

■ 発行日　2012/6/21
■ 発行元　株式会社 アープ

本書は、Web でのテスト自動化における Test Design の一連の操作方法まとめたものです。
Test Design のメニューの説明やより詳細な使い方については、「ユーザーズガイド」を参照してください。

目次

1. はじめに 1
2. 環境構築 2
2.1. Seleniumのサイトについて 2
2.2. Selenium IDEのダウンロード 3
2.3. Selenium Client Driversのダウンロード 3
3. Test Designプロジェクトの作成 4
3.1. Test Designプロジェクトの作成について 4
3.2. Selenium Client Driversの設定 5
4. Webアプリケーションのテストコード作成とテスト 7
4.1. 初期表示 7
4.2. コードの作成 8
4.3. テスト実行 11
5. コード自動生成 12
5.1. Selenium IDE Test Design formatterのセットアップ 12
5.2. Selenium IDE でコード生成 14
6. サンプルコード 18
6.1. 「Test1」サンプル 18
6.2. 「Test2」サンプル 21
7. 参考文献 23

1. はじめに

Test Design は Selenium(※1) を使って Web アプリケーションのテストを行うことができます。
C#で記述されたコードから Selenium の「Selenium Client Drivers」ライブラリを実行することで、Web ブラウザに対して文字列の入力や、ボタンのクリック、リンクのクリック等を行います。また、Web サーバーからのレスポンスを取得して期待値との違いをチェックし、ログ出力・画面キャプチャーの取得等を行うことができます。

このチュートリアルでは、Test Design から Web アプリケーションをテストする方法について、サンプルコードを用いて説明します。

◎Web でのテストは、以下の手順で行います。

１）環境構築
- Selenium IDE のダウンロード
- Selenium Client Drivers のダウンロード

２）Test Design のプロジェクト作成
- Test Design のプロジェクト作成
- Selenium Client Drivers の設定

３）Web アプリケーションのテストコード作成とテスト実行

※1 Selenium については参考文献を参照。

◆本書で例として使用しているブラウザは「Firefox」です。
OS は Windows7 を使用しています。

2. 環境構築

本章では、Test Design から Web をテストするための環境構築について説明します。
以下に本章の要点を示します。

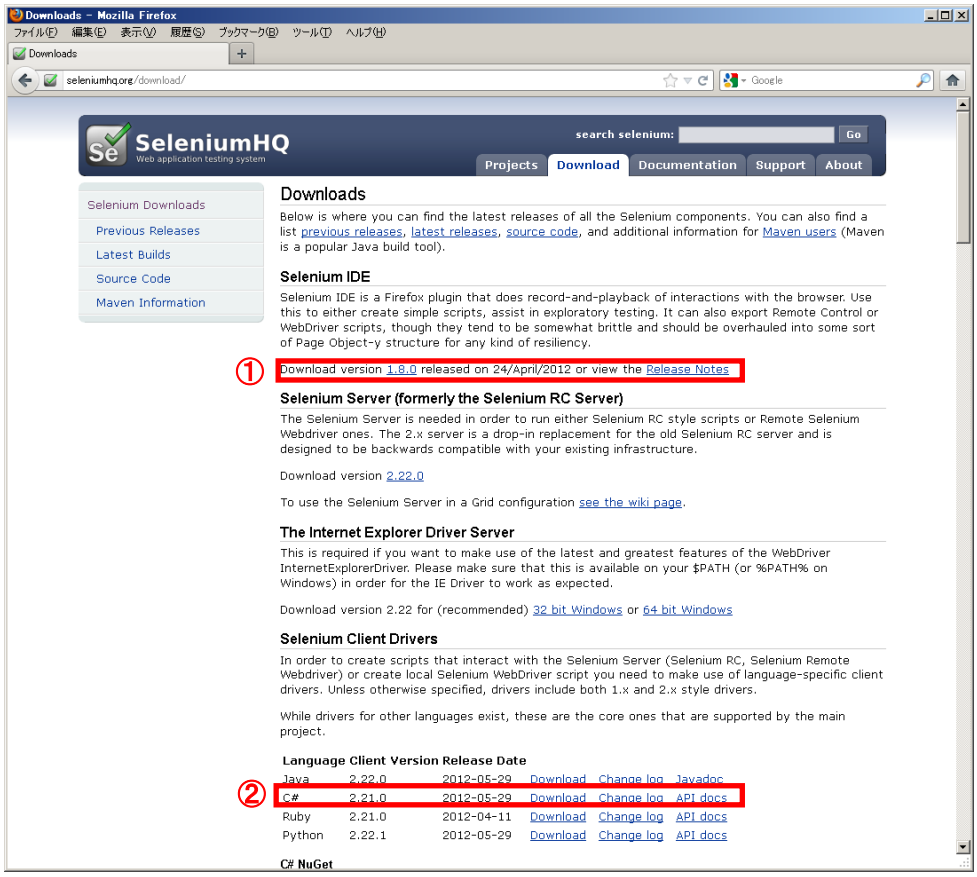
①Selenium IDE のダウンロード
②Selenium Client Drivers のダウンロード

2.1. Selenium のサイトについて

Selenium は Web ブラウザに対して自動的に文字列の入力や、ボタンのクリック、リンクのクリックを行うテストツール群です。

Web ブラウザから以下のサイトにアクセスし、「Selenium IDE」「Selenium Client Drivers」のダウンロードを行ってください。

http://seleniumhq.org/download/

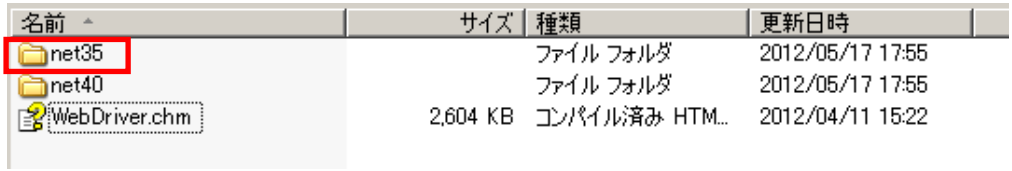


2.2.Selenium IDE のダウンロード

環境構築では、初めに「Selenium IDE」のインストールを行います。
Selenium サイトの Selenium IDE の項目内にある「Download version ...」リンクからダウンロードできます（①）。
リンクをクリックすると、「selenium-ide-x.x.x.xpi」がダウンロードされ、自動的にインストールされます。
Selenium IDE を利用すると、WEB サイトから Test Design 用のテストコードを自動生成できます。(Selenium IDE を利用したコード作成は後述)

