Indy RUG – January 2004

Choosing a Test Automation Framework

Mike Kelly

Agenda

- Introduction
- Record and Playback
- Frameworks 101
  - Test Script Modularity
  - Test Library Architecture
  - Keyword-Driven or Table-Driven Testing
  - Data-Driven Testing
  - Hybrid Test Automation
- Questions

Record and Playback

Perceived Advantages

- Ease of creation
- Low level of technical skill required

Perceived Limitations

- Usually limited to GUI applications
- Time to maintain
- Traditionally not powerful
- Problems with cutting edge technologies or custom developed controls

What is a Test Framework?

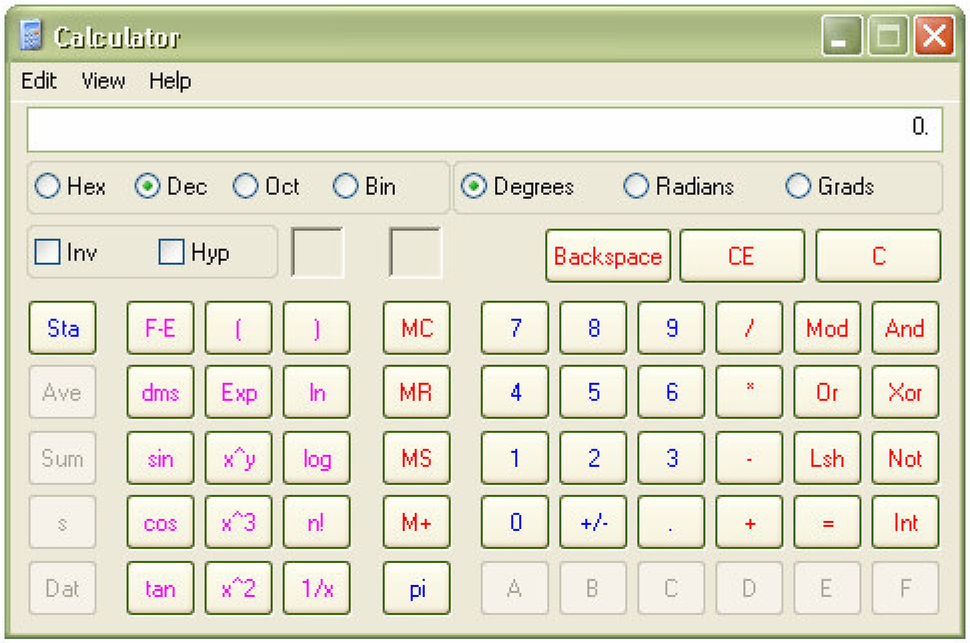
- Set of Assumptions, Concepts, and Practices
- Facilitates use of best practices
- Facilitates reuse
- Makes test code easier to maintain

Test Script Modularity

- Simplest of frameworks
- Small independent scripts
- Hierarchical structure
- Implemented entirely in Robot

Test Script Modularity

Example


Calculator
Edit View Help
0.
Hex Dec Oct Bin
Degrees Radians Grads
Inv Hyp
Backspace CE C
Sta F-E ( ) MC 7 8 9 / Mod And
Ave dms Exp ln MR 4 5 6 * Or Xor
Sum sin x^y log MS 1 2 3 - Lsh Not
s cos x^3 n! M+ 0 +/- . + = Int
Dat tan x^2 1/x pi A B C D E F

