使用します。
Set Td = MyDB.TableDefs("Authors")
Set rs = Td.OpenRecordset
rs.MoveFirst
For i = 1 To 10
    Browser("Book Club").Page("Search Books").WebEdit("Author Name").Set
rs("Author")
    Browser("Book Club").Page("Search Books").WebButton("Search").Click
Next
```

▶ **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、メッセージをテスト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、テストを作成し、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 **SetTOPProperty** ステップを使用してテストを実行するとき、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、デ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进行处理します。

▶ **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、**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。

また、Web ページ、Web フレーム、Web ファイルの XML ドキュメントの内容を検査する XML チェックポイントを作成できます。QuickTest は XML 出力とスキーマ検証もサポートしています。

詳細については、第 23 章「XML の検査」、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。

次のテストは、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、このテストは 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。

▶ **応答していないアプリケーションを強制終了するにはどうすればよいですか。**

次のいずれかのステップをテストに追加することにより、QuickTest でのテストの実行中に任意の標準のアプリケーションを強制終了できます。

- ▶ SystemUtil.CloseProcessByName "app.exe"
- ▶ SystemUtil.CloseProcessByWndTitle "Some Title"

▶ **実行セッション中にクリップボードを介したコピーや貼り付けはできますか。**

Clipboard オブジェクトを使用して、QuickTest の実行セッション中にテキストのコピー、切り取り、貼り付けを実行できます。

Clipboard オブジェクトでは、次のように、Visual Basic で使用できる Clipboard オブジェクトと同じメソッドをサポートしています。

- ▶ Clear
- ▶ GetData
- ▶ GetText
- ▶ SetData
- ▶ SetText

これらのメソッドの詳細については、<http://msdn.microsoft.com/en-us/library/ms172962.aspx> を参照してください。

Clipboard オブジェクトの使用例を次に示します。

```
Set MyClipboard = CreateObject("Mercury.Clipboard")
MyClipboard.Clear
MyClipboard.SetText "TEST"
MsgBox MyClipboard.GetText
```

テストの保守

- ▶ **アプリケーションに変更が加えられた場合、テス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テストの保守方法は、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つの大きなテストを作成するのではなく、テストを小さなグループに分けて作成すべき主な理由の1つがこれです。

また、QuickTest のアクションを使用して、よりモジュール化され、より効果的なテストを設計できます。テストを機能ごとにいくつかのアクションに分割します。アプリケーションに変更が加えられたら、特定のアクションだけを変更するだけでよく、テストのほかの部分は変更せずに済みます。可能なかぎり、複数のテストにまったく同じスクリプトを作成するのではなく、再利用可能なアクションの呼び出しを挿入するようにします。こうすることで、元の再利用可能なアクションに変更を加えるだけで、そのアクションを呼び出すすべてのテストに変更が自動的に適用されます。詳細については、第 16 章「高度なアクション機能を使用した作業」を参照してください。

同じテスト・オブジェクトが含まれるテストやアクション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、チェックポイントまたは ActiveScreen 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、1111 ページ「更新モード・オプションを使ったテストの更新」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・モードでテストを実行すると、QuickTest によってテストが実行された後、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、1094 ページ「メンテナンス実行ウィザードを使ったテストの実行」を参照してください。

▶ **テストの記録終了後、ActiveScreen 情報を増やしたり減らしたりできますか。**

記録後に ActiveScreen に保存された情報が、テストを編集するには不十分な場合や、ActiveScreen 情報が不要になり、テストのサイズを小さくする場合に、テストに保存されている ActiveScreen 情報の量を変更する方法はいくつかあります。

▶ テストによって使用されるディスク容量を減らすには、**[名前を付けて保存]** を選択して **[ActiveScreen ファイルを保存する]** チェック・ボックスをクリアすることで ActiveScreen 情報を削除します。詳細については、330 ページ「テストの保存」を参照してください。

▶ Windows または Java アプリケーションの記録時に ActiveScreen の情報をすべて保存しないことを選択した場合、次の方法のどれかを使用して、ActiveScreen に格納される情報を増やすことができます。

[オプション] ダイアログ・ボックスの [ActiveScreen] 表示枠で、ActiveScreen のキャプチャ設定が必要な量の情報をキャプチャするように設定されているか確認します。次に以下のことを行います。

▶ **[更新モード]** を実行し、既存のすべてのステップについて必要な量の情報を ActiveScreen に保存します。**[更新モード]** オプションの詳細については、1111 ページ「更新モード・オプションを使ったテストの更新」を参照してください。

▶ ActiveScreen に追加するオブジェクトが含まれるステップを再記録します。ステップを再記録するには、記録するステップの前のステップを選択し、テストで選択された位置と一致するようにアプリケーションを配置してから、記録を開始します。あるいは、追加するステップの前のステップでテストにブレークポイントを設定し、そのブレークポイントまでテストを実行します。これにより、ステップを記録する場所へ移動できます。ブレークポイントの設定の詳細については、1071 ページ「ブレークポイントの設定」を参照してください。

Windows アプリケーションで ActiveScreen に保存される情報量の変更に関する詳細については、1229 ページ「ActiveScreen オプションの設定」を参照してください。

▶ **古いテス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。

ローカライズされたアプリケーションのテスト

▶ **あるアプリケーションについて、ローカライズされたいくつかのバージョンをテストしています。それぞれのバージョンには、ローカライズされたユーザ・インタフェース文字列が含まれています。QuickTest で効率的なテストを作成するには、どのようにすればよいですか。**

これらのユーザ・インタフェース文字列は、グローバル環境変数リストにあるパラメータを使ってパラメータ化できます。グローバル環境変数リストは、変数と、それに対応する値のリストで、任意のテストからアクセスできます。詳細については、第 24 章「値のパラメータ化」を参照してください。

▶ **あるアプリケーションについて、ローカライズされたいくつかのバージョンをテストしています。テストの際に、アプリケーションの言語に応じて異なるデータを効率的に入力するにはどうすればよいですか。**

テストの反復を 1 回だけ実行する場合、あるいはアクションまたはテストのすべての反復で変数の値を変えない場合は、環境変数を使用し、テストの実行ごとにアクティブな環境変数ファイルを切り替えます。

テストまたはアクションの反復を複数回実行し、入力データを反復ごとに変更する場合は、アプリケーションのローカライズ・バージョンごとに外部データ・テーブルを作成します。テスト対象を別のローカライズされたバージョンに変更するときには、[テストの設定] ダイアログ・ボックスの [リソース] 表示枠でテスト用のデータ・テーブル・ファイルを別のデータ・テーブル・ファイルに切り替えます。データ・テーブルを使った作業の詳細については、第 42 章「データ・テーブルを使った作業」を参照してください。テストのデータ・テーブル・ファイルの選択に関する詳細については、1263 ページ「テストのためのリソース設定の定義」を参照してください。

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、どうすればよいですか。

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。
- ▶ テスト内のアクションの数を最小限に抑えます。理想的には、テストに含めるアクションの数は数十個程度にします。
- ▶ 関数を再利用可能なアクションとして格納するのではなく関数ライブラリに格納します。
- ▶ テストを「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テストを実行するため、テス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、1243 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。
- ▶ テストを編集する際に ActiveScreen を使用していない場合は、テストの編集中に ActiveScreen を非表示にしておくことで、編集時の応答時間を改善します。これを行うには、[表示] > [ActiveScreen] を選択するか、または [ActiveScreen] ツールバー・ボタンをクリックして、ActiveScreen を非表示に切り替えます。詳細については、第 2 章「QuickTest の概要」を参照してください。

- ▶ 情報をキャプチャして ActiveScreen に保存する条件とその量を指定します。キャプチャする情報が多いほど、さまざまな ActiveScreen オプションを使用してテストにステップを追加するのが容易になります。しかし、キャプチャする情報が多いと、記録や編集を行うのに時間がかかるようになります。パフォーマンスを向上させるために、次の ActiveScreen オプションを選択できます。
- ▶ Windows アプリケーションをテストしている場合は、あらゆるステップの ActiveScreen 情報をすべて保存する、特定のステップの ActiveScreen 情報だけを保存する、ActiveScreen のキャプチャを完全に無効化する、といった選択ができます。この設定は、[オプション] ダイアログ・ボックスの [ActiveScreen] 表示枠で行います。詳細については、1229 ページ「ActiveScreen オプションの設定」を参照してください。
- ▶ Web アプリケーションをテストしている場合は、ActiveScreen でのすべてのステップの画面キャプチャを無効にします。[オプション] ダイアログ・ボックスの [ActiveScreen] 表示枠で、[ユーザ定義レベル] をクリックして [ActiveScreen キャプチャのユーザ定義設定] ダイアログ・ボックスを開きます。

次に、[ActiveScreen のキャプチャを無効にする] オプションを選択します。これで記録時間を短縮できます。[オプション] ダイアログ・ボックスの [ActiveScreen] 表示枠の詳細については、1229 ページ「ActiveScreen オプションの設定」を参照してください。