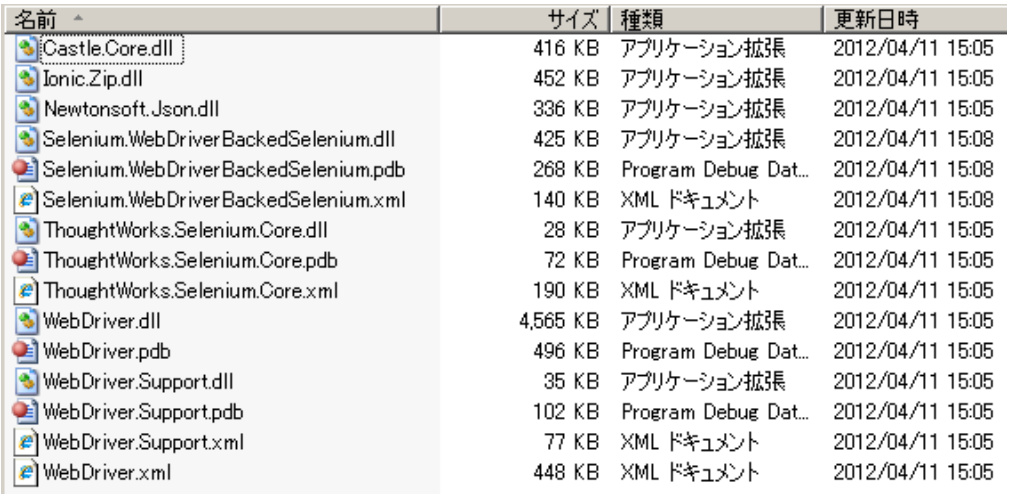
2.3.Selenium Client Drivers のダウンロード

次に、「Selenium Client Drivers」のダウンロードを行います。
Selenium サイトの 「Selenium Client Drivers」 の項目内にあるリンクからダウンロードできます。「C#」版をダウンロードしてください。(②)
「selenium-dotnet-2.xx.0.zip」をダウンロード後、zip ファイルを解凍してください。
「net35」フォルダ内のファイルを Test Design で使用します。



名前	サイズ	種類	更新日時
net35		ファイル フォルダ	2012/05/17 17:55
net40		ファイル フォルダ	2012/05/17 17:55
WebDriver.chm	2,604 KB	コンパイル済み HTM...	2012/04/11 15:22

(Selenium-dotnet-1.1.8.zip の展開後)



名前	サイズ	種類	更新日時
Castle.Core.dll	416 KB	アプリケーション拡張	2012/04/11 15:05
Ionic.Zip.dll	452 KB	アプリケーション拡張	2012/04/11 15:05
Newtonsoft.Json.dll	336 KB	アプリケーション拡張	2012/04/11 15:05
Selenium.WebDriverBackedSelenium.dll	425 KB	アプリケーション拡張	2012/04/11 15:08
Selenium.WebDriverBackedSelenium.pdb	268 KB	Program Debug Dat...	2012/04/11 15:08
Selenium.WebDriverBackedSelenium.xml	140 KB	XML ドキュメント	2012/04/11 15:08
ThoughtWorks.Selenium.Core.dll	28 KB	アプリケーション拡張	2012/04/11 15:05
ThoughtWorks.Selenium.Core.pdb	72 KB	Program Debug Dat...	2012/04/11 15:05
ThoughtWorks.Selenium.Core.xml	190 KB	XML ドキュメント	2012/04/11 15:05
WebDriver.dll	4,565 KB	アプリケーション拡張	2012/04/11 15:05
WebDriver.pdb	496 KB	Program Debug Dat...	2012/04/11 15:05
WebDriver.Support.dll	35 KB	アプリケーション拡張	2012/04/11 15:05
WebDriver.Support.pdb	102 KB	Program Debug Dat...	2012/04/11 15:05
WebDriver.Support.xml	77 KB	XML ドキュメント	2012/04/11 15:05
WebDriver.xml	448 KB	XML ドキュメント	2012/04/11 15:05

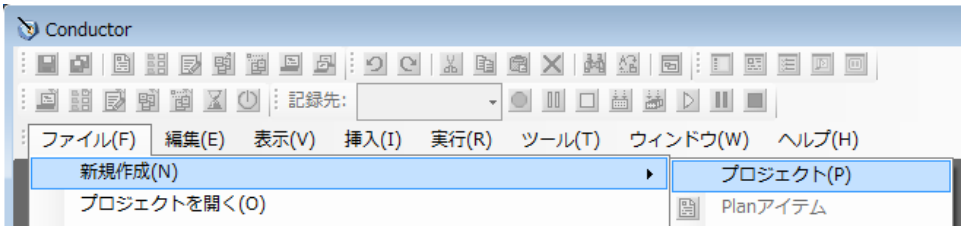
(net35 フォルダ内のファイル)

3. Test Design プロジェクトの作成

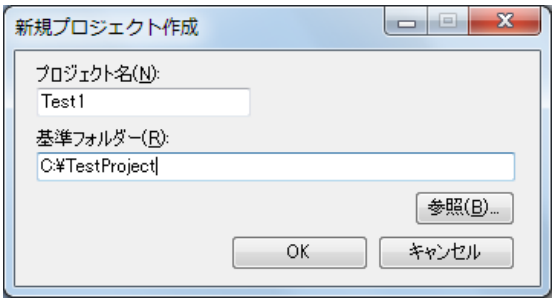
環境構築後、Selenium Client Drivers の設定をします。
Web のテスト用に Test Design プロジェクトを作成し、Selenium Client Drivers の設定を行ってください。

