

NEW

Visual Studio.NET/.NET Framework関連ツール新製品レビュー

PRODUCTS



QuickTest Professional 6.0

.NET対応の機能テスト自動化ツール



株式会社テクノジックアート
http://www.tech-arts.co.jp/
岡本 正輝 OKAMOTO, Masaki

問
合
先

マーキュリー・インタラクティブ・ジャパン株式会社

TEL : 03-5402-9300

URL : http://www.mercury.co.jp/products/quicktestpro/

FAX : 03-5425-2288

MAIL : info_jp@mercuryinteractive.com

Technology Tools

- Visual Basic .NET
- Visual C# .NET
- SQL Server 2000
- Oracle 9i
- Access 2002
- ASP.NET
- Internet Information Services
- Other:

Visual Studio .NET 2002
(.NET Framework 1.0)

Environment

ターゲットOS
Windows 98/Me/NT4.0/2000/XP

テストサポート環境
開発環境/プラットフォーム:
.NET (Windows フォーム/ Web フォーム)、
J2EE

Web関連テクノロジー:
HTML、DHTML、JavaScript、XML、
WSDL、UDDI

ブラウザ:
IE5.0以降、Netscape 4.7以降、
AOL6.0

ERP/CRM:
Oracle E-Business Suite他

マルチメディア:
Flash 5.0他

*) QuickTest Professional 6.0 (以下QTP)
は、.NET Framework 1.0のみ対応です。.NET
Framework 1.1対応は、QTP6.5 (2004年
1月30日提供開始) からになります。

価格

QuickTest Professional 6.0 :
840,000円
(インストールPC固定、1ユーザー) ~

QuickTest Professional .NET Add-In 6.0 :
200,000円
(インストールPC非固定、1ユーザー) ~

機能テストとは？

今回みなさんにご紹介するのは、「QuickTest Professional 6.0」(以下QTP)という、機能テストの自動化を促進する製品です。さて、「機能テストの自動化」と聞いて、ピンと来ない方もいらっしゃるのではないのでしょうか。そこで、製品の説明に先立ち、まずはこのキーワードを詳しく取り上げてみたいと思います。

そもそも“機能テスト”と言われても、どのようなテストを指しているのか明確でないように思います。そこで、本稿でいう“機能テスト”とは「ユーザーの認識できる機能をテストすること」つまり、「受け入れテストの際に行なわれる機能テストを指すもの」とします。

みなさんが何らかの検索機能をテストする場合を想像してみてください。「検索画面でキーワードを入力して、検索ボタンを押し、その検索結果をチェックする」というようなテスト内容を思い浮かべるのではないのでしょうか。今挙げたような、画面を介するテストは“機能テスト”の具体例です。

自動化の必要性

では“機能テストの自動化”について考えてゆきましょう。まずは、現在までのシステム開発で行なわれてきた機能テストを思い出してみてください。.NET Frameworkを基盤におくシステムの構成としては、WebベースまたはC/S形式のものが大半を占めています。このような構成のシステムであれば、当然、操作画面にはGUIを採用することになるでしょう。

さて、このようなGUIを備えたシステムに対して機能テストを行なう場合、みなさんは、どのようなテスト手法を採ってききましたか。

「テスターが画面を操作し、その操作に対する表示内容を目で確認する」

といった方法ではないのでしょうか。もし、このような手動のテスト手法を採った場合、テスターが画面の操作手順を間違えたり、あるいは表示内容の正/不正の判断を誤ったり、といったミスを生む危険性があります。これらのような人為的ミスは、テストの精度低下を招き、システムの品質向上を難しくします。

では、このような問題には、どのように対処すれば良いのでしょうか。実は、QTPのような「機能テスト自動化ツール」(以降“テストツール”と呼ぶ)を用いると、このような問題を解決に導くことが容易になるのです。

■自動テスト手法の特徴

ここまでで、手動テスト手法の問題点を、ある程度把握していただけたかと思えます。今度は、自動テスト手法の特徴を見てみましょう。

特徴1▶ 画面操作を正確に行なえる

特徴2▶ 表示内容についての検査を正確に行なえる

特徴3▶ テストツール向けにテスト内容をカスタマイズする必要がある

3つ目に挙げた特徴ですが、これは自動テスト手法において注意すべき点だと言えます。ただし、テストツールとしてQTPを採用した場合には、この点が大きな問題となることは少ないでしょう(後述しますが、QTPではこの作業の負荷を減少させる工夫がなされています)。

QuickTest Professionalの機能

では、さっそくQTPについて見てみましょう。QTPは、以下の3つの特徴を持っています。

特徴 1 テスト作成のためのGUI環境を提供

QTP向けのテスト手順(以降“テストスクリプト”と呼ぶ)を作成するためのGUI環境を提供しています。この環境により、よほど複雑なテスト内容でない限りは、マウス操作主体のシンプルなユーザーインターフェ

イスのみを介して、テストスクリプトを作成することができます。

特徴 2 テストを拡張するための多彩な機能

QTPでは、一度作成したテストスクリプトに対して後からテスト内容を追加し、テストそのものを拡張することができます。詳細については、後述します。

特徴 3 テスト結果のレポートを自動生成

テストスクリプトを自動実行した後に、その結果をレポートとして生成する機能を提供しています。

なお、QTPを利用して.NETアプリケーション(VB.NET/C#/C++)のテストを行なうには、QTPの他に、.NET用アドイン「QuickTest Professional .NET Add-In 6.0」(別売り)が必要です。

以降では、QTPおよびQuickTest Professional .NET Add-In 6.0を開発用マシンにインストール済みという前提で話を進めてゆきます。

■テスト作業の流れ

では、どのような流れでテストを行なえば良いのでしょうか。テストツールを用いたテスト作業は、おおまかには表1のようになります。

この流れを踏まえ、QTPを使用した場合の手順を見てみましょう。QTPでのテスト作業は、以下の5つの段階に分かれます。

表1: テストツールを用いたテスト作業の流れ

テスト作業	概要
①テストの作成	アプリケーションに対する一連の操作をツールに記録させる。記録した操作と操作の間に、表示データの検査を設定する
②テストの実行	作成したテストを、ツールに自動実行させる
③テスト結果の確認	ツールにより出力されたテスト結果を確認する

作業 1 操作を記録するための準備

QTPとテスト対象アプリケーションの諸設定について、そのときのテスト内容に合っているかどうかを確認します。

作業 2 アプリケーションに対する操作の記録

ある特定の機能をテストするためには、ある特定の画面に行かなくてはなりません。さらにその画面上で、チェックボックスやオプションボタンを適切に設定し、機能呼び出さなくてはなりません。QTPを利用して、それらの操作を記録します。

作業 3 テストの拡張

操作の記録を終えた時点では、“テストの枠組みができあがった”にすぎません。このテストの枠組みに対して、QTPに多数用意されている機能を利用してさまざまな拡張を行ないます。たとえば、「画面表示中のデータへの検査を、操作と操作の間に追加する」といったテスト拡張はその一例です。つまりこの作業は、手動テストの際に、「テスターが目で見えて確認していた部分」に当たります。

現段階ではたいしたメリットは感じないかもしれませんが、実はこの例はQTPが提供する強力なテスト拡張のひとつです(もちろんその理由は、“見間違いを避けられる”というだけではありません)。この点については、以降の「実際にテストしてみよう」で確認することにします。また、この他のテスト拡張についても後述します。