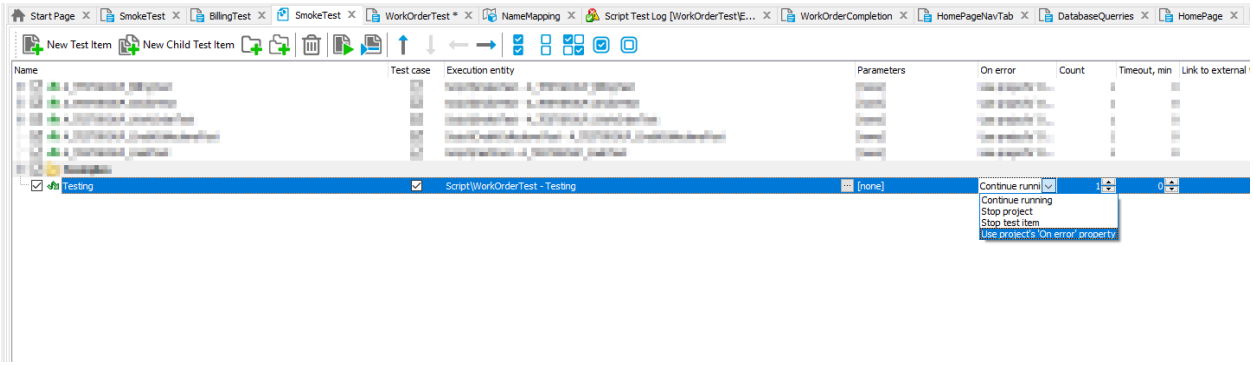
Custom “On Error Actions” by having a “Raise exception” option in the on error column

In the project window there is a drop down under the column of On Error. This field determines what TestComplete does when it encounters an error. The option of this are Continue running, Stop test item, Stop project, or Inherit from project. I would like to add one option to this called Raise exception.



This would allow users to do their own error handling by wrapping their entire test function associated with the test item in a try catch statement.

One use case for this would be running a test multiple times with different data (ie: Account Number). When an error occurs we would like it to move onto the next account not continue executing with this same account so continue running will not work in this case. If we use stop test item we will not be able to execute the accounts after it. This is why there should be a 5th custom option that will let you handle it yourself. This is how a test case would be structured with multiple data points (ie: account numbers) if this feature existed:

```
def MultipleDataPointTest():
  accounts = ReadAccountsNumberDataBase(querry)
  for account  in accounts:
      try:

        #Arrange
        testCaseAccountNumber = int(account[0])
        Log.PushLogFolder(Log.CreateFolder("Test data: " +
str(testCaseAccountNumber)))

        #Act
```

```
        TestStep1()
        TestStep2()
        TestStep3()

        #Assert
        confirmValue = getValue()
        if(not confirmValue):
          Log.Error("Test Failed")
        Log.Error("Test Passed")

        CloseOpenBrowsers()
        Log.PopLogFolder()
      except Exception as e:
        CloseOpenBrowsers()
        Log.Warning("The error was: " + str(e))
        Log.Warning("Moving on to next account.")
        Log.PopLogFolder()
        continue
```

This problem arises because Smart Bears has there own error handling and when there is an error in one of the TestComplete modules it is already handled. See code below for what kind of errors throw exception when test item on continue running:

```
#generic function to call possible functions errors
def Testing():
  try:
    #replace with any below to see result
    Error2()
  except Exception as e:
    Log.Warning("The error was: " + str(e))
    pass

# generic python error
# WILL raise exception
def Error1():
    Log.Message(badvar)
    Log.Message("Still going")
```

```
# Click on Object that does not exist
# will NOT raise exception
def Error2():
    Aliases.browser.Click()
    Log.Message("Still going")

# Enter .Keys() on Object that does not exist
# will NOT raise exception
def Error3():
    Aliases.browser.Keys('[Enter]')
    Log.Message("Still going")


# Log.Error(string)
# will NOT raise exception
def Error4():
    Log.Error("Bad thing")
    Log.Message("Still going")

# Purposeful execution raising
# WILL raise exception
def Error5():
  Log.Error("Bad thing")
  raise Exception
  Log.Message("Still going")
```