3.1. Test Design プロジェクトの作成について

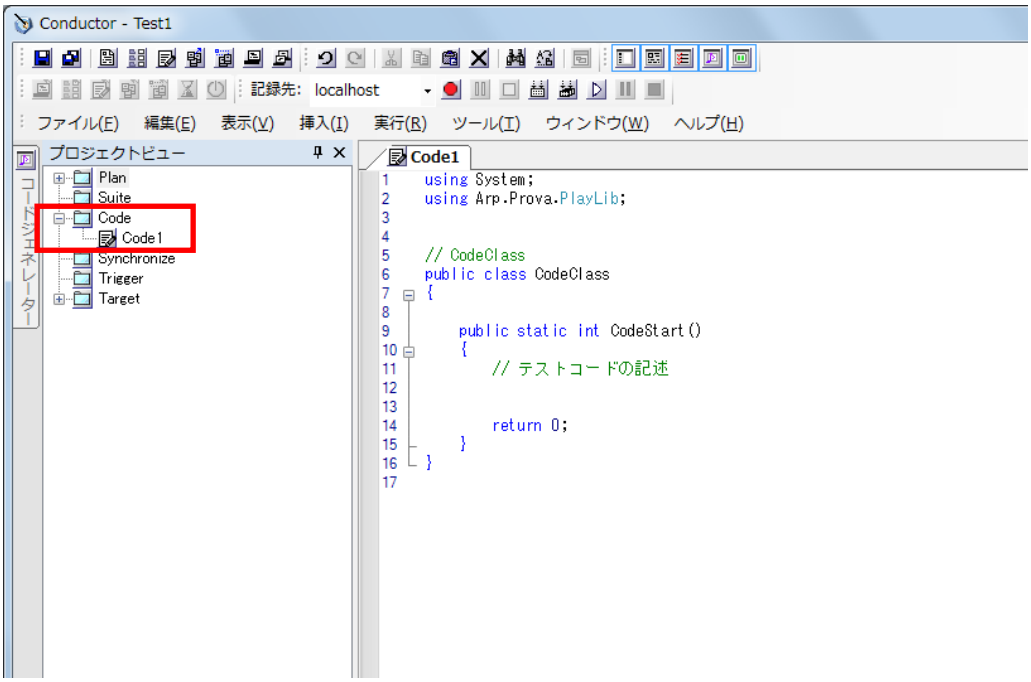
(Step 1) Conductor を起動し、プロジェクトを新規作成します（ファイル＞新規作成＞プロジェクト）。
Conductor については、「ユーザーズガイド」または Conductor のヘルプを参照してください。



(Step 2) プロジェクト名を入力し、プロジェクトを保存する基準となるフォルダを選択します。
例として、プロジェクト名に**[Test1]**、基準フォルダに**[C:¥TestProject]**を入力し[OK]ボタンを押下します。
[C:¥TestProject¥Test1]以下にプロジェクトが作成されます。



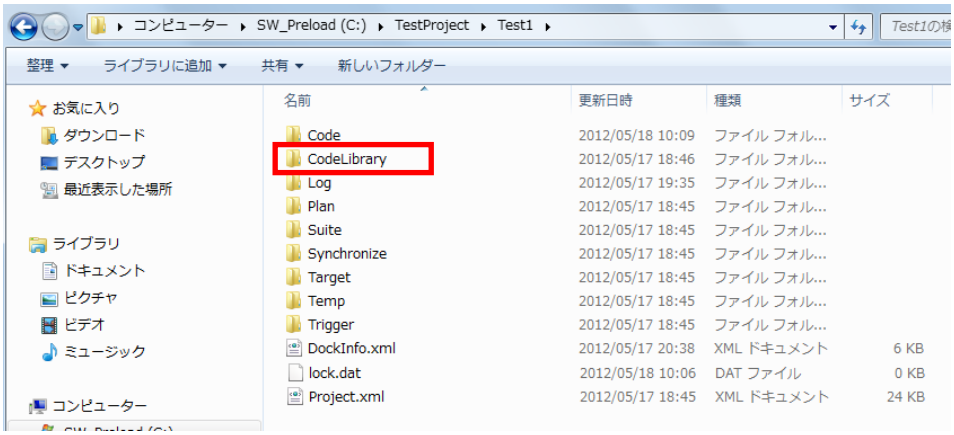
(Step 3)プロジェクトができると、Conductor の Code フォルダに初期コード（Code1）が表示されます。



3.2.Selenium Client Drivers の設定

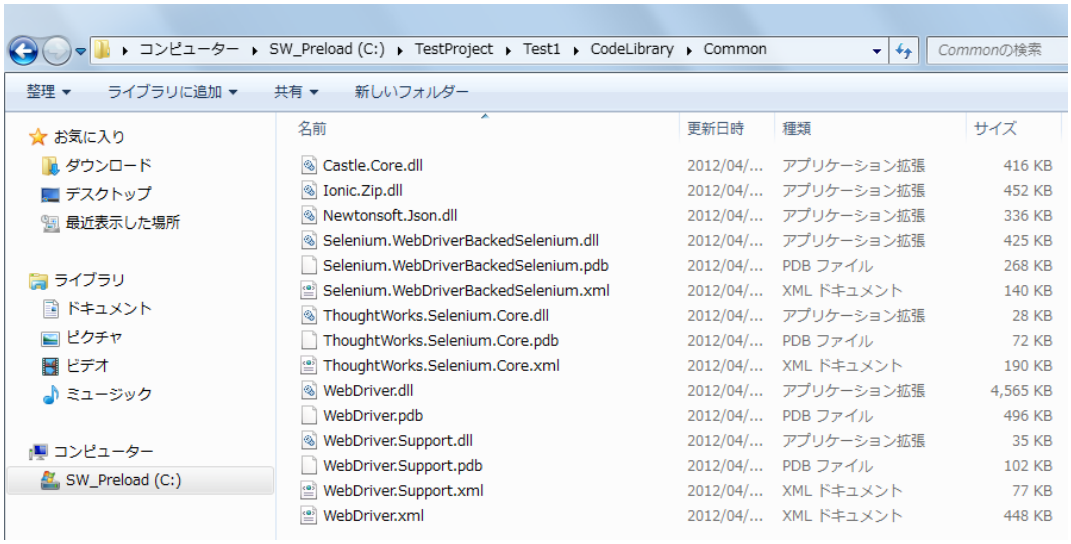
Selenium Client Drivers を Test Design のプロジェクトに保存します。

(Step 1) Test Design プロジェクトの作成で作成したプロジェクトの CodeLibrary フォルダ内に、Common フォルダを作成してください。



(Step 2) Selenium Client Drivers のダウンロードでダウンロードしたファイルを、Common フォルダへ、コピーしてください。

例では、**[C:¥TestProject¥Test1¥CodeLibrary¥Common]**に Selenium Client Drivers がコピーされます。



