**Set up multiple browser testing**

This task describes some of the techniques you can use to enable effective cross-browser testing for your application or Web page.

**Tip:**For a use-case scenario related to this task, see [Using descriptive programming for multiple browser testing - Use-case scenario](https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Addins_Guide/Wrk_Multi_Browsers_Use-case_DescriptiveProgramming.htm).

In this topic:

* [Prerequisite- turn off auto updates for the browsers](https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Addins_Guide/Wrk_Multi_Browsers_How2.htm#mt-item-0)
* [Configure the Record and Run settings to launch a browser](https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Addins_Guide/Wrk_Multi_Browsers_How2.htm#mt-item-1)
* [Use the BROWSER\_ENV environment variable to launch a browser](https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Addins_Guide/Wrk_Multi_Browsers_How2.htm#mt-item-2)
* [Launch a browser with a test parameter](https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Addins_Guide/Wrk_Multi_Browsers_How2.htm#mt-item-3)
* [Launch a browser using a data table parameter](https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Addins_Guide/Wrk_Multi_Browsers_How2.htm#mt-item-4)
* [Launch a Browser using a WebUtil.LaunchBrowser step](https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Addins_Guide/Wrk_Multi_Browsers_How2.htm#mt-item-5)
* [Dynamically load an object repository during the test run](https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Addins_Guide/Wrk_Multi_Browsers_How2.htm#mt-item-6)
* [Add steps for browser specific behavior](https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Addins_Guide/Wrk_Multi_Browsers_How2.htm#mt-item-7)

Prerequisite- turn off auto updates for the browsers

To ensure that you are testing on the specific browser version you want, turn off the automatic update feature for your browser.

https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Resources/_TopNav/Images/_TopNav_mt-back-to-top.png [Back to top](https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Addins_Guide/Wrk_Multi_Browsers_How2.htm#top)

Configure the Record and Run settings to launch a browser

Using the Record and Run settings, change the browser on which you run the test for each test run.

1. Select **Record > Record and Run Settings**.
2. In the Record and Run Settings dialog box, select the **Web** tab.
3. In the Web tab, select the **Open the following address when a record or run session begins:** option.
4. In the web address drop-down list, enter a Web address to open or select a Web address from the drop-down list.
5. In the **Open the following browser when a record or run session beings:** drop-down list, select the browser on which you want to run your test.

**Note:**If you choose Apple Safari, you must provide additional connection information. For details, see [Connect to a remote Mac computer](https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Addins_Guide/Remote_Connection_Setup.htm)

When you start the test run, the specified browser opens the Web address entered.

https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Resources/_TopNav/Images/_TopNav_mt-back-to-top.png [Back to top](https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Addins_Guide/Wrk_Multi_Browsers_How2.htm#top)

Use the BROWSER\_ENV environment variable to launch a browser

Using the **BROWSER\_ENV** environment value, change the browser to launch each test run, and in some cases specify a specific browser version (if installed).

1. Select **File > Settings**.
2. In the Settings dialog box, select the **Environment** node.
3. In the Environment node, from the **Variable type** drop-down list, select **User-defined**.
4. In the user-defined variables list, click the **Add** button https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Addins_Guide/Images/sl_add_application.gif.
5. In the Add New Environment Parameter dialog box, in the **Name** field, enter **BROWSER\_ENV** (case-sensitive).
6. In the **Value** field, enter the value (case-sensitive) for the browser to open:

|  |  |
| --- | --- |
| **IE** | Opens the installed version of Internet Explorer. |
| **IE64** | Opens the installed 64-bit version of Internet Explorer. |
| **CHROME** | Opens the installed version of Google Chrome. |
| **CHROME\_HEADLESS** | Opens the locally installed version of Headless Chrome.  Supported for Chrome versions 60 and higher. |
| **FIREFOX** | Opens the installed version of Firefox. |
| **FIREFOX64** | Opens the latest version of 64-bit Mozilla Firefox that is both installed on the computer and supported by UFT. |
| **SAFARI** | Opens Safari on the remote Mac computer connected to UFT. |
| **EDGE** | Opens the installed version of Microsoft Edge with the Edge Agent for Functional Testing already enabled. |
| **CHROME\_EMULATOR** | Opens Chrome in emulated mode with the specified device. |
| **PHANTOMJS** | Opens the locally installed version of the PhantomJS web toolkit. |

