

ADOBE® FLEX™ 3

MERCURY QUICKTEST PROFESSIONAL™を使用した
ADOBE® FLEX アプリケーションのテスト

Fx

Mercury QuickTest Professional™ を使用した Adobe® Flex アプリケーションのテスト

© 2008 Adobe Systems Incorporated. All rights reserved.

Mercury QuickTest Professional™ を使用した Adobe® Flex アプリケーションのテスト

本マニュアルがエンドユーザー使用許諾契約を含むソフトウェアと共に提供される場合、本マニュアルおよびその中に記載されているソフトウェアは、エンドユーザー使用許諾契約にもとづいて提供されるものであり、当該エンドユーザー使用許諾契約の契約条件に従ってのみ使用または複製することが可能となるものです。当該エンドユーザー使用許諾契約により許可されている場合を除き、本マニュアルのいかなる部分といえども、Adobe Systems Incorporated (アドビシステムズ社) の書面による事前の許可なしに、電子的、機械的、録音、その他いかなる形式・手段であれ、複製、検索システムへの保存または伝送を行うことはできません。本マニュアルの内容は、エンドユーザー使用許諾契約を含むソフトウェアと共に提供されていない場合であっても、著作権法により保護されていることにご留意ください。

本マニュアルに記載される内容は、あくまでも参照用としてのみ使用されること、また、なんら予告なしに変更されることを条件として、提供されるものであり、従って、当該情報が、アドビシステムズ社による確約として解釈されてはなりません。アドビシステムズ社は、本マニュアルにおけるいかなる誤りまたは不正確な記述に対しても、いかなる義務や責任を負うものではありません。

新しいアートワークを創作するためにテンプレートとして取り込もうとする既存のアートワークまたは画像は、著作権法により保護されている可能性のあるものであることをご留意ください。当該アートワークまたは画像を新しいアートワークに許可なく取り込んだ場合、著作権者の権利を侵害することがあります。従って、著作権者から必要なすべての許可を必ず取得してください。

例として使用されている会社名は、実在の会社・組織を示すものではありません。

Adobe、Adobe ロゴ、Flex、Flex Builder は、アドビシステムズ社の米国およびその他の国における登録商標または商標です。ActiveX and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

All other trademarks are the property of their respective owners.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

This product contains either BISAFE and/or TIPEM software by RSA Data Security, Inc.

The Flex Builder 3 software contains code provided by the Eclipse Foundation ("Eclipse Code"). The source code for the Eclipse Code as contained in Flex Builder 3 software ("Eclipse Source Code") is made available under the terms of the Eclipse Public License v1.0 which is provided herein, and is also available at <http://www.eclipse.org/legal/epl-v10.html>.

Adobe Systems Incorporated, 345 Park Avenue, San Jose, CA 95110-2704, USA.

Notice to U.S. government end users. The software and documentation are "Commercial Items," as that term is defined at 48 C.F.R. §2.101, consisting of "Commercial Computer Software" and "Commercial Computer Software Documentation," as such terms are used in 48 C.F.R. §12.212 or 48 C.F.R. §227.7202, as applicable. Consistent with 48 C.F.R. §12.212 or 48 C.F.R. §227.7202-1 through 227.7202-4, as applicable, the Commercial Computer Software and Commercial Computer Software Documentation are being licensed to U.S. Government end users (a) only as Commercial items and (b) with only those rights as are granted to all other end users pursuant to the terms and conditions herein. Unpublished-rights reserved under the copyright laws of the United States. For U.S. Government End Users, Adobe agrees to comply with all applicable equal opportunity laws including, if appropriate, the provisions of Executive Order 11246, as amended, Section 402 of the Vietnam Era Veterans Readjustment Assistance Act of 1974 (38 USC 4212), and Section 503 of the Rehabilitation Act of 1973, as amended, and the regulations at 41 CFR Parts 60-1 through 60-60, 60-250, and 60-741. The affirmative action clause and regulations contained in the preceding sentence shall be incorporated by reference.

目次

第 1 章: QuickTest Professional の操作

テストプロセスの概要	1
テスト作成の概要	2
テストの記録	5
テストスクリプトについて	6
操作について	7
チェックポイントの使用	8
共通のメソッドとプロパティの使用	9
Flex テストの再生	10

第 2 章: 高度な概念

コンテナの操作	11
Repeater オブジェクトの操作	11
データ駆動型でリストベースのコントロールの操作	11
トラブルシューティング	12
Flex を使用した自動化されたテストの制限	16

第 1 章 : QuickTest Professional の操作

Mercury QuickTest Professional™ を操作して、Adobe Flex アプリケーションをテストできます。

トピック

テストプロセスの概要.....	1
テスト作成の概要.....	2
テストの記録.....	5
テストスクリプトについて.....	6
操作について.....	7
チェックポイントの使用.....	8
共通のメソッドとプロパティの使用.....	9
Flex テストの再生.....	10

テストプロセスの概要

大規模で複雑なアプリケーションでは、ユーザーの操作によってランタイム例外が作成されないこと、およびデータサービスへの接続が正しく機能していることを確認するためにテストを行う必要があります。

本マニュアルでは、テストプロセスに QA テスト担当者と Flex 開発者という 2 つのロールがあることを想定しています。QA テスト担当者は Flex アプリケーションの開発方法を理解していません。また、Flex ソースコード、Flex コンパイラ、Flex マニュアルにもアクセスできないと想定します。これに対して、Flex 開発者は Mercury QuickTest Professional (QTP) など、テストツールの使用方法を必ずしも理解しているわけではありません。

開発者向けの Flex マニュアルには、テスト可能なアプリケーションを作成するための一連のガイドラインが含まれています。これらのガイドラインでは、テスト可能なアプリケーションのコンパイル方法、テストを行うためのカスタムコンポーネントの計測方法、テストプロセスが円滑に機能するようにアプリケーションを記述する方法について説明しています。ガイドラインの詳細については、『Adobe Flex 3 開発ガイド』を参照してください。

Flex アプリケーションをテストする場合は、事前に Flex 開発者が SWF ファイルをコンパイルしておきます。開発者は SWF ファイルを埋め込んだ HTML ファイルも用意します。これは HTML ラッパーと呼ばれます。この 2 つのファイルを任意の Web サーバーにデプロイします。

一部の Flex アプリケーションでは、Web サービスやリモートオブジェクトなど、外部のリソースを使用する必要があります。このため、Web サーバーがこれらのリソースにアクセスできることを確認してください。アクセス許可ファイルをターゲットサーバーに追加して、Flex アプリケーションのサーバーへのアクセスを許可することが必要な場合もあります。詳しくは、Flex 開発者に問い合わせてください。

テストを実行するには、QTP を起動し HTML ラッパーを要求します。QTP は他のターゲットの種類の場合と同様に、Flex アプリケーションとのやり取りを記録します。その後、Web サーバーがアクセス可能な限り、Flex アプリケーションに対してテストを繰り返し実行できます。

Flex ファイルのデプロイ

Flex アプリケーションをテストする前に、アプリケーションとそれをサポートするファイルを、QTP がアクセスできる場所にデプロイする必要があります。これらのファイルは、以下のとおりです。

- アプリケーションの SWF ファイル
- HTML ラッパーおよびラッパーファイル（ラッパーファイルには、SWF ファイル、JavaScript ファイル、およびデバッグなどの機能をサポートするその他のファイルが含まれます）
- RSL SWC ファイル
- モジュールの SWF ファイル