（コピー後の CodeLibrary¥Common フォルダ）

4. Web アプリケーションのテストコード作成とテスト

本章では、テストコードを作成し、Web アプリケーションのテストを実行します。
テストコードは、言語 C#を使って作成します。作成したテストコードは Player を使って実行することができます。(Player については、「ユーザーズガイド」、Player のヘルプを参照してください)

以下に本章の要点を示します。

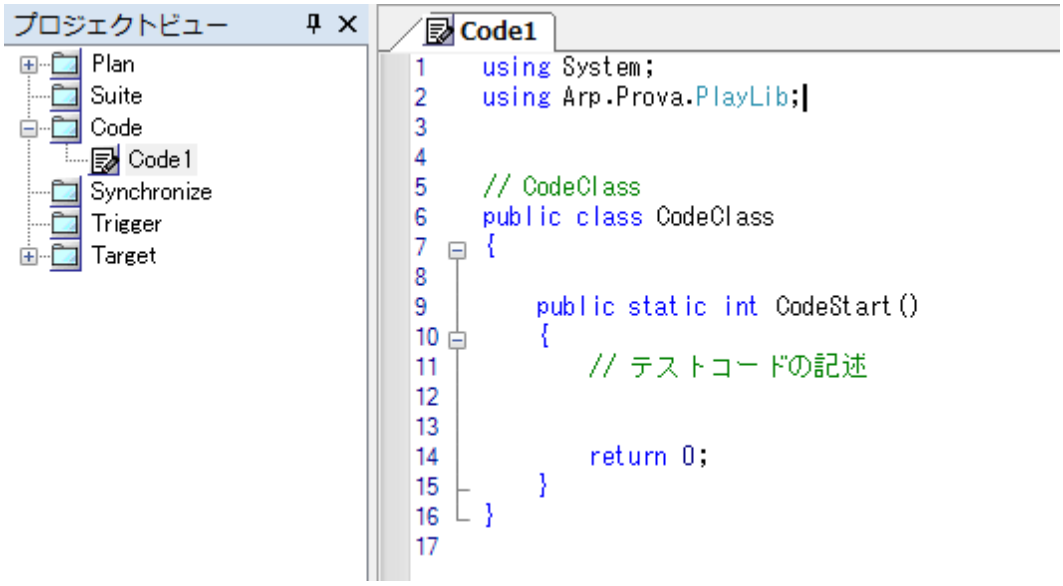
①テストコード作成方法

②テストコード実行

4.1. 初期表示

プロジェクト作成後、プロジェクトビューの Code フォルダに[Code1]があります。テストコードは、[Code1]もしくは新規作成した Code アイテムに追記する形で作成します。(Code アイテムについては「ユーザーズガイド」を参照してください。)

初期状態は以下のようになっています。



4.2.コードの作成

サンプルを作成します。サンプルの内容は、以下の通りです。

①Test Design のトップページを表示



②[Test Design とは]をクリックし、製品の紹介ページを表示します。

③ブラウザを閉じます。

(Step 1)ファイル Code1 に下記の様にコードを記述してください。

1)using 記述

2)Selenium ドライバ変数の記述

3)メソッド名の記述

以上記述したサンプルを示します。

```
using System;
using System.IO;                        // 1)
using System.Text;                      // 1)
using Arp.Prova.PlayLib;
using OpenQA.Selenium;                  // 1)
using OpenQA.Selenium.Firefox;          // 1)
using OpenQA.Selenium.Support.UI;       // 1)

public class CodeClass
{
    private static IWebDriver driver;   // 2)
    private static string baseURL;      // 2)

    public static int CodeStart()
    {
        // テストコードの記述
        Process();                      // 3)
        return 0;
    }
```

(Step 2)Web アプリケーションを実行するプログラムを記述します。前述したサンプルの下に追記してください。
本サンプルプログラムは、ブラウザは Firefox を使用しています。

4) FirefoxDriver をインスタンス化します。(※2)
5) テストするサイトの URL をセットします。
6) Firefox を表示し、サイトを表示します。
7) “Test Design とは”と記述されたリンクを探しクリックを行います。
8) Firefox を閉じます。

以上の内容を記述したサンプルを示します。

```
    //
    //  Test Design Process Code(C#)
    //
    private static void Process()
    {
        try
        {
            driver = new FirefoxDriver();                          // 4)
            baseURL = "http://www.testdesign.jp/";                 // 5)
            driver.Navigate().GoToUrl(baseURL);                    // 6)
            driver.FindElement(By.LinkText("Test Design とは")).Click();    // 7)
        } catch ( Exception e ) {
            PlayLib.TestLog( e.Message );
        } finally {
            if( driver != null )
            {
                driver.Quit();                                     // 8)
            }
        }
    }
}
```

※2
Firefox 以外にも、Driver は提供されていますが、本チュートリアルでは Firefox を推奨します。
その他・・・InternetExplorerDriver (Windows のみ)
ChromeDriver
など

4.3. テスト実行

コードが出来上がったら、Test Design の実行を行います。「実行ボタン」を押下してテストを実行します。



以下のページが順に表示されます。

①http://www.testdesign.jp を表示します。



②「Test Design とは」リンクを押下します。



5. コード自動生成

本章では Selenium IDE を使って、Test Design のコードを自動生成する方法を示します。

◆Selenium IDE は Firefox ブラウザ上のみで動作します。

以下に本章の要点を示します。

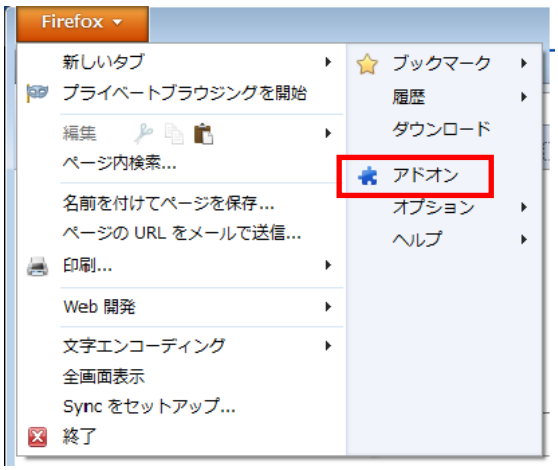
①Selenium IDE　Test Design formatter のセットアップ