1. Click **OK** to save the Name and Value of the variable.
2. In the Test Settings dialog box, click **Apply** and **OK** to save the variable and close the dialog box.

https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Resources/_TopNav/Images/_TopNav_mt-back-to-top.png [Back to top](https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Addins_Guide/Wrk_Multi_Browsers_How2.htm#top)

Launch a browser with a test parameter

Instruct UFT to automatically launch a specific browser for a test run using the Record and Run Settings:

1. Select **Record > Record and Run Settings**.
2. In the Record and Run Settings dialog box, select the **Web** tab.
3. In the Web tab, select the **Open the following when a record and run session begins:** option.
4. (Optional) In the **Address** field, enter the address to which to open the browser.

You can leave the **Browser** drop-down list as the default value.

1. In the **Parameter type** drop-down list, select the parameter type: **Global Data Table** or **Test Parameter**.
2. In the **Parameter Name** field, enter the parameter name. (The default parameter name is **Browser**.)
3. Click **Apply** to save the changes and **OK** to close the dialog box.

UFT automatically adds a column to the Global tab in the Data pane or a test parameter.

1. Before running the test, do one of the following, depending on the parameter type selected:

|  |  |  |
| --- | --- | --- |
| **For a Global Data Table parameter** | In the **Global** tab of the Data pane, set the value of the parameter. | **Value to use:**   * + **IE.** Opens Internet Explorer.   + **IE64.** Opens a 64-bit version of Internet Explorer.   + **CHROME.** Opens Google Chrome.   + **CHROME\_HEADLESS**. Opens Headless Chrome.   + **FIREFOX.** Opens Mozilla Firefox.   + **FIREFOX64.** Opens the latest version of 64-bit Mozilla Firefox that is both installed on the computer and supported by UFT.   + **SAFARI.** Opens Safari on the remote Mac computer connected to UFT (defined in the Web tab of the Record and Run Settings dialog box or in the **REMOTE\_HOST**environment variable).   + **EDGE.** Opens the installed version of Microsoft Edge with the Edge Agent for Functional Testing already enabled.   + **CHROME\_EMULATOR.** Opens Chrome in emulated mode with the specified device.   + **PHANTOMJS:** Opens the locally installed version of the PhantomJS web toolkit.   For details on adding mobile-relevant parameters, see [Define Mobile Record and Run Settings](https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Addins_Guide/Record_Settings_Mobile.htm). |
| **For a Test Parameter** | * 1. In the toolbar, click the **Run** button https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Addins_Guide/Images/sl_run_icon.gif.   2. In the Run dialog box, select the **Input Parameters** tab.   3. In the **Value** column for the parameter, enter the value of the parameter |

https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Resources/_TopNav/Images/_TopNav_mt-back-to-top.png [Back to top](https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Addins_Guide/Wrk_Multi_Browsers_How2.htm#top)

Launch a browser using a data table parameter

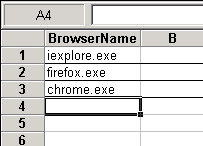
1. (Optional) Create a reusable action to use in all your tests for launching the browsers.
2. In the Data pane, open the **Global** tab.
3. In the Global tab, double-click the header of the first column in the table or the first column of the column where you want to store the parameter.
4. In the Change Parameter Name dialog box, enter the name for the parameter and click **OK**.

For example, you could name this parameter **BrowserName** (to identify it as the name of the browser to open).

