Regular Expressions in UFT

A **regular expression** is a string that describes or matches a set of strings. It is often called a **pattern** as it describes a set of strings.

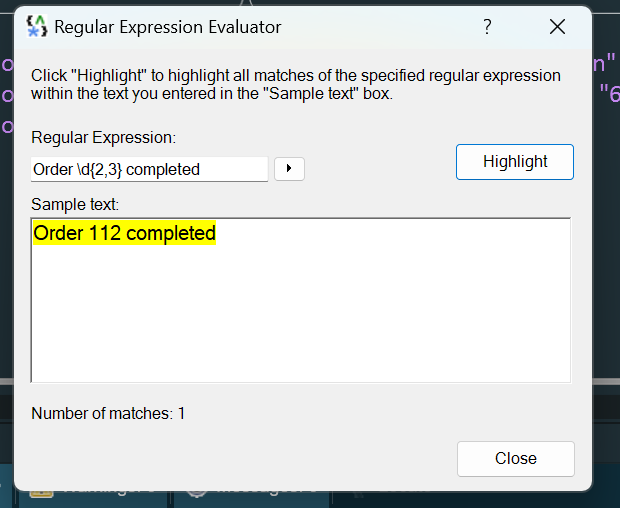
When working on particular applications let’s say one of the property values called *id* is dynamic, that is, on every run you notice that the value of *id* property is getting changed. So when you recorded your script, the id property value was REUFT101. When you replayed the same script, value of *id* changed to REUFT120, on next iteration it changed to REUFT105. You can easily identify a pattern here whereby the last two digits of *id* value are getting changed.  
Can we write a single string that can match ALL these dynamic values? Yes, and the answer lies with regular expressions. You can write **REUFT1\d\d** The first five characters were fixed so we need not *regularize* them, however, the last two characters were dynamic digits. We need to ensure that our regular expression can match any single digit between 0 to 9. The regular expression for that is \d**.** Since we need to match two digits we have used it twice \d\d

## Regular expressions can be used in

* Object Repository for object recognition
* Checkpoints

## Regular Expression Evaluator

 UFT One provides regular expression evaluator right inside UFT’s IDE. This reg ex evaluator is accessible through Tools > Regular Expression Evaluator



Example:

Try on User Registration

# Debugging in UFT

Debugging purpose is to identify the root cause of the problem

* Insert Breakpoints

Breakpoint is used to pause the script execution and inspect the variable values / application behaviour

* Update values in runtime using Console

## Debugging using console

Changing the value in console window

## Slow debugging session

During a run session, UFT One normally runs steps quickly. When debugging, you may want to slow the test run speed to enable more effective debugging.

To slow the test run speed:

1. Browse to the Test Runs pane of the Options dialog box (**Tools > Options > GUI Testing** tab **> Test Runs** node).
2. Modify the **Delay each step execution** by value with the number of milliseconds you want UFT One to wait between each step.

## Step into, out of, or over a specific GUI step

* **Step Into -** Runs only the current step in the active document.

In the toolbar, press the Step Into button .

When debugging a GUI test, if the current step calls another action or a function, the called action or function is displayed in the document pane. The test or function library pauses at the first line of the called action or function.

* **Step Out -** Continues the run to the end of the function, or user code file, returns to the calling test, component, or function library, and then pauses the run session at the next line (if one exists).

In the toolbar, press the Step Out button .

* **Step Over** - If the current step calls a user-defined function, the called function is executed in its entirety, but the called function script is not displayed in the document pane. The run session then returns to the calling document and pauses at the next step

In the toolbar, press the **Step Over** button .