②Selenium IDE でコード生成

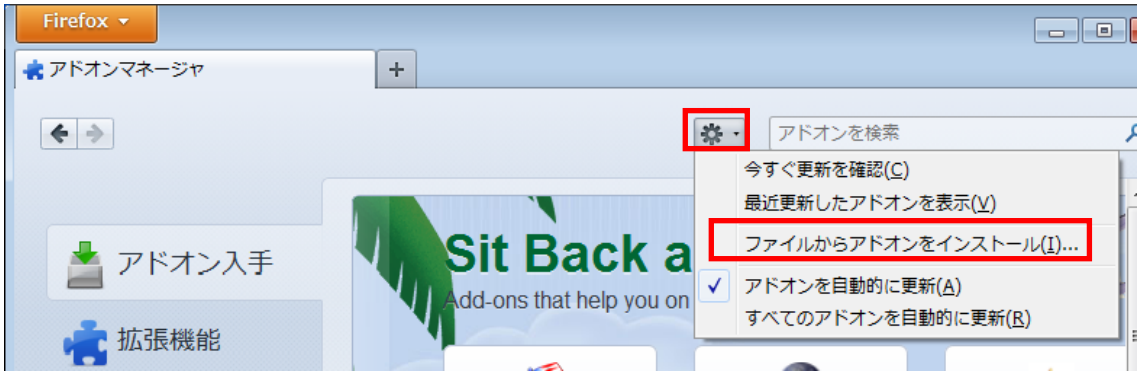
5.1. Selenium IDE Test Design formatter のセットアップ

(Step 1) Firefox ブラウザからアドオンマネージャの画面を表示します。

［メニュー］→［アドオン］

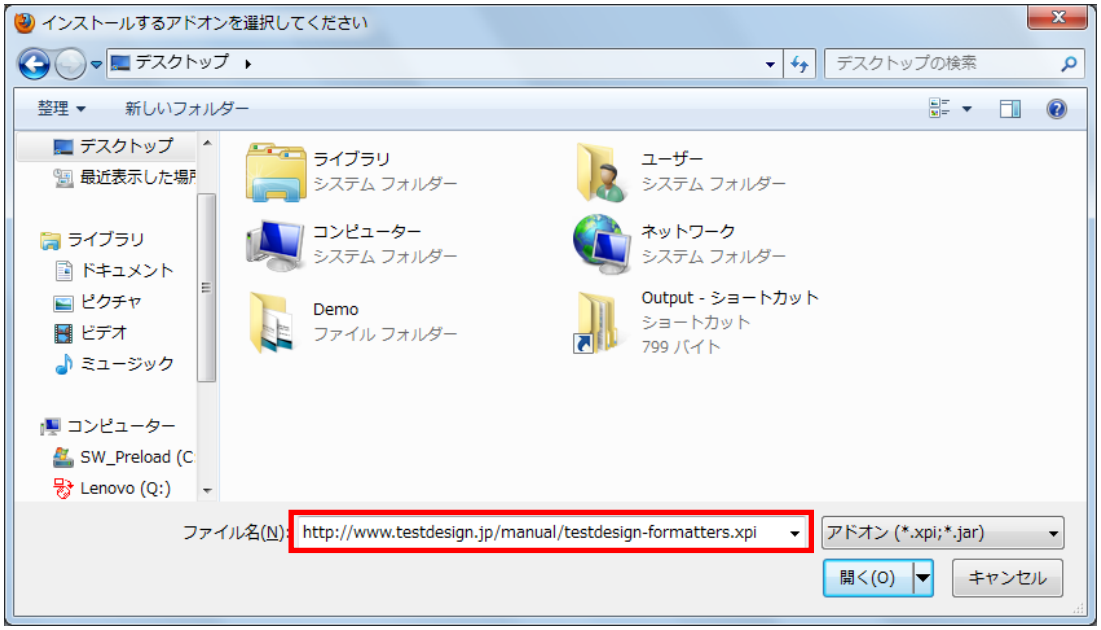


(Step 2)「アドオンツール」を押下し、「ファイルからアドオンをインストール」を選びます。

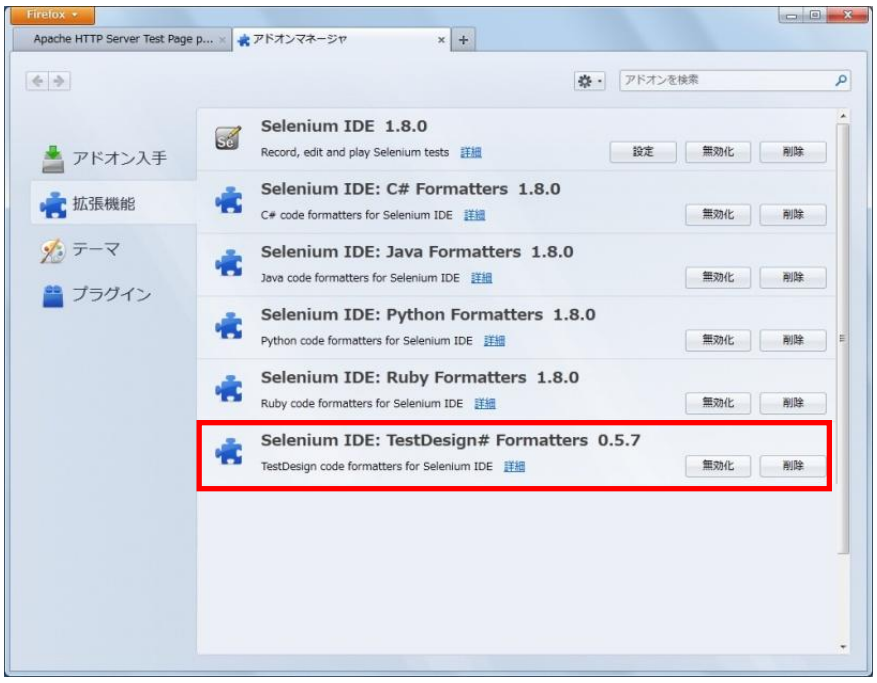


(Step 3)「インストールするアドオンを選択してください」ウィンドウが表示されたら、以下のＵＲＬを「ファイル名」に指定して「開く」ボタンを押下してください。インストールが始まります。
Firefox の再起動を促すメッセージが表示されます。指示に従ってください。