- ▶ 新規テストを保存する場合、または [名前を付けて保存] を使用してテストを新しい名前で保存する場合は、[上書き保存] または [名前を付けて保存] ダイアログ・ボックスの [ActiveScreen ファイルを保存する] オプションをクリアにすることで、テストでキャプチャされた ActiveScreen ファイルを保存しないように選択します。これは、テストの設計が完了し、テスト実行のためだけにテストを使用する場合に特に便利です。ActiveScreen ファイルのないテストは、速く開き、占有するディスク容量が極めて少なくなります。[オプション] ダイアログ・ボックスの [ActiveScreen] 表示枠の詳細については、1229 ページ「ActiveScreen オプションの設定」を参照してください。

ヒント : ActiveScreen ファイルなしでテストを保存した後で ActiveScreen ファイルを回復する必要が生じた場合は、必要なステップを再び記録するか、**[更新モード]** オプションを使用してテストのすべてのステップの画面を再キャプチャします。詳細については、1111 ページ「更新モード・オプションを使ったテストの更新」を参照してください。

- ▶ テスト結果として、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、1246 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、テス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、953 ページ「テスト全体の実行」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テスト結果や古いテスト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、1000 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。

QuickTest が使用するディスク容量を減らすには、どうすればよいですか。

QuickTest が使用するディスク容量を減らすには、次のいずれかを行います。

- ▶ 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、1246 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。

- ▶ 情報をキャプチャして ActiveScreen に保存する条件とその量を指定します。キャプチャする情報が多いほど、さまざまな ActiveScreen オプションを使用してテストにステップを追加するのが容易になります。しかし、キャプチャする情報が多いと、記録や編集を行うのに時間がかかるようになります。パフォーマンスを向上させるために、次の ActiveScreen オプションを選択できます。
- ▶ Windows アプリケーションをテストしている場合は、あらゆるステップの ActiveScreen 情報をすべて保存する、特定のステップの ActiveScreen 情報だけを保存する、ActiveScreen のキャプチャを完全に無効化する、といった選択ができます。この設定は、[オプション] ダイアログ・ボックスの [ActiveScreen] 表示枠で行います。詳細については、1229 ページ「ActiveScreen オプションの設定」を参照してください。
- ▶ Web アプリケーションをテストしている場合は、ActiveScreen でのすべてのステップの画面キャプチャを無効にします。[ActiveScreen] 表示枠で、[ユーザ定義レベル] をクリックして [ActiveScreen キャプチャのユーザ定義設定] ダイアログ・ボックスを開きます。次に、**[ActiveScreen のキャプチャを無効にする]** オプションを選択します。これで記録時間を短縮できます。[オプション] ダイアログ・ボックスの [ActiveScreen] 表示枠の詳細については、1229 ページ「ActiveScreen オプションの設定」を参照してください。

- ▶ 新規テストを保存する場合、または [名前を付けて保存] を使用してテストを新しい名前で保存する場合は、[上書き保存] または [名前を付けて保存] ダイアログ・ボックスの **[ActiveScreen ファイルを保存する]** オプションをクリアにすることで、テストでキャプチャされた ActiveScreen ファイルを保存しないように選択します。これは、テストの設計が完了し、テスト実行のためだけにテストを使用する場合に特に便利です。ActiveScreen ファイルのないテストは、使用するディスク容量が極めて少なくなります。

ヒント： ActiveScreen ファイルなしでテストを保存した後で ActiveScreen ファイルを回復する必要が生じた場合は、必要なステップを再び記録するか、**[更新モード]** オプションを使用してテストのすべてのステップの画面を再キャプチャします。詳細については、1111 ページ「更新モード・オプションを使ったテストの更新」を参照してください。

推奨されるテストの長さがありますか。

テストの長さについて公式の制限はありませんが、テストをアクションに分割し、可能な限り、再利用可能なアクションを使用することを推奨します。アクションに含むステップ数は、200 ～ 300 までにしてください。数十までが理想的です。詳細については、第 15 章「アクションを使った作業」を参照してください。

付録 C

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 (1555 ページ)
- ▶ パッケージ設定ファイルについて (1556 ページ)
- ▶ データ・ファイルの作成 (1558 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 (1559 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 43 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>
```



```
</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 D

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、524 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（1562 ページ）
- ▶ カスタム・ビットマップ・コンペアラの開発（1565 ページ）
- ▶ チュートリアル：カスタム・コンペアラの作成（1574 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（1585 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、535 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、1563 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テストとビットマップ・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発見されましたが、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（1570 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、1566 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、1568 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、1574 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、1585 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。

- ▶ 1566 ページ「入力の受け取りとビットマップの比較」
- ▶ 1566 ページ「比較結果を QuickTest に提供」
- ▶ 1567 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（1570 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（1570 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、535 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 (1572 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テス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、1573 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。

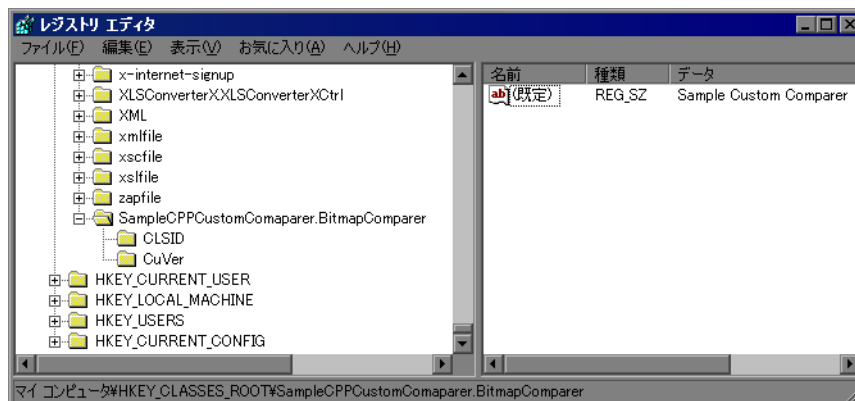
たとえば、次のような場合です。

- ▶ 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 (1582 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、1585 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・コンペアラ的设计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、1027 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、530 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、1566 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、1567 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、1585 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・コンペアラ的设计が完了したら、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・コンペアラを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、1569 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** が表示されません。
- 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、1585 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あるドキュメント・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 19 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，比較の後にカスタム・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

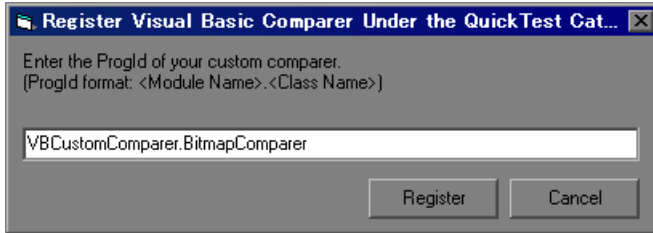
カスタム・コンペアラの DLL をビルドしたら、コンピュータにインストールするために、**regsvr32.exe** プログラムで実行します。

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、1568 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、1569 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

この索引は、『HP QuickTest Professional ユーザーズ・ガイド』の第1巻と第2巻両方の項目を含みます。

A

ActionIteration, 環境変数 658
Active Server Page テクノロジ 1544
ActiveScreen 380
 更新 383
 標準のキャプチャ設定 1233
 ファイルの保存と削除 330, 332
 保存される情報を増やす/減らす 1548
ActiveScreen キャプチャのユーザ設定ダイアログ・ボックス 1233
 Web オプション 1242
 一般オプション 1236
 キャプチャのレベル・オプション 1237
ActiveScreen のキャプチャ設定 1233
API, Windows の使用 891
API ベースのテキスト認識 749
Application Management, QuickTest との統合 1511
ASCII 1190
