



Secrets of Test Automation Frameworks

Presented by
Paul M. Grossman
Senior Consultant

- **Experience**
 - 10 Years in Manual Software Testing
 - 6 Years Experience with HP WinRunner
 - 3 Years Experience with HP QuickTest Professional
 - 3 Years Experience with HP Quality Center
- **Certified**
 - HP Certified Instructor
 - HP Accredited Integration Specialist (AIS) Quality Center 9.2
 - HP QuickTest Professional 9.0 Certified Product Specialist
 - HP Software Universe, Mercury World, PSQT Track Speaker
- **QuickTest Professional 10 Beta Tester**



Combined thought leadership

More capability

More flexibility



Who We Are Now

- Technisource is the 7th largest IT solutions and services firm in North America
- 8,000+ employees & consultants
- Annual revenues in excess of \$600 M
- 60 branch offices across 29 states



Session Goals

- Automation Pitfalls & Lessons Learned
- Architectures & Abstraction Layers
- Tips & Tricks for Robust Execution
- Regular Expressions
- Explore the SAFFRON framework
- Dynamic Code Execution
- Single Source Documentation
- Examine several "Tools of the Trade"



The Myth

Buy the automation testing tool
and then...



Magic Happens!

Year 1: QA Project Manager Wants Automation. Tool Purchase Approved.
Project Manager Accepts Offer At New Company.

Year 2: New QA Manager Hires Lead Developer with C++ Experience.
Project Goal Set: “100% Automation”.

Year 3: Junior Visual Basic Developer Joins Team.
Lead Developer Moves To New Department.
Junior Developer Informed Project Is “High Priority Side Job”.

Year 4: Junior Developer Retires.

Results: 0 Defects found. Negative Return On Investment.

New General Manager Renewed Predecessor’s Automation Mandate.

Top 5 Project Pitfalls

- 1. Lack of Planning**
 - 2. Lack of Visible Progress**
 - 3. Lack of Trained Resources**
 - 4. Lack of Time Commitment**
 - 5. Lack of Buy-In**
 - 1. Developers**
 - 2. Testers**
 - 3. Management**
- **100% Automation = Unfocused Scattershot Development**
 - **Too much Maintenance to demo any success.**
 - **Resource with experience in a different programming language.**
 - **Automation by definition is Automatic, right?**
 - **Spread Tool Compatibility Rumors**
 - **Fearing Loss of Job Security**
 - **Want Tangible Metrics to Show ROI**

Mitigation Strategies

- 1. Lack of Planning**
 - 2. Lack of Visible Progress**
 - 3. Lack of Trained Resources**
 - 4. Lack of Time Commitment**
 - 5. Lack of Buy-In**
 - 1. Developers**
 - 2. Testers**
 - 3. Management**
- **Committed to 10% Automation of Existing Manual Tests.**
 - **Proof of Concept & Demo of Recent Success.**
 - **Hands-on Tool Training approved following Demo**
 - **Full Time Project Commitment (Initially Declined)**
 - **Repeat Defects Manually**
 - **Automate "Painful" Test Setup**
 - **Tracked Defect Severity as well as Execution Time**

Lessons Learned

2002 Automation Expo Lessons Learned

- **Take a “Best Bang For The Buck” approach**
 - Automate Common Features First
 - Automate Easy Features First
- **Use Functions / Actions / BPT Components**
 - Decrease Maintenance & New Test Development Time
- **A Defined Coding Standard**
 - Project Continues to be Supported
- **Framework Engine**
 - Non-Programmers Create Workflows
 - Best Practices are Internalized



Source: Hans Buwalda,
Jamie Mitchell,
Automation Handbook, Linda Hayes

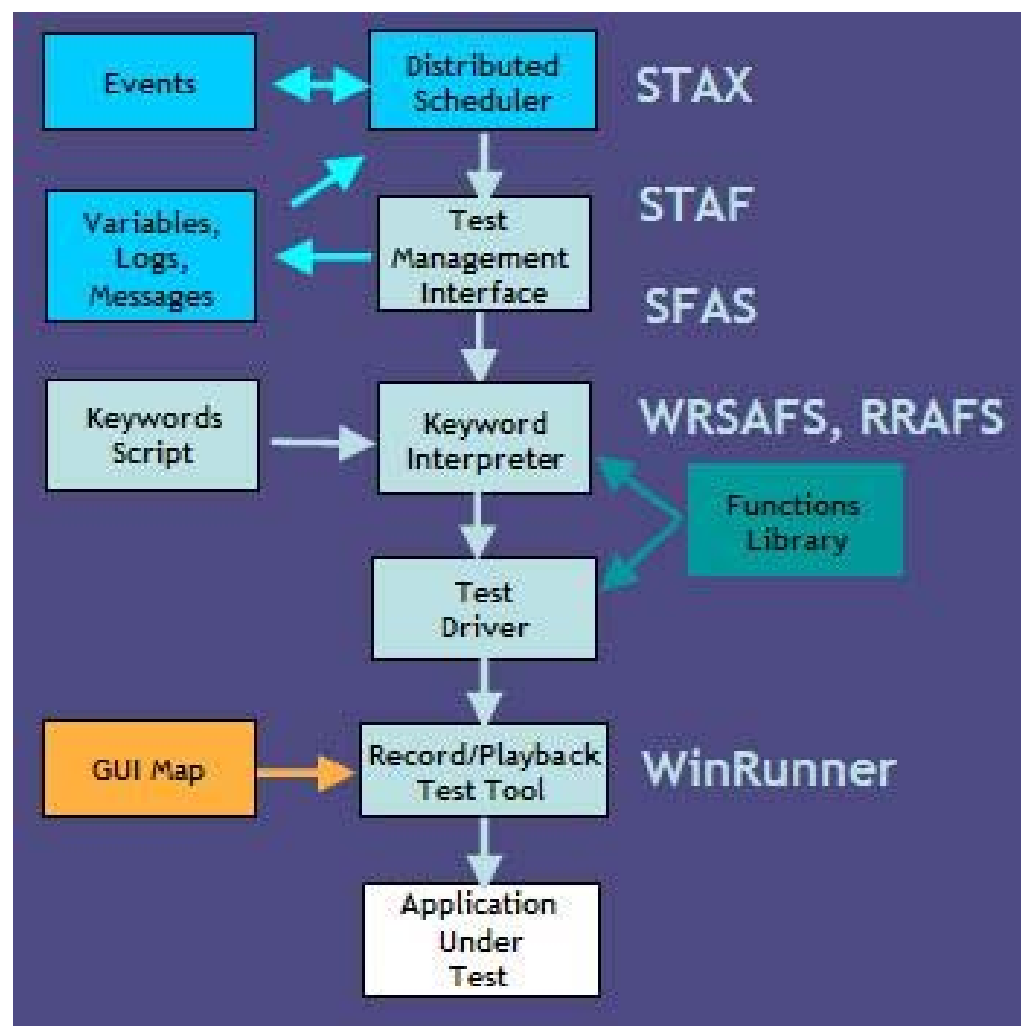
What is an Automation Framework?

What is a Framework?

- *"A **Test Automation Framework** is a set of assumptions, concepts and tools that provide support for automated software testing. The main advantage ... is the low cost for maintenance. "* Source: Wikipedia
- Abstraction
 - Non-Technical Resources create Automated Test Scripts
 - Hides Complexity
 - Internalizing Best Practices
 - Bolsters Robust Execution

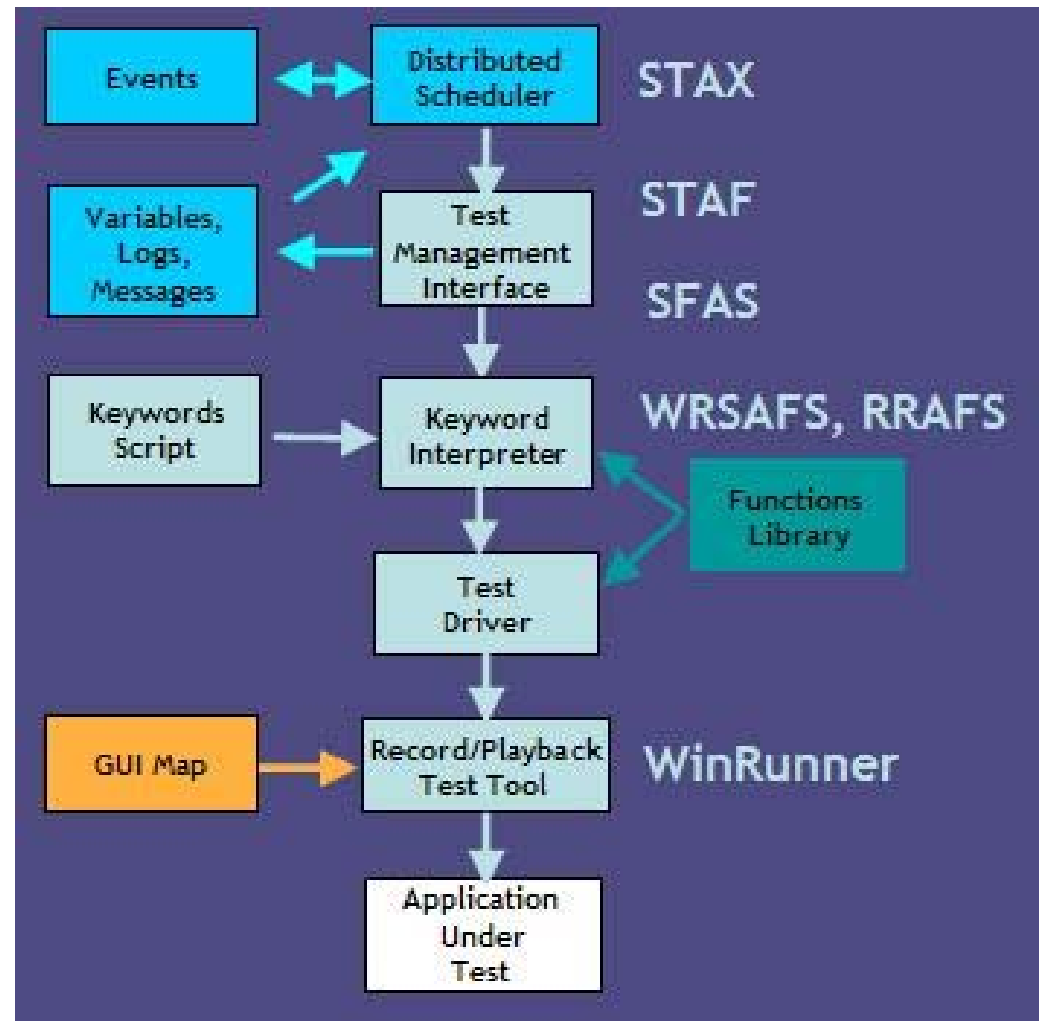
What is a Framework?

- The AUT is controlled by the Test Tool
- Has Repository of Objects Definitions
- A Keyword Library with pre-defined Functions, Actions or BPT Components drive the AUT.



What is a Framework?

- A Driver script may be implemented to use data to vary test script execution.
- Driver Scripts reduce the need to interact with "messy" script code, but does not minimize script coding.
 - Test Assets are managed in central repository.