The column name of the selected column is renamed to reflect the renamed parameter.

1. In the data table, enter the .exe names for the browsers you want to open.

For example, if you need to run the test on Internet Explorer, Firefox, and Chrome, you would enter **iexplore.exe**, **firefox.exe**, and **chrome.exe** in the first three rows of the column, respectively:



1. Add a test step with the following format:

SystemUtil.Run DataTable("<parameter name>", dtGlobalSheet), <address to open the browser to>

For example, if you wanted to open up to the Mercury Tours site, you could enter the following:

SystemUtil.Run DataTable("BrowserName", dtGlobalSheet), http://newtours.demoaut.com

**Note:**If you add this step to a reusable action which is then called by other tests, the relevant rows must be added in the Global tab of all tests that call that action.

https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Resources/_TopNav/Images/_TopNav_mt-back-to-top.png [Back to top](https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Addins_Guide/Wrk_Multi_Browsers_How2.htm#top)

Launch a Browser using a WebUtil.LaunchBrowser step

If you are using Business Process Testing to test your web applications, add use a **WebUtil.LaunchBrowser**step to launch the appropriate browsers as needed within each component.

Provide an argument for the Browser parameter, which is the same as the environment variables for Web- based environments:

* **CHROME**
* **CHROME\_EMULATOR**
* **CHROME\_HEADLESS**
* **EDGE**
* **FIREFOX**
* **FIREFOX64**
* **IE**
* **IE64**
* **PHANTOMJS**
* **SAFARI**

**Note:**If you are using business process tests created from UFT versions 12.53 and earlier, the **Web.txt**function library (found in your test's application area) does not contain the LaunchBrowser function. As a result, you will not be able to use the LaunchBrowser method in keyword GUI components.

To use this function, copy the following code into the Web.txt function library:

Public Function LaunchBrowser(Browser, device\_model, device\_manufacturer, device\_ostype, device\_osversion)

WebUtil.LaunchBrowser Browser, device\_model, device\_manufacturer, device\_ostype, device\_osversion

End Function

https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Resources/_TopNav/Images/_TopNav_mt-back-to-top.png [Back to top](https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Addins_Guide/Wrk_Multi_Browsers_How2.htm#top)

Dynamically load an object repository during the test run

If your test requires you to have different object repositories for each browser type, load the relevant object repositories as part of the test run without having to manually configure anything before the test run:

1. In the Data pane, open the **Global** tab.
2. In the Global tab, double-click the header of the first column in the table or the first column of the column where you want to store the parameter.
3. In the Change Parameter Name dialog box, enter the name for the parameter and click **OK**.

For example, you could name this parameter **Browser** (to identify it as the name of the browser on which to run the test).

The column name of the selected column is renamed to reflect the renamed parameter.

1. In the data table, enter the names for the browsers on which you want to run the test.
2. Add a test step with the following format:
3. If DataTable("<data table parameter>") = <Browser 1> Then
4. RepositoriesCollection.Add "<location to object repository>"
5. ElseIf DataTable("<data table parameter>") = <Browser 2> Then
6. RepositoriesCollection.Add "<location to object repository>"

End If

1. Add the additional steps for the application/Web page.

When the test runs, the appropriate object repository loads, and the test steps use the objects in the loaded object repository.

https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Resources/_TopNav/Images/_TopNav_mt-back-to-top.png [Back to top](https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Addins_Guide/Wrk_Multi_Browsers_How2.htm#top)

Add steps for browser specific behavior

If you need to add steps to perform browser specific behavior in the course of the test, use test parameters to create steps for this behavior.