Test Script Modularity

Record and Playback

```
Sub Main

    Dim Result As Integer

    'Initially Recorded: 1/17/2002  2:21:29 PM
    'Script Name: Modularity - Record and Playback

    Window SetContext, "Caption=Calculator", ""
    PushButton Click, "Text=5"
    PushButton Click, "Text=+"
    PushButton Click, "Text=4"
    PushButton Click, "Text=="

    Result = EditBoxVP (CompareProperties, "ObjectIndex=1", "VP=Add VP")

    PushButton Click, "Text=CE"
    PushButton Click, "Text=5"
    PushButton Click, "Text=-"
    PushButton Click, "Text=4"
    PushButton Click, "Text=="

    Result = EditBoxVP (CompareProperties, "ObjectIndex=1", "VP=Subtract VP")

    PushButton Click, "Text=CE"
    PushButton Click, "Text=5"
    PushButton Click, "Text=/"
    PushButton Click, "Text=4"
    PushButton Click, "Text=="

    Result = EditBoxVP (CompareProperties, "ObjectIndex=1", "VP=Divide VP")

End Sub
```

Framework

```
Sub Main

    'Initially Recorded: 1/17/2002  2:26:44 PM
    'Script Name: Modularity - Test Case

    CallScript "Modularity - Add"

    CallScript "Modularity - Subtract"

    CallScript "Modularity - Divide"

End Sub
```

Test Script Modularity

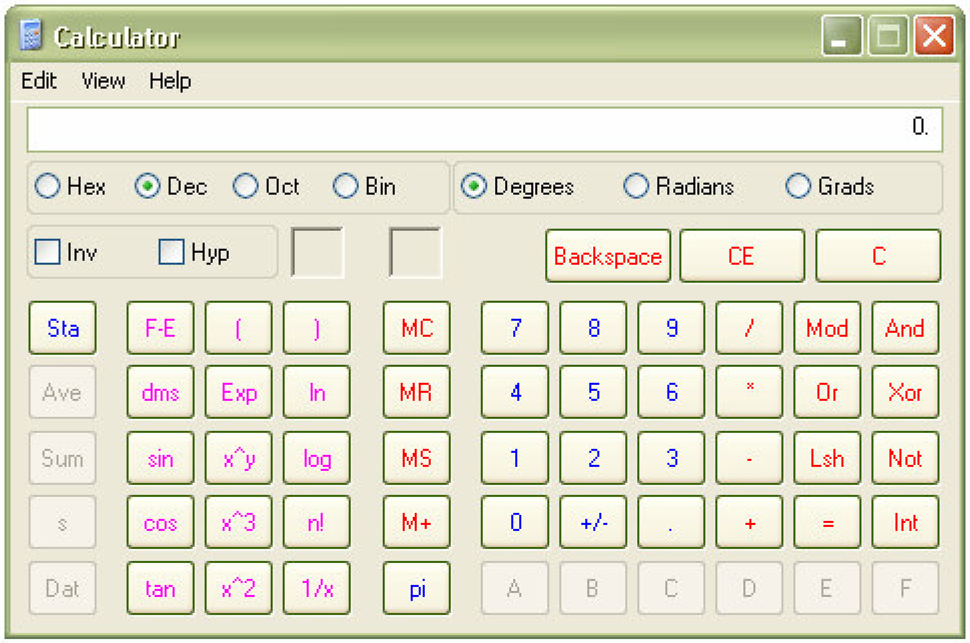
- Simple to implement
- Low level of technical knowledge required
- Easier to read and debug code
- Shorter maintenance cycles

Test Library Architecture

- Similar to Test Script Modularity
- Independent procedures or functions
- May be implemented entirely in Robot using SQABasic libraries
- May be implemented outside of Robot using APIs, DLLs, or other custom test extensions

Test Library Architecture

Example


Calculator
Edit View Help
0.
Hex [✓ Dec] Oct Bin
[✓ Degrees] Radians Grads
Inv Hyp
Backspace CE C
Sta F-E ( ) MC 7 8 9 / Mod And
Ave dms Exp ln MR 4 5 6 * Or Xor
Sum sin x^y log MS 1 2 3 - Lsh Not
s cos x^3 n! M+ 0 +/- . + = Int
Dat tan x^2 1/x pi A B C D E F

