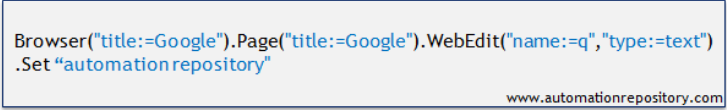
Descriptive Programming in QTP – The Complete Guide:

**Descriptive Programming is the type of programming where you DON’T use Object Repository while writing your test scripts.**

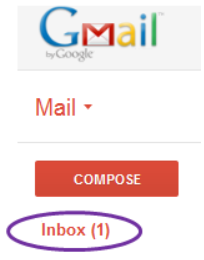
Exa.



**So when you say that you want to write your script a such a manner that it doesn’t use object repository, it actually means that you have to write the object part of your statement in a different way such that it doesn’t go to Object Repository to find the object properties.**

### **Using Regular Expressions in Descriptive Programming:**

Consider the following scenario – You have to write a script where you have to click on the inbox link in GMail. But you would have noticed that the number of unread emails is also displayed together with the Inbox link in GMail. Also, the number of unread emails changes very frequently.



So how do you handle such scenarios when you use Object Repository? Yes, you use Regular Expression when specifying the object property in OR. Well, you can use the same regular expressions concept with Descriptive Programming also. Let’s see how this is done.

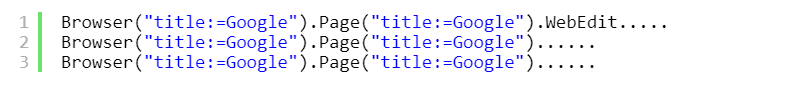
Now when you don’t have any unread mails in GMail, the inbox link will be displayed as ‘Inbox’ but in case there are any unread mails, it is appended with the Inbox text as Inbox(2), Inbox(14) etc. Here you will notice that the common text in both the cases is ‘Inbox’. The most basic regular expression that we can use here is **Inbox.\*** . And for the link, the regular expression would look like **Link(“innertext:=Inbox.\*”)**. The complete code is shown below –

Browser("title:=Gmail.\*").Page("title:=Gmail.\*").Link("innertext:=Inbox.\*").Click

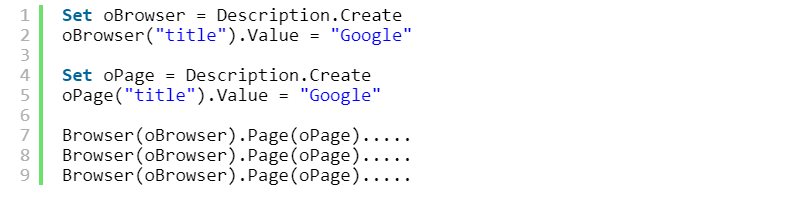
In the above code, we have used regular expression for Browser and Page title also. This is done because when you open any mail in Gmail, some text gets appended after Gmail in the title. So this code will work from Inbox as well as from some other page in Gmail.

### **Using Description Object in Descriptive Programming**

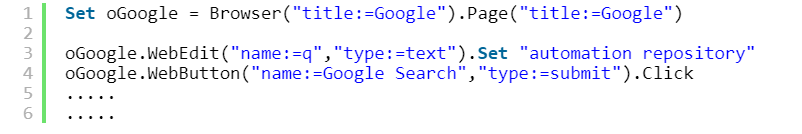
In the first part of [Descriptive Programming in QTP](http://www.automationrepository.com/2012/03/descriptive-programming-in-qtp-the-complete-guide-part1/) series, we covered the most basic method of using Descriptive Programming where you specify the object property and value directly with the statement. This approach is shown below.



Here you can notice that the object property for browser and page objects are specified multiple times. Now suppose that the this property gets changed at some point in the future. In such a scenario, you would have to do a lot of rework to update the scripts. To avoid this, you can use an alternative method where you can specify the object properties and methods in Description Object and then use this object in your statement. Below code highlights this approach.



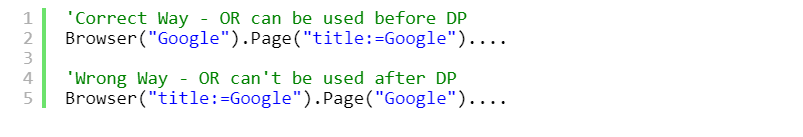
Alternatively, you can use one more method to shorten your code –



In both the above cases, the object property and value is written at only a single place. So if there is any change in the object property, it needs to be done at few places and thus the amount of rework is significantly reduced.

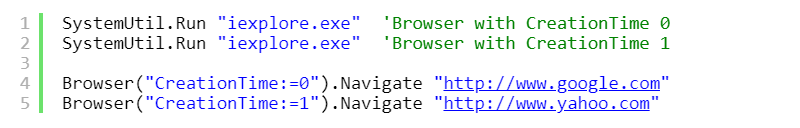
### **Combining both Object Repository and Descriptive Programming**

You can combine both Object Repository and Descriptive Programming approaches in your scripts. While doing so you have to take care of one thing. In the same statement, you can start with OR and then use DP, but you cant do it the reverse way. That is, you can’t first use DP and then move over to OR. The below code illustrates this concept.

**Using Ordinal Identifiers in Descriptive Programming**

Many a times when you write scripts, you might need to include ordinal identifiers (index, creation time, location) in the object repository. You can use ordinal identifiers in Descriptive Programming also. Let’s see an example for the same.

**Example: Ordinal Identifiers in Descriptive Programming.**In this example,the code will open 2 internet explorer browsers and open different URLs in both the browsers.



Based on the Creation Time ordinal identifier, the above code opens google.com in the first browser and yahoo.com in the second browser.