1. In the canvas, select an action.
2. In the Properties pane, select the **Parameters** tab.
3. In the Parameters tab, click the **Add** button.
4. In the parameters grid, provide a name for the parameter. For example, you could name the parameter **ActiveBrowser** to show that the value of the parameter represents the browser currently in use.
5. Add steps to the test. You can use the value of the parameter by using the Parameter object:
6. Select Case Parameter("<parameter name>")
7. Case "<Browser 1>"
8. 'Do something specific for browser 1
9. Case "<Browser 2>"
10. 'Do something specific for browser 2

End Select

**Note:**Add additional Case statement as needed for each browser type.

When the test runs, the test steps run as specified in the necessary **Case** statement.

# Using descriptive programming for multiple browser testing - Use-case scenario

One of the challenging parts of cross-browser testing of your applications or Web pages is the object identification of objects in different browser types. Since each browser type can read the HTML code of your application and translate this differently, UFT may have trouble identifying the same objects in different browsers.

One technique that you can use when UFT is not identifying objects correctly is descriptive programming. When you insert a programmatic description into your test instead of the actual test object name, UFTsearches for the object in your application matching the description.

In this use-case scenario, see how UFT can find a problematic object using description instead of the test object name for the object (as stored in the object repository.).

In your application, you are trying to test this area of your application, containing a number of edit fields:

**Running the test on multiple browsers**

After you create a single test of your application or Web page to use in different browsers, you still must run it to actually test the application or Web page. You have a number of options on how to run the test across different browser types.

In this topic:

* [Manually configure the browser type](https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Addins_Guide/Wrk_Multi_Browsers_RunTests.htm#mt-item-0)
* [Instruct UFT to open a browser defined with a parameter](https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Addins_Guide/Wrk_Multi_Browsers_RunTests.htm#mt-item-1)
* [Use a test or data table parameter](https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Addins_Guide/Wrk_Multi_Browsers_RunTests.htm#mt-item-2)
* [Use a WebUtil.LaunchBrowser step](https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Addins_Guide/Wrk_Multi_Browsers_RunTests.htm#mt-item-3)

Manually configure the browser type

UFT provides you the opportunity to select the browser type before each test run. You can do this in one of the following places:

* **The Web tab of the Record and Run Settings dialog box.**

In the Web tab, you can select the browser type from the drop-down list. Then, when you run the test, UFTopens the appropriate browser and runs the test.

* **A user-defined environment variable specified in the Environment pane of the Test Settings dialog box.**

UFT uses the **BROWSER\_ENV** environment variable, and the requisite values for each browser type to enable you to set this variable before each test run. When you enter a value for the **BROWSER\_ENV** variable, UFT automatically opens up the necessary browser (ignoring any other browser launch settings).

However, this requires manual intervention by the person running the test, and does not enable you to run subsequent tests of the application or Web page on the different browser types in sequence.

For details, see [Configure the Record and Run settings to launch a browser](https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Addins_Guide/Wrk_Multi_Browsers_How2.htm#Wrk_Multi_Browsers_How2_ConfigureRRSettings) or [Use the BROWSER\_ENV environment variable to launch a browser](https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Addins_Guide/Wrk_Multi_Browsers_How2.htm#Wrk_Multi_Browsers_How2_SetBROWSER_ENV)

https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Resources/_TopNav/Images/_TopNav_mt-back-to-top.png [Back to top](https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Addins_Guide/Wrk_Multi_Browsers_RunTests.htm#top)

Instruct UFT to open a browser defined with a parameter

Instead of manually setting the browser before each test run (which thereby defeats the purpose of automated testing), you can insert a parameter into a test step that defines the browser to open. (You can also define a single reusable action that opens the browser, which can be reused in all the tests of your application or Web page.)

The values for this parameter (which are the **.exe** programs for each browser) are then defined in the Data pane. When UFT reaches this test step, it reads the data pane and decides which browser needs to open based on the selected data.

This removes the need for you or another person to manually configure settings or variables in a test before running the test, enabling you automatically test your application or Web page on all browser types and/or versions. If you create a reusable action with the step that opens the browser, the parameter and data must be added to every test that calls this external action.