http://www.testdesign.jp/manual/testdesign-formatters.xpi



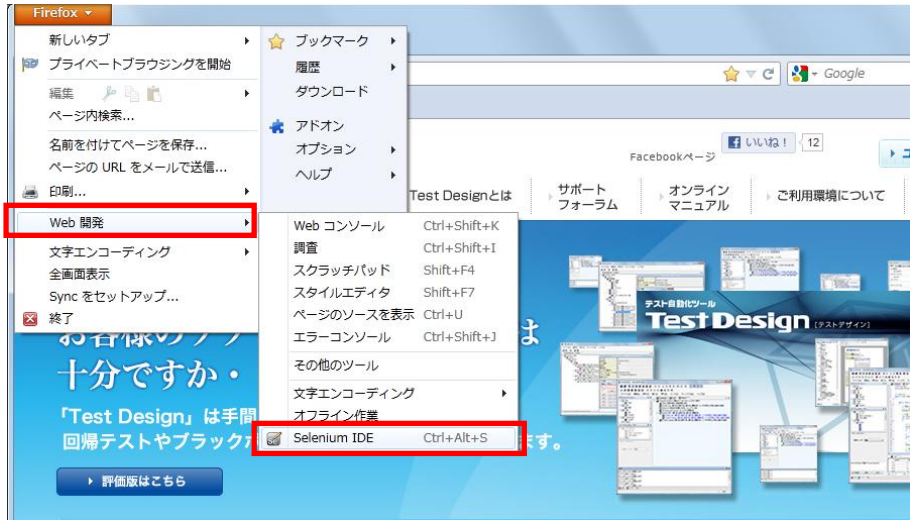
(Step 4)インストール後、「拡張機能」の画面に、「Selenium IDE: TestDesign# Formatters」が追加されます。これで、Selenium IDE を利用してコードを自動生成できるようになりました。



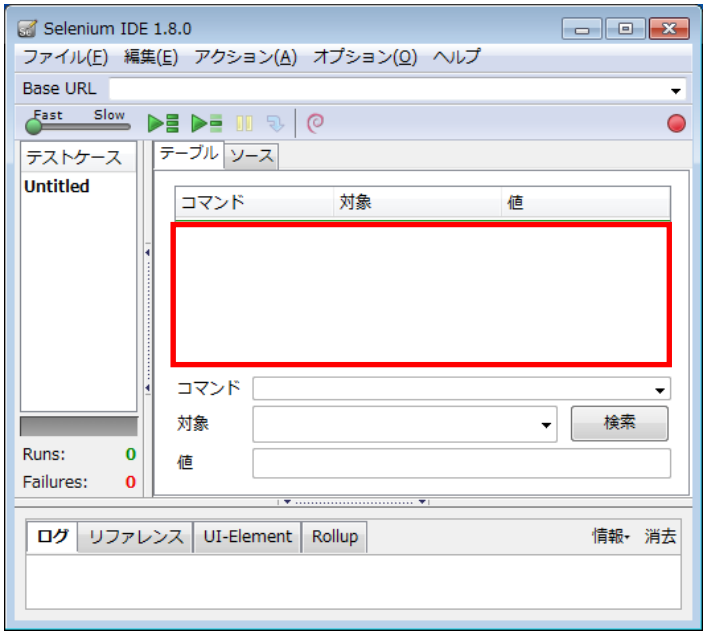
5.2.Selenium IDE でコード生成

Selenium IDE を起動し、Web アプリケーションを操作します。操作した内容が記録されるので、それをエクスポートして、Test Design のコードを生成することができます。

(Step 1) Firefox のメニューから、Selenium IDE を実行します。
[Web 開発] → [Selenium IDE]



(Step 2) Selenium IDE が起動します。ブラウザの操作を行って記録をとります。



●Selenium IDE を起動すると、自動的に「記録モード」になります。
ブラウザにアクセスして、記録したい操作を行ってください。赤枠部分に記録した内容が表示されます。(※自動記録できない操作もあります)