Where Do We Start?

Architecture Options

- **Capture / Replay**
- **Framework Based**
- **Data-Driven**
- **Keyword-Driven**
- **Dialog-Driven**

- **Event-Driven**
- **Random**
- **Exhaustive**
- **Hybrid**
- **Exploratory**

Where Do We Start?

Architecture Options

- **Capture / Replay**

- **High Maintenance**
- **Low Readability**

<pre> Global Variables Assumed the system is running default 3224 & 768 LaunchURL: describes the url Dim gScreenResolution as Integer = 3224 Dim gScreenResolution as Integer = 768 ALT URL - Can be read from QC Dim gURL Dim gURL = gURL = "" Used for base access to debugging code Dim RunCode: RunCode = True at the end of the script, should always be true Uncomment = False Dim gURL Dim gURL = gURL = "" Used for base access to debugging code Dim RunCode: RunCode = True at the end of the script, should always be true Uncomment = False Dim gURL Dim gURL = gURL = "" Used for base access to debugging code Dim RunCode: RunCode = True at the end of the script, should always be true Uncomment = False Dim gURL Dim gURL = gURL = "" Used for base access to debugging code Dim RunCode: RunCode = True at the end of the script, should always be true Uncomment = False Dim gURL Dim gURL = gURL = "" Used for base access to debugging code Dim RunCode: RunCode = True at the end of the script, should always be true Uncomment = False </pre>	<pre> Global Variables Assumed the system is running default 3224 & 768 LaunchURL: describes the url Dim gScreenResolution as Integer = 3224 Dim gScreenResolution as Integer = 768 ALT URL - Can be read from QC Dim gURL Dim gURL = gURL = "" Used for base access to debugging code Dim RunCode: RunCode = True at the end of the script, should always be true Uncomment = False Dim gURL Dim gURL = gURL = "" Used for base access to debugging code Dim RunCode: RunCode = True at the end of the script, should always be true Uncomment = False Dim gURL Dim gURL = gURL = "" Used for base access to debugging code Dim RunCode: RunCode = True at the end of the script, should always be true Uncomment = False Dim gURL Dim gURL = gURL = "" Used for base access to debugging code Dim RunCode: RunCode = True at the end of the script, should always be true Uncomment = False Dim gURL Dim gURL = gURL = "" Used for base access to debugging code Dim RunCode: RunCode = True at the end of the script, should always be true Uncomment = False </pre>	<pre> Global Variables Assumed the system is running default 3224 & 768 LaunchURL: describes the url Dim gScreenResolution as Integer = 3224 Dim gScreenResolution as Integer = 768 ALT URL - Can be read from QC Dim gURL Dim gURL = gURL = "" Used for base access to debugging code Dim RunCode: RunCode = True at the end of the script, should always be true Uncomment = False Dim gURL Dim gURL = gURL = "" Used for base access to debugging code Dim RunCode: RunCode = True at the end of the script, should always be true Uncomment = False Dim gURL Dim gURL = gURL = "" Used for base access to debugging code Dim RunCode: RunCode = True at the end of the script, should always be true Uncomment = False Dim gURL Dim gURL = gURL = "" Used for base access to debugging code Dim RunCode: RunCode = True at the end of the script, should always be true Uncomment = False Dim gURL Dim gURL = gURL = "" Used for base access to debugging code Dim RunCode: RunCode = True at the end of the script, should always be true Uncomment = False </pre>	<pre> Global Variables Assumed the system is running default 3224 & 768 LaunchURL: describes the url Dim gScreenResolution as Integer = 3224 Dim gScreenResolution as Integer = 768 ALT URL - Can be read from QC Dim gURL Dim gURL = gURL = "" Used for base access to debugging code Dim RunCode: RunCode = True at the end of the script, should always be true Uncomment = False Dim gURL Dim gURL = gURL = "" Used for base access to debugging code Dim RunCode: RunCode = True at the end of the script, should always be true Uncomment = False Dim gURL Dim gURL = gURL = "" Used for base access to debugging code Dim RunCode: RunCode = True at the end of the script, should always be true Uncomment = False Dim gURL Dim gURL = gURL = "" Used for base access to debugging code Dim RunCode: RunCode = True at the end of the script, should always be true Uncomment = False Dim gURL Dim gURL = gURL = "" Used for base access to debugging code Dim RunCode: RunCode = True at the end of the script, should always be true Uncomment = False </pre>	<pre> Global Variables Assumed the system is running default 3224 & 768 LaunchURL: describes the url Dim gScreenResolution as Integer = 3224 Dim gScreenResolution as Integer = 768 ALT URL - Can be read from QC Dim gURL Dim gURL = gURL = "" Used for base access to debugging code Dim RunCode: RunCode = True at the end of the script, should always be true Uncomment = False Dim gURL Dim gURL = gURL = "" Used for base access to debugging code Dim RunCode: RunCode = True at the end of the script, should always be true Uncomment = False Dim gURL Dim gURL = gURL = "" Used for base access to debugging code Dim RunCode: RunCode = True at the end of the script, should always be true Uncomment = False Dim gURL Dim gURL = gURL = "" Used for base access to debugging code Dim RunCode: RunCode = True at the end of the script, should always be true Uncomment = False Dim gURL Dim gURL = gURL = "" Used for base access to debugging code Dim RunCode: RunCode = True at the end of the script, should always be true Uncomment = False </pre>
--	--	--	--	--

```
SystemUtil.Run "C:\Program Files\Internet Explorer\EXPLORE.EXE", "", "C:\Documents and Settings\grosspa", "open"
Browser("Technisource").Navigate "https://www.technisource.com"
```

'Login – Repeated in every test

```
Browser("Technisource").Dialog("Login").WinEdit("User name:").Set "user"
Browser("Technisource").Dialog("Login").WinEdit("Password:").Set "pwd1"
Browser("Technisource").Dialog("Login").WinButton("OK").Click
```

```
Browser("Technisource").Dialog("Connect to inside.spherion.com").Activate
```

Problem: Every Test Has Identical Code.

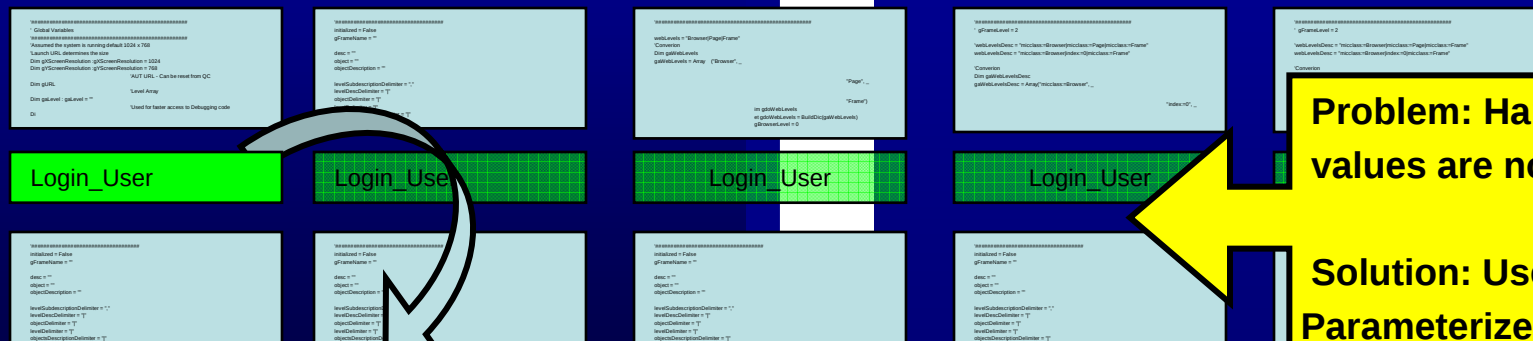
Solution: Function Based Framework

Where Do We Start?

Architecture Options

- **Framework-Based (Function Library)**

- **Lower Maintenance**
- **Hidden Complexity**
- **No Scalability**



Problem: Hard-coded values are not scalable.

Solution: Use Parameterized values

Public Function Login_User

```
Browser("Technisource").Dialog("Login").WinEdit("User name:").Set "user"
Browser("Technisource").Dialog("Login").WinEdit("Password:").Set "pwd1"
Browser("Technisource").Dialog("Login").WinButton("OK").Click
```

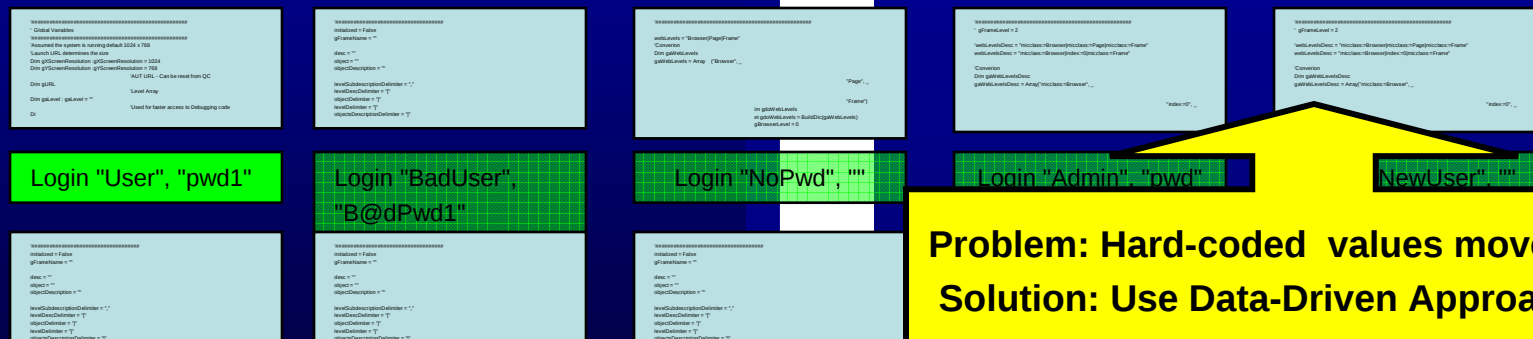
End Function

Where Do We Start?

Architecture Options

- Framework-Based (Function Library)

- Lower Maintenance
- Some Scalability



Public Function Login (strUser, strPassword)

Browser("Technisource").Dialog("Login").WinEdit("User name:").Set **strUser**

Browser("Technisource").Dialog("Login").WinEdit("Password:").Set **strPassword**

Browser("Technisource").Dialog("Login").WinButton("OK").Click

End Function

Where Do We Start?

Architecture Options

• Data-Driven

• Highly Scalable

Function Login

Login [User]

Login [BadUser]

Login [NoPwd]

UserName

Password

Network

"User"

"pwd1"

NA

"BadUser"

"B@dpwd"

NA

"NoPwd"

""

NA

Data Comes from
Data Table, Data File
or SQL Query.

Public Function Login

```
Browser("Technisource").Dialog("Login").WinEdit("User name:").Set DataTable ("UserName")
Browser("Technisource").Dialog("Login").WinEdit("Password:").Set DataTable ("Password")
Browser("Technisource").Dialog("Login").WinButton("OK").Click
```

End Function

Where Do We Start?