- SWC ファイル、スタイルの SWF ファイル、およびイメージ、ビデオ、サウンドファイルなどのヘルパーファイル Flex アプリケーションと HTML ラッパーファイルを HTTP サーバー上の任意の場所にデプロイします。QTP からこの場所にアクセスできるようにしてください。

場合によっては、Flex アプリケーションで使用されるファイルはリモートサーバーに存在します。この場合は、いかなるファイアウォールを介しても、必ずそのファイルにアクセスできるようにする必要があります。さらに、リモートサーバーには、リモートアクセスを許可するためのクロスドメイン許可ファイルが必要な場合があります。詳しくは、Flex 開発者と相談してください。

必要に応じて、HTML ラッパーをパスワードで保護したり、QTP へのアクセスに認証メカニズムを導入したりできます。詳しくは、QTP のマニュアルを参照してください。

Flex にインストールされるテストファイルについて

QTP 向けの Flex のサポートには、テストコンピュータ上に 2 つの ActiveX プラグインをインストールする必要があります。これらのプラグインは、必要な Flex と QTP 間の通信レイヤーを提供します。一方のプラグインは QTP 内部で実行され、他方はブラウザ内部で実行されます。ブラウザプラグインは、Windows 上の Microsoft Internet Explorer 6 でのみ署名および実行されるように設計されています。これらのプラグインは連携して、Adobe Flash Player と Flex フレームワークが、記録モード中に確実に QTP との適切なスクリプトのやり取りを送信できるようにします。

同様に、これらのプラグインは再生中に QTP がスクリプトコマンドを Flash Player と Flex に送信するためのメカニズムを提供します。ブラウザプラグインは Flash Player の `flash.external.ExternalInterface` クラスを使用するため、このプラグインには `flash.external.ExternalInterface` を実行するための適切なセキュリティコンテキストが必要です。SWF ファイルが Web サーバーから実行される場合、Flash Player は自動的に `flash.external.ExternalInterface` を有効にします。SWF ファイルがローカルファイルシステムから実行される場合は、`flash.external.ExternalInterface` を有効にするため、またブラウザプラグインが正常に機能するために、SWF ファイルが信頼できることが必要です。プラグインのロードに問題がある場合、原因が Flash Player のデバッグログファイルに書き込まれます。

テスト環境について

Flex コンポーネントの名前はコンポーネントの特徴を表しています。例えば、ボタンは Flex プログラミング環境では `Button` コントロールで表されます。テスト環境では、テスト可能なコントロールの名前には、それぞれ Flex の接頭辞が付きます。例えば、`Button` コントロールは QTP スクリプト内では `FlexButton` となります。これにより、異なる種類のアプリケーションのコントロールがスクリプト内で混同されるのを防ぐことができます。

すべてのビジュアル Flex コンポーネントは、品質管理テストではスクリプトで記述できます。コンポーネントのプロパティとメソッドの多くは記録して再生できます。各メソッドには、一定数のプロパティが格納されます。

テストを最初に作成したときには、ビジュアルコントロールは通常コンテナ内で参照されます。Flex プログラムによって使用されるすべてのコンテナが、テスト環境に反映されるとは限りません。例えば、1 組のボタンが水平ボックスコンテナ (HBox) 内部にある場合、その HBox コントロールはテストスクリプト内に表示されない場合があります。スクリプト内のコントロール数を制限すると、スクリプトが読みやすくなり、詳細情報を失うことなく階層の簡潔性を維持できます。

`tea.html` ドキュメントには、テストできる Flex コンポーネントの完全な一覧が含まれています。このドキュメントでは、それらのコンポーネントのテスト可能なメソッドとプロパティも表示できます。

テスト作成の概要

テストを記録する場合、QTP はユーザーが実行した各アクションに関連付けられたテストスクリプトの行を記録します。各行は通常、画面上に表示されるコンポーネントに対して実行されたアクションを表します。これらのコンポーネントはオブジェクト呼ばれ、オブジェクトに対して実行されるアクション (ボタンのクリックなど) は操作と呼ばれます。

テストの基本構成要素はテストオブジェクトです。オブジェクトは QTP のオブジェクトリポジトリに格納されます。QTP がオブジェクトを識別する方法、およびオブジェクトをオブジェクトリポジトリに追加する方法を理解する必要があります。詳細については、「[オブジェクトの識別](#) (7 ページ) を参照してください。

テストの再生用の記録を作成するには、テストオブジェクトに対する一連の操作やイベントを記録します。QTP によって認識されるイベントを把握することが重要です。各オブジェクトに対してサポートされているイベントのリストにイベントを追加する開発者は、『Adobe Flex 3 開発ガイド』を参照してください。

多くの QTP テストでは、チェックポイントを使用して出力の値を既知の値と比較します。Flex では、QTP で使用されるチェックポイントの種類のサブセットをサポートしています。詳細については、「[チェックポイントの使用](#)」(8 ページ)を参照してください。

アプリケーションのテスト中に問題が発生する場合は、「[トラブルシューティング](#)」(12 ページ)を参照してください。このセクションでは、エラーコードとその意味、および QTP で Flex プラグインを使用する場合に発生する可能性のある一般的な問題の一部について説明します。

QTP オブジェクトモデルについて

テストを作成する前に、QTP と Flex のオブジェクトモデルがどのように統合されているかを理解することが重要です。「テストオブジェクトモデル」は、アプリケーションで使用されるオブジェクトを表現するために QTP が使用するオブジェクトの種類のセットです。テストオブジェクトモデルは、QTP テストオブジェクトをアプリケーション内の Flex オブジェクトにマッピングします。例えば、Flex アプリケーション内の Button コントロールは、QTP オブジェクトモデル内で FlexButton として認識されます。

QTP オブジェクトモジュールに表示される Flex オブジェクトの完全なリストについては、ドキュメント『QTP Object Type Information』を参照してください。このドキュメントは、Mercury Quick Test Professional で Flex アプリケーションをテストするときを使用できる操作とオブジェクトのプロパティについて説明しています。このドキュメントは、次の場所からダウンロードできます。

http://www.adobe.com/go/flex3_qtp_object_type_reference

QTP テストオブジェクトはテスト中にプロパティを格納し、イベントを記録します。各オブジェクトには、QTP が記録できる独自のプロパティとメソッドのセットがあります。テストオブジェクトでは、対応する Flex オブジェクトのすべてのイベントとプロパティがサポートされるわけではありませんが、ユーザーの操作に最もよく関連付けられるイベントがサポートされています。

「オブジェクトリポジトリ」は、テスト内で使用されるすべてのオブジェクトのリストです。オブジェクトをテストに追加するには、そのオブジェクトをオブジェクトリポジトリに追加する必要があります。オブジェクトをテストから削除するには、そのオブジェクトをオブジェクトリポジトリから削除する必要があります。

「操作」は、オブジェクトに対して実行するアクションです。これらは、Flex イベントに相当しますが、通常はより高いレベルにあります。例えば、ボタンがクリックされると、QTP は「クリック」操作を記録します。click イベントを本質的に構成する mouseDown および mouseUp イベントは記録しません。

各操作には、イベントに関する情報を定義するパラメータがあります。この情報には、操作が発生したときに押されていたキーやリスト内で選択されていたアイテムのインデックスなど、詳細な内容が含まれています。