例として、以下の操作を記録します。

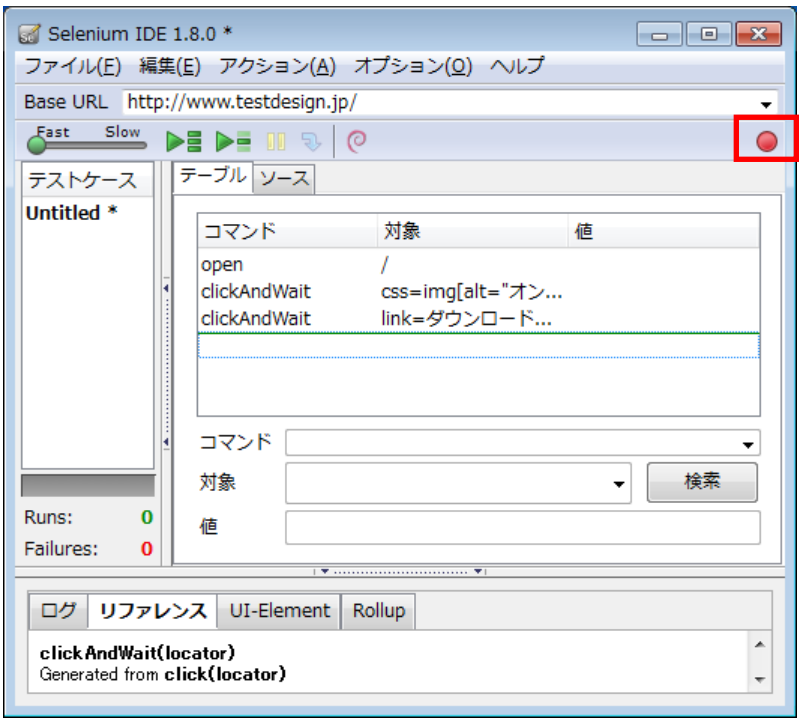
例）http://www.testdesign.jp のサイトをアクセス

↓

「オンラインマニュアル」をクリック

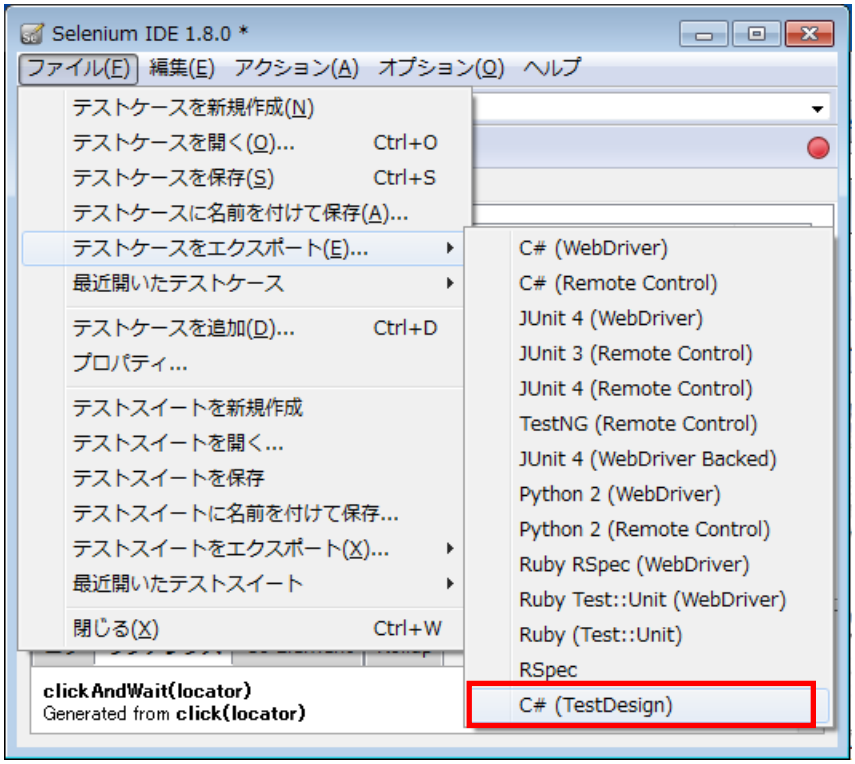
↓

「ダウンロードページへ」をクリック

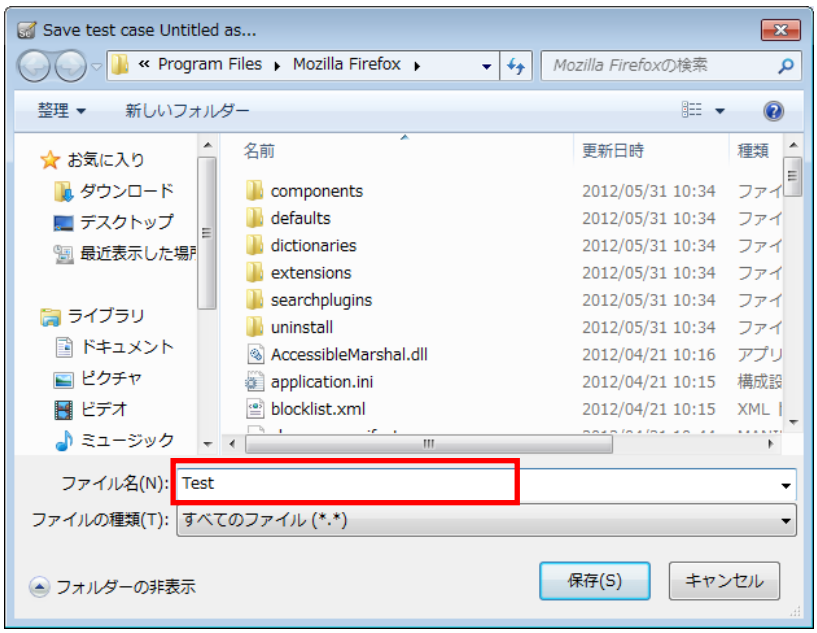


をクリックし、自動記録を終了します。

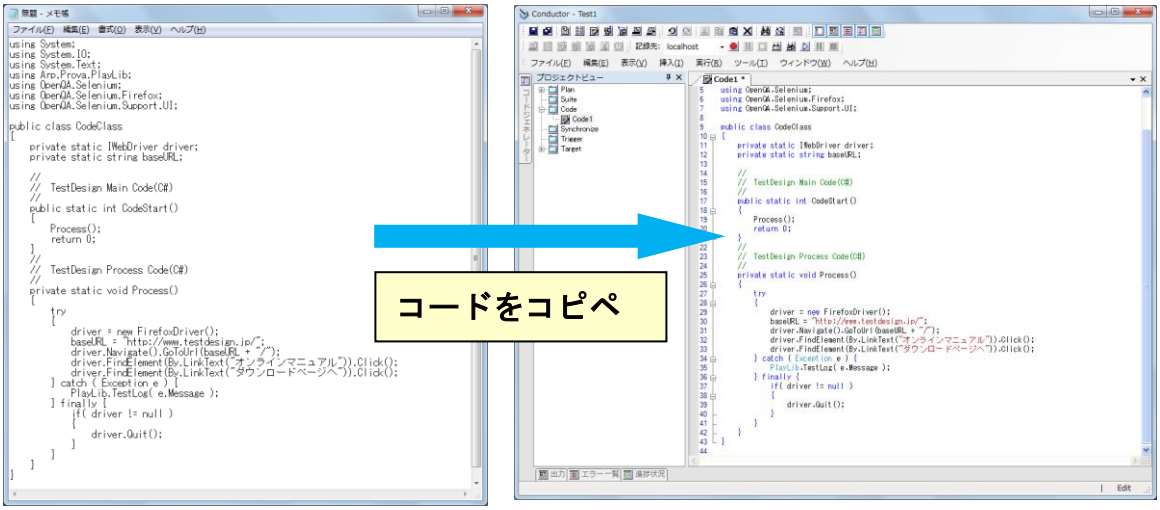
(Step 3) 記録した内容を、Test Design のコードとしてエクスポートします。
［ファイル］→［テストケースをエクスポート］→［C#(Test Design)］を選択。



ファイル名をつけて保存します。
例）ファイル名を「Test」とします。



(Step 4) エクスポートしたソースコードを Test Design へ登録します。
(Step 3)でエクスポートしたファイル（1）をメモ帳等のエディタで開き、ソースコードを Test Design の Code(2)へコピー＆ペーストします。



(1)　　(2)

(Step 5) コード作成後は、Conductor からテストの実行を行ってください。

6. サンプルコード

本章では、Test Design と Selenium を連携したサンプルコードを示します。
「Test Design の Web ページ」にて、本章のサンプルを使用したサンプルプロジェクトを公開しています。(http://www.testdesign.jp/manual/Sample_for_Web.zip)