Architecture Options

- Keyword-Driven**

Login "User"

Search_For "Pink"

SelectFromList "Payment", "Cash"

- Multiple Approaches**

- Functions Only
- Actions (QTP)
- BPT Components (QC)

- Coarseness**

- Small: Select "Cash"
- Large: Find "Last Order"

Global Data Table

Result	Keyword	Type
	Login	User
	Verify_Cart_Is	Empty
	Search_For	Pink

Local Data Table

User_Type	UserName	Password	Network
User	"User"	"pwd1"	
Bad User Name	"BadUser"	"B@o yd"	NA
No Password	"NoPassword"		NA

Default is the
"Happy Path"

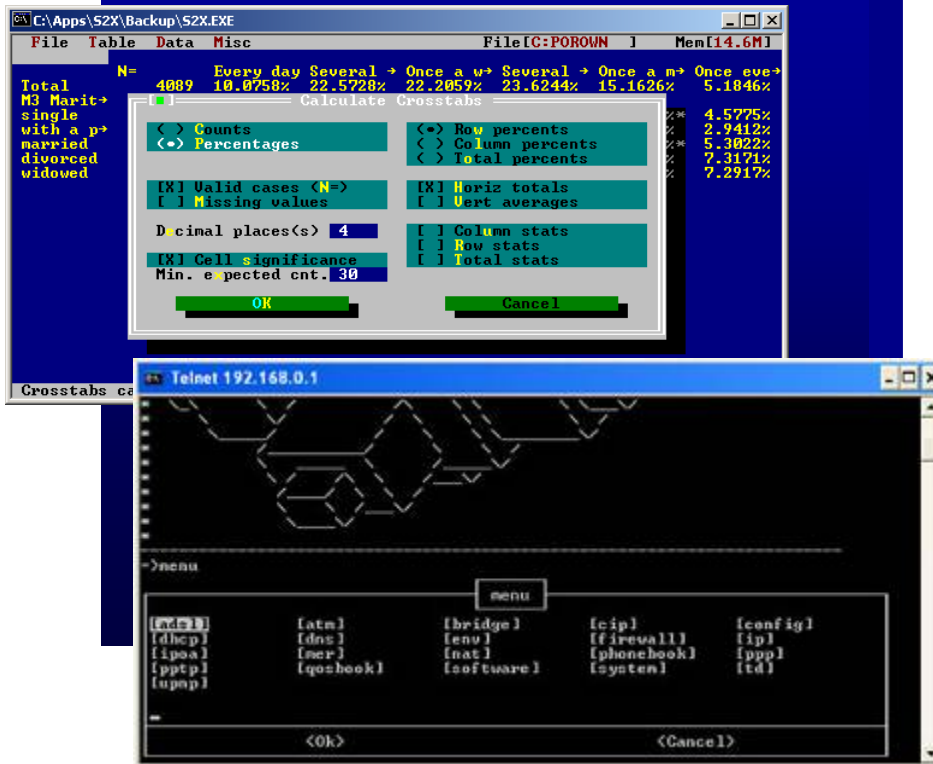
Public Function Login (UserName)

```
Find_User_In_DataTable ("Value", GlobalDataTable)
Browser("Technisource").Dialog("Login").WinEdit("User name:").Set DataTable ("UserName", LocalDataTable)
Browser("Technisource").Dialog("Login").WinEdit("Password:").Set DataTable ("Password", LocalDataTable)
Browser("Technisource").Dialog("Login").WinButton("OK").Click
End Function
```

Where Do We Start?

Architecture Options

- Dialog-Driven

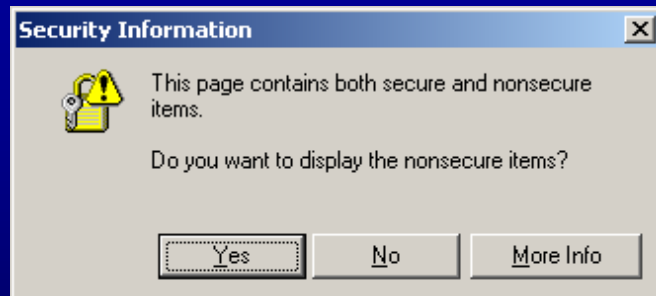


- Legacy DOS or Terminal Emulation (Mainframe) Apps
 - Whole Windows
 - Pages
 - Tabs
- Generic as Possible
- Avoid Trivial Dialogs
- Avoid Dialogs with Many Fields

Where Do We Start?

Architecture Options

- **Event-Driven**



- **This document failed to print**

Document name: 'Microsoft PowerPoint - Secre...'
Printer name: 'Canon iR2200-3300 PS3'
Time sent: 6:45:35 AM 9/21/2008

Click here to open the print queue, and then for assistance, click Troubleshooter on the Help menu.

- **Unknown Preconditions**
 - Object Property State
 - AJAX Page Updates
 - SOA Testing
- **Recovery Scenarios**
 - Popup Dialogs
 - Application Errors
 - QTP Run Errors
- **Con: Can slow Execution**

Where Do We Start?

Architecture Options

- Exhaustive

- Large Combinations
- Time Consuming

	AB	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y											
1	AUTO TESTING MATRIX																																		
2	OLD Application																																		
3					Test Cases - name example																														
4	Name				Description				TEST01	TEST02	TEST03	TEST04	TEST05	TEST06	TEST07	TEST08	TEST09	TEST10	TEST11	TEST12	TEST13	TEST14	TEST15	TEST16	TEST17	TEST18	TEST19								
236	Automobile Insurance																																		
237	Automobile Insurance				Liability				X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X								
238	Automobile Insurance				Personal Injury Protection				X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X								
239	Automobile Insurance				Additional Personal Injury Protection						X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X								
240	Automobile Insurance				Medical Payment				X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X								
241	Automobile Insurance				Uninsured Motorists				X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X								
242	Automobile Insurance				Comprehensive				X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X								
243	Automobile Insurance				Specified Perils							X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X								
244	Automobile Insurance				Collision				X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X								
245	Automobile Insurance				Towing						X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X								
246	Automobile Insurance				Fleet										X	X	X	X	X	X	X	X	X	X	X	X	X								
247	Automobile Insurance				Non-Fleet				X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X								
248	Automobile Insurance				Federal Terrorism				X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X								
249																																			
250	Automobile Insurance																																		
251	Automobile Insurance				Non-Owned Auto Coverage				X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X								
252	Automobile Insurance				Hired Physical Damage				X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X								
253	Automobile Insurance				Broadended PIP							X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X								
254	Automobile Insurance				Driver Other Car							X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X								

Where Do We Start?

Architecture Options

- Hybrid
- Smart Monkey Testing

- Random + Keyword Driven
- Pro: Duplicate Random Path

Global Data Table

Result	Action	Data	Comments		
Pass	Search_For	"Pink"	Random Value		
Warning	Logout		"Logout" Button Disabled. App returned to home state.		
Pass	Login	"User"	Random Value		

Where Do We Start?

Architecture Options

- Hybrid Testing with QTP
- Record + Playback
- Framework Driven
 - Functions
 - Subs
- Keyword-Driven
 - Objects
 - Function, Actions
 - BPT Components

- Data-Driven
 - Data Tables
 - SQL Queries
 - Text Files
- Event-Driven
 - Recovery Scenarios
- Random
 - Random Parameterized Values

Where Do We Start?

Automation Design Options

Team Skill Level

- **Keyword View**
- **Expert View**
- **Business Process Testing**

Abstraction Layers

- **Functions**
- **Actions or**
- **BPT Components**

Pitfalls of Automation

- 100% Automation Mandate
- Poor Selection Criteria
- No Abstraction Layer
- Unorganized Development
- Automation is not Development
- Lack of Robust Code Execution



Be Realistic!

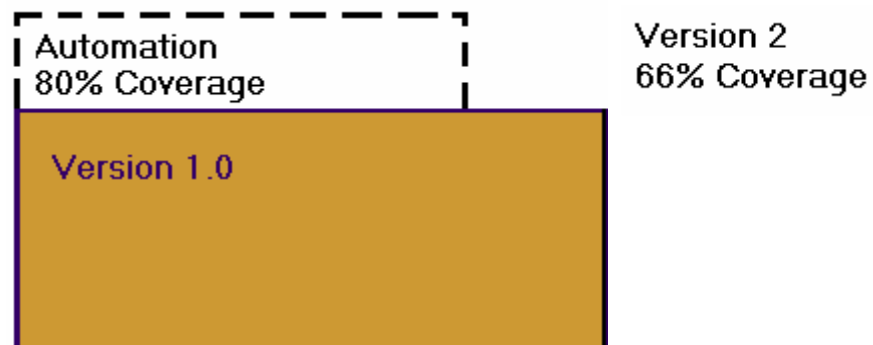
- Automation is Code Development
 - Don't assume management understands this.
 - Follow Development's lead
- Project Analysis
 - Prioritized list of what can be automated
 - What can not, and should not be, automated
- Some clients have Achieved 80% - 90% Automation but...
 - Attained over a number of years
 - 40% - 60% Automation can make a significant impact*

What can you do with unrealistic goals?

**Source: Automation Handbook, Linda Hayes*

Avoiding the “100% Automation” Trap

- Define “100% Automation”
 - Is it existing manual tests?
 - Requirements coverage?
 - Code Coverage?
- New releases will have more functionality, constantly decreasing code coverage
- A percentage of the code could be unreachable code, such as uncalled functions.



Solution: Let's do the math

Developer to Automated Tester Ratio?

Number of months ahead?

How much of the App is complete now?