ASP ファイル 1544
attribute/ <プロパティ名>表記法 890

B

Business Process Monitor, QuickTest との統合 1511
Business Process Testing 1489
 役割 1490
 ワークフロー 1493
Business Process Testing でのワークフロー 1493
Business Process Testing の役割 1490

C

CGI スクリプト 1543
Close メソッド 877
COM 1544

CompareBitmaps メソッド 1570
ControllerHostName, 環境変数 658
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 119

D

Debug from Step 1068
Dictionary オブジェクト 491
Dim ステートメント
 エキスパート・ビューおよび関数ライブラリ 859
Do...Loop ステートメント
 エキスパート・ビューおよび関数ライブラリ 882
Domain コマンド・ライン・オプション 1004
DOS コマンド, テスト内で実行 891

E

Excel, Microsoft Excel を参照
Excel の数式
 値をパラメータ化するための 1205
 チェックポイントでの 1206
 データ・テーブルでの 1204

ExecuteFile 関数 947
ExecuteFile ステートメント 921
Exist ステートメント 824
Exit プロパティ 1541
eXtensible Markup Language (XML) 1544

F

FAQ 1537
For...Each ステートメント
 エキスパート・ビューおよび関数ライブラリ 882

索引

For...Next ステートメント

エキスパート・ビューおよび関数ライ
ブラリ 881

FromDate コマンド・ライン・オプション 1004

G

GetDefaultConfigurationString メソッド 1570

GetHelpFilename メソッド 1570

GetROProperty メソッド 888

GroupName, 環境変数 658

H

HP Application Management, QuickTest との統
合 1511

HP Micro Player 993

HP Quality Center, Quality Center を参照

HP ソフトウェアの Web サイト xxv

I

IBitmapCompareConfiguration インタフェース 1570

If...Then...Else ステートメント

エキスパート・ビューおよび関数ライ
ブラリ 884

IntelliSense 837, 900

IVerifyBitmap インタフェース 1570

J

JavaScript 1390

Java アプリケーション

キャプチャのレベル・オプション 1237

L

LoadRunner, QuickTest との統合 1511

LocalHostName, 環境変数 658

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 1004

M

Mercury Tours, サンプル・アプリケーション 16

META タグ 1543

Microsoft Excel 1190, 1204

Microsoft Query

データベース・チェックポイントのた
めのデータベースの選択 587, 1204

Microsoft の Visual Basic スクリプト言語 12

MinSize コマンド・ライン・オプション 1005

N

Name コマンド・ライン・オプション 1005

O

Object プロパティ, 実行環境オブジェクト・
メソッド 890

OCR 749

ODBC, データベース・チェックポイントの
ためのデータベースの選択 1204

Option Explicit ステートメント 945

Oracle アプリケーション

キャプチャのレベル・オプション 1240

OSVersion, 環境変数 658

OS, 環境変数 658

output.txt ログ・ファイル 1526

P

Parameter 予約済みオブジェクト 1271

Password コマンド・ライン・オプション 1006

PathFinder.Locate, ステートメント 1229

Print ユーティリティ・ステートメント 817

ProductDir, 環境変数 658

ProductName, 環境変数 658

ProductVer, 環境変数 658

Project コマンド・ライン・オプション 1006

Q

QA エンジニア, 自動化エンジニアを参照

QCUtil オブジェクト 1404

Quality Center 1397

Connectivity Add-in 1404

QuickTest からの接続 1399

QuickTest での使用 13

QuickTest との統合 1404, 1443

依存関係タブ 1437

環境変数ファイル 657

関連付けられている関数ライブラリ 920

切断 1403

相対パス 1432

テストの保存 332

テスト・プロセスの管理 13

テストを開く 1407

データ・テーブル 1200

- バージョン・コントロール 1461
 - バージョン・コントロール管理 1462
 - 不具合の報告
 - 自動 1011
 - 手動 1009
 - プロジェクトへのテストの保存 1406
 - ライブラリ・タブ 1435
 - リモートでの QuickTest テストの実行 1422
 - 履歴タブ 1436
 - Quality Center 9.x 1477
 - バージョン・コントロール 1478
 - Quality Center OTA 1404
 - Quality Center からの切断 1403
 - Quality Center からテストを開くダイアログ・ボックス 1478, 1408
 - Quality Center への QuickTest の接続 1399
 - Quality Center への接続 - サーバへの接続ダイアログ・ボックス 1400
 - QuickTest
 - HP のアプリケーション管理製品およびパフォーマンス・テスト製品との統合 1511
 - アクセス権, 必要 15
 - 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 1387
 - 開始 19, 20
 - 製品情報 75
 - 説明 3
 - ソフトウェアの更新 16
 - レイアウト 1123
 - カスタマイズ 1123
 - QuickTest Print Log ウィンドウ 817
 - QuickTest Professional Asset Upgrade Tool for Quality Center 329, 337, 1365, 1407
 - QuickTest Professional のバージョン情報ウィンドウ 75
 - QuickTest アセット・ビューア 1456
 - QuickTest ウィンドウ
 - アクション・ツールバー 23, 46
 - オートメーション・ツールバー 44
 - 欠落リソース表示枠 38
 - 使用可能なキーワード表示枠 34
 - 情報表示枠 37
 - ステータス・バー 25
 - 挿入ツールバー 45
 - タイトル・バー 25
 - タスク表示枠 42
 - タブの移動 1124
 - ツール・ツールバー 45
 - テーマ 27
 - テスト・フロー表示枠 41
 - デバッグ・ツールバー 23
 - デバッグ・ビューア表示枠 35
 - データ・テーブル 25
 - 表示ツールバー 46
 - 標準ツールバー 44
 - 標準のレイアウトの復元 1131
 - 表示枠の移動 1124
 - 表示枠の自動非表示 1129
 - ファイル・ツールバー 25
 - 複数のドキュメント 1146
 - プロセス・ガイダンス表示枠 39
 - 編集ツールバー 45
 - メニュー・バー 23
 - リソース表示枠 40
 - ルック・アンド・フィール 27
 - レイアウトのカスタマイズ 1123
 - QuickTest オートメーション・リファレンス 1394
 - QuickTest テストを開くダイアログ・ボックス 328, 330
 - QuickTest 内のステップへ移動, テスト結果ウィンドウから 984
 - QuickTest のアセット・バージョン 1462
 - QuickTest の開始 20
 - QuickTest のテスト・バージョン 1462
 - QuickTest ヘリモート・アクセス 1422
- R
- Recursive コマンド・ライン・オプション 1006
 - RegisterUserFunc ステートメント 939
 - ResultDir, 環境変数 658
 - RGB の許容範囲, ビットマップ・チェックポイント 524
- S
- SAP Gui for Windows アプリケーションキャプチャのレベル・オプション 1238
 - ScenarioId, 環境変数 659
 - Server コマンド・ライン・オプション 1007
 - Setting オブジェクト 1290
 - SetTOProperty メソッド 209

索引

Set ステートメント

エキスパート・ビューおよび関数ライ
ブラリ 858

SGML 1544

Silent コマンド・ライン・オプション 1007

SQL ステートメント画面, データベース・
チェックポイントの作成 588

SystemTempDir, 環境変数 659

SystemUtil.Run メソッド 877

T

TestDirector, Quality Center *を参照*

TestDirector, Quality Center *を参照*

TestDir, 環境変数 659

TestIteration, 環境変数 659

TestName, 環境変数 659

Test コマンド・ライン・オプション 1007

TSL 関数の呼び出し, QuickTest からの 1505

U

Unicode 4

UnregisterUserFunc ステートメント 929, 939, 941

UntilDate コマンド・ライン・オプション 1008

Update Run ダイアログ・ボックス 1115

UpdatingActiveScreen, 環境変数 659

UpdatingCheckpoints, 環境変数 659

UserName, 環境変数 659

User コマンド・ライン・オプション 1008

V

VALUE 関数, データ・テーブル・ワークシ
ートの 1204

VBScript 1390

関連付けられている関数ライブラリ
Quality Center 920

構文 856

構文エラー 862

構文を自動的に拡張 845, 901

テキストの書式設定 861

ドキュメント 878

VBScript 構文を自動的に拡張する 845, 901

Visual Basic 1390

Visual C++ 1390

Visual Studio.NET 1390

UserId, 環境変数 659

W

W3C の「Web Content Accessibility Guidelines」6

WaitProperty ステートメント 820

Wait ステートメント 824

Web

Web オブジェクトへのキーボード・コ
マンドの送信 1545

Web Web

高度な問題, FAQ 1543

Web コンテンツ・アクセシビリティ・チェッ
クポイント

テスト結果の 1042

Web コンテンツ, 動的な 1541

While ステートメント, エキスパート・ビュー
および関数ライブラリにおける 883

Windows API 891

Windows アプリケーション

キャプチャのレベル・オプション 1240

Windows コマンド・ライン・オプション 1003

Windows ダイアログ・ボックス 1146

WinRunner

QuickTest からの TSL 関数の呼び出し 1505

QuickTest からのテストの呼び出し 1502

関数の引数, QuickTest からのパラメー
タの引き渡し 1508

作業 1501

テスト, QuickTest からのパラメータの
引き渡し 1504

テスト結果内の WinRunner ステップの
表示 1013

WinRunner 関数の呼び出しダイアログ・ボッ
クス 1506

WinRunner テストの呼び出しダイアログ・
ボックス 1502

With ステートメント

"With" ステートメント作成結果ウイン
ドウ 813

エキスパート・ビュー 809

既存のアクションのために生成 812

削除 814

自動生成, 記録中 811

手作業で入力 885

wscript.exe 1391

X

XML

- XML をテキストとして編集ダイアログ・ボックス 621
- オブジェクトとメソッド 631
- オブジェクト・リポジトリからエクスポート 247
- オブジェクト・リポジトリとしてインポート 246
- 出力値の結果