すべてのコントロールに操作が伴うわけではありません。例えば、FlexRule コントロールには操作がありません。これらのコントロールも、オブジェクトリポジトリに追加し、チェックポイントを使用してオブジェクトのプロパティをテストできます。FlexRepeater や FlexDisplayObject など、操作を持たないその他のコントロールは、通常、テストでやり取りする必要がない低レベルオブジェクトです。

自動化されたテストでは、各コンポーネントを永続的で変化しないプロパティのセットによって識別する必要があります。主要なプロパティ automationName は、人間が判読でき、画面上の対応するものと容易に関連付けられます。このプロパティのセットは総称して「オートメーション ID」と呼ばれます。QTP はリポジトリにオートメーション ID を格納します。コンポーネントの ID を変更すると、リポジトリ内に複数のバージョンのオブジェクトが存在することになります。さらに、QA エンジニアはこれらの ID をデータベースからの ID で変更して、データ駆動型のスクリプトを作成することができます。

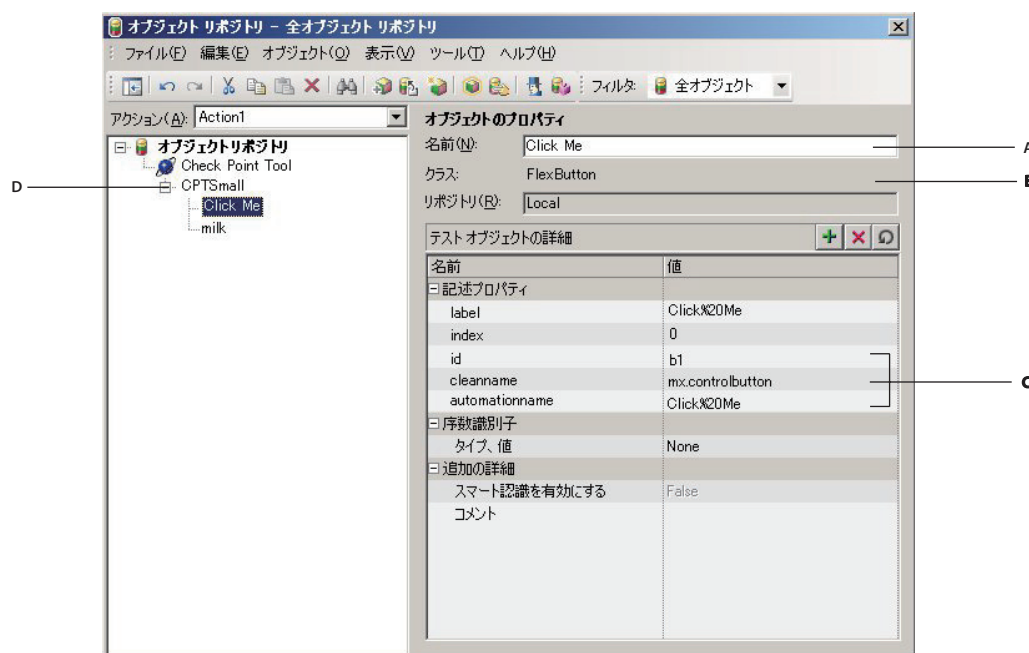
QTP では特にリストアイテムをコンポーネントとして扱うことを避けます。これはリストアイテムの識別子が編集可能なリスト内で変更される可能性があり、識別子が繰り返される（例えば、テキストが繰り返される行を持つデータグリッド）ことがあるためです。また、リストアイテムに対して、リポジトリ内にエンTRIES を作成するか、またはリポジトリ内のエンTRIES を要求すると、データ駆動型スクリプトにとって大きな負荷となる可能性があります。これはリポジトリ内にすべてのリストアイテムを配置すること、およびリポジトリ内の多くのオブジェクトを変更する必要があるためです。

オブジェクトの保存

QTP を使用すると、Flex アプリケーションとのやり取りを記録して再生するか、または自分でスクリプトを記述することができます。一般的には、Flex アプリケーションでのユーザーのアクションを QTP で記録して、後でその記録されたスクリプトを変更する方が簡単です。記録セッション中に、QTP はユーザーが対話した Flex オブジェクトをオブジェクトリポジトリに追加します。そうでない場合は、各オブジェクトについて、自分で記述したスクリプトで使用する前に別途定義する必要があります。

テスト中に QTP が Flex コントロールを検出すると、そのオブジェクトを表現するプロパティのセットを保存し、そのセットへの参照をオブジェクトリポジトリ内に格納します。QTP オブジェクトリポジトリ内の Flex コントロールの定義には、QTP が Flex コントロール（例えば Button）をテストオブジェクト（例えば FlexButton）にマッピングするのに十分な情報が含まれています。

次の例は、QTP リポジトリエディタ内の Flex Button コントロールを示しています。



A. テストオブジェクトの名前 B. テストオブジェクトモデルのクラス C. このオブジェクトを識別するために使用される Flex プロパティ D. アプリケーション階層

テストを実行し Flex コントロールと対話すると、QTP は以下のことを行います。

- 1 Flex オブジェクトがマッピングされているテストオブジェクトクラスを識別します。
- 2 オブジェクトの現在の状態を読み取り、オブジェクトの説明プロパティと値のリストを保存します。
- 3 スクリプト内でオブジェクトの識別に使用する一意の ID を割り当てます。QTP は、Flex コントロールの外観（ボタンのラベルなど）やコントロールの Flex id に基づいた複数の説明プロパティを使用できます。
- 4 オブジェクトに対して実行された操作を、適切なテストオブジェクトメソッドを使用して記録します。

テストが実行されると、QTP はコンポーネントのプロパティを名前と値のペアとして保存します。保存されるプロパティは、オブジェクトリポジトリ内での識別に必要なものに限られます。これらのプロパティには id とインデックスが含まれています。チェックポイント、データテーブルなどの方法を使用して、その他のオブジェクトのプロパティ（TextInput コントロールのテキストの値、Button コントロールの背景の色など）を保存および取得できます。

アプリケーション階層について

オブジェクトリポジトリ内では、オブジェクトは階層構造で格納されます。階層のルートはブラウザです。次の要素は Flex アプリケーションです。その下に、アプリケーションを構成するコンテナとコントロールがあります。

QTP のオブジェクトリポジトリとキーワードビューでは、ツリー構造に似た形式でアプリケーション内のコンテナのレベルが表現されます。一般に、この階層はアプリケーションが表示される方法に基づきます。例えば、MyApp Flex アプリケーション内に Change Settings (設定の変更) というラベルが付いた TabNavigator のビュー内部に Submit (送信) というラベルが付いた Button がある場合、階層は次のように表示されます。

```
Browser
|
Application ("MyApp")
|
Panel ("Change Settings")
|
Button ("Submit")
```

QTP スクリプトでは、ドット表記シンタックスを使用してオブジェクトにアクセスします。各レベルはドットで区切られます。Button を参照するには、次のようなパスを使用します。

```
Browser("Home").FlexApplication("MyApp").FlexPanel("Change Settings").FlexButton("Submit")
```

QTP スクリプトで Flex アプリケーションを表す方法について詳しくは、「[テストスクリプトについて](#)」(6 ページ) を参照してください。

HBox や Panel などのコンテナは TabNavigator コンテナに含まれますが、TabNavigator 自身は階層内に表示されません。スクリプトを読みやすくするために、テストに影響を与えない Flex コンテナがあれば階層から削除します。