For task details on how to set up automatic opening of browser types from UFT, see [Launch a browser using a data table parameter](https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Addins_Guide/Wrk_Multi_Browsers_How2.htm#Wrk_Multi_Browsers_How2_UseDataParam).

https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Resources/_TopNav/Images/_TopNav_mt-back-to-top.png [Back to top](https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Addins_Guide/Wrk_Multi_Browsers_RunTests.htm#top)

Use a test or data table parameter

In the Record and Run Settings dialog box (Web tab), you can instruct UFT to use either a test parameter or data table parameter to launch the browser. You set the parameter at the beginning of the test run (for a test parameter) or insert the **BROWSER\_ENV** value in the Data table in the Data pane.

Then, when UFT runs the test, it launches the correct browser according to the values you inserted.

For details, see [Launch a browser with a test parameter](https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Addins_Guide/Wrk_Multi_Browsers_How2.htm#Wrk_Multi_Browsers_How2_ConfigureRRSettingsforBrowser).

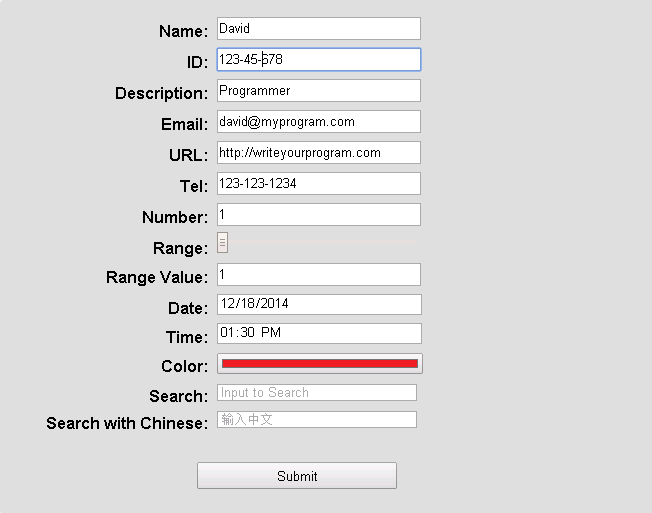
https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Resources/_TopNav/Images/_TopNav_mt-back-to-top.png [Back to top](https://admhelp.microfocus.com/uft/en/14.03/UFT_Help/Content/Addins_Guide/Wrk_Multi_Browsers_RunTests.htm#top)

Use a WebUtil.LaunchBrowser step

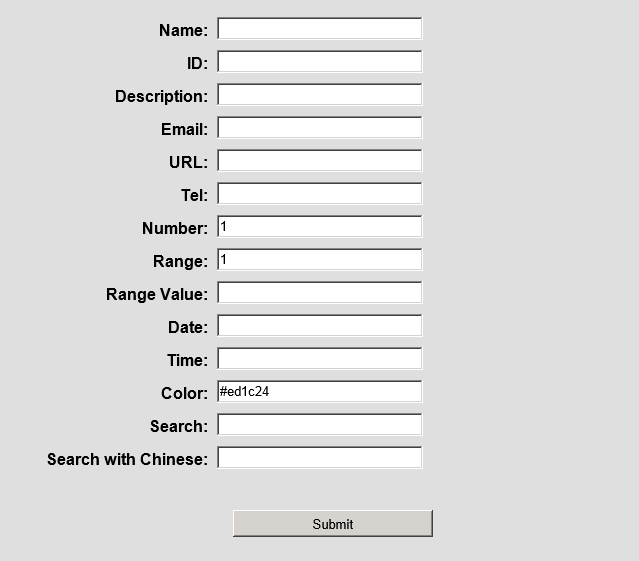
If you are using Business Process Testing to test your web applications, you should use a **WebUtil.LaunchBrowser** step to launch the appropriate browsers as needed within each component.