 - 属性の詳細 1053
 - 分析 1049
- チェックポイント 601
 - Web ページ/フレーム 604
 - XPath 631
 - 結果の分析 630, 1031
 - スキーマ妥当性チェック・ダイアログ・ボックス 626
 - スキーマの追加ダイアログ・ボックス 629
 - スキーマの編集ダイアログ・ボックス 629
 - テスト・オブジェクト用 612
 - 名前空間 603, 631, 727
 - ファイル用 609
 - 変更 630
- チェックポイントの結果
 - 属性の詳細 1034
 - チェックポイント・サマリ 1033
- XML 構造
 - インポート 622, 740
 - 更新 622, 740
 - 実行の更新モードを使った更新 622, 740
- XML 出力値 680
- XML 出力値結果ウィンドウ 1051
- XML 出力のプロパティ・ダイアログ・ボックス 736
- XML チェックポイント結果ウィンドウ 1032
- XML チェックポイントの結果
 - 要素の値ダイアログ・ボックス 1041
- XML チェックポイントのプロパティ・ダイアログ・ボックス 616
- XML チェックポイント (ファイルから)・ダイアログ・ボックス 609
- XML 値, 出力 727
- XML をテキストとして編集ダイアログ・ボックス, XML チェックポイント 621

Z

Zip ファイル

- テストのインポート 338
- テストのエクスポート 338

あ

アイコン

- 大きいまたは小さい表示 1143

アクション 429, 469

- 欠落の呼び出し 1172
- 値, 「アクションの値, 共有」を参照
- 値の共有 490

Dictionary オブジェクトの使用 491

環境変数の使用 491

グローバル・データ・テーブルの使用 490

外部 432

概要 430, 469

キーワード・ビューへの追加 396

基本構文を使用した呼び出し 492

欠落に対する呼び出しの削除 1176

欠落に対する呼び出しの割り当て 1173

構文 492

再利用可能な 432

再利用不可能な 432

作業についてのガイドライン 443

削除 464

作成 441

ステップからの実行 954

挿入

既存 470

コピー 471

呼び出し 474

ダイアグラム 430, 469, 470

テスト・フロー 439

テスト・フロー表示枠 435

テンプレート 466

名前の変更 461

ネスト 458, 480

パラメータ, 「アクション・パラメータ」を参照

パラメータ化データ, 格納場所 456

パラメータの構文 493

パラメータの設定 450

複数, テストの 431

プロパティ 437

プロパティの設定 445

- 分割 459
- 戻り値の格納の構文 494
- アクションから実行 438
- アクションからデバッグ 438
- アクション間でのデータの引渡し 433
- アクション・タブ, データ・テーブル 433
- アクション・ツールバー, キーワード・ビュー 46, 439
- アクション・データ・シート 433, 1188
- アクションの値, 共有
 - Dictionary オブジェクトの使用 491
 - 環境変数の使用 491
 - グローバル・データ・テーブルの使用 490
- アクションの選択ダイアログ・ボックス 472, 475
- アクションのネスト 458, 480
- アクションの分割 459
- アクションの分割ダイアログ・ボックス 460
- アクションの呼び出し
 - 欠落 1169
 - 実行プロパティ 485
 - パラメータ値 487
 - 反復 485
 - プロパティ 485
- アクション・パラメータ 464, 480, 634, 643
 - オプションの設定 644
 - ガイドライン 483
 - 出力値の格納 681, 692
- アクションまで実行 438
- アクション・リスト 439
- アクセス権
 - Quality Center に必要 15
 - QuickTest の実行に必要 15
- アセット
 - 定義 1430
 - バージョン・コントロールからのチェックアウト 1465
 - バージョン・コントロールへのチェックイン 1468
 - バージョン・コントロールへの追加 1465
- アセットのアップグレード 1365, 1407, 329, 337
- アセット比較ツール 1446
 - 色設定ダイアログ・ボックス 1455
 - オプション 1449
 - ショートカット・メニュー・コマンド 1454
 - 凡例 1453
 - 開く 1447
- アセット・ビューア 1456
 - 長いドキュメント 1460
 - 開く 1456
- 値
 - オブジェクトのプロパティの指定 165
 - オブジェクトのプロパティ値の表示 199
 - オブジェクト・プロパティの標準設定の復元 167, 170
 - 出力 677
 - 設定 763
 - 入力 408
 - パラメータ化 633
- 値カラム, キーワード・ビュー 392
- 値設定オプション・ダイアログ・ボックス 638, 767
- 値セル 408
- 値の設定 763
- 値の設定領域 764
- 値の動的なリスト, メソッド引数 840
- 値のリスト, メソッド引数 840
- 値のリスト, メソッド引数に使用可能な値, メソッド引数 392, 789
- アドイン
 - Quality Center 内の QuickTest テストとの関連付け 1412
 - QuickTest 5
 - テストへの関連付け 1256
- アナログ記録 372, 375
- アプリケーション
 - QuickTest から起動 1141
 - 実行 877
 - 閉じる 877
 - ローカライズされたバージョンのテスト 1549
- アプリケーション, サンプル 16
- アプリケーションのクラッシュ・トリガ 1326
- アプリケーションの静止画像, キャプチャと表示 990
- アプリケーション・プロセスの終了 1338, 1341
- アプリケーション領域
 - 回復シナリオ, 削除 1358
- 暗号化, パスワード 410
- い
- 依存関係
 - 使用グリッド 1438

- 使用者グリッド 1438
- 定義 1430
- 依存関係タブ, Quality Center 1437
 - 使用グリッド 1441
 - 使用者グリッド 1438
- 一次リポジトリ 252
- 一次リポジトリ表示枠 258
- 一般オプション 899
- 一般>テキスト認識表示枠 750
- 移動ダイアログ・ボックス 846
- 色
 - オブジェクト・リポジトリ結合ツールでの設定 269
 - オブジェクト・リポジトリ比較ツールでの設定 302
 - キーワード・ビューでの設定 422
- 色設定ダイアログ・ボックス, アセット比較ツール 1455
- 印刷
 - 関数ライブラリ 918
 - テスト 339
- 印刷ダイアログ・ボックス
 - テスト結果ウィンドウ 996
- 印刷プレビュー・ダイアログ・ボックス 994
- 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 117
 - プログラムの記述 873
- インポート
 - XML ファイルからオブジェクト・リポジトリ 246
 - zip ファイルからのテストのインポート 338
- え
- エージェント, リモート 1423
- エキスパート・ビュー 829, 1539
 - アクションの戻り値の構文 494
 - アクション・パラメータの構文 493
 - アプリケーションの実行 877
 - アプリケーションの終了 877
 - 一般カスタマイズ・オプション 899
- エレメントの強調表示 902
- 基本アクション構文 492
- 説明 831
- チェックポイント 834
- テキストの検索 850
- テキストの置換 852
- パラメータについて 835
- 見映えのカスタマイズ 897
- エクスポート
 - zip ファイルへのテストのエクスポート 338
 - オブジェクト・リポジトリから XML ファイルへ 247
 - 画面レコーダ・ムービー 993
 - 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 195
- エディタ・オプション・ダイアログ・ボックス 899
- エラー, VBScript の構文 862
- エラー時の動作, テストのオプション 1260
- お
- オートフィルのリスト・ダイアログ・ボックス 1198
- オートメーション
 - Application オブジェクト 1392
 - オブジェクト・モデル 1387
 - オブジェクト・リポジトリ 248
 - 開発環境 1390
 - 言語 1390
 - タイプ・ライブラリ 1390
 - 定義 1388
 - テスト用スクリプトの生成 1256
- オートメーション・エンジニア, Business Process Testing における役割 1491
- オートメーション・ツールバー, QuickTest ウィンドウ 44
- オブジェクト
 - オブジェクト・リポジトリからの削除 156
 - 識別 81
 - テスト・オブジェクトを参照。
 - ドラッグ・アンド・ドロップ 34, 1155
 - ナビゲートして学習の使用による追加 230
 - 認識 107
 - 表示操作 81
 - プロパティ, 実行環境 889
 - メソッド, 実行環境 889
- オブジェクト・スパイ 98, 101
- オブジェクトの構成画面 1310
- オブジェクトの状態トリガ 1326
- オブジェクトの選択画面 1330
- オブジェクトの選択ダイアログ・ボックス 403

- オブジェクトの認識
 - 自動スクリプトの生成 122
 - 標準設定の復元 121
- オブジェクトの認識ダイアログ・ボックス 109
- オブジェクトのプロパティ値
 - 指定または変更 165
 - 表示 199
 - 標準設定の復元 167, 170
- オブジェクトのプロパティと値の設定画面 1333
- オブジェクトの割り当てダイアログ・ボックス 133
- オブジェクト・フィルタの定義ダイアログ・ボックス 147
- オブジェクト・プロパティ, ネイティブ・メソッド 890
- オブジェクト・モデル
 - オートメーション 1387
 - 定義 1388
- オブジェクト・リポジトリ
 - XML からのインポート 246
 - XML へのエクスポート 247
 - アクションとの関連付け 451
 - 以前のバージョンから変換 222
 - オートメーションの使用による管理 248
 - オブジェクトの検索 161
 - オブジェクトのコピー, 貼り付け, および移動 153
 - オブジェクトの削除 156
 - オブジェクトの追加 138
 - 管理 212
 - 関連付けの管理 201
 - 共有 94
 - 欠落 1169
 - 作成 221
 - 選択 93
 - テストの設定 1263
 - 閉じる 225
 - 開く 222
 - 変更 228
 - 保存 223
 - 未割り当て 1180
 - ローカル 93
 - ローカル・オブジェクトのエクスポート 195
 - ローカル・オブジェクトのエクスポートと置換 195
- オブジェクト・リポジトリ・ウィンドウ 185
 - オブジェクトの詳細領域 192
 - オプション 190
 - テスト・オブジェクトの詳細 164
 - について 184
 - フィルタ・ツールバー 189
 - 編集ツールバー 187
 - ボタン 187
- オブジェクト・リポジトリ結合ツール 251
 - 一次リポジトリ表示枠 258
 - 色の設定 269
 - ウィンドウ 254
 - 解決方法のオプション表示枠 258
 - 矛盾の解決方法の設定 267
 - ターゲット・リポジトリのフィルタ処理 286
 - ターゲット・リポジトリ表示枠 256
 - 二次リポジトリ表示枠 258
 - ビューの変更 256
 - 矛盾 281
 - 矛盾の解決 284
- 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 141
- オブジェクト・リポジトリの種類 90
- オブジェクト・リポジトリ比較ツール 291
 - 色の設定 302
 - ウィンドウ 294
 - 相違点のタイプ 301
 - 統計情報 306
 - リポジトリの同期化 309
 - リポジトリのフィルタ処理 308
 - リポジトリ表示枠 295
- オブジェクト・リポジトリ・マネージャ 214
- オプション・ステップ 961
 - 設定 962
 - 標準設定 963
- オプション・ダイアログ・ボックス 1220
 - ActiveScreen 表示枠 1229
 - 一般>テキスト認識表示枠 750
 - 一般表示枠 1223
 - 実行>画面キャプチャ表示枠 1246
 - 実行表示枠 1243
 - スクリプトの生成オプション 1223, 1393
 - ノード 1220
 - フォルダ表示枠 1226
- オンライン・ドキュメント xxiii

オンライン・リソース xxv

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 258

ガイドライン

ユーザ定義関数 945

開発環境 1390

外部アクション

定義 432

データの格納場所 456