Developers * Months = Today's % of Automation

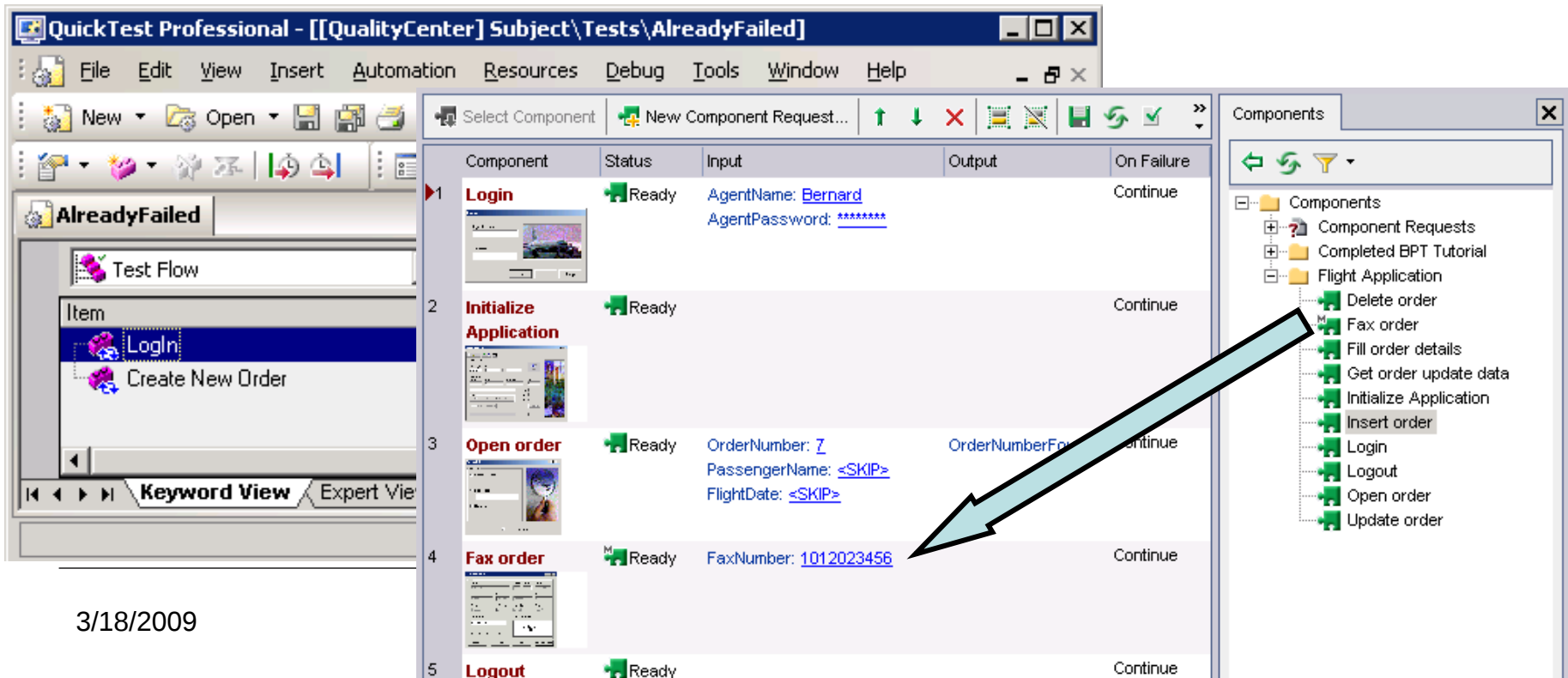
Is this beyond the end of the project?

Test Selection Criteria

- Iterative Tests
- Time Consuming
- Tedious and Precise
- Avoid Visual and User Acceptance Tests

Abstraction Layers

- QTP
 - Reusable Actions
 - Low & High Abstraction
 - Many Supporting Tools
- Business Process Testing Components
 - Reusable
 - High Level Abstraction
 - End-to-End Scenarios



The screenshot displays the QuickTest Professional (QTP) interface. The main window shows a test flow diagram with steps: Login, Initialize Application, Open order, Fax order, and Logout. The 'Keyword View' tab is active, showing the details of the 'Fax order' step. A large blue arrow points from the 'Fax order' step in the test flow to the 'Fax order' component in the 'Components' list on the right.

Component	Status	Input	Output	On Failure
1 Login	Ready	AgentName: Bernard AgentPassword: *****		Continue
2 Initialize Application	Ready			Continue
3 Open order	Ready	OrderNumber: Z PassengerName: <SKIP> FlightDate: <SKIP>	OrderNumberFor	Continue
4 Fax order	Ready	FaxNumber: 1012023456		Continue
5 Logout	Ready			Continue

The 'Components' list on the right includes:

- Component Requests
- Completed BPT Tutorial
- Flight Application
 - Delete order
 - Fax order
 - Fill order details
 - Get order update data
 - Initialize Application
 - Insert order
 - Login
 - Logout
 - Open order
 - Update order

Abstraction Layers

- Custom and Overridden Object Methods (SIFL)
 - Reusable
 - Internalize Best Practices
 - Fast

```
1. BrowserDefault.Navigate "http://newtours.mercuryinteractive.com"
2. BrowserDefault.Link("innerText:=Cruises").Click
3. Set oTable = BrowserDefault.WebTable("innerText:=Cruise Itinerary.*")
4. sDeparture = oTable.GetXWhereYisString("Departure", "Port of Call", "Skagway, Alaska")
5. assertStringEquals sDeparture, "5 pm"
```

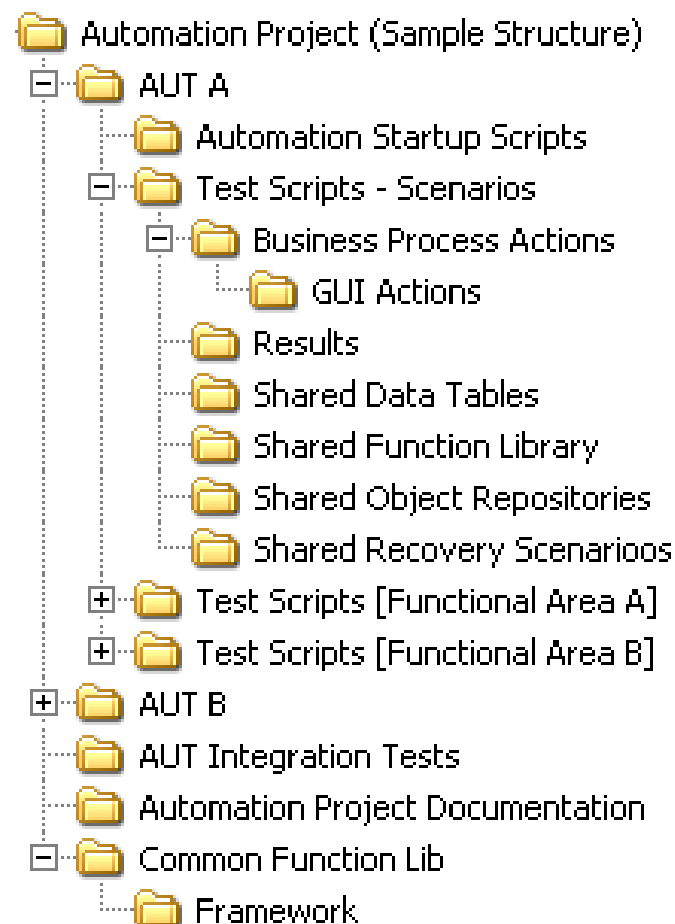
- Supporting Function Library (SAFFRON)
 - Reusable
 - Increased Readability
 - Internalize Best Practices

Visual Basic:

```
1. BrowseTo "http://google.com/"
2. EnterTextIn "q", "Hello World!"
3. Activate "WebButton", "Google Search"
4. Activate "Link", "Hello, World Page"
```

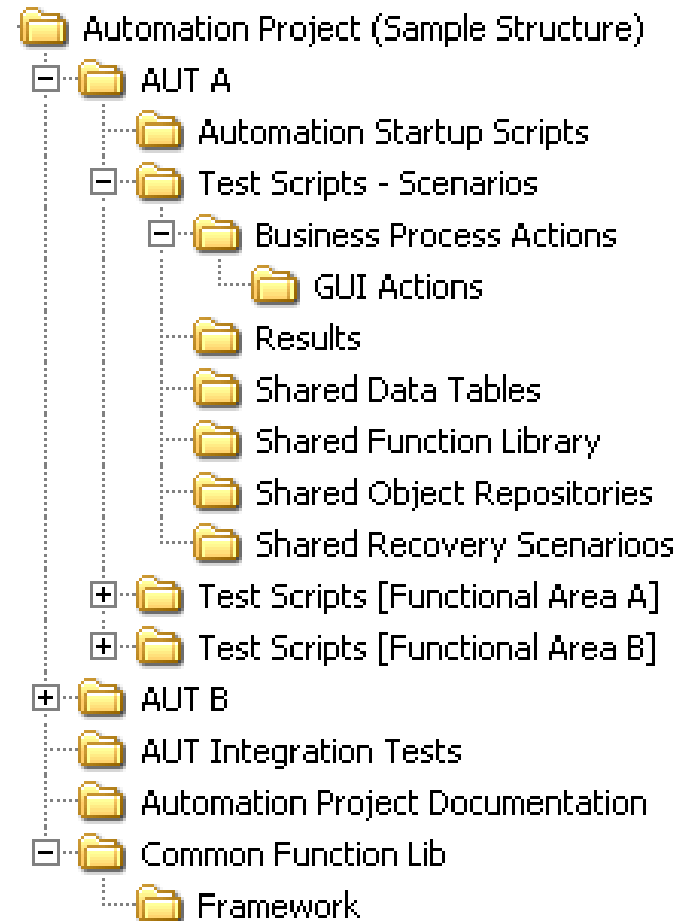
Organized Development

- Modular Organization
 - Applications Under Test
 - Test Categories
 - Functional
 - Scenarios
 - Integration
 - Shared Project Resources
 - Objects
 - Function Libraries
 - Recovery Scenarios



Organized Development

- Startup Scripts
- 3 Level Modular Test Design
 - Scripts
 - Business Process
 - GUI Interaction
- Common Framework
- Project Documentation



Follow Development's Lead

Naming Conventions
Option Explicit
Unit Testing

Naming Conventions

- Why are Naming Conventions so Important?

Print **"Hello World!"** *' Prints 'Hello World!': My First Program!*

a = **1** : b = **2**

c = a + b

Print c *' Prints '5': My First Arithmetic Application!*

a = **"Base"** : b = **"ball"**

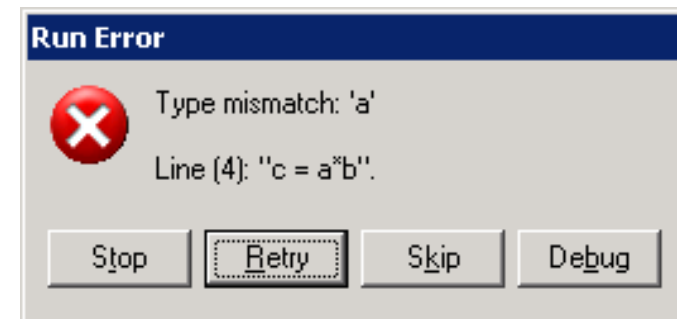
c = a + b

Print c *' Prints 'Baseball': My First String Concatenation!*

a = **"Base"** : b = **"ball"**

c = a * b

Print c *'Multiply
'My First Bug!*



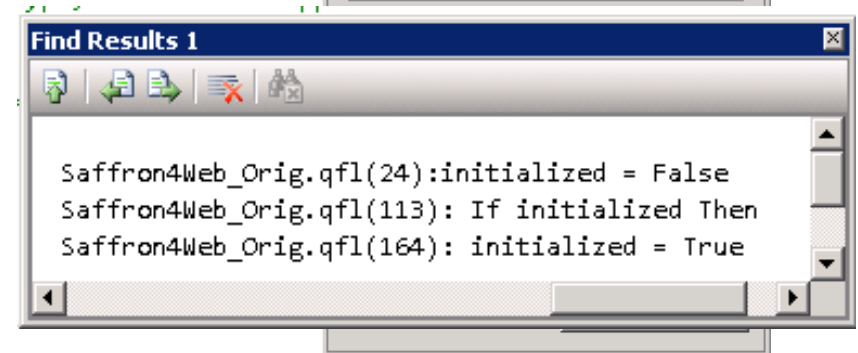
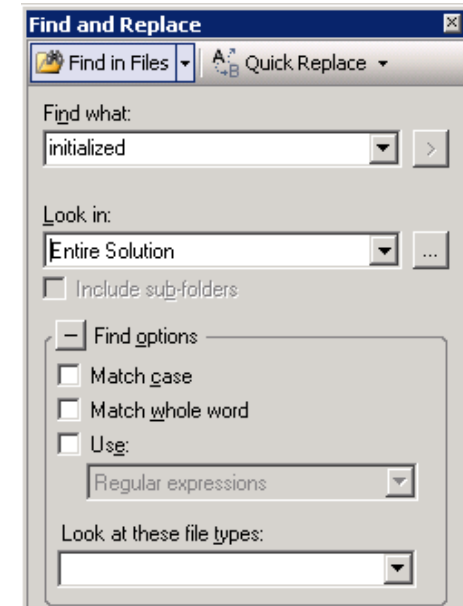
Naming Conventions

- Use Hungarian Notation for variables
 - **Function** fOpen (strPath)
 - strMenu = "**New > File**"
- *strString* *strCity*
- *intInteger* *intAge*
- *lngLong* *lngChinaPop*
- *objObject* *objBrowser*
- *arrArray* *arrBirthdays*
- *dicDictionary*
- *gGlobalVariable*
- *goGlobalObject*
- *gdGlobalDictionary*
- *dpDescriptiveProgramming*
- *rgxRegularExpression*
- *estrExecutableString*

Naming Conventions

What if a variable changes it's expected data type?