6.1. 「Test1」サンプル

(Windows のダイアログ連携)
Test Design の Web ページから「testdesign-formatters.xpi」ファイルを
ダウンロードしてくるサンプルです。

以下にサンプルコードの内容を示します。

1. Test Design の Web ページ"http://www.testdesign.jp"にアクセスします。
2. 「オンラインマニュアル」ページから testdesign-formatters.xpi をダウンロードします。
3. ダウンロードの際に表示されるウィンドウ（名前を付けて保存する・・・など）によってログに出力されるメッセージが違います。
   「保存ファイル名を入力してください」ウィンドウが表示されました。」など

   ※保存先や名前を指定しないでダウンロードした場合は、
   「ファイルをダウンロードして保存しました。」とログに表示されます。

   ※実行後ログに赤い×が表示される場合があると思いますが、目的のウィンドウの有無を確認し、その結果を受けて後の処理を変えるための確認で、このサンプルにおいてはチェックの失敗ではありません。

●「Test1」サンプルコード

```
using System;
using System.IO;
using System.Text;
using Arp.Prova.PlayLib;
using OpenQA.Selenium;
using OpenQA.Selenium.Firefox;
using OpenQA.Selenium.Support.UI;

public class CodeClass
{
    private static IWebDriver driver;
    private static string baseURL;
    //
    //  TestDesign Main Code(C#)
    //
    public static int CodeStart()
    {
                    Process();
        return 0;
    }
    //
    private static void Process()
    {
        try
        {
            driver = new FirefoxDriver();
            baseURL = "http://www.testdesign.jp";

            driver.Navigate().GoToUrl(baseURL + "/p/");
            driver.FindElement(By.CssSelector("img[alt=¥"オンラインマニュアル¥"]")).Click();

            //testdesign-formatters.xpi をダウンロード
            driver.Navigate().GoToUrl("http://www.testdesign.jp/manual/testdesign-formatters.xpi");

            //保存ファイルダイアログが出力されている場合
            if (PlayLib.CheckWindowExist("testdesign-formatters.xpi を開く", true, 1) ) {
                PlayLib.Window("testdesign-formatters.xpi を開く").Click(286, 317);
                PlayLib.Sleep(1000);

                //保存ファイルダイアログが出力されている場合
                if (PlayLib.CheckWindowExist("保存ファイル名を入力してください...", true, 1) ) {
                    PlayLib.Window("保存ファイル名を入力してください...").Button("保存(S)").Click(42, 14);
                    PlayLib.Sleep(1000);
                    PlayLib.TestLog("「保存ファイル名を入力してください」ウィンドウが表示されました。");
                }
                else
                {
                    PlayLib.TestLog("「保存ファイル名を入力してください」ウィンドウは表示されませんでした。");
                }

                //保存ファイルダイアログが出力されている場合
                if (PlayLib.CheckWindowExist("名前を付けて保存の確認", true, 1) ) {
                    PlayLib.Window("名前を付けて保存の確認").Button("はい(Y)").Click(47, 15);
                    PlayLib.TestLog("「名前を付けて保存の確認」ウィンドウが表示されました。");
                }
                else
                {
                    PlayLib.TestLog("「名前を付けて保存の確認」ウィンドウは表示されませんでした。");
                }
```

```
                //ダウンロードマネージャが表示されている場合クリアする。
                if (PlayLib.CheckWindowExist("ダウンロードマネージャ", true, 1) ) {
                    PlayLib.Window("ダウンロードマネージャ").Close();
                }
                else
                {
                    PlayLib.TestLog("「ダウンロードマネージャ」は表示されませんでした。");
                }

                //通常サンプルでは、ファイル名・保存先を指定せずにダウンロードして終了します。
                PlayLib.TestLog("ファイルをダウンロードして保存しました。");
            }
        } catch ( Exception e ) {
            PlayLib.TestLog( e.Message );
        } finally {
            if( driver != null ) {
                 driver.Quit();
            }
        }
    }
}
```

6.2.「Test2」サンプル

（ページを表示した場合の文字列の存在チェック）

Test Design の Web ページで、目的の文字列をチェックするサンプルです。

「Test Design」の Web ページなので、「Test Design とは」の文言のチェックは成功しますが「SetROBO とは」の文言のチェックでは、わざと失敗します（その文言はページにないため）

以下にサンプルコードの内容を示します。

1. Test Design の Web ページ"http://www.testdesign.jp"にアクセスします。
2. 開いたページに「Test Design とは」という文字があるかどうかチェックします（成功する）
3. 開いたページに「SetROBO とは」という文字があるかどうかチェックします（失敗する）

●「Test2」サンプルコード

```
using System;
using System.Collections;
using System.IO;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading;
using Arp.Prova.PlayLib;
using OpenQA.Selenium;
using OpenQA.Selenium.Firefox;
using OpenQA.Selenium.Chrome;
using OpenQA.Selenium.IE;
using OpenQA.Selenium.Support.UI;

public class CodeClass
{
    private static IWebDriver driver;
    private static string baseURL;

    public static int CodeStart()
    {
        Demo();
        return 0;
    }
    //
    //  TestDesign Main Code(C#)
    //
    private static void Demo() {
        try
        {
            driver = new FirefoxDriver();
            baseURL = "http://www.testdesign.jp/";
            driver.Navigate().GoToUrl(baseURL);

            driver.FindElement(By.LinkText("Test Design とは")).Click();
            //ページの HTML を取得する。
            string resultComment = driver.PageSource;

            int resultPosition = 0;
            //文字列が存在するパターン
            resultPosition = resultComment.IndexOf("Test Design とは");
            PlayLib.CheckValue(resultPosition,-1,false);
            //文字列が存在しないターン
            resultPosition = resultComment.IndexOf("SetROBO とは");
            PlayLib.CheckValue(resultPosition,-1,false);
        } catch ( Exception e ) {
            PlayLib.TestLog( e.Message );
        } finally {
            if( driver != null ) {
               driver.Quit();
            }
        }
    }
}
```

7. 参考文献

１） Seleniumについては以下のURLを参考にしてください。
http://seleniumhq.org/
2012/5/18 現在 selenium-ide-1.1.8.xpi　が最新です。

All Selenium projects are licensed under the Apache 2.0 License.

更新履歴

Ver.	更新日時	章	貢	詳細
1.00	2012-6-21			新規