外部関数, スクリプトからの実行 947

回復後のテスト実行オプション画面 1344

回復後のテスト実行のオプション 1317

回復シナリオ 1317

欠落の削除 1183

欠落の特定 1181

コピー 1354

削除 1353

テストからの削除 1358

テストとの関連付け 1355

バージョンの比較 1446, 1447

ファイル 1321

プロパティの設定 1358

プロパティの表示 1352, 1358

変更 1353

保存 1349

無効化 1359

回復シナリオ・ウィザード 1324

オブジェクトの選択画面 1330

オブジェクトのプロパティと値の設定画面 1333

回復後のテスト実行オプション画面 1344

回復シナリオ・ウィザードの完了画面 1348

回復操作画面 1337, 1338

回復操作 - 関数の呼び出し画面 1343

回復操作 - プロセスの終了画面 1341

回復操作 - ボタン, またはキーを押す画面 1339

関数の呼び出し画面 1343

テスト実行エラー画面 1334

トリガ・イベントの選択画面 1326

名前と記述画面 1346

プロセスの終了画面 1341

プロセスの選択画面 1335

ボタン, またはキーを押す画面 1339

ポップアップ・ウィンドウの条件を指定画面 1328

回復シナリオ・ウィザードの完了画面 1348

回復シナリオ・マネージャ・ダイアログ・ボックス 1321

回復操作 1317

Microsoft Windows の再起動 1338

アプリケーション・プロセスの終了 1338

関数呼び出し 1338

キーボードまたはマウスの操作 1338

外部リソース

Quality Center にテストを保存 332

学習, オブジェクト 230

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, Business Process

Testing における役割 1490

カスタマイズ

ツールバーとメニュー 1133

カスタマイズ・ダイアログ・ボックス 1133

オプション・タブ 1143

コマンド・タブ 1136

ツール・タブ 1141

ツールバー・タブ 1139

カスタム・コンペアラ

インストール 1568

作成 1565

説明を付ける 1567

登録 1568

「ビットマップ・チェックポイントのカスタマイズ」を参照

仮想オブジェクト 1297

削除 1315

定義 1302

仮想オブジェクト・ウィザード 1303

仮想オブジェクト・マネージャ 1315

仮想オブジェクト・マネージャ・ダイアログ・ボックス 1301

画像チェックポイント

画像内容の比較 521

プロパティの値の編集 521

画像チェックポイント・プロパティ・ダイアログ・ボックス 518

画面レコーダ・タブ, テスト結果ウィンドウ 991

カラム, キーワード・ビューでの表示 420

索引

- 環境, テスト 5
 - 環境変数 1271
 - 組み込み 658, 1271
 - 出力値の格納 682, 695
 - ファイル, Quality Center での 657
 - 環境変数, ユーザ定義 1275
 - エクスポート 1277
 - 外部 654
 - 内部 653
 - 表示 1275
 - 変更 1275
 - 関数
 - コード
 - 仕上げ 937
 - 挿入 937
 - ユーザ定義 907
 - 関数コードのプレビュー 936
 - 関数定義 ジェネレータ 928
 - 関数コードのプレビュー 936
 - 関数に説明を付ける 934
 - 関数の定義 928
 - 関数の登録 929
 - 説明 924
 - 開く 926
 - 関数に説明を付ける 934
 - 関数の登録 929
 - 関数の引数, QuickTest から WinRunner へのパラメータの引き渡し 1508
 - 関数呼び出し
 - ドラッグ・アンド・ドロップ 34, 1155
 - 関数ライブラリ 907
 - 一般オプション 899
 - 一般オプションのカスタマイズ 899
 - エレメントの強調表示 902
 - 管理 909
 - 関連付けの変更 923
 - 関連付けられているものを使用した作業 920
 - 現在のものを関連付け 922
 - 作業 1381
 - 実行の一時停止 1070
 - スクリプト・エディタで 1381
 - スクリプト・エディタで閉じる 1386
 - スクリプト・エディタでの作成 1384
 - スクリプト・エディタでの編集 1384
 - スクリプト・エディタでの保存 1385
 - スクリプト・エディタで開く 1382
 - 説明 12, 30
 - テキストの検索 850
 - テキストの置換 852
 - テスト 910
 - テストの指定 1263
 - デバッグ 917, 1061
 - ナビゲーション 914
 - バージョンの比較 1446, 1447
 - 開く 911, 919
 - プロパティ 1373
 - 編集 916
 - 保存 912
 - 見映えのカスタマイズ 897
 - 読み取り専用, 編集 917
 - 関連 ID, 使用グリッド 1441
 - 関連詳細, 使用グリッド 1442
 - 関連タイプ, 使用グリッド 1441
 - 関連付け
 - Quality Center で作成したテストとアドイン 1414
 - オブジェクト・リポジトリとアクション 451
 - 関数ライブラリ 920, 922, 923
 - 共有オブジェクト・リポジトリ 201
 - テストへのアドイン 1256
 - 関連名, 使用グリッド 1442
- ## き
- キー・カラム 553, 598
 - キーの割り当て
 - エキスパート・ビューでの 903
 - 関数ライブラリでの 903
 - キーボード・コマンド, Web オブジェクトへの送信 1545
 - キーボード・ショートカット
 - エキスパート・ビューでの 903
 - 関数ライブラリでの 903
 - キーワード・ビューでの 419
 - キーワード駆動型のテスト
 - QuickTest の設定 353
 - アプリケーションの分析 348
 - オートメーション・インフラストラクチャ 348
 - オブジェクト・リポジトリの設定 350
 - 概要 342

- 関数ライブラリの作成 352
- テスト・ステップの作成 354
- テストの作成 354
- テストの実行 356
- テストのトラブルシューティング 356
- 方法 348
- キーワード・ビュー 27, 387, 389
 - カラム, 説明 391
 - カラム, 表示 420
 - キーボードのキー 419
 - ステップ, 削除 418
 - ステップ, 追加 396
 - ステップ, ブロックの後への追加 414
 - ステップ, 変更 415
 - 表示オプション 420
 - フォントと色 422
- 技術情報 xxv
- 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 85
 - テスト・オブジェクトを参照。
- 既存のアクション, 挿入 470
- 既存の出力値の追加ダイアログ・ボックス 746
- 既存のチェックポイントの追加ダイアログ・ボックス 502
- キャプチャのレベル・オプション 1237
 - Java アプリケーション 1237
 - Oracle アプリケーション 1240
 - SAP Gui for Windows アプリケーション 1238
 - Windows アプリケーション 1240
 - ターミナル・エミュレータ・アプリケーション 1241
- 矛盾の解決, オブジェクト・リポジトリ結合ツール 284
- 矛盾の解決方法
 - 結合したオブジェクト・リポジトリ 284
 - 設定, オブジェクト・リポジトリ結合ツール 267
- 行範囲の変更ダイアログ・ボックス 720
- 共有オブジェクト・リポジトリ 90, 94
 - アクションとの関連付け 451
 - 関連付けの管理 201
 - 結合 251
 - バージョンの比較 1446, 1447
 - 比較 291
 - 未割り当て 1180
 - ローカル・リポジトリからの更新 273
 - 共有オブジェクト・リポジトリ・ウィンドウ 219
 - 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 289
 - 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 204
- 記録
 - アナログ 372
 - 時間, 向上 1550
 - 低レベル 372, 1538
 - テスト 368
- く
 - クエリ・ファイル, データベース・チェックポイントの ODBC/Microsoft Query を使った作業 1204
 - 作成 587, 1204
 - クッキー 1543
 - 組み込み環境変数 658, 1271
 - グローバル・データ・シート 433, 1188
 - グローバル・データ・シートとアクション・データ・シート, 選択 433
 - グローバル・データ・テーブル・パラメータ 651
 - グローバル・テスト・オプション 1219
- け
 - 計算
 - エキスパート・ビュー 880
 - 関数ライブラリでの 880
 - 結果, 「実行結果」参照
 - 結果セット 584
 - 結果の詳細タブ, テスト結果ウィンドウ 972
 - 結果の場所タブ, 実行ダイアログ・ボックス 958
 - 結合
 - 共有オブジェクト・リポジトリ 251
 - ローカル・オブジェクト・リポジトリ 273
 - 欠落アクションの特定ダイアログ・ボックス 1173, 1176
 - 欠落リソース 1169
 - 欠落リソース表示枠 38
 - 説明 1169
 - フィルタ処理 1171
 - 未割り当ての共有オブジェクト・リポジトリ 1180
 - 未割り当てのリポジトリ・パラメータ 1184

索引

権限

Quality Center に必要 15
QuickTest の実行に必要 15

言語 1390

言語サポート, Unicode 4

検索ダイアログ・ボックス

エキスパート・ビュー 850
オブジェクト・リポジトリ結合ツール 287
オブジェクト・リポジトリ比較ツール 310
テスト結果 987

検索と置換ダイアログ・ボックス 157

こ

更新を確認 1223

構文

アクション 492
アクションの戻り値 494
アクション・パラメータ 493

構文エラー, VBScript 862

項目カラム, キーワード・ビュー 391

項目セル 399

項目, 選択

アプリケーションからの 403
共有オブジェクト・リポジトリからの 400
項目リストからの 400

項目リスト 400

コマンド

オブジェクト・リポジトリ結合ツール 261
オブジェクト・リポジトリ比較ツール 298
ツールバーまたはメニューに追加 1136

コマンド・ライン・オプション

Domain 1004
FromDate 1004
Log 1004
MinSize 1005
Name 1005
Password 1006
Project 1006
Recursive 1006
Server 1007
Silent 1007
Test 1007
UntilDate 1008
User 1008
を使用したテスト結果の削除 1003

コメント

エキスパート・ビュー 879
関数ライブラリ 879
キーワード・ビュー 818

コメント・カラム, キーワード・ビュー 393

コメント・タブ, タスク表示枠 1164

コレクション, 仮想オブジェクト 1297

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

実行 951
実行結果。実行結果を参照
ステップ

移動 416
管理 416
削除 418
追加 396

コンポーネントの実行 951

さ

サーバ

サーバ側の接続 1543
切断, Quality Center からの 1403
リダイレクト 1543

サーバのリダイレクト 1543

最初にお読みください xxiii

再利用可能なアクション 432

再利用不可能なアクション 432

サイレント・テスト・ランナー 1523

ダイアログ・ボックス 1523

削除

アクション 464
オブジェクト・リポジトリのオブジェ
クト 156
テスト結果 1000
ブレイクポイント 1072
リポジトリ・パラメータ 237

サマリ・カラム, キーワード・ビュー 393

サンプル・アプリケーション, Mercury Tours 16

し

シート・メニュー・コマンド, データ・テー
ブル 1195

時間, ユーザ定義書式の設定 1197

システム・カウンタ

設定 1284

- 有効化 1284
- システム・カウンタ, 結果 1054
- システム モニタ・タブ 1054
 - 結果のエクスポート 1054
- 実行オプション, オプション・ダイアログ・ボックスの 1243
- 実行>画面キャプチャ表示枠 1246
- 実行環境
 - オブジェクト 889
- 実行環境オブジェクト 81, 85
 - プロパティと操作の表示 98
- 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。 88
- 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 967