- Solution #1: Convert it.
 - VarType
 - cBool, cByte, cCur, cDate, cDble, cLng, cSng, cStr
 - IsArray, isObject, isDate
 - Challenge: `isNumeric` ("123") 'evaluates to...?
- Solution #2: Change It
 - Not with Search and Replace
 - Test Design Studio – Find in Files



Option Explicit

Option Explicit

- Con: Forces Variable Declaration, More Up Front Work.
- Pro: Reduces Errors Due To Variable Misspellings
- Con: Must Be In Every Function Library

How Many Variables Are In The Code Below?

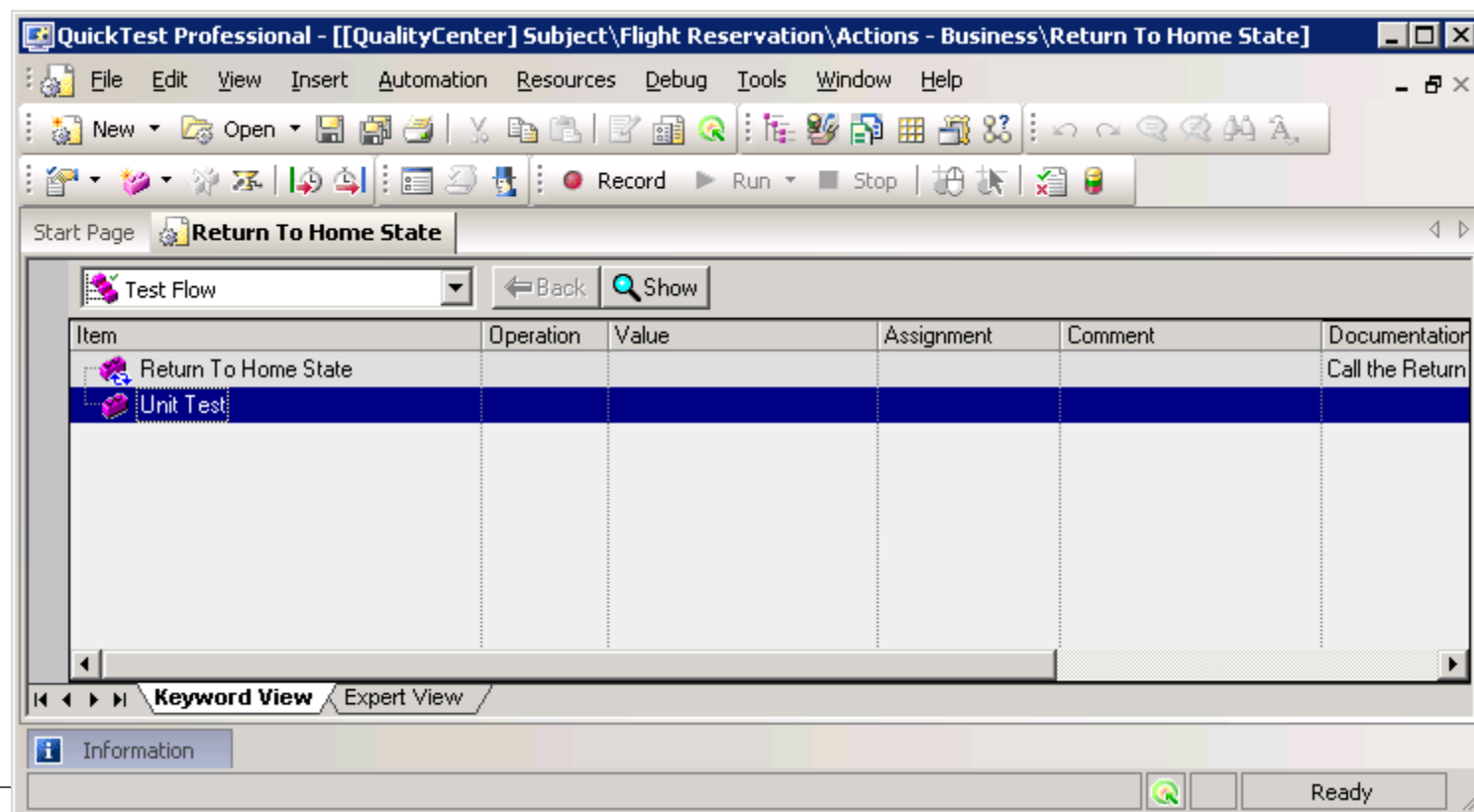
Option Explicit : Dim sReceipt, micStatus, sDescription
sReceipt = "Complete"

IF sReciept = "" THEN	<i>' Variable is undefined: 'sReciept'</i>
micStartus = micPass	<i>' Variable is undefined: 'micStartus'</i>
sReciept = "Blank"	<i>' Variable is undefined: 'sReciept'</i>
Else	
micStartus = micFail	<i>' Variable is undefined: 'micStartus'</i>
Receipt = "Printed"	<i>' Variable is undefined: 'Receipt'</i>
End If	

Reporter micStatus, sDescription, sReceipt *' Variable is undefined: 'sDescription'*

Unit Testing

- Best Practice: Create Individual Action Scripts
- One Reusable Action named after the Script
- One Non-Reusable Action named 'Unit Test'



Best Practices

Object Verification

Assume Failure

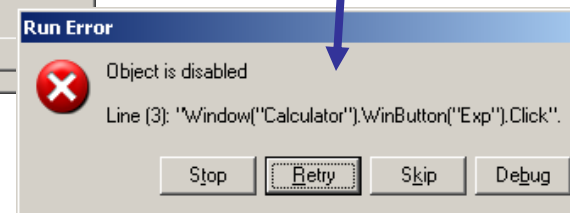
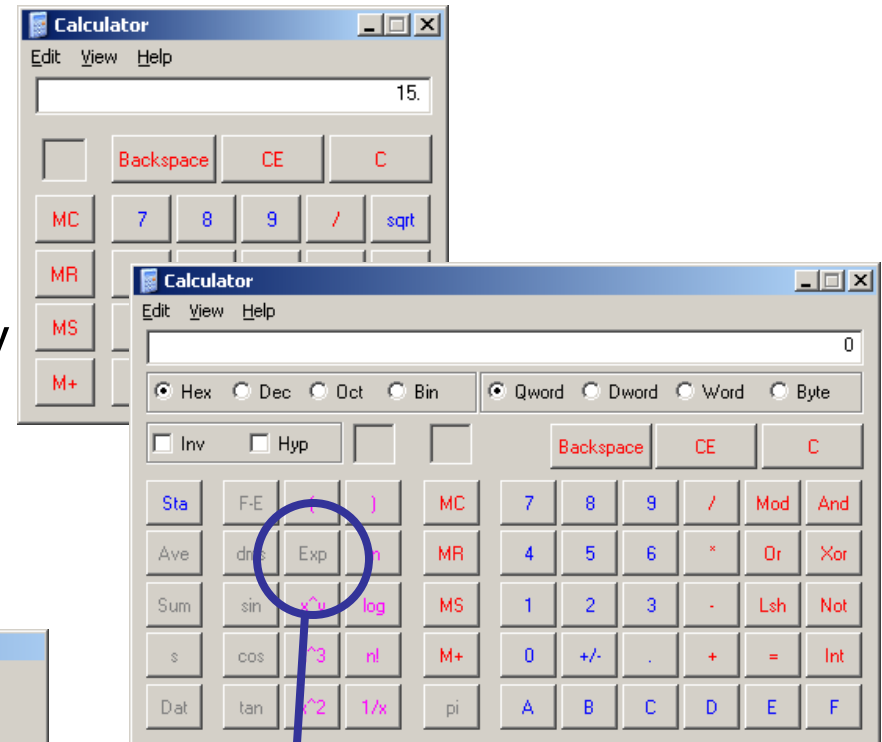
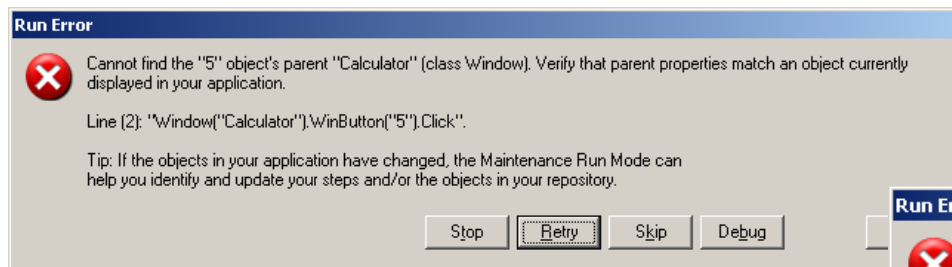
Return to Home State