アプリケーション階層は、通常ツリー形式のレイアウトで画面上に内容が表示されるので、表示階層とも呼ばれます。表示階層はオートメーション階層と必ずしも同じではありません。オートメーション階層は QTP スクリプト内でオブジェクトが参照されるときオブジェクトの階層です。画面上 (例えば、境界上やウィンドウ内) に表示されるコンテナは表示階層の一部ですが、オートメーション階層の一部ではないことがあります。

通常、コンテナが画面上に表示されない場合は、テスト階層にも出現しません。例えば、単独の HBox コンテナは多くの場合コンテンツの配置に使用されますが、視覚的な表現形式を持たないため、テストスクリプト内には出現しません。ただし、HBox が TabNavigator の子である場合は例外です。この場合、HBox はそのコンテンツを一意に識別するために必要なので、階層の一部となります。TabNavigator および他のコンテナの使用方法について詳しくは、「[コンテナの操作](#)」(11 ページ) を参照してください。

テストの記録

テストを QTP に記録するには、アプリケーションの SWF ファイルをコンパイルして HTML ラッパーファイルを作成する必要があります。メインのラッパーファイルは、アプリケーションをページ上のオブジェクトとして定義して埋め込み、ページが要求されたときに Flash Player が呼び出されるようにします。アプリケーションの SWF ファイルを直接要求することはできません。この HTML ラッパーは自分自身で記述することも、コンパイラで生成することもできます。

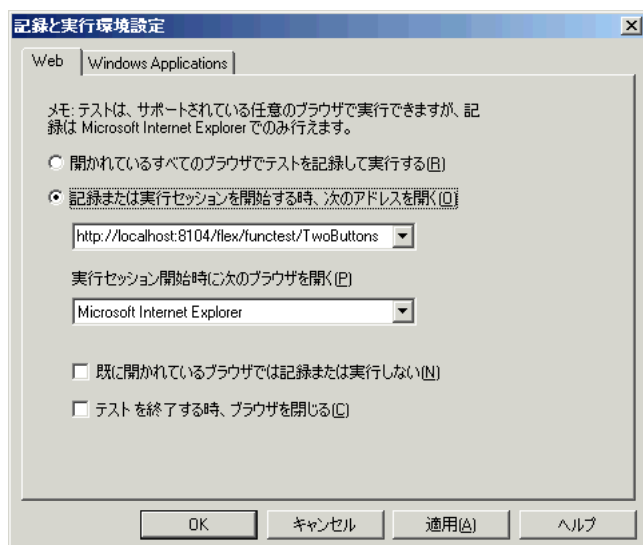
QTP が、アプリケーションの SWF ファイルを埋め込むラッパーファイルを参照するようにしてください。

Adobe Flex Builder またはコマンドラインコンパイラを使用して SWF ファイルを生成できます。SWF ファイルをプリコンパイルしたら、SWF ファイルと HTML ラッパーファイル、およびランタイム共有ライブラリ (RSL)、グラフィックや他の SWF ファイルなどの追加アセットを、Web サーバー上のディレクトリにデプロイします。

テストを記録するには、現在開いているブラウザで QTP がテストを実行するように指定するか、アプリケーションの場所を URL として指定します。後者の場合には、QTP がブラウザを起動すると、指定した要求 URL をブラウザに渡します。「記録と実行環境設定」ダイアログボックスで使用する方法を決定します。

一般に、Web サーバーを使用して HTML ラッパーを QTP クライアントに提供する必要があります。ファイルシステム上の HTML ラッパーファイルは要求できません。

開いているブラウザウィンドウで現在実行しているアプリケーションに対して QTP にテストを実行するには、「開かれているすべてのブラウザでテストを記録して実行する」オプションをオンにします。QTP がブラウザを開いて URL を要求するようにするには、次の例に示すように 2 番目のオプションを選択します。



テストを再生すると、QTP はブラウザを開いて同じ URL を要求します。

必ず、HTML ファイルかアプリケーションの SWF ファイルを埋め込むその他のラッパー ファイル タイプを指定してください。SWF ファイルを直接要求することはできません。

テストスクリプトについて

QTP は、キーワードビューと呼ばれるオブジェクトレベルのビューと、エキスパートビューと呼ばれるテキストのみのビューのステップにテストを記録します。各ビューは編集可能で、一方を編集すると、他方が QTP によって更新されます。例えば、エキスパートビュー内のあるステップを削除すると、キーワードビューから該当するステップが QTP によって削除されます。ステップの追加、チェックポイントの追加または削除についても同じ動作が行われます。

エキスパートビューは、テスト内の各ステップを VBScript のアクションとして表します。VBScript はドット表記を使用して親オブジェクトと子オブジェクトを区切ります。各行の最後にはアクションが記述され、そのアクションを定義するパラメータが後に続いています。

次の例は、エキスパートビューでのマウスクリックを示しています。

```
Browser("Main Page").FlexApplication("MyApp").FlexButton("b2").Click
```

次の例は、Flex アプリケーションでのエキスパートビューのシンタックスを示しています。

```
Browser("document_title").FlexApplication("app_name").test_object("name").operation  
[parameters]
```

`document_title` プロパティは、HTML ページの `<HEAD>` ブロック内の `<TITLE>` タグによって定義されます。Flex アプリケーションの場合、HTML ラッパー内の `<TITLE>` タグのデフォルト値は MXML ファイルへの URL です。ページにタイトルが設定されていない場合、QTP は `Browser` を使用します。

`app_name` プロパティは、HTML ページの `<object>` タグに埋め込まれた SWF ファイルの ID プロパティと同じです。

`test_object` は、Flex コントロールに対応する QTP のテストオブジェクトです。例えば、`Button` コントロールの QTP テストオブジェクトは `FlexButton` です。`FlexButton` は `name` プロパティを受け取ります。このプロパティはオブジェクトトリポジットりで表されるそのオブジェクトの QTP 名を示します。この名前は通常、その外観 (ラベルなど) に基づいていますが、Flex MXML ソースコード内の `Flex id` と同じになる場合もあります。

`operation` プロパティは、QTP が記録するオブジェクトによって送出されたイベントです。このプロパティにはパラメータを指定しないか、または任意の数のパラメータを指定できます。パラメータの数と型はイベントを送出しているコントロールの種類によって異なります。操作中に押し下げられたキー、リストエントリのインデックスまたはテキストフィールドに入力されたテキストなどの場合があります。例えば、ユーザーが `FlexTextArea` オブジェクトに文字を入力すると、ユーザーが入力したテキストはその操作の終了後に表示されます。ユーザーが `FlexTextArea` オブジェクトに数字の 42 を入力した場合、エキスパートビューには次の例のようなステートメントが表示されます。

```
Browser("Main Page").FlexApplication("MyApp").FlexTextArea("ta1").Input "42"
```

ユーザーが `FlexTextArea` オブジェクト内のテキストを選択した場合、QTP は `Select` 操作を記録します。この操作はカーソルで区切られた 2 つのパラメータを受け取ります。最初のパラメータは選択内容の開始文字の位置で、2 番目のパラメータは終了文字の位置です。ユーザーが `FlexTextArea` オブジェクト内の最初の 4 文字を選択した場合、エキスパートビューには次の例のようなステートメントが表示されます。

```
Browser("Main Page").FlexApplication("MyApp").FlexTextArea("ta1").Select 0,3
```

オブジェクトの識別

テストオブジェクトは、QTP のテストスクリプトの構成要素です。テストで使用される各オブジェクトはオブジェクトリポジトリに格納されます。Flex コントロールはこれらの QTP テストオブジェクトにマッピングされます。