 - WinRunner ステップの表示 1013
 - 印刷 996
 - 印刷前のプレビュー 994
 - 検索 983, 987
 - コマンド・ライン・オプションによる削除 1003
 - 削除, テスト結果削除ツール 1000
 - 実行時データ・テーブル 1049
 - 出力値 1048
 - 手動による不具合の報告 1009
 - スキーマ 1015
 - チェックポイント 1022
 - テスト結果ウィンドウ 969
 - 特定の実行の表示 979
 - パラメータ化された値 1046
 - 表示のカスタマイズ 1015
 - ファイルにエクスポート 997
 - 不具合の自動報告 1011
 - メッセージの送信 815
 - 有効化とフィルタ処理 894
- 実行結果を分析。実行結果を参照
- 実行時
 - 設定, 追加と削除 1293
 - データ・テーブル 1186
- 実行時データ・テーブル 1049
- 実行セッション
 - 一時停止 1070
 - 回復シナリオの無効化 1359
 - テスト・オブジェクトを使った作業 208
 - 認識プロパティの変更 209
 - プログラミングによるテスト・オブジェクトの作成 209
- 実行セッションの一時停止 1070
- 実行セッションのムービー
 - Quality Center での結果の表示 988
 - エクスポート 993
 - キャプチャ 1246
 - キャプチャと表示 991
 - キャプチャのオプション設定 1246
 - テスト結果からの削除 992
- 実行ダイアログ・ボックス 953
- 実行プロパティ, アクションの呼び出しの設定 485
- 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 877
- 出力
 - XML 値 727
 - 値 677
 - データベース値 722
 - テキスト値 697, 699
 - プロパティ値 683
- 出力タイプ 690
 - アクション・パラメータ 692
 - 環境変数 695
 - データ・テーブル 693
 - テスト・パラメータ 692
- 出力値
 - QuickTest でサポート 1534
 - XML 727, 736
 - アクション・パラメータまたはテスト・パラメータへの格納 681
 - オブジェクトのプロパティ 687
 - 環境変数への格納 682
 - 既存の追加 745
 - 結果の表示 1048
 - 定義 677
 - データ・テーブルへの格納 681
 - データベース 722, 724, 726
 - テーブル 708, 712, 717
 - テキスト 697, 701
 - テキストの作成 699
 - テキスト領域 699
 - 表示 683
 - 標準 683
 - 編集 683

- 出力値カテゴリ
 - XML 出力値 680
 - データベース出力値 680
 - テキスト出力値 679
 - テキスト領域出力値 679
 - 標準出力値 679
- 出力パラメータのプロパティ・ダイアログ・ボックス 687
- 手動ステップ 415
- 手動テスト 425
- 使用可能なキーワード表示枠 34, 1155
- 使用グリッド 1438
- 条件ステートメント 801
 - キーワード・ビューでの使用 427
- 使用者グリッド 1438
- ショートカット・キー
 - QuickTest 47
 - キーワード・ビューでの 419
 - 表示 1143
- 情報表示枠 37
- ショートカット
 - QuickTest 47
 - エキスパート・ビューでの 903
 - オブジェクト・リポジトリ結合ツール 261
 - オブジェクト・リポジトリ比較ツール 298
 - 関数ライブラリでの 903
 - メニュー項目の 47
- 初期化スクリプト 1389
- 書式メニュー・コマンド, データ・テーブル 1197
- 序数識別子 115
 - テスト・オブジェクトの指定 180
- 所有者 ID, 使用者グリッド 1439
- 所有者詳細, 使用者グリッド 1440
- 所有者タイプ, 使用者グリッド 1439
- 所有者名, 使用者グリッド 1440
- 新規アクションの呼び出しを挿入ダイアログ・ボックス 441
- 新規結合ダイアログ・ボックス 271

- す
- 数式
 - 値をパラメータ化するための 1205
 - チェックポイントでの 1206
 - データ・テーブルでの 1204
- 数式, データ・テーブルでの 1204
- スキーマ, 実行結果の 1015
- スキーマ妥当性チェック・ダイアログ・ボックス, XML チェックポイント 626
- スキーマの追加ダイアログ・ボックス, XML チェックポイント 629
- スキーマの編集ダイアログ・ボックス, XML チェックポイント 629
- スクリプト・エディタ 1363
 - ウィンドウのカスタマイズ 1367
 - 関数ライブラリ 1381
 - テスト 1377
 - 表示領域 1375
 - フロー表示枠 1369
 - メイン・ウィンドウ 1366
 - リソース表示枠 1372
- スクリプト, テスト, テストを参照
- スクリプトの生成オプション 1393
- スタート・ページ 31
- ステータス・バー
 - QuickTest ウィンドウ 25
 - オブジェクト・リポジトリ結合ツール 259
 - オブジェクト・リポジトリ比較ツール 297
- ステートメント, キーワード・ビューでの使用 415
- ステートメントの自動補完 837, 900
- ステップ
 - 移動 416
 - オプション 961
 - キーワード・ビューからの削除 418
 - キーワード・ビューでの表示プロパティ 425
 - キーワード・ビューでの変更 415
 - キーワード・ビューへの追加 396
 - 削除 418
 - 手動 415
 - 挿入 783
 - 追加 396
 - ブロックの後への追加 414
- ステップから実行 954
- ステップ・コマンド 1064
- ステップ・ジェネレータ 781, 783
- ステップ・ジェネレータ・ダイアログ・ボックス 786
- ステップでオブジェクトを選択ダイアログ・ボックス 400
- ステップの移動 416
- ステップまで実行 1068

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 120, 122

情報の分析 1019

設定 123

テストの実行中に無効化する 1261

スマート認識プロパティ・ダイアログ・ボックス 128

せ

正規表現 769

エキスパート・ビューおよび関数ライブラリでの使用 855

関数ライブラリでの使用 855

チェックポイントでの使用 771

定義 772

定数 765

バックスラッシュ (¥) 774

プロパティ値 770

製品情報ウィンドウ 75

製品情報ボタン 75

設計時データ・テーブル 1185

セッション ID 1543

接続文字列, データベース・チェックポイントの指定 588

絶対パス 322

設定 437

設定タブ, データベース・チェックポイントのプロパティ・ダイアログ・ボックス 595

セルの識別タブ, データベース・チェックポイントのプロパティ・ダイアログ・ボックス 597

そ

相違点のタイプ

オブジェクト・リポジトリ比較ツール 301

操作

項目リストからの選択 399, 400

実行環境オブジェクト 88

ステップ用に選択 407

テスト・オブジェクト 88

ネイティブ 88

引数 408

操作カラム, キーワード・ビュー 392

操作セル 407

相対パス 322

Quality Center 1432

挿入ツールバー, QuickTest ウィンドウ 45

ソフトウェア・アップデート 16

た

ターゲット・リポジトリ 252

保存 289

ターゲット・リポジトリ表示枠 256

ターミナル・エミュレータ・アプリケーション

キャプチャのレベル・オプション 1241

タイトル・バー, QuickTest ウィンドウ 25

タイプ・ライブラリ 1390

タイムアウト

設定 1261

テキスト・チェックポイントの指定 577

標準チェックポイントの指定 517

タスク・エディタ・ダイアログ・ボックス 1167

タスク, 管理 42

タスク・タブ, タスク表示枠 1161

タスク表示枠 42, 1160

コメント・タブ 1164

タスク・タブ 1161

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オプション 1422

単一テキスト・ブロック・モード, テキスト認識 752

ち

チェックアウト・コマンド 1465

チェックアウト・コマンド, Quality Center 9.x 1479

チェックイン・コマンド 1465, 1468

チェックイン・コマンド, Quality Center 9.x 1479, 1481

チェックポイント

QuickTest でサポート 1532

XML 601

エキスパート・ビューでの 834

オブジェクト 512

画像 518

既存の追加 501

失敗 1091

新規追加 500

数式の使用 1206

説明 499

タイプ 505

- 定義 321, 499
- データベース 583
- テーブル 537, 538, 542
- テキスト 559, 560
- テキスト領域 563
- パラメータ化 667
- ビットマップ 523
- 標準, テキストの検査 579
- 変更 518, 521
- 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 514
 - データベースの検査 542
- 遅延, ステップの編集 837
- 遅延の入力, ステップの編集時 837
- 置換ダイアログ・ボックス
 - エキスパート・ビュー 852
 - 関数ライブラリ 852
- つ
- 通貨, ユーザ定義書式の設定 1197
- ツールチップ
 - 表示 1143
- ツール・ツールバー, QuickTest ウィンドウ 45
- ツールバー
 - QuickTest ウィンドウ
 - アクション 46
 - オートメーション 44
 - 挿入 45
 - ツール 45
 - デバッグ 23, 45
 - 表示 46
 - 標準 44
 - ファイル 25
 - 編集 45
 - オブジェクト・リポジトリ結合ツール 260
 - オブジェクト・リポジトリ比較ツール 298
 - 表示と非表示 1139
 - 標準設定 1139
 - ツールバーとメニュー
 - カスタマイズ 1133
 - ツールバー・ボタン
 - テキスト・ラベルの表示 1139
 - 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 1245
- 定数値オプション・ダイアログ・ボックス 766
- 定数値オプション・ボタン 766
- 定数値, 定義 763
- ディスク容量, 保存 1550
- 低レベル記録 372, 378, 1538
- データ駆動テスト 634, 681
- データ・シート
 - アクション 1188
 - グローバル 1188
 - グローバルとアクション, 選択 433
 - ローカル 1188
- データ・テーブル
 - 結果の表示 1049
 - 出力値の格納 681, 693
 - テーブル・カラム 648
 - テーブルの行 648
 - パラメータ, オプションの設定 649
- データ・テーブルの仕様 1192
- データ・テーブルのワークシートの関数 1204
- データ・テーブル・ワークシートの TEXT 関数 1204
- データ・ドライバ 670
- データベース
 - Microsoft Query/SQL ステートメントによるクエリの作成 587
 - ODBC/Microsoft Query でのクエリの作成 1204
 - SQL ステートメント画面 588
 - 結果セット 584
 - 接続文字列 588
 - チェックポイントの作成 584
 - 手作業での SQL ステートメントの定義 585
- データベース・クエリ・ウィザード 585
- データベース出力値 680, 722, 724
- データベース出力値のプロパティ・ダイアログ・ボックス 724
- データベース・チェックポイント 583
 - 値のタイプの指定 595
 - 一般情報 592
 - 期待データの指定 594
 - 結果の分析 1025
 - 説明 583
 - セルの識別の設定の指定 597
 - セルの指定 593

- 変更 599
- データベース・チェックポイントのプロパティ・ダイアログ・ボックス 589
- 期待データ・タブ 594
- 設定タブ 595
- セルの識別タブ 597
- データベースでの内容のプロパティの検査 584
- データ・メニュー・コマンド, データ・テーブル 1197
- テーブル出力値 712
 - 行の範囲の変更 720
 - 出力オプションの変更 720
 - テーブルの内容タブ 714
 - テーブルのプロパティ・タブ 714
- テーブル出力値のプロパティ・ダイアログ・ボックス 712
 - テーブルの内容タブ 712
 - プロパティ・タブ 718
- テーブル・チェックポイント
 - 値のタイプの指定 551
 - 一般オプション 545
 - 期待データの指定 550
 - 結果の分析 1025