Self-Closing Dialogbox

Object Verification

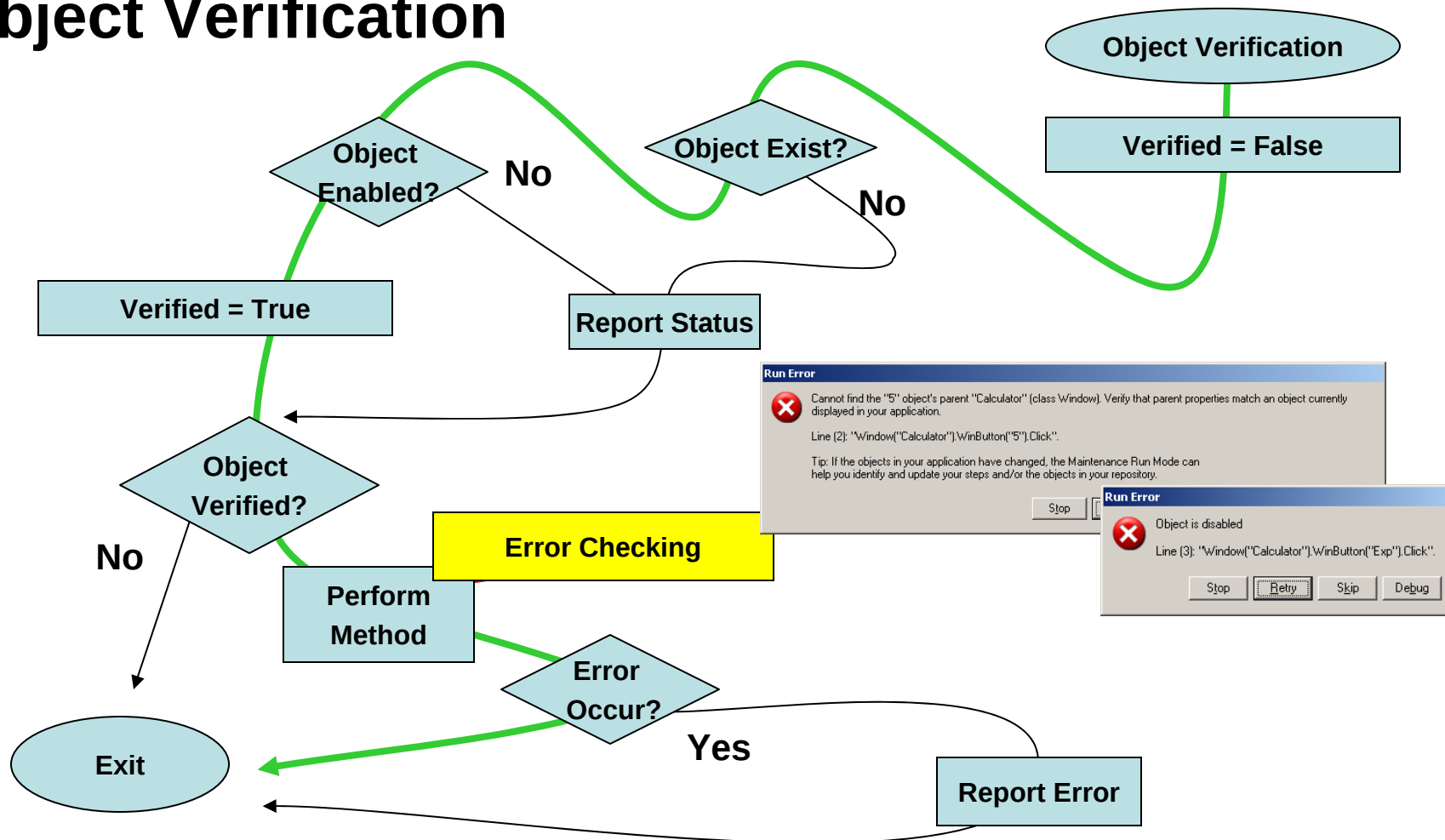
Robust Test Execution

- Problem: Object Not Found
 - Multiple Instances of AUT
 - New Release of Application
 - Application Closes Unexpectedly
- Run Errors Stops Execution



Tips & Tricks

Object Verification



Object Verification

```
1: 'Check the object was found and is enabled
2: 'before executing a method.
3:
4: Public Function VerifyObject (oObject)
5:     If oObject.Exist = True Then
6:
7:         If oObject.Enabled = True Then
8:             VerifyObject = True
9:         Else
10:             VerifyObject = False
11:             'Report Error: Object not enabled
12:         End If
13:
14:     Else
15:         VerifyObject = False
16:         'Report Error: Object not found
17:     End If
18: End Function
```

Assume Failure

Assume Failure

```
1: 'Check the object was found and is enabled
2: 'before executing a method.
3:
4: Public Function VerifyObject (oObject)
5:     VerifyObject = False
6:     Status = "Does not Exist"
7:     If oObject.Exist = True Then
8:
9:         Status = "Is not Enabled"
10:        If oObject.Enabled = True Then
11:            VerifyObject = True
12:        End If
13:
14:    End If
15:    'Report : Object Status
16: End Function|
```

```
19:
20:
21:
22:
23:
24: Public Function RobustClick (oObject)
25:     If VerifyObject (oObject) Then
26:         on error resume next
27:         oObject.click
28:
29:         If Err <> 0 Then
30:             'Report Error
31:         End If
32:         On error goto 0
33:     End If
34: End Function
```

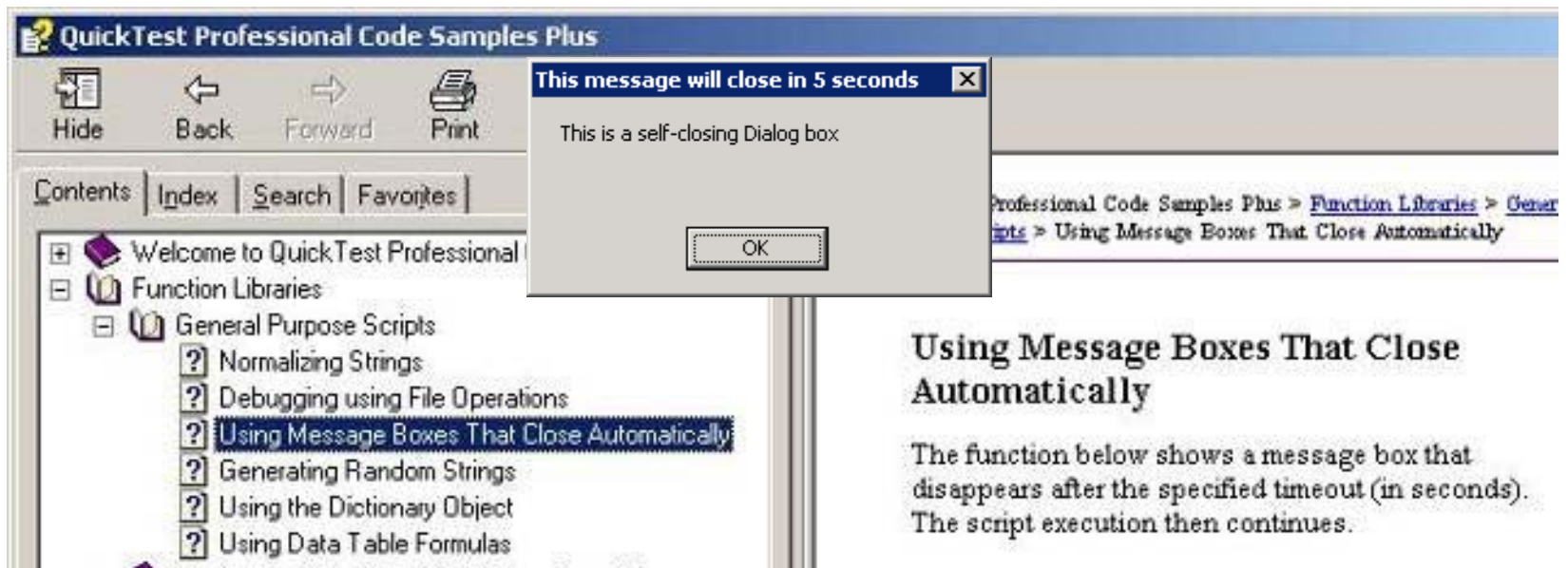
Return to Home State

n To Home State

```
graph TD; Start([Return To Home State]) --> AppRunning{App Running?}; AppRunning --> Launch[Launch App  
(Navigate to URL)]; AppRunning --> AppStable{App Stable?}; AppStable --> CloseApp1[Close App]; AppStable --> UserLoggedIn{User Logged In?}; UserLoggedIn --> Logout[Log Out  
Log In User]; UserLoggedIn --> HomeState{Home State?}; HomeState --> HomeButton{Home Button?}; HomeButton --> ClickHome[Click Home Button]; ClickHome --> GAFFalse{gAlreadyFailed = False?}; GAFFalse --> Exit([Exit]); GAFFalse --> NextAttempts([Next of 5 Attempts]); NextAttempts --> AppRunning; GAFFalse --> GAFTrue[gAlreadyFailed = True]; GAFTrue --> AppRunning;
```

The flowchart illustrates the process of returning to the home state. It begins with a decision point "App Running?". If the app is running, it proceeds to "Launch App (Navigate to URL)". If the app is stable, it checks if the user is logged in. If logged in, it checks if the user wants to log out or log in again. If the user is not logged in, it checks if they want to click the home button. Both paths lead to a decision point "gAlreadyFailed = False?". If false, it leads to "Exit". If true, it leads to "Next of 5 Attempts", which loops back to "App Running?". A final box at the bottom indicates "gAlreadyFailed = True", which also loops back to "App Running?".

- **Self Closing Message Boxes**
 - High Visibility
 - Allows Robust Test Execution



***Start > Programs > Quick Test Professional > Documentation >
“Quick Test Professional Code Samples”***

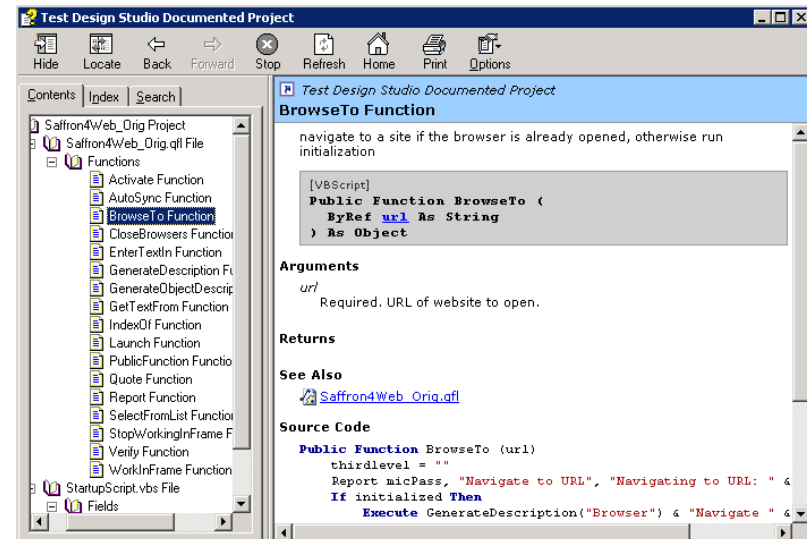
Self Closing Dialog Demo

Let's Take a Break

Coming Up...

Regular Expressions

The Regulator



S.A.F.F.R.O.N.