テストの記録時、QTP はユーザーが対話した各オブジェクトをオブジェクトリポジトリに追加して、スクリプト内のテストオブジェクト名によってそのオブジェクトを参照します。QTP が使用する識別子は、リポジトリ内のテストオブジェクトの `name` プロパティです。このプロパティは、Flex アプリケーションのソースコード内で Flex 開発者が指定する `Flex id` と同じになる場合があります。例えば、"myCombo" の `Flex id` を持つ `ComboBox` コントロールは、`myCombo` という名前でもオブジェクトリポジトリに格納されます。QTP スクリプトでは、次の例に示すようにこのコントロールを参照します。

```
FlexComboBox("myCombo")
```

ただし、Flex オブジェクトが常にその `Flex id` プロパティによってスクリプト内で識別されるわけではありません。多くの場合、テストオブジェクト名は画面上に表示される名前と一致します。例えば、`FlexButton` オブジェクトの QTP テストオブジェクト名は、そのオブジェクトの `label` プロパティの値であって `id` プロパティの値ではありません。例えば、次の `FlexButton` は QTP オブジェクトリポジトリ内で「Click Me」として識別されます。



QTP のエキスパートビューでは、この `FlexButton` オブジェクトは次の例で示すように参照します。

```
FlexButton("Click Me")
```

`FlexButton` オブジェクトにラベルがない場合、QTP はテストオブジェクト名に対して `Button` のツールヒントを使用します。ラベルもツールヒントもない場合、QTP はソースコードの `id` プロパティを使用します。Flex 開発者が `id` を指定していない場合、QTP はインデックスを使用します。これらのすべてを省略すると、QTP は独自の番号付け方式を使用してオブジェクトを識別します。

テストオブジェクト名を決定するためのプロパティの適用順序は、オブジェクトによって異なります。テストオブジェクト名が `Flex id` プロパティと異なるこの他のオブジェクトには、`FlexAccordionHeader`、`FlexButtonBarButton`、`FlexCheckBox`、`FlexLink`、`FlexPopUpButton`、`FlexRadioButton`、`FlexScrollThumb`、`FlexSimpleButton`、`FlexSliderThumb`、`FlexTab` などがあります。

操作について

記録中にユーザーが Flex アプリケーションと対話すると、アクションが QTP スクリプトに格納されます。これらのアクション（ボタンのクリックなど）は、テストスクリプトのシナリオを定義します。QTP では、これを操作と呼びます。Flex では、それらをイベントと呼びます。各操作は、Flex アプリケーション内のコントロールとの対話を定義します。通常、Flex アプリケーションとの複雑な対話（フォームに入力して送信する、複数のオブジェクトをドラッグしてある特定の場所にドロップするなど）を実行するには、操作を組み合わせて使用します。

ボタンなどの Flex コントロールでは多くのイベントが定義されます。これらのイベントには、`click`、`showToolTip`、`focusIn` などの一般的なもののほか、`creationComplete`、`render`、`invalid` などのユーザーからほとんど直接呼び出されないものもあります。Flex イベントには `mouseMove`、`keyDown`、`focusOut` などの詳細なアクションを表すものもあります。

最もよく使用される Flex イベントは、QTP では操作として利用可能です。ただし、QTP ですべてのイベントが記録されるわけではありません。QTP は、重要な意味を持つユーザーの操作、または論理的な最小単位の操作を記録します。つまり、`mouseDown` と `mouseUp` イベントの組み合わせではなく `click` イベントが記録され、`ComboBox` の `mouseDown`、`open`、`drag`、`mouseUp` および `close` のイベントシーケンスではなく `select` イベントが記録されます。

テストの作成時、記録可能なイベントがすべて記録されるわけではありません。例えば、多くのコントロールにある `mouseMove` イベントはデフォルトでは記録されません。このイベントがデフォルトで記録されれば、すべてのマウスの移動操作に対してスクリプトに新しいステップが含まれることになり、テストに大きな負荷を与えます。テストスクリプトに手動でこのイベントを追加して、QTP にこのイベントを再生させることができます。各 Flex コンポーネントについて使用可能な操作のリストを確認するには、ドキュメント『QTP Object Type Information』を参照してください。

場合によっては、QTP はこれらのイベントをより小さなイベントのシーケンスとして再生することがあります。例えば、`click` は呼び出された状況に応じて、`mouseDown` と `mouseUp` イベントのペア、または `keyDown` と `keyUp` イベントのペアとして再生されます。

QTP は各操作について、操作の特性を定義するパラメータのセットを格納します。これらのパラメータには、マウスがクリックされたときに押し下げられていたキーや、ドラッグ & ドロップ操作での移動先の X および Y 位置などが含まれます。操作パラメータには、キーボードやマウスなど、操作の発生元も記録されます。

Flex コントロールの操作を記録する際、QTP は操作の実行方法を記録します。イベントの発生元によって再生が異なる場合があるので、QTP はイベントを引き起こしたのがマウスかキーボードかを記録します。例えば、ボタンがマウスを使用してクリックされると、`mouseDown`、`mouseUp` および `click` の各イベントが送出されます。一方、ボタンにフォーカスがあるときにスペースバーを押すことでボタンがクリックされた場合は、`keyDown`、`keyUp` および `click` の各イベントが送出されます。

スクリプトを読みやすくするために、Flex プラグインはスクリプト内に記録されるイベントパラメータの数を最小限に抑えます。共通情報の記録を削減するために、QTP は各操作に対してデフォルトのパラメータを持っています。ボタンクリック操作の種類を表すデフォルトの値は「`mouse`」です。操作パラメータがデフォルト値と同じ値の場合、QTP はその値を記録しません。例えば、次の値は記録されません。

```
FlexButton("buttonId").Click "mouse"
```

QTP は次の値のみを記録します。

```
FlexButton("buttonId").Click
```

多くの場合 `FlexButton` のクリックはキーボードではなくマウスを使用して行われるため、デフォルトのアクションはマウスクリックとなっています。

チェックポイントの使用

チェックポイントを使用すると、オブジェクトを検査して、オブジェクトに所定の値が含まれているかどうかに応じて成功または失敗を判定することができます。テスト中、QTP はチェックポイントで現在値を期待値と比較します。両方の値が一致しない場合、テストを失敗とし、一致する場合は、テストを成功とします。

Flex QTP プラグインでは、次のチェックポイントの種類がサポートされています。

- ビットマップ
- 標準
- テーブル

Flex QTP プラグインでは、次のチェックポイントはサポートされていません。

- アクセシビリティ
- データベース
- イメージ
- ページ
- テキスト
- XML

標準チェックポイント

標準チェックポイントを使用するとコントロールのプロパティを検査できます。オブジェクトリポジトリ内の任意の Flex オブジェクトを検査できます。テスト可能なプロパティには、オブジェクトの状態についての情報（TextArea の text プロパティなど）とオブジェクトの外観についての情報（bgcolor スタイルプロパティ、height および width プロパティなど）の両方が含まれています。

使用可能なプロパティは Flex コントロールの完全なプロパティセットの一部ですが、最もよく使用される各種のプロパティは含まれています。プロパティの一覧については、[tea.html](#) ドキュメントを参照してください。

次の例では、アイテムの実際の値が期待値に一致するかどうかを検証する Check メソッドを使用したスクリプトの一部が示されています。

```
Browser("FlexStore").FlexApplication("flexstore").FlexCanvas("Support").
  FlexForm("index:4").Check CheckPoint("index:4")
```