 - 作成 538
 - 説明 537
 - セルの識別の設定の指定 552
 - セルの指定 548
 - テーブルの内容タブ 546
 - テーブルのプロパティ・タブ 546
 - 変更 556
- テーブル・チェックポイントのプロパティ・ダイアログ・ボックス 542
 - 期待データ・タブ 550
 - テーブルの内容タブ 544
 - プロパティ・タブ 554
- テーブルの内容タブ
 - テーブル出力値のプロパティ・ダイアログ・ボックス 712
 - テーブル・チェックポイントのプロパティ・ダイアログ・ボックス 544
- テーブル・プロパティ
 - 検査対象の指定 555
 - 出力するプロパティの選択 719
- テキスト, 検査
 - テキスト領域チェックポイントの使用 563
- テキスト出力値 679
- 作成 697
- 指定 701
- テキスト出力パラメータのプロパティ・ダイアログ・ボックス 701
- テキスト選択の設定ダイアログ・ボックス 570
- テキスト・チェックポイント 559, 560
 - 後のテキストの指定 576
 - オプションの設定 570
 - 結果の分析 1029
 - 検査対象テキストの指定 573
 - タイプ 559
 - タイムアウトの指定 577
 - テキスト選択の設定 570
 - 標準チェックポイント 579
 - 変更 578
 - 前のテキストの指定 574
- 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 565
- テキスト値, 出力 697, 699
- テキスト認識 749
 - ガイドライン 754
 - サポート環境 756
 - 使用事例シナリオ 758
 - 単一テキスト・ブロック・モード 752
 - 複数テキスト・ブロック・モード 753
- テキスト認識表示枠, オプションダイアログ・ボックス 750
- テキスト領域チェックポイントのプロパティ・ダイアログ・ボックス 565
- テキスト領域出力値 679
 - 作成 699
- テキスト領域出力値のプロパティ・ダイアログ・ボックス 701
- テスト
 - Quality Center プロジェクトへの保存 1406
 - Quality Center での管理 13, 1397
 - Quality Center でのテンプレート・テストを使用した作成 1416
 - Quality Center プロジェクトで開く 1407
 - Quality Center への接続 332
 - QuickTest での起動 328
 - アクションから実行 438
 - 圧縮 338
 - 印刷 339
 - 解凍 338

- 回復シナリオの関連付け 1355
- 回復シナリオの削除 1358
- 回復シナリオの無効化 1359
- 外部リソースと一緒に保存 332
- 拡張 321
- 管理 327
- 旧バージョンのテストを開く 329
- 記録 365, 368
- 更新 1111
- コンポーネントとの比較 1499
- 作業 1377
- 作成 315, 328, 341
- 実行 951
- 実行, オプション・ステップの使用 961
- 「実行結果」参照
- 実行の一時停止 1070
- スクリプト・エディタで閉じる 1381
- スクリプト・エディタでの編集 1379
- スクリプト・エディタでの保存 1380
- スクリプト・エディタで開く 1377
- ステップからの実行 954
- 設定 437
- ダイアグラム 430, 469, 470
- チェックポイント「チェックポイント」を参照。
- テスト・ステップについて 319
- デバッグ 1061
- 名前を付けて保存 332
- バージョン・コントロールからの
チェックアウト 1465
- バージョン・コントロールへのチェック
イン 1468
- バージョン・コントロールへの追加 1465
- バージョンの比較 1446
- バージョンの表示と比較 1443
- パラメータ化, 例 665
- プロパティ 1370, 1373
- 保守 1091
- 保存 330
- ローカル 432
- テスト・オブジェクト 81, 85
 - アプリケーション内での強調表示 160
 - 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 153
 - オブジェクト・リポジトリ内の検索
157, 161
- 管理 137
- 記述プロパティの削除 179
- 検索 157
- 識別 81
- 実行セッションでの 208
- 実行セッションでの作成 209
- 序数識別子の指定 180
- 新規の定義 150
- 新規プロパティの定義 177
- 選択
 - アプリケーションからの 403
 - 共有オブジェクト・リポジトリからの 400
 - 項目リストからの 400
- 追加
 - オブジェクト・リポジトリへの 138
 - 記述プロパティ 173
 - ドラッグ・アンド・ドロップ 184, 230
 - 名前の変更 171
 - プログラムの記述を使った作成 209
 - プロパティ値, 置換 157
 - プロパティ値の取得と設定 887
 - プロパティと操作の表示 98
 - プロパティの表示 199
- 変更
 - 実行セッション中のプロパティ 209
 - 実行セッションでの 209
 - 名前 171
 - プロパティ 164
 - ローカル・リポジトリへのコピー 197
- テスト・オブジェクトの選択
 - アプリケーションからの 403
 - 共有オブジェクト・リポジトリからの 400
 - 項目リストからの 400
- テスト・オブジェクトの操作 88
- テスト・オブジェクトのプロパティ 88
 - 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 88
- テスト・オプション
 - 個別のテストの設定 1251
 - 実行時 1293
 - 取得 1292
 - すべてのテストの設定 1219
 - 設定 1290
 - テスト実行中 1289

- 復元 1292
- テスト結果ウィンドウ 969
 - QuickTest 内のステップへ移動 984
 - 画面レコーダ・タブ 991
 - 結果の詳細タブ 972
 - システム モニタ・タブ 1054
 - 実行結果ツールバー 975
 - 実行結果ツリー 971
 - テーマ 977
 - 見映えのカスタマイズ 977
- テスト結果削除ツール 1000
 - コマンド・ラインからの実行 1003
- 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 975
- テスト結果ツリー 971
- テスト実行エラー画面 1334
- テスト実行エラー・トリガ 1326
- テスト実行時間, 向上 1550
- テスト実行ログ 1526
- テスト・セット 1420
- テスト・データベースの維持 1389
- テストの圧縮 338
- テストの解凍 338
- テストの実行 951
 - Quality Center プロジェクトからの 1419
 - Update Run ダイアログ・ボックス 1115
 - WinRunner テストの実行 1502
 - アクションから 438
 - オプション・ステップの使用 961
 - 期待結果の更新 1111
 - 結果の表示 978
 - 高度な問題 1537
 - 実行ダイアログ・ボックス 953
 - ステップから 954
- テストの設定ダイアログ・ボックス 1252
 - 回復表示枠 1279
 - 環境表示枠 1271
 - 実行表示枠 1259
 - スクリプトの生成オプション 1393
 - ノード 1252
 - パラメータ表示枠 1269
 - プロパティ表示枠 1255
 - リソース表示枠 1263
 - ローカル・システム・モニタ表示枠 1284
- テストの同期化 819
 - オブジェクトの表示を待機 824
 - タイムアウト値の変更 825
 - 同期化ポイント 820
 - 特定のプロパティ値の待機 820
- テストの複数のアクション 431
- テストの保守 1091
- テストの保存ダイアログ・ボックス 1406
- テスト・バッチ, 実行 963
- テスト・パラメータ 634, 643
 - オプションの設定 644
 - 出力値の格納 681, 692
 - ステップでの使用 1271
- テスト・フロー (アクション) 439
- テスト・フロー表示枠 41
 - アクション 41, 435
- テスト・プロセス 6
 - テスト結果の分析 11, 347
 - テストの作成 7, 9, 342, 345
 - テストの実行 10, 346
 - 不具合の報告 11, 347
- テスト・リソース, 欠落 1169
- テストを開くダイアログ・ボックス 1410
- テストをリソースと保存ダイアログ・ボックス 334
- デバッグ
 - Debug from Step 1068
 - アクションからデバッグ 438
 - アクションまで実行 438
 - 関数ライブラリ 917, 1061
 - コマンドの実行 1082
 - 式の確認 1074
 - 実行の一時停止 1070
 - ステップまで実行 1068
 - テスト 1061
 - テスト, 例 1086
 - ブレークポイント
 - 削除 1072
 - 設定 1071
 - 無効化と有効化 1071
 - 変数へのアクセス 1079
- デバッグ・ツールバー, QuickTest ウィンドウ 23, 45
- デバッグ・ビューア 35
- デバッグ・ビューア表示枠 35, 1073
 - ウォッチ・タブ 1074

コマンド・タブ 1082
 変数タブ 1079
 テンプレート, アクション 466
 テンプレート・テスト 1412, 1414
 データ・テーブル 25, 35, 1185
 Quality Center での使用 1200
 アクション・タブ 433
 アクション・データ・シート 1188
 オートフィルのリスト 1198
 グローバル・タブ 433
 個別テストの反復オプション 1260
 シート・メニュー・コマンド 1195
 実行時 1186
 仕様 1192
 書式メニュー・コマンド 1197
 数式の使用 1204
 スクリプティング関数, 使用 1208
 設計時 1185
 データ・シート 1188
 データのインポート, さまざまな形式
 での 1190
 データ・メニュー・コマンド 1197
 テーブルの編集 1190
 バージョンの比較 1446, 1447
 場所 1189
 ファイル・メニュー・コマンド 1194
 編集メニュー・コマンド 1195
 保存 1189
 メニュー・コマンドの使用 1193
 ローカル・データ・シート 1188
 ワークシートの関数 1204

 と
 同期化タイムアウト
 設定 1261
 同期化ポイント
 作成 820
 挿入 821
 同期化ポイントの追加ダイアログ・ボックス 821
 統計情報ダイアログ・ボックス 280
 比較ツール 306
 動的な Web コンテンツ 1541
 動的に生成される URL や Web ページ 1542
 ドキュメント, オンライン xxiii
 ドッキング表示枠 1129
 トラブルシューティングと技術情報 xxv

トランザクション 1518
 終了 1521
 挿入 1520
 測定 1518
 定義 1518
 トランザクション終了ボタン 45
 トランザクションの開始ダイアログ・ボック
 ス 1520
 トランザクションの終了ダイアログ・ボック
 ス 1521
 トリガ
 アプリケーションのクラッシュ 1326
 イベント 1317
 オブジェクト状態 1326
 テスト実行エラー 1326
 ポップアップ・ウィンドウ 1326
 トリガ・イベントの選択画面 1326
 ドキュメントのみのオプション 425

 な
 ナビゲートして学習オプション 230
 名前
 テスト・オブジェクトの変更 171
 名前と記述画面 1346

 に
 二次オブジェクト・リポジトリ 252
 二次リポジトリ表示枠 258
 入力パラメータタブ, 実行ダイアログ・ボッ
 クス 960
 認識プロパティ 81, 85
 表示 98
 認証
 Quality Center への接続 1399

ね
 ネイティブな操作 88
 表示 98
 ネイティブなプロパティ 88
 表示 98
 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 1220
 ノード, テストの設定ダイアログ・ボックス 1252

は

バージョン
 比較 1446
 表示と比較 1443
 バージョン管理 1462
 バージョン管理, Quality Center 9.x 1478
 バージョン・コントロール 1461, 1462
 Quality Center 9.x 1478
 アセットのチェックアウト 1465
 アセットの追加 1465
 コマンド 1464
 チェックアウトの取り消し 1469
 テストのチェックイン 1468
 バージョンの履歴 1470
 ベースラインの履歴 1473
 バージョンの比較 1446
 バージョンの履歴
 ベースラインの履歴との比較 1476
 バージョンの履歴ダイアログ・ボックス 1470
 パーセント, ユーザ定義書式の設定 1197
 場所プロパティ, オブジェクトの識別に使用 117
 バス, 絶対と相対 322
 パスワード, 暗号化 410
 パスワード・エンコーダ・ダイアログ・ボ