Dynamic Code Execution

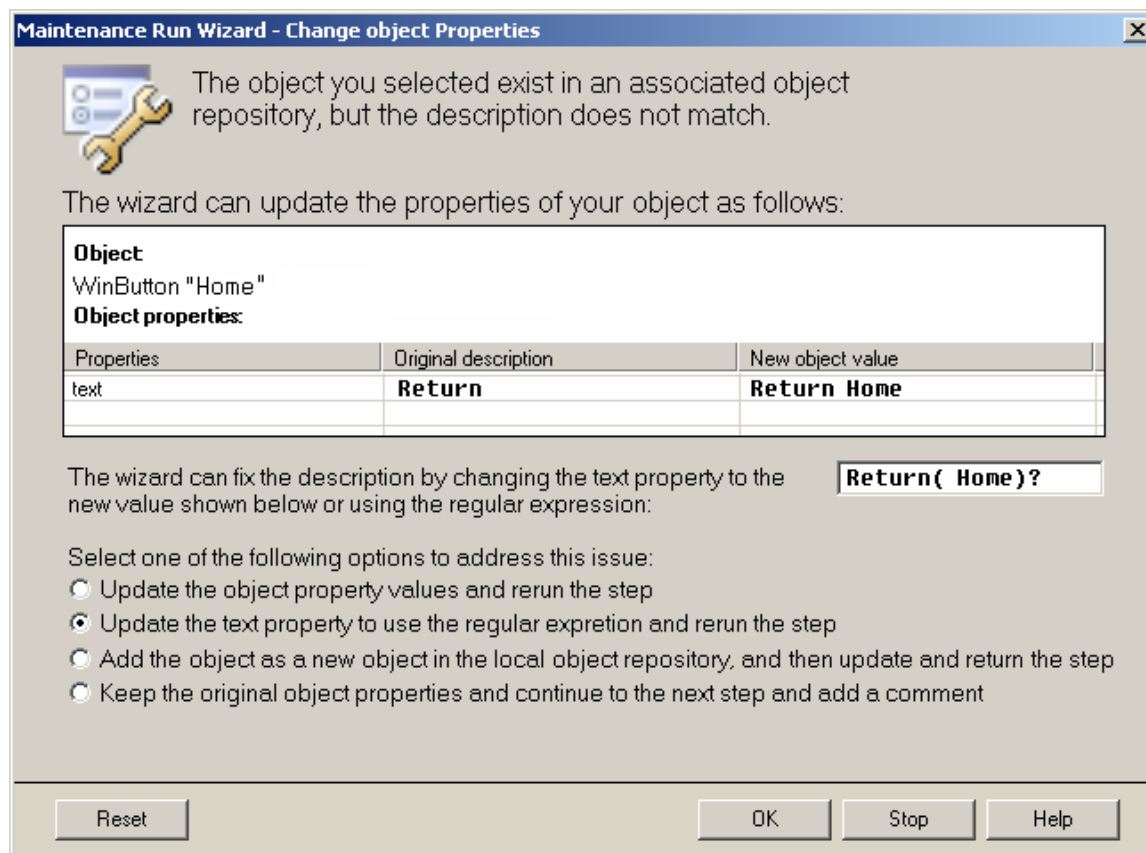
Regular Expressions with The Regulator

Tools of the Trade

The Regulator



Regular Expression Testing Tools

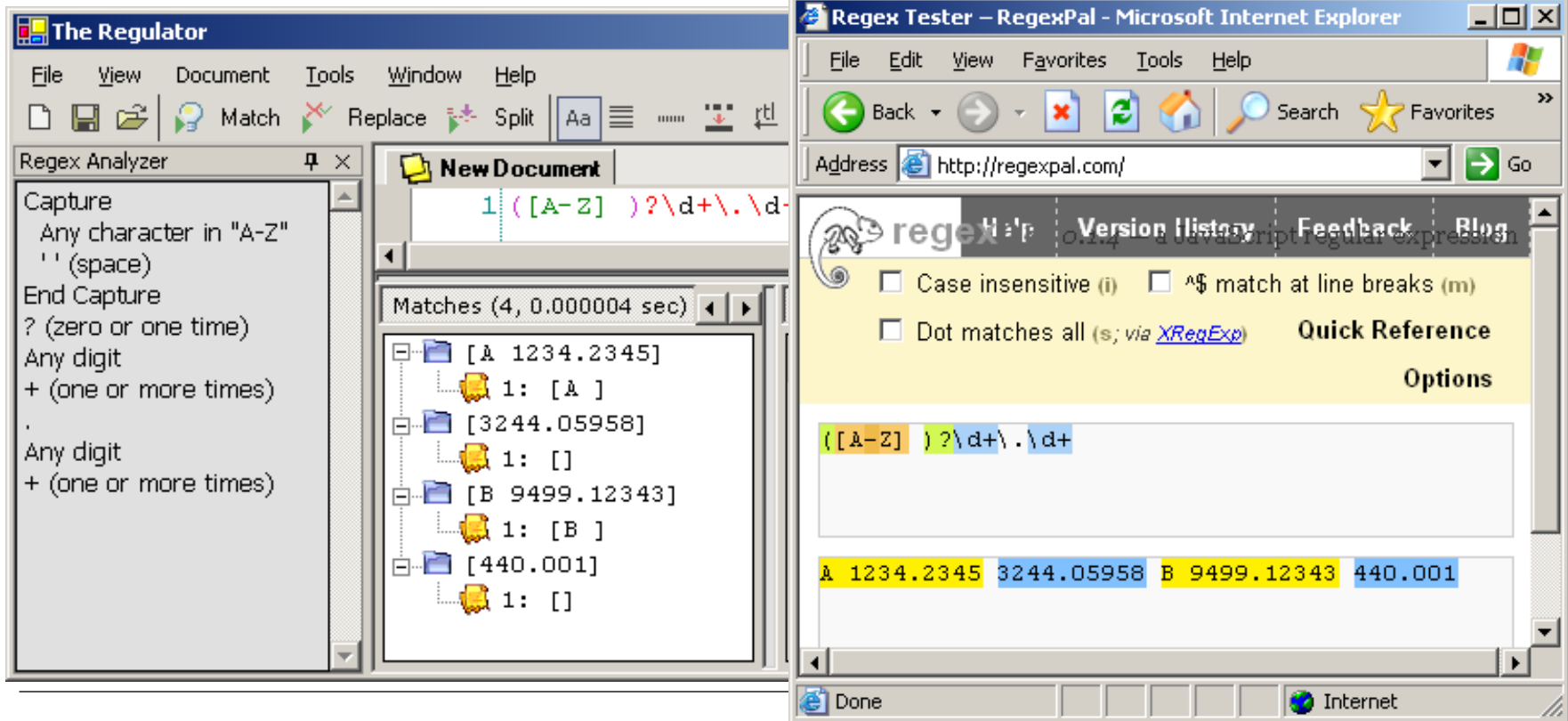


QTP 9.5: Maintenance Run Mode Features Automatic RegEx Generation

Regular Expressions

- When are Regular Expressions Used?
 - Primary Use: Captures matching text from a string

"A 1234.2345 3244.05958 B 9499.12343 440.001"



The left screenshot shows 'The Regulator' software. The 'Regex Analyzer' window displays the pattern `([A-Z])?\d+\.\d+`. The 'Matches' window shows four matches: `[A 1234.2345]`, `[3244.05958]`, `[B 9499.12343]`, and `[440.001]`. Each match is broken down into its components: the first capture group (the letter), the second capture group (the space), and the third capture group (the number). The right screenshot shows 'Regex Tester - Regexpal' in a Microsoft Internet Explorer browser. The address bar shows `http://regexpal.com/`. The 'regexpal' logo is visible. The 'Options' section includes checkboxes for 'Case insensitive (i)', 'Dot matches all (s; via XRegExp)', and '^\$ match at line breaks (m)'. The 'Quick Reference' section shows the pattern `([A-Z])?\d+\.\d+`. The 'Matches' section shows the same four matches as the left screenshot, with the first capture group highlighted in yellow, the second in blue, and the third in green.

Regular Expressions

- Why use Regular Expressions?

To reduce Maintenance when ...

- Developers change Object Properties Release to Release
- In Descriptive Programming with Inconsistent Object Design

Release 0.1a



Text Property "Return"

Release 0.1b



Text Property "Return Home"

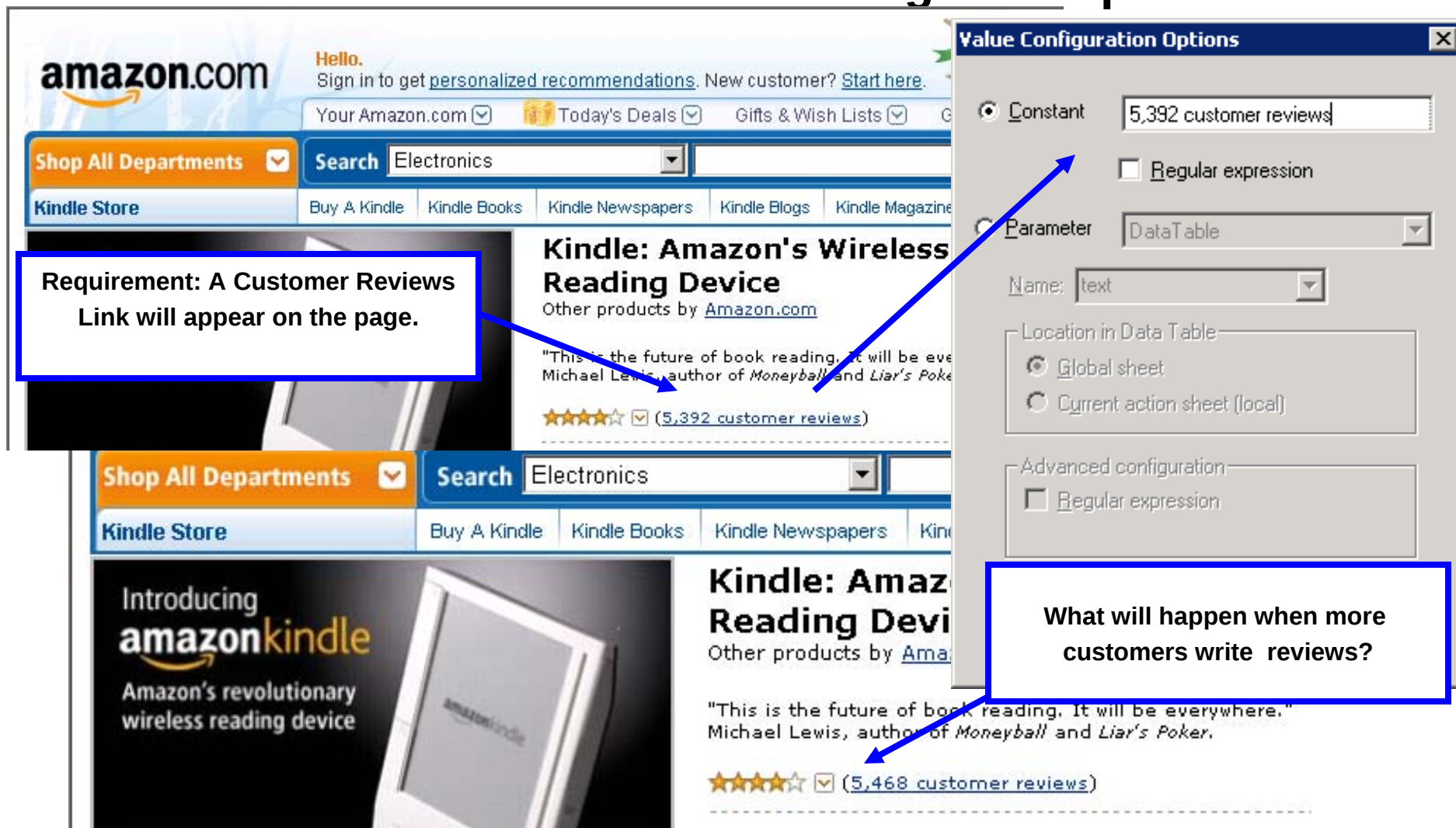
Release 0.2a



Text Property "Return"

Return(Home)?