詳しくは、『QuickTest Professional Object Model Reference』を参照してください。

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を使用する場合は、Flex オブジェクトの外観の違いを認識して確認することが必要です。この違いは、テストオブジェクトのタブフォーカス、mouseover イベントに対するオブジェクトの応答によって生じる可能性があります。フォーカスを持ち周囲にハロー効果があるオブジェクトを持つアプリケーションの領域にビットマップチェックポイントを追加する場合は、テスト中そのオブジェクトがフォーカスを確実に持つようにする必要があります。

次の例では、FlexButton テストオブジェクトについて、フォーカスを持っていない場合、フォーカスを持っている場合、およびオブジェクト上にマウスが置かれている場合の違いを示しています。



FlexImage オブジェクトには視覚上のフォーカスが置かれることはないため、画像のチェックに対してビットマップチェックポイントを使用する場合は、上述の問題は生じません。

テーブルチェックポイント

テーブルチェックポイントを使用すると、リストベースのコントロールとコンテナのプロパティを検査できます。テーブルチェックポイントは Flex のリストベースのコントロールすべてに対して使用できます。リストベースのコントロールには、List、DataGrid、TileList および Tree があります。テーブルチェックポイントは、すべてのコンテナに対して使用でき、内包するすべての子に対するリストが生成されます。

共通のメソッドとプロパティの使用

Flex テストオブジェクトでは、すべての QTP テストオブジェクトに共通のメソッドとプロパティがサポートされています。共通のメソッドを以下に示します。

- CaptureBitmap
- Check
- CheckProperty
- ChildObjects
- GetROProperty
- GetTOPProperty

共通のプロパティを以下に示します。

- Exist
- Object

共通のメソッドとプロパティはキーワードビューとエキスパートビューで使用できます。次の例では、エキスパートビューで `myList` コントロールの `visible` プロパティが `false` に設定されているかどうかをチェックするために `CheckProperty()` メソッドが使用されています。

```
Browser("My Page").FlexApplication("MyApp").FlexCheckBox("myList").CheckProperty "visible", false
```

上記の各メソッドとプロパティについては、『QuickTest Professional Object Model Reference』を参照してください。

Flex テストの再生

テストを再生するためにユーザーがブラウザを開く必要はありません。QTP がブラウザを起動します。QTP から要求する生成済み HTML ラッパーが必要です。アプリケーションの SWF ファイルを直接要求しないでください。

場合によっては、再生が意図したとおりに実行されないことがあります。多くの場合、これは同期の問題です。例えば、エフェクトの再生が QTP の予想より時間がかかる場合があります。これは、エフェクトの再生中に QTP がスクリプト内の次のステップに移動しようとしている可能性があります。Web サービス呼び出しなどの非同期メソッドを使用してデータを取得する場合にも別の問題が発生する可能性があります。Web サービス処理からデータを受信していない場合でも、スクリプトの実行が継続されるためです。

これらの問題の解決方法としては、待機時間を挿入することやオブジェクトの存在を確認してから実行を継続することなどがあります。詳しくは、「[再生の同期に関する問題の解決](#)」(14 ページ)を参照してください。

第 2 章 : 高度な概念

Flex アプリケーションをテストするために、Mercury QuickTest Professional™ と併用できる高度な手法があります。

トピック

コンテナの操作	11
Repeater オブジェクトの操作	11
データ駆動型でリストベースのコントロールの操作	11
トラブルシューティング	12
Flex を使用した自動化されたテストの制限	16

コンテナの操作

一般に、Mercury QuickTest Professional (QTP) では、テストスクリプト内のネストしたコントロールに関する詳細情報の量が削減されます。テストの結果やコントロールの識別に影響を与えないコンテナはスクリプトから削除されます。このことは、HBox、VBox、Canvas コンテナなど、レイアウトのみに使用されるコンテナに適用されますが、ViewStack、TabNavigator または Accordion コンテナ内で使用されている場合は除きます。この場合、それらのコンテナはナビゲーションを提供するために階層に追加されます。

ネストしたコンテナを使用するときには、ID の競合に注意してください。複数のタブを持つコンテナ (Accordion、TabNavigator コンテナなど) で、複数のタブに同じ名前を付けることができます。これらのコンテナではタブのラベルから ID が作られるため、ID が重複している可能性があります。

QTP では、ユーザーがレイアウトコンテナ上のイベントを実行しない限り、スクリプト内のレイアウトコンテナは記録されません。これにより、テストオブジェクトのネスト階層が深くなりすぎることを防ぎ、テストスクリプトが読みやすくなります。

アプリケーション階層について詳しくは、「[アプリケーション階層について](#)」(4 ページ) を参照してください。

ボタン、テキストフィールド、スクロールバーなどのコントロールの上にコンテナがある場合、通常、クリックイベントが発生しなくなります。ただし、クリックイベントをトラップする最上位のコンテナに関連付けられているリスナーがあると、コントロールをクリックできなくなります。このような問題を回避するために、コンテナをコントロールの上に配置しないでください。

Repeater オブジェクトの操作

Repeater オブジェクトは QTP オブジェクト階層およびスクリプト内に表示され、Repeater オブジェクトが作成した子オブジェクトが含まれます。実際の Flex アプリケーションでは、これらの子オブジェクトはメインコンテナの子であっても、Repeater オブジェクトの子ではありません。両方のモデルが考慮される QTP のテーブルチェックポイントでは、子オブジェクトはメインコンテナおよび Repeater オブジェクトの子として表示されます。

データ駆動型でリストベースのコントロールの操作

オートメーションで正常に動作するには、リストベースのコントロールのカスタムアイテムレンダラーオブジェクトはデータ駆動型であることが必要です。テーブルのチェックポイントを生成する場合、データは 1 つのアイテムレンダラーにプッシュされ、次に値が照会されます。アイテムレンダラーがデータ駆動型でない場合、正しいデータが生成されません。

リストベースのコントロールに表示されているデータが複数のテスト間で変更される場合には、インデックスに基づいた記録のアプローチを使用してください。この状況では、値に基づいて記録されません。

トラブルシューティング

QTP を使用して Flex アプリケーションをテストする場合、いくつかの一般的な問題とその解決方法があります。一般的な問題は、以下のことを確認すると解決できる場合もあります。

- `include-libraries` フラグを使用して、`automation.swc`、`automation_agent.swc`、`qtp.swc` ファイルをアプリケーションにリンクさせます。(これらのライブラリを含めるか含めないかは、ライブラリを含める場合と含めない場合の SWF ファイルのサイズを比較して判断できます。これらのライブラリとのコンパイルで、アプリケーションのサイズがより大きくなります。)
- Process Explorer (<http://www.microsoft.com/technet/sysinternals/Processesandthreadsutilities.msp> 参照) を使用して、TEAPluginIE.dll および TEAPluginQTP.dll がロードされているかどうかを確認してください。記録が開始されたら、これらの DLL を検索する必要があります。TEAPluginQTP.dll がロードされ、TEAPluginIE.dll がロードされない場合、IE セキュリティ設定に問題がある可能性があります。TEAPluginIE は、QTP エージェントによって作成および使用される ActiveX コントロールを呼び出します。
- Flash Player のログファイルを表示します。ログファイルは、`C:\Documents and Settings\user_name\Application Data\Macromedia\Flex Player\Logs` にあります。user_name は、現在の Windows ユーザー名です。エラーメッセージの有無を確認してください。