 クス 411
 バックスラッシュ (¥) 774
 パフォーマンス, 向上 1550
 パフォーマンス・テスト製品, QuickTest との
 統合 1511
 パラメータ
 WinRunner 関数への引渡し 1508
 WinRunner テストへの引渡し 1504
 アクション 464, 480, 645
 アクションのガイドライン 483
 アクションの設定 450
 アクションの呼び出しの構文 493
 エキスパート・ビュー 835
 親アクション 645
 環境変数, ユーザ定義 1273, 1275
 先行するアクション呼び出しの出力 645
 テスト 645
 テストの指定 1269
 未割り当てのオブジェクト・リポジト
 リの処理 1184
 リポジトリ 233
 管理 234

欠落 1169
 削除 237
 追加 235
 変更 237
 割り当て 204
 パラメータ・オプション・ダイアログ・ボッ
 クス 644
 パラメータ・オプション・ボタン 765
 パラメータ化
 値 633
 データ・ドライバの使用 670
 テスト, 例 665
 メソッド 636
 リポジトリ・パラメータを使用するプ
 ロパティ値 239
 パラメータ化アイコン 638, 640, 767
 パラメータ化された値, テスト結果の表示 1046
 パラメータ化の例 665
 パラメータ・タイプ 653
 アクション・パラメータ 634
 環境変数パラメータ 653
 タイプ 653
 データ・テーブル・パラメータ 647
 テスト・パラメータ 634
 乱数パラメータ 663
 パラメータ値
 アクションの呼び出し 487
 定義 763
 パラメータの定義, 標準設定 765, 768
 反復 485, 647
 個別テストのオプション 1260

ひ

比較
 共有オブジェクト・リポジトリ 291
 引数, 定義 928
 ピクセルの許容範囲, ビットマップ・チェッ
 クポイント 524
 ビジネス・アナリスト
 Business Process Testing における役割 1490
 ビジネス・コンポーネント, 概要 14
 ビジネス・プロセス・テスト 1494
 概要 14
 実行 1498
 日付, ユーザ定義書式の設定 1197
 必須プロパティ, 設定 109

索引

- ビットマップ・コンペアアラ、「カスタム・コンペアアラ」を参照
- ビットマップ・チェックポイント 523
 - RGB の許容範囲 524
 - カスタマイズ 524, 1561
 - 作成 526
 - の結果の分析 1027
 - ピクセルの許容範囲 524
 - 変更 526
- ビットマップ・チェックポイントのカスタマイズ 1561
 - API 1570
 - サンプル 1585
 - チュートリアル 1574
- 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 530
- 非表示モード 1425
- 表現
 - エキスパート・ビューおよび関数ライブラリでの使用 855
- 表示ツールバー 46
- 標準出力値 679
 - 作成 683
 - 指定 687
- 標準設定のオブジェクトの認識設定 121
- 標準設定のオプション・ステップ 963
- 標準設定のパラメータの定義 765, 768
- 標準設定のプロパティ, 変更 81, 164
- 標準チェックポイント
 - 結果の分析 1024
 - タイムアウトの指定 517
- 標準ツールバー, QuickTest ウィンドウ 44
- 表示領域
 - スクリプト・エディタ 1375
- 表示枠
 - 移動 1124
 - 欠落リソース 38
 - 自動非表示 1129
 - 使用可能なキーワード 34
 - 情報 37
 - タスク 42
 - テスト・フロー 41
 - デバッグ・ビューア 35
 - ドッキング 1129
 - フローティング 1129
 - プロセス・ガイダンス 39
 - リソース 40
 - レイアウトのカスタマイズ 1124
- ふ
- ファイル・ツールバー, QuickTest ウィンドウ 25
- ファイル・メニュー・コマンド, データ・テーブル 1194
- フィルタ
 - オブジェクトの定義 147
- 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 308
 - ターゲット・リポジトリ 286
- 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 286
 - オブジェクト・リポジトリ比較ツール 308
- フィルタ・プロパティ (スマート認識) 123
- フォント, キーワード・ビューでの設定 422
- 不具合の報告
 - 自動 1009
 - 手動 1009
- 不具合, 報告 1009
 - 自動化, テスト中 1011
 - テスト結果からの 1009
- 復元, QuickTest の標準設定のレイアウト 1223
- 複雑な値 766
- 複数テキスト・ブロック・モード, テキスト認識 753
- 複数のドキュメントを使った作業 1146
- ブックマーク 847
- ブレイクポイント
 - キーワード・ビューでの使用 427
- 削除 1072
- 設定 1071
- 説明 1070
- 無効化と有効化 1071
- フローティング表示枠 1129
- フロー表示枠
 - スクリプト・エディタ 1369
- プログラミング 1539
 - VBScript 856
 - エキスパート・ビューおよび関数ライブラリ 829

- 関数ライブラリ 829
- コメント 818
- 実行セッション中のメッセージの表示 817
- 条件ステートメント 801
- ステップ・ジェネレータ 781, 783
- テスト結果へのメッセージの送信 815
- メッセージの生成 815
- ループ・ステートメント 806
- プログラムの記述 209, 865
 - With ステートメント 869
 - インデックス・プロパティ 873
 - オブジェクトに対するチェックの実行 874
 - 記述オブジェクト 870
 - ステートメント 866
 - 変数 866
- プロジェクト (Quality Center)
 - 接続 1399
 - 切断 1403
 - テストの保存 1406
 - テストを開く 1407
- プロセス・ガイダンス 1213
 - 開始 1212
 - 表示枠 1210
- プロセス・ガイダンス表示枠 39
- プロセスの選択画面 1335
- プロパティ 437, 1370, 1373
 - CreationTime 119
 - アクションの設定 445
 - アクションの呼び出しの設定 485
 - インデックス 117
 - 回復シナリオの表示 1352, 1358
 - キーワード・ビューのステップの表示 425
 - 実行環境オブジェクト 889
 - テスト・オブジェクト 88
 - テスト・オブジェクト記述からの削除 179
 - テスト・オブジェクト記述の追加 173
 - テスト・オブジェクトの新規プロパティの定義 177
 - 認識プロパティを参照。
 - ネイティブ 88, 889
 - 場所 117
 - 標準 81, 164
- 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・タブ
 - テーブル出力値のプロパティ・ダイアログ・ボックス 718
 - テーブル・チェックポイントのプロパティ・ダイアログ・ボックス 554
- プロパティ値
 - テスト・オブジェクト記述での指定 239
 - 同期化ポイント 820
- プロパティの追加と削除ダイアログ・ボックス 173
- 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 111, 128
- へ
- 米国リハビリテーション法の第 508 条, 「Web Content Accessibility Guidelines」 6
- ベースライン 1472
- ベースラインの履歴
 - バージョンの履歴との比較 1476
- ベースラインの履歴ダイアログ・ボックス 1473
- 変更
 - ライセンス 16
- 編集ツールバー, QuickTest ウィンドウ 45
- 編集メニュー・コマンド, データ・テーブル 1195
- 変数
 - 環境 1271
 - 環境変数, ユーザ定義も参照
 - グローバル・スコープ内で一意 945
- ほ
- 補足プロパティ, 設定 109
- ボタン
 - ツールバーまたはメニューに追加 1136
- ボタンの表示ダイアログ・ボックス 1135
- ポップアップ・ウィンドウ・トリガ 1326
- ポップアップ・ウィンドウの条件を指定画面 1328
- め
- メソッド
 - 実行環境オブジェクト 889
 - 新規の追加または振る舞いの変更 939
 - 操作を参照。
 - ネイティブ 889
 - ユーザ定義 939
- メソッドの登録 939

索引

メソッドの登録解除, UnregisterUserFunc ステートメントの使用 943

メッセージ

実行セッション中の表示 817

生成 815

テスト結果への送信 815

メニュー

新規作成 1136

メニュー・バー, Quick Test ウィンドウ 23

メンテナンス実行ウィザード

スマート認識画面 1108

メンテナンス実行モード 1094

も

文字セット・サポート, Unicode 4

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 133

プロパティへのアクセス 890

メソッド 939

ユーザ定義オブジェクト, 割り当て 133

ユーザ定義関数 907

ガイドライン 945

関数定義ジェネレータ 924

関数定義ジェネレータでのコードのレビュー 936

仕上げ 937

説明を付ける 934

追加の生成 936

ツールチップの追加 934

登録 929

ユーザ定義設定

元に戻す 1236

ユーザ定義の外部環境変数 654

ユーザ定義の数値の書式, 設定 1197

ユーザ定義の内部環境変数 653

優先順位

回復シナリオの設定 1358

指差しマーク

使い方のヒント 99

よ

要素の値ダイアログ・ボックス

XML チェックポイントの結果 1041

予約済みオブジェクト 920

ら

ライセンス情報 16

変更 16

ライブラリ・タブ, Quality Center 1435

乱数パラメータ 663

り

リソース

管理 40, 1149

定義 1430

テスト内で欠落 1169

バージョン・コントロールからの

チェックアウト 1465

バージョン・コントロールへのチェッ

クイン 1468

バージョン・コントロールへの追加 1465

リソースおよび依存関係モデル

概要 1431

用語集 1430

リソース表示枠 40, 1149, 1372

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 516, 520, 532, 545, 570, 592, 617, 688, 704, 714, 725, 737

リポジトリの関連付けダイアログ・ボックス 201

リポジトリの種類 90

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 309

リポジトリ・パラメータ 233

値のパラメータ化 239

管理 234

削除 237

追加 235

変更 237

割り当て 204

リポジトリ・パラメータ・ダイアログ・ボックス 239

リポジトリ・パラメータの管理ダイアログ・ボックス 234

リポジトリ・パラメータの追加ダイアログ・ボックス 235

リモート・エージェント 1423

リモート・エージェントの設定ダイアログ・ボックス 1425

利用可能な値, メソッド引数 840

履歴タブ, Quality Center 1436

る

ループ・ステートメント 806
 キーワード・ビューでの使用 427

れ

レイアウト
 QuickTest ウィンドウのカスタマイズ 1123
 タブの移動 1124
 標準設定の復元 1131
 表示枠の移動 1124
レポート, 「テスト結果ウィンドウ」を参照
レポートの挿入ダイアログ・ボックス 815
レポート, フィルタの 894
レポートをエクスポート・ダイアログ・ボックス 997

ろ

ローカライズ 653, 1189
ローカライズされたアプリケーション, テスト 1549
ローカル・オブジェクトのエクスポートと置換 195
ローカル・オブジェクト・リポジトリ 90, 93
 エクスポートと置換 195
 結合 273
 へのオブジェクトのコピー 197
ローカル・システム・モニタ表示枠 1284
ローカル・データ・シート, 「アクション・データ・シート」を参照
ローカル・データ・テーブル・パラメータ 652
ローカル・テスト 432
ローカル・パラメータ 408
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 195

わ

割り当て
 欠落アクション 1172
 欠落アクションの呼び出し 1173
 未割り当てのオブジェクト・リポジトリ 1180
 未割り当てのリポジトリ・パラメータ 1184

ユーザ定義オブジェクト 133
リポジトリ・パラメータ 204
割り当てカラム, キーワード・ビュー 392

