Recent

Topics

Tutorials

Highlights

Settings

Feedback (http://community.

Sign Out

Building a test reporter

We have seen in the previous recipe that `Action` can accumulate dat
by adding information to its `Details` field. The `Status` field of `Actio`
set from within the code, generally the `Run` method. Our reporting
mechanism leverages the fact that the `Actions` are objects, and that
they will use the native UFT reporter as their target output.

Getting ready

From the **File** menu, navigate to **New** | **Function Library**, or use the
Alt + *Shift* + *N* shortcut. Save the file as
`C:\Automation\Lib\lib.Reporter.vbs`.

How to do it...

Write the following code in the function library you created:

```
Function ReportActionStatus(ByRef p)
    ' ------------------------------------------------
    ' Function    :    ReportActionStatus
    ' Purpose     :    Reports an event to the UFT reporter with the data
    ' Args        :    ByRef p
    ' Returns     :    N/A
    ' ------------------------------------------------
    Reporter.ReportEvent GetNormalizedStatus(p.Status.Value), TypeName(p)
End Function


Function GetStatusText(ByVal iStatus)
    ' ------------------------------------------------
    ' Function    :    GetStatusText
    ' Purpose     :    Returns the text associated with a status
    ' Args        :    ByVal iStatus
    ' Returns     :    "success", "failure"
    ' ------------------------------------------------
    Dim sStatus

    Select Case CInt(iStatus)
        Case 0, 2, 4 'micPass, micDone, micInfo
            sStatus = "success"
        Case Else 'micFail, micWarning
            sStatus = "failure"
    End Select

    GetStatusText = sStatus
End Function


Function GetNormalizedStatus(ByVal iStatus)
    ' ------------------------------------------------
    ' Function    :    GetNormalizedStatus
    ' Purpose     :    Returns the status as 0 or 1
    ' Args        :    ByVal iStatus
    ' Returns     :    0 or 1
    ' ------------------------------------------------
    GetNormalizedStatus = micPass
    If CLng(iStatus) <> CLng(micPass) Then
        GetNormalizedStatus = micFail
    End If
End Function
```

The main function here is `ReportActionStatus`, which accepts the
reference to an object that is built with the action design pattern and has
both `Status` and `Details` as public fields.

The `GetStatusText` function is used to standardize the report message
according to the status. The `GetNormalizedStatus` function limits the
status to `0` (success) and `1` (failure), and it is used to standardize the
accumulated statuses that can be sent from an Action or other function.

Finally, the `PrintReportInfo` function is just used to log messages to
the UFT Reporter without affecting the results, as shown:

```
Function PrintReportInfo(ByVal sSender, ByVal sMessage)
    ' -------------------------------------------------
    ' Function   :   PrintReportInfo
    ' Purpose    :   Reports an info event to the UFT reporter and log
    ' Args       :   ByVal sSender
    '                ByVal sMessage
    ' Returns    :   N/A
    ' -------------------------------------------------
    Print sSender & ": " & sMessage
    Reporter.ReportEvent micInfo, sSender, sMessage
End Function
```

How it works...

The `ReportActionStatus` function accepts the reference to an `Action` type object, which implements the design pattern as defined. With this reference, it has access to the public fields of `Action`, and so it retrieves and formats the data before sending it to the UFT reporter. This way, we make the reporting easier, as `Action` can just accumulate what it finds and the actual reporting in a single step.