一般的なトラブルシューティング

ユーザーの環境と使用しているアプリケーションに関して、以下の情報を確認することで、他の一般的な問題を解決できます。

- ActiveX バージョンの Flash Player 9 のデバッグ版がインストールされています。
- Flex プラグインがインストールされています。QTP では、起動中に現在のプラグインが一覧表示されます。
- テストの自動化をサポートする新しい `frameworks.swc` ファイルを使用して、Flex 3 でアプリケーションがコンパイルされています。
- アプリケーションが Microsoft Internet Explorer バージョン 6 以上で実行されています。
- アプリケーションが、`id` 属性セットを持つ `<object>` タグを使用して、HTML ページ (つまり、ラッパー) を介してロードされています。`id` 属性に、ピリオドやハイフンが含まれません。
- アプリケーションが、Web サーバー、または信頼できる SWF ファイルからローカルにロードされています。

必要なアプリケーションが正しいバージョンかどうかを判断するには、プログラムの追加と削除コントロールパネルに移動し、現在インストールされているプログラムの一覧を確認してください。

QTP で Flex アプリケーションとのやり取りが記録されない場合は、プラグインがインストールされていることを確認してください。最も一般的な問題を以下に示します。

- Flash Player のバージョンが正しくありません。
- Internet Explorer のバージョンが正しくありません。
- Internet Explorer でプラグインが無効になっています。
- QTP プラグインがインストールされていません。
- Flex プラグインが依存関係にある DLL がシステムに存在しません。こうした DLL には、MSVCR71.DLL、MSVCP71.DLL などがあります。

問題は次のような方法で判断できます。

- インストールされているプラグインの一覧を表示します。
- ログファイルを表示します。詳しくは、「[Flash Debug Player を使用したログ機能](#)」(13 ページ) を参照してください。

イベントの記録

QTP でイベントを記録する場合、以下のような問題が発生する場合があります。

- QTP がイベントを記録しない場合は、アプリケーションがオートメーションライブラリを使用してコンパイルされたことを確認します（実行時ロードを使用していない場合）。
- クリックイベントが記録されない場合は、コンテナがコントロールと重なり、クリックがバブリングできなくなっている可能性があります。可能であれば重なっているコンテナを削除します。
- あるコンポーネントをクリックしたときに、クリックイベントが別のコンポーネントで記録されることがあります。この場合、対象のコンポーネントが複合オブジェクトの一部である可能性があります。それを包むオブジェクトが子を正しく公開していないか、所有しているプロパティが正しく設定されていない可能性があります。

ログ機能

問題の原因を判断するために、Flash Debug Player および QTP のログファイルを表示できます。

Flash Debug Player を使用したログ機能

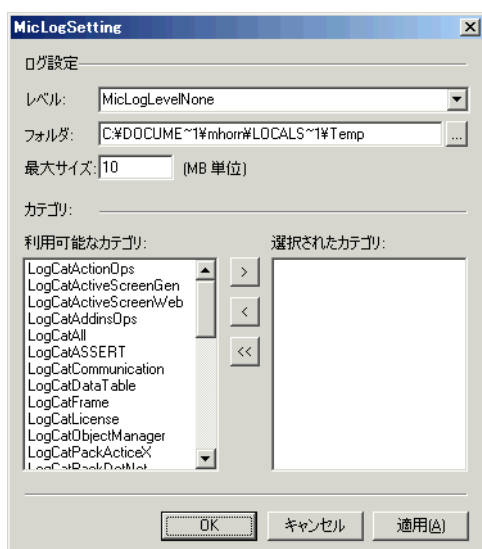
Flash Debug Player は、プラグインと ActionScript のエラーを flashlog.txt ファイルに書き込みます。ファイルのデフォルトの場所は C:\Documents and Settings\ユーザー名です。ログ機能を有効にするには、mm.cfg を設定する必要があります。詳しくは、『Adobe Flex 3 アプリケーションの構築と展開』の「デバッグ版の Flash Player の構成」（161 ページ）を参照してください。

QTP を使用したログ機能

QTP ログ機能を構成して、アプリケーションのテストに関連した QTP エラーを記録できます。ログに記録後、QTP ログファイルを表示して、問題と思われる原因を判断します。

QTP ログファイルの記録を有効にする

- 1 コマンドラインダイアログボックスを開きます。
- 2 QTP の bin ディレクトリに移動します。デフォルトの場所は C:\Program Files\Mercury Interactive\QuickTest Professional\bin です。
- 3 MicLogSetting.exe を実行します。MicLogSetting ダイアログボックスが表示されます。



- 4 レベルドロップダウンリストから、レベルとして「MicLogLevelDebug2」を選択します。
- 5 フォルダフィールドで、ログファイルが保存されているフォルダを選択します。
- 6 「利用可能なカテゴリ」リストから「LogCatPackTEA」を選択し、右矢印ボタンをクリックして「LogCatPackTEA」を「選択されたカテゴリ」リストに移動します。

- 7 「OK」をクリックして、QTP ログレベルへの変更を保存します。
- 8 QTP を実行します。Flex アプリケーションとのやり取りを通常どおりに記録します。
- 9 QTP を終了します。
- 10 潜在的な問題がないか QTP ログファイルを確認します。

ログファイルは C:\Program Files\Mercury Interactive\QuickTest Professional\MicLogFile_QTPro_pid1912_11_32_32.html と同じような場所に生成されます。プラグインが実行中であることを確認するには、以下の内容がログにあるかどうかを調べます。

```
Read Flex environment and matching ProgId TEAPluginQTP.TEAFlexAgentQTP
```

注意: QTP ログファイルは巨大なサイズになる可能性があります。QTP ログファイルは、他の方法では解決できない問題が発生した場合のみに使用することをお勧めします。

再生の同期に関する問題の解決

場合によっては、テストスクリプトの再生が画面上の表示と同期していないことがあります。例えば、QTP は、エフェクトの終了や、オブジェクトのテスト前に返されるデータ結果を待機しません。そのため、QTP で次のようなエラーが記録される可能性があります。

- テストスクリプト再生の整合性が取れません。
- 「詳細不明」のエラーが発生します。
- エラーになってはいけな箇所です。テストスクリプトに深刻なエラーが発生します。

これらのタイミングに関する問題には、以降のセクションで説明するいくつかの解決方法があります。

スクリプトへの待機時間の付加

Wait ステートメントを使用して、指定した秒数だけスクリプトの再生を一時停止できます。スクリプト内の一時停止する操作に、Wait (num_seconds) をパラメータとして渡します。

次の例では、Milk FlexCheckBox コントロールがクリックされる前に 10 秒間待機します。

```
Browser("My Page").FlexApplication("MyApp").FlexCheckBox("Milk").Click Wait(10)
```

Wait ステートメントの使用方法について詳しくは、QTP のマニュアルを参照してください。

オブジェクトが存在するかどうかの確認

コントロールの操作を実行する前に、オブジェクトが存在するかどうかを確認できます。これには、Exist ステートメントを使用できます。次の例では、Milk FlexCheckBox コントロールをクリックする前にその存在を確認します。

```
If Browser("My Page").FlexApplication("MyApp").FlexCheckBox("Milk").Exist Then
    Browser("My Page").FlexApplication("MyApp").FlexCheckBox("Milk").Click
End If
```