For each step, you provide an argument for the Browser parameter, which is the same as the environment variables for Web- based environments:

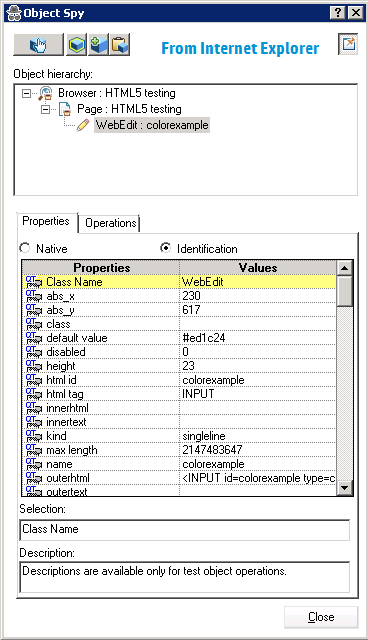
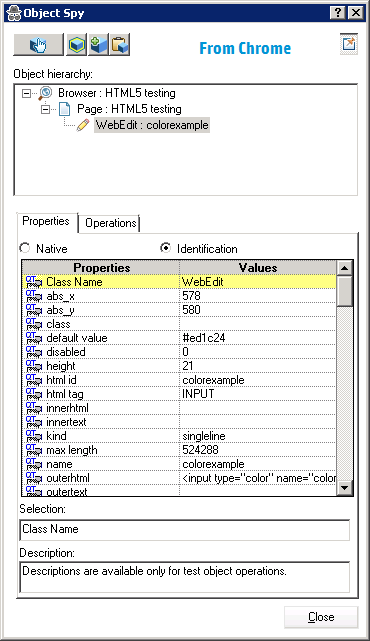
* **CHROME**
* **CHROME\_EMULATOR**
* **CHROME\_HEADLESS**
* **EDGE**
* **FF<VersionNumber>**
* **FIREFOX**
* **FIREFOX64**
* **IE**
* **IE64**
* **PHANTOMJS**
* **SAFARI**



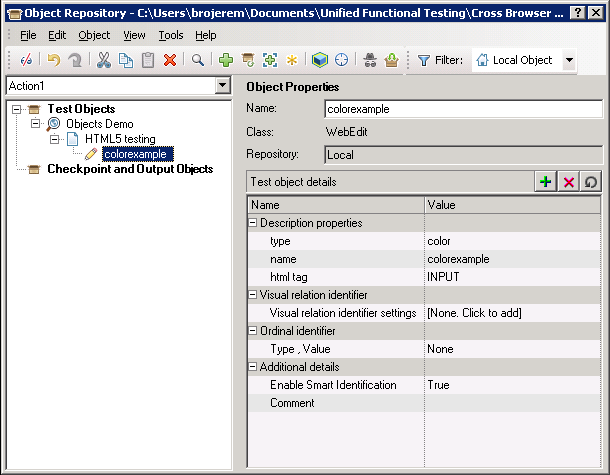
In Chrome and Firefox, the application area displays as seen above. However, in Internet Explorer, the window has a different appearance, particularly the **Color** field:



Even though the visual appearance is different, a closer look shows that the object properties of the **Color** field are basically the same:

In the object repository, the Color field object is recognized as a WebEdit object with the name **colorexample**(as it was by the Object Spy):

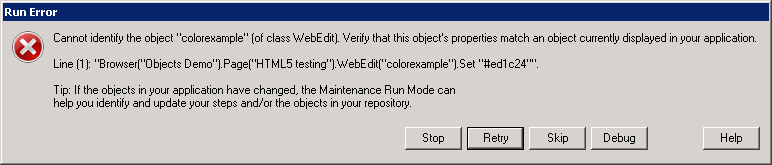


Based on this, when you insert a test step for this object, it is displayed like this:

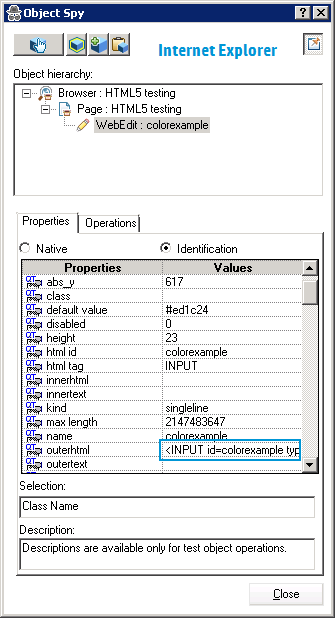
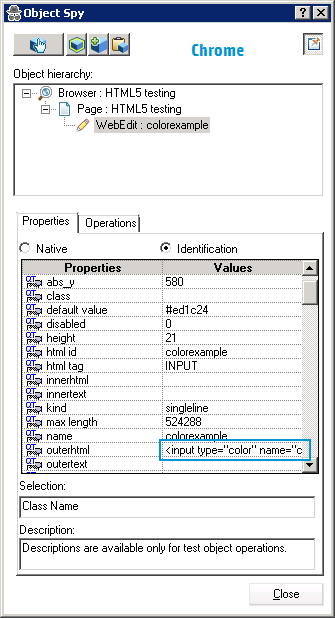
Browser("Objects Demo").Page("HTML5 testing").WebEdit("colorexample").Set

However, when you run the test step, there are varying results:

* The step runs on Chrome and Firefox without problem.
* The step fails on Internet Explorer:



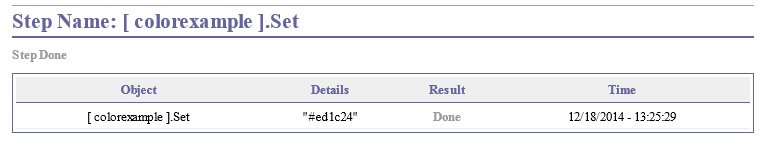
A closer look at the properties in the Object Spy for the **colorexample** object shows slight differences in the properties between Internet Explorer and Chrome:

In this case, the property difference is causing UFT to not identify the object in Chrome. As a result, modify the step using descriptive programming:

Browser("Objects Demo").Page("HTML5 testing").WebEdit("name:=colorexample", "htmltag:=INPUT").Set "#ed1c24"

Using this statement, when the test runs, the step runs without a problem:



Thus, by changing the test step to use descriptive programming, UFT is able to identify the object and run the test step across all browsers.

**Testing on headless browsers**

UFT has the following known issues when working with PhantomJS and Headless Chrome.

|  |  |
| --- | --- |
| **Supported Chrome versions** | Headless Chrome is supported only in Chrome versions 60 or higher. |
| **Recording** | Recording on PhantomJS toolkits or Headless Chrome is not supported. |
| **Data driving** | If you are data driving your test, we recommend adding a **Browser.CloseAllTabs** method to the end of each iteration to prevent failures in your test runs. |
| **Non-supported features** | The following features and objects are not supported when working with PhantomJS apps:   * Recording * Object Spy * Highlight in Application * Navigate and Learn * Add Objects to Local button in the Object Spy * Drag and Drop functionalities |
| **Non-supported objects** | The following test objects are not supported:   * **about://\*** pages * iFrame or cross-domain iFrame objects * Objects created with Web 2.0 technologies (ASAP .NET AJAX, Dojo, Google Web Tools, jQueryUI, SiebelOpenUI and YahooUI) * **PhantomJS only:** Browser dialogs, such as Alert, confirmation, or prompt dialogs |
| **Test Objects and Methods** | **PhantomJS only:**   * When running tests with the CaptureBitmap method with a large page source, a general run error is thrown. * The following test object methods and application objects are not supported:   + **Link** object created by the **role=link** property   + **WebAudio**   + **WebButton** created by the **role=button** property   + **WebList** created by the **role=list** property   + **WebMenu**   + **WebTable** created by the **role=table** or **role=grid** property   + **WebTabStrip**   + **WebTree**   + **WebVideo**   **PhantomJS / Headless Chrome:**  The following test object methods and application objects are not supported:   * **Browser.ClearCache** * **Browser.DeleteCookies** * **Browser.HandleDialog** * **Browser.Home** * **Browser.IsSiblingTab** * **Browser.Object** * **Browser.Stop** * **ViewLink** * **WebFile** |
| **Replay type** | The **Mouse** replay type is not supported. Use **Event** replay type instead. |
| **Running tests** | Maintenance Run Mode is not supported. |