Regular Expressions



Requirement: A Customer Reviews Link will appear on the page.

Value Configuration Options

- ☒ Constant: 5,392 customer reviews
- ☐ Regular expression
- ☐ Parameter: DataTable
- Name: text
- Location in Data Table:
 - ☒ Global sheet
 - ☐ Current action sheet (local)
- Advanced configuration:
 - ☐ Regular expression

What will happen when more customers write reviews?

Regular Expressions



amazon.com Hello. Sign in to get [personalized recommendations](#). New customer? [Start here](#).

Your Amazon.com Today's Deals Gifts & Wish Lists Gift Cards

Shop All Departments Search Electronics

Kindle Store Buy A Kindle Kindle Books Kindle Newspapers Kindle Blogs Kindle

Introducing **amazonkindle**
Amazon's revolutionary wireless reading device

Kindle: Amazon's Wireless Reading Device
Other products by [Amazon.com](#)

"This is the future of book reading. It's the future of *Moneyball* and..."

★★★★★ (5,392 customer reviews)

Value Configuration Options

☒ Constant

☒ Regular expression

☐ Parameter

Name:

Location in Data Table

☒ Global sheet

☐ Current action sheet (local)

Advanced configuration

☐ Regular expression

OK Cancel Help

Regex: **(\d\d?\d?,?)+ customer reviews**

Matches:

- "10 customer reviews"
- "1234 customer reviews"
- "1,234,567 customer reviews"

Non-Match: 1 customer review

Regular Expressions

- How to read a *Regular Expression*?

Regex: (\d \d? \d? ,?)+ customer reviews

Translation:

"One group..."

()

"or more..."

+

"of at least one Digit..."

\d

"or possibly Two Digits..."

\d?

"but no more than Three Digits..."

\d?

"possibly followed by a comma..."

,?

All followed by " customer reviews"

customer reviews

Matches: 10 customer reviews

1,234 customer reviews

12,345,678 customer reviews

- Pattern Matching

Literal Characters

*Regex: **Cat***

Matches: Cat

Non-Matches: cat, Cut, Dog, Cate

- (Dot) Matches any character, except a new line.

*Regex: **C . t*** **Challenge: Matches and Non-Matches**

Matches: Cat, Cut, C t

Non-Matches: cat, Dog, Ca

Quantifiers – Match the Previous Character

- * 0 or More Times**
- + 1 or More Times**
- ? 0 or 1 Time**

Regular Expressions

* Star - Match Preceding Character Zero or More times

- * **Any character**
Followed by Zero or More of any characters.

Regex: **St.***

Matches: Street, St. Cloud, Sty, St

Non-Matches: sty (lowercase)

Regular Expressions

+ Plus - Match Preceding Character One or More times

2 Characters – Second of which is a Quatifier

s + **Literal lowercase "s"**
Followed by One or More lowercase "s".

Regex: Des+ert

Matches: Desert, Dessert, Desssert

Non-Matches: Deert

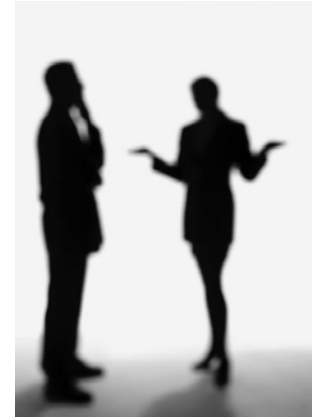
Regular Expressions

? - Match Preceding Character Zero or One time

Regex: **customer reviews?** Literal 'Customer Review'
Possibly followed
by an 's'

Matches: customer review, customer reviews

Non-Matches: customer reviews? (literal question mark)



Grouping and Character Classes

() "The Whole Group"

(|) "This Group or That"

[] "One of a Few"

[^] "Any but These"

Regular Expressions

()? – Match Preceding Group Zero or One time

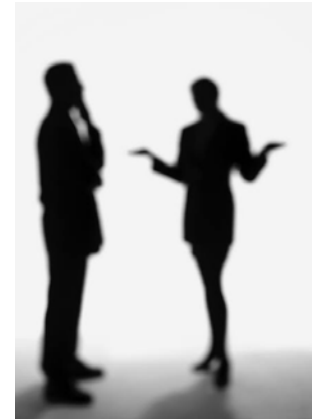
Regex: **Maybe(not)?**

Literal 'Maybe'

Possibly Followed by ' not'

Matches: Maybe, Maybe not

Non-Matches: maybe, Maybe Not, Maybe Not Not



| "Or" Match this group or that group

Regex: **P(oe|un)t** **Literal 'P'**
Followed by 'oe' or 'un'
Followed by a literal 't'

Matches: Poet, Punt

Non-Matches: Put, Poe, Pun, poet, punt

Challenge: Match Poet, poet, Punt and punt

[Pp](oe|un)t

Regular Expressions

[] Character Class - Match one of the group

Regex: **P[eo]t** **Literal 'P'**

Followed by 'e' or 'o'

Followed by a literal 't'

Matches: Pet, Pot

Non-Matches: Put, pet, Poet



Regular Expressions

[^] (Caret) Negative Character Class
Match Any Character Except These

Regex: **P[^eo]t**

Literal 'P'

**Followed by anything
but 'e' or 'o'**

Followed by a literal 't'

Matches: Pit, Pat, P&t,...

Non-Matches: Pet, Pot, Poet



Regular Expressions

- Anchors

- ^ "Must Start with..."

- **^Calc.*** Matches: Calculator
Non-Matches: New Calculator

- \$ "and stops here."

- **.*Explorer\$** Matches: Microsoft Internet Explorer
Non Matches: Microsoft Internet Explorer Beta

Challenge: Create a Regex starting with "Policy" or "policy"

^[Pp]olicy.*

Matches: Policy form, policy form

Non-Matches: Insurance policy, Insurance policy form



Regular Expressions

- Ranges (Dash)

[0-9] Any Number

- .Challenge: Matches 10 through 99
Non-Matches: 0 – 9

[1-9][0-9]

[A-Z] Upper case only

- Challenge: Valid Uppercase Test Letter Grades
[A-DF]

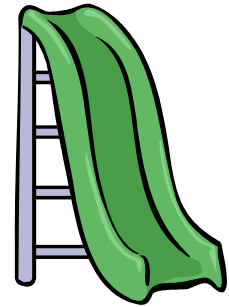
Note: **[a-Z]** Is not Valid, but **[a-zA-Z]** is Valid



What if the string we are trying to match contains RegEx metacharacters?

"Calculator (Beta)"

Regular Expressions



Matching Literal Characters

- "Calculator (Beta)"
- Regex **Calculator (Beta)**

Matches: Calculator Beta Non-Match: Calculator (Beta)

Regex **Calculator \ (Beta \)**

Matches: Calculator (Beta)

- Challenge: Match "St. Cloud"

St \. Cloud

- Matches St. Cloud
- NonMatches: Stu Cloud

Reference Books



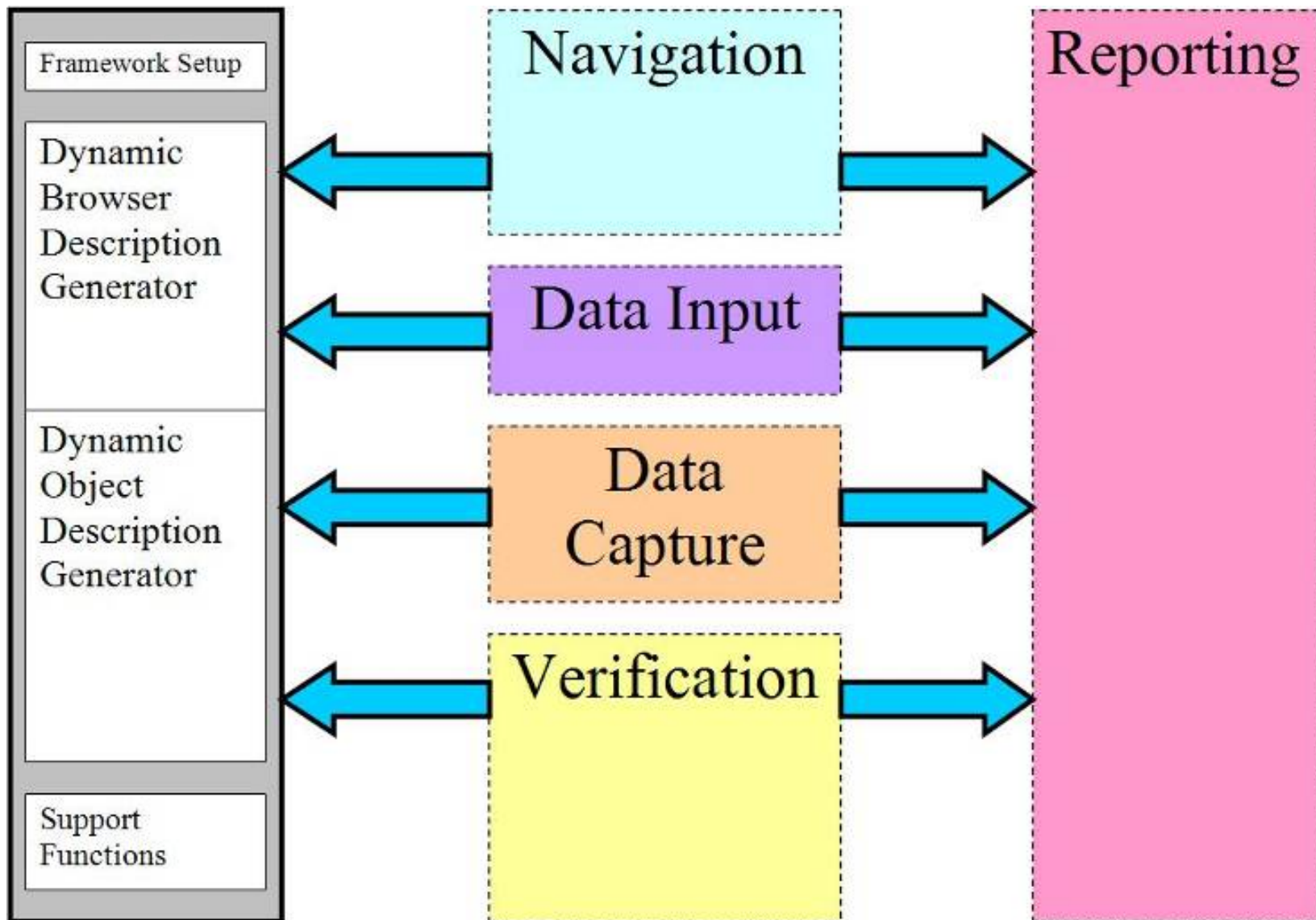
SAFFRON

Sample Automation Framework

With Dynamic Code Execution