コントロールの Exist 共通プロパティの値を確認することもできます。Exist ステートメントおよび Exist 共通プロパティについて詳しくは、QTP のマニュアルを参照してください。

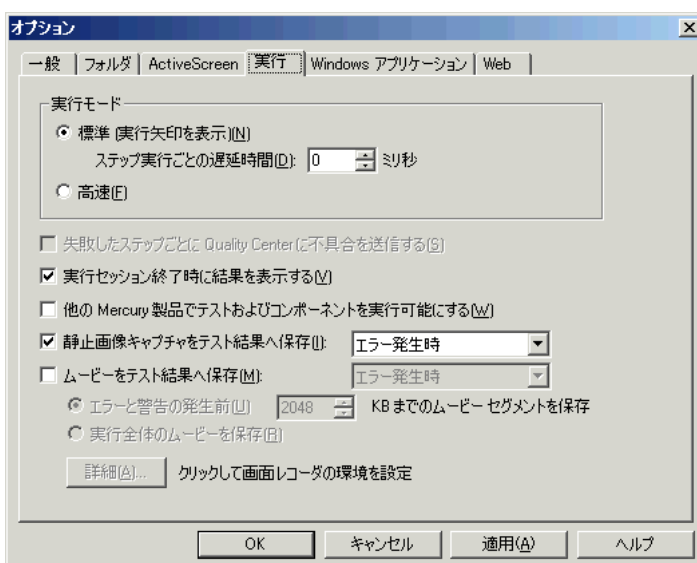
グローバルな再生時間の遅延

グローバルな実行遅延を設定して、すべてのイベントの再生速度を遅らせるように QTP を設定できます。この方法は最後の手段として検討してください。まず、「スクリプトへの待機時間の付加」(14 ページ)の説明にあるように、テスト内で問題のある箇所の前に Wait ステートメントを挿入してみます。大規模なテストスクリプトでは、各イベントの起動時間が増加すると、たとえそれが数分の 1 秒であったとしても、テスト全体の実行時間が大幅に増加する可能性があります。そのため、スクリプトが適切に実行される範囲で最小値を見つけるようにしてください。

グローバルな遅延を付加することを決定した場合は、QTP を設定して、スクリプト内の各ステップをミリ秒単位で遅らせることから始め、エラーが発生しなくなるまで徐々に遅延時間を増加させていきます。

テスト内の各ステップに遅延を付加する

- 1 ツール/オプションを選択します。
- 2 オプションダイアログボックスで、「実行」タブを選択します。



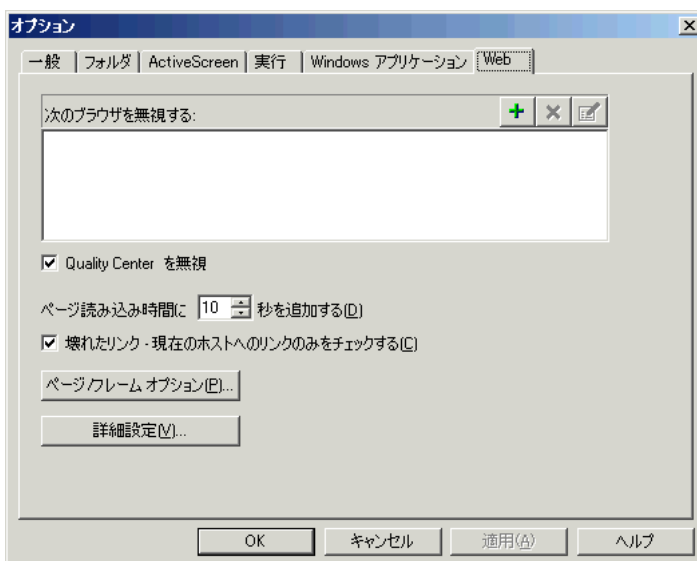
- 3 「実行モード」で「標準」オプションを選択します。これはデフォルトのオプションです。
- 4 「ステップ実行ごとの遅延時間」オプションの値を数ミリ秒単位で増加させます。デフォルト値は 0 です。
- 5 「OK」をクリックして変更内容を保存します。
- 6 テストに戻ります。それでも問題が発生する場合は、遅延時間を増加させてテストを再実行します。

起動テスト時間の遅延

アプリケーションがブラウザ内で起動するときに、アプリケーションを変更または再コンパイルする場合は、ブラウザの起動からテスト開始までの遅延を考慮してください。テストが直ちに開始されアプリケーションがすべてのオブジェクトのインスタンス化を完了していない場合、QTP がエラーを記録する可能性があります。

テストの起動の遅延

- 1 ツール/オプションを選択します。
- 2 オプションダイアログボックスで、「Web」タブを選択します。



- 3 「ページ読み込み時間に n 秒を追加する」オプションの値を、QTP がテストの実行を開始する前にアプリケーションが確実にコンパイルされるのに十分な値まで増加させます。デフォルト値は 10 です。

テストが失敗しても最後まで実行

スクリプトからテストの成功または失敗を判断できず、スクリプトを最後まで実行する状況が発生する場合があります。特に、この状況はデータテーブル内の式を変更するテストについて報告されてきました。

テストが失敗しても最後まで実行させる QTP メソッドを記述するには、次の例に示すようにキーワードを作成します。

```
Public Function WidthGreaterThan(test_object, minWidth)
actual = test_object.GetROProperty("width")
a = CInt(actual)
b = CInt(minWidth)
If a > b Then
Msg = "WidthGreaterThan", "Width is greater than: " &minWidth
Reporter.ReportEvent micPass, Msg
Else
Msg = "Width " & actual & " is not greater than " & minWidth
Reporter.ReportEvent micFail, "WidthGreaterThan", Msg
End If
End Function
RegisterUserFunc "FlexButton", "WidthGreaterThan", "WidthGreaterThan"
Browser("FlexStore").FlexApplication("flexstore").FlexCanvas("Products").
FlexButton("Remove from cart_11").WidthGreaterThan 10
Browser("FlexStore").FlexApplication("flexstore").FlexCanvas("Products").
FlexButton("Remove from cart_11").WidthGreaterThan 1099
Browser("FlexStore").FlexApplication("flexstore").FlexCanvas("Products").
FlexButton("Remove from cart_11").WidthGreaterThan 100
```

Flex を使用した自動化されたテストの制限

Flex と QTP との統合においては、以下の制限があります。

- Flex のテストオブジェクトは、オブジェクトの認識ダイアログボックス（ツール／オブジェクトの認識）に表示されません。このため、一般的には、QTP がオブジェクト `id` を取得する方法の設定、または QTP がどのプロパティを保存するかをカスタマイズしません。Flex が "TEAFlex.xml" ファイル内のオブジェクトを識別するために使用するプロパティを設定できます。
- Web 設定（ツール／Web イベント記録の設定）は Flex アプリケーションには適用されません。ここでは、マウスイベントの反応を指定できます。例えば、クリック操作に対して、`mouseDown` と `mouseUp` 操作の組み合わせとして記録するように QTP に指示します。
- Flex のテストオブジェクトは、『QuickTest Object Model Reference』（ヘルプ／QuickTest Professional ヘルプ／「目次」タブ／QuickTest Object Model Reference）には含まれません。このリファレンスには、すべてのテスト可能なメソッドとプロパティ、サンプル、ID プロパティの一覧が表示されます。Flex のすべてのテストオブジェクト、イベントおよびイベントの値が記述されている `tea.html` ファイルを参照できます。