Test Library Architecture

Record and Playback

```
Sub Main

    Dim Result As Integer

    'Initially Recorded: 1/17/2002  2:21:29 PM
    'Script Name: Modularity - Record and Playback

    Window SetContext, "Caption=Calculator", ""
    PushButton Click, "Text=5"
    PushButton Click, "Text=+"
    PushButton Click, "Text=4"
    PushButton Click, "Text=="

    Result = EditBoxVP (CompareProperties, "ObjectIndex=1", "VP=Add VP")

    PushButton Click, "Text=CE"
    PushButton Click, "Text=5"
    PushButton Click, "Text=-"
    PushButton Click, "Text=4"
    PushButton Click, "Text=="

    Result = EditBoxVP (CompareProperties, "ObjectIndex=1", "VP=Subtract VP")

    PushButton Click, "Text=CE"
    PushButton Click, "Text=5"
    PushButton Click, "Text=/"
    PushButton Click, "Text=4"
    PushButton Click, "Text=="

    Result = EditBoxVP (CompareProperties, "ObjectIndex=1", "VP=Divide VP")

End Sub
```

Framework

```
'$Include "Library.sbh"

Sub Main

    'Initially Recorded: 1/17/2002  3:07:15 PM
    'Script Name: Library - Test Case

    LibraryAdd "5", "4", "9."

    LibrarySubtract "5", "4", "1."

    LibraryDivide "5", "4", "1.25"

End Sub
```

Test Library Architecture

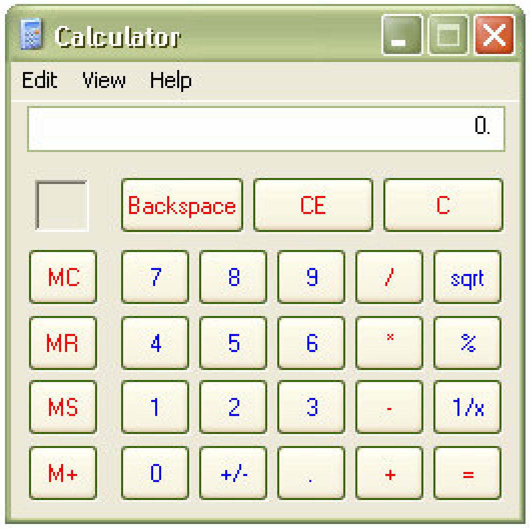
- Knowledge of SQABasic language and of basic programming concepts required
- Easier to read and debug code
- Much shorter maintenance cycles
- Highest level of reuse
- Higher creation cost

Keyword-Driven or Table-Driven

- Requires definition of data tables and keywords
- Makes tests independent of automated test tool
- Look and feel similar to manual tests

Keyword-Driven or Table-Driven

Example



Keyword-Driven or Table-Driven

Window	Control	Action	Arguments
Calculator	Menu		View, Standard
Calculator	Pushbutton	Click	1
Calculator	Pushbutton	Click	+
Calculator	Pushbutton	Click	3
Calculator	Pushbutton	Click	=
Calculator		Verify Result	4
Calculator		Clear	
Calculator	Pushbutton	Click	6
	Pushbutton	Click	-
	Pushbutton	Click	3
	Pushbutton	Click	=
		Verify Result	3

```
Main Script / Program
     Connect to data tables
     Read in row and parse out values
     Pass values to appropriate functions
     Close connection to data tables

Menu Module
     Set focus to window
     Select the menu pad option
     Return

Pushbutton Module
     Set focus to window
     Push the button based on argument
     Return

Verify Result Module
     Set focus to window
     Get contents from label
     Compare contents with argument value
     Log results
     Return
```

Keyword-Driven or Table-Driven

- Knowledge of programming required
- Self documenting test cases
- Tool independent
- Very high level of reuse
- High creation cost
- Open source versions available:

  http://safsdev.sourceforge.net/

Data-Driven Framework

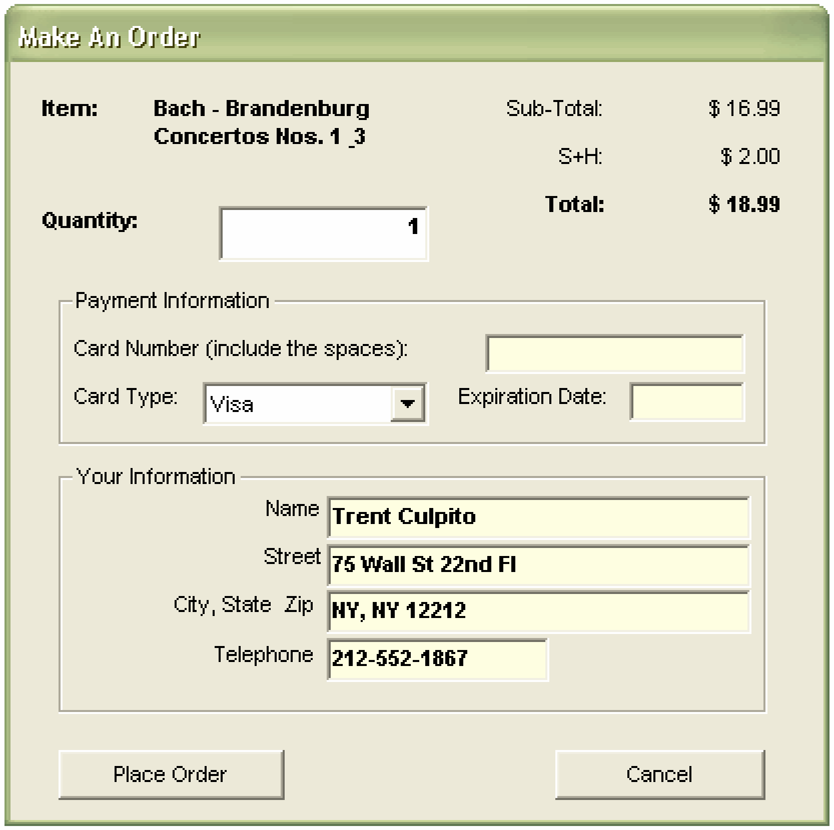
- Simple to implement
- Use of data files for runtime data
- Script is for navigation and to “drive” the data

- Can be implemented using datapools in TestManager
- Or can be implemented using ODBC sources, CVS files, Excel spreadsheets, DAO objects, ADO objects, etc… – Anything you want!

Data-Driven Framework

Example

Test the Expiration Date field



Item:	**Bach - Brandenburg Concertos Nos. 1_3**	Sub-Total:	$ 16.99
		S+H:	$ 2.00
Quantity:	1	**Total:**	**$ 18.99**

Payment Information

Card Number (include the spaces):

Card Type: Visa

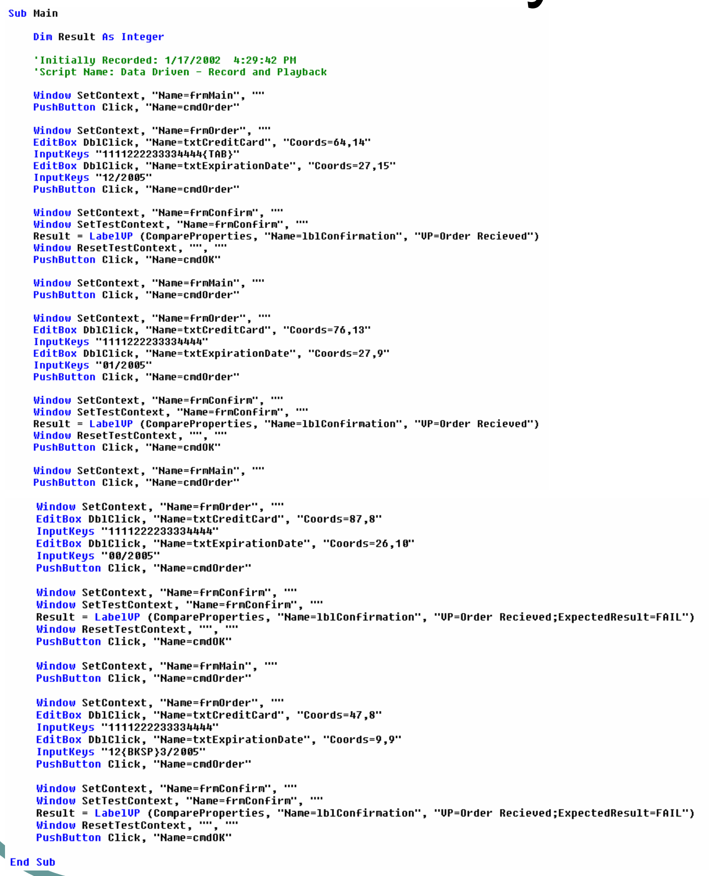
Expiration Date:

Your Information

Name	**Trent Culpito**
Street	**75 Wall St 22nd Fl**
City, State Zip	**NY, NY 12212**
Telephone	**212-552-1867**

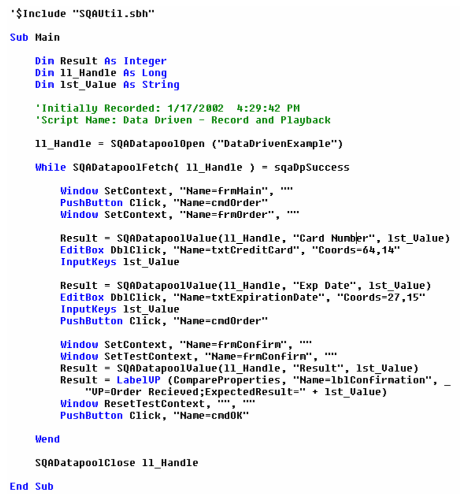
Data-Driven Framework

Record and Playback



```
Sub Main

    Dim Result As Integer

    'Initially Recorded: 1/17/2002  4:29:42 PM
    'Script Name: Data Driven - Record and Playback

    Window SetContext, "Name=frmMain", ""
    PushButton Click, "Name=cmdOrder"

    Window SetContext, "Name=frmOrder", ""
    EditBox DblClick, "Name=txtCreditCard", "Coords=64,14"
    InputKeys "1111222233334444{TAB}"
    EditBox DblClick, "Name=txtExpirationDate", "Coords=27,15"
    InputKeys "12/2005"
    PushButton Click, "Name=cmdOrder"

    Window SetContext, "Name=frmConfirm", ""
    Window SetTestContext, "Name=frmConfirm", ""
    Result = LabelVP (CompareProperties, "Name=lblConfirmation", "VP=Order Recieved")
    Window ResetTestContext, "", ""
    PushButton Click, "Name=cmdOK"

    Window SetContext, "Name=frmMain", ""
    PushButton Click, "Name=cmdOrder"

    Window SetContext, "Name=frmOrder", ""
    EditBox DblClick, "Name=txtCreditCard", "Coords=76,13"
    InputKeys "1111222233334444"
    EditBox DblClick, "Name=txtExpirationDate", "Coords=27,9"
    InputKeys "01/2005"
    PushButton Click, "Name=cmdOrder"

    Window SetContext, "Name=frmConfirm", ""
    Window SetTestContext, "Name=frmConfirm", ""
    Result = LabelVP (CompareProperties, "Name=lblConfirmation", "VP=Order Recieved")
    Window ResetTestContext, "", ""
    PushButton Click, "Name=cmdOK"

    Window SetContext, "Name=frmMain", ""
    PushButton Click, "Name=cmdOrder"

    Window SetContext, "Name=frmOrder", ""
    EditBox DblClick, "Name=txtCreditCard", "Coords=87,8"
    InputKeys "1111222233334444"
    EditBox DblClick, "Name=txtExpirationDate", "Coords=26,10"
    InputKeys "00/2005"
    PushButton Click, "Name=cmdOrder"

    Window SetContext, "Name=frmConfirm", ""
    Window SetTestContext, "Name=frmConfirm", ""
    Result = LabelVP (CompareProperties, "Name=lblConfirmation", "VP=Order Recieved;ExpectedResult=FAIL")
    Window ResetTestContext, "", ""
    PushButton Click, "Name=cmdOK"

    Window SetContext, "Name=frmMain", ""
    PushButton Click, "Name=cmdOrder"

    Window SetContext, "Name=frmOrder", ""
    EditBox DblClick, "Name=txtCreditCard", "Coords=47,8"
    InputKeys "1111222233334444"
    EditBox DblClick, "Name=txtExpirationDate", "Coords=9,9"
    InputKeys "12{BKSP}3/2005"
    PushButton Click, "Name=cmdOrder"

    Window SetContext, "Name=frmConfirm", ""
    Window SetTestContext, "Name=frmConfirm", ""
    Result = LabelVP (CompareProperties, "Name=lblConfirmation", "VP=Order Recieved;ExpectedResult=FAIL")
    Window ResetTestContext, "", ""
    PushButton Click, "Name=cmdOK"

End Sub
```

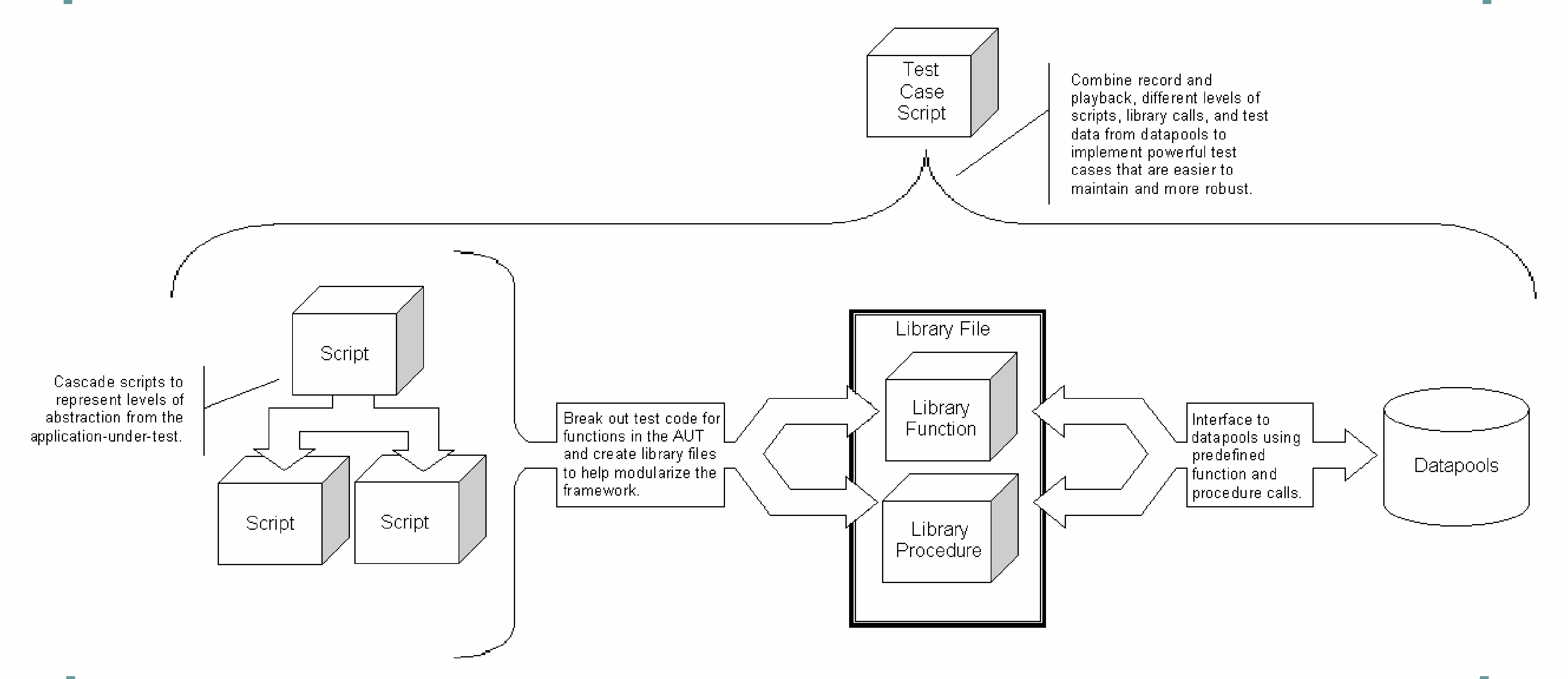
Framework



```
'$Include "SQAUtil.sbh"

Sub Main

    Dim Result As Integer
    Dim ll_Handle As Long
    Dim lst_Value As String

    'Initially Recorded: 1/17/2002  4:29:42 PM
    'Script Name: Data Driven - Record and Playback

    ll_Handle = SQADatapoolOpen ("DataDrivenExample")

    While SQADatapoolFetch( ll_Handle ) = sqaDpSuccess

        Window SetContext, "Name=frmMain", ""
        PushButton Click, "Name=cmdOrder"
        Window SetContext, "Name=frmOrder", ""

        Result = SQADatapoolValue(ll_Handle, "Card Number", lst_Value)
        EditBox DblClick, "Name=txtCreditCard", "Coords=64,14"
        InputKeys lst_Value

        Result = SQADatapoolValue(ll_Handle, "Exp Date", lst_Value)
        EditBox DblClick, "Name=txtExpirationDate", "Coords=27,15"
        InputKeys lst_Value
        PushButton Click, "Name=cmdOrder"

        Window SetContext, "Name=frmConfirm", ""
        Window SetTestContext, "Name=frmConfirm", ""
        Result = SQADatapoolValue(ll_Handle, "Result", lst_Value)
        Result = LabelVP (CompareProperties, "Name=lblConfirmation", _
            "VP=Order Recieved;ExpectedResult=" + lst_Value)
        Window ResetTestContext, "", ""
        PushButton Click, "Name=cmdOK"

    Wend

    SQADatapoolClose ll_Handle

End Sub
```

- Little to no increased knowledge of programming required
- Somewhat self documenting test cases
- Reduction of required test code
- Drastically lowers test case creation cost

Hybrid Test Automation


Test Case Script
Combine record and playback, different levels of scripts, library calls, and test data from datapools to implement powerful test cases that are easier to maintain and more robust.
Cascade scripts to represent levels of abstraction from the application-under-test.
Script
Script
Script
Break out test code for functions in the AUT and create library files to help modularize the framework.
Library File
Library Function
Library Procedure
Interface to datapools using predefined function and procedure calls.
Datapools

Questions

Additional Information:

Mike Kelly, “Choosing a Test Automation Framework.” *Rational Developer Network*

Cem Kaner, "Improving the maintainability of automated test suites." *Software QA*, Volume 4, #4, 1997. Available at Kaner.com

John Earles, “Framework Architectures! Make Way for One More Silver Bullet.” *Rational Developer Network*

Bret Pettichord, “Seven Steps to Test Automation Success.” Available at Pettichord.com

Carl Nagle, “Software Automation Framework Support” http://safsdev.sourceforge.net/

Mike Kelly, “Using Cost-Benefit Analysis to Compare Different Test Structure for Rational Robot.” *Rational Developer Network*