6. **WebUtil** Object

In [UFT 14.01](https://www.learnqtp.com/uft-14-01-enhancements/) update, HPE introduced two new methods for *WebUtil*Object. **LaunchBrowser**and**LaunchMobileBrowserWithID**

Using **LaunchBrowser**method you can launch applications across desktop and mobile devices. The syntax for **LaunchBrowser** is

WebUtil.LaunchBrowser Browser, [device\_model, device\_manufacturer, device\_ostype, device\_osversion]

where:

* **Browser:** Name of browser to be launched. Ex: **CHROME, FIREFOX, IE**
* **device\_model:**The model of the selected device.
* **device\_manufacturer:**The name of manufacturer of the selected device.
* **device\_ostype:**The OS running on the device.
* **device\_osversion:**The OS version running on the selected device.

Parameters in square brackets are optional and they are meant to be used for mobile browsers only.

**Example:**

The following example launches the Chrome browser on an iOS Apple 5s device.

WebUtil.LaunchBrowser "MOBILE\_CHROME", "Apple\_5s", "Apple", "IOS", "10.1.3"

Using **LaunchMobileBrowserWithID**method you can launch a mobile browser on an iOS or an Android device using the Mobile Center’s device ID.

WebUtil.LaunchMobileBrowserWithID Browser, device\_ostype, device\_id

where

* **Browser:** Name of browser to be launched. Ex: **MOBILE\_HPWEB, MOBILE\_CHROME, MOBILE\_SAFARI**
* **device\_ostype:**The OS running on the device.
* **device\_id:**The id assigned to the device by Mobile Center.

**Example:**

WebUtil.LaunchMobileBrowserWithID "MOBILE\_CHROME", "IOS", "02"

*If you want to keep track of further articles on UFT (QTP). I recommend you to*

Headless browsers provide automated control of a web page in an environment similar to popular web browsers, but are executed via a [command-line interface](https://en.wikipedia.org/wiki/Command-line_interface) or using network communication. They are particularly useful for [testing](https://en.wikipedia.org/wiki/Functional_testing) web pages as they are able to render and understand HTML the same way a browser would, including styling elements such as page layout, colour, font selection and execution of [JavaScript](https://en.wikipedia.org/wiki/JavaScript) and [AJAX](https://en.wikipedia.org/wiki/Ajax_(programming)) which are usually not available when using other testing methods

[Headless Chrome](https://developers.google.com/web/updates/2017/04/headless-chrome) is a way to run the Chrome browser in a headless environment without the full browser UI. One of the benefits of using Headless Chrome (as opposed to testing directly in Node) is that your JavaScript tests will be executed in the same environment as users of your site. Headless Chrome gives you a real browser context without the memory overhead of running a full version of Chrome

Headless testing is a way of running browser UI tests without the head, which in this case means that there’s no browser UI, no GUI of any sorts. This is useful since when running tests, especially in a CI environment, there is nobody “watching” the visuals, so there is no need to have the extra overhead of the browser GUI.

One of the biggest reasons for using a headless browser/carrying out headless testing is performance, since it lets you run tests more quickly in a real browser environment. Headless browsers avoid draw operations, which handle rendering of the UI and their various pixels on the screen. With headless testing, we ignore those draw operations and the headless engines just run the same tests in the background without a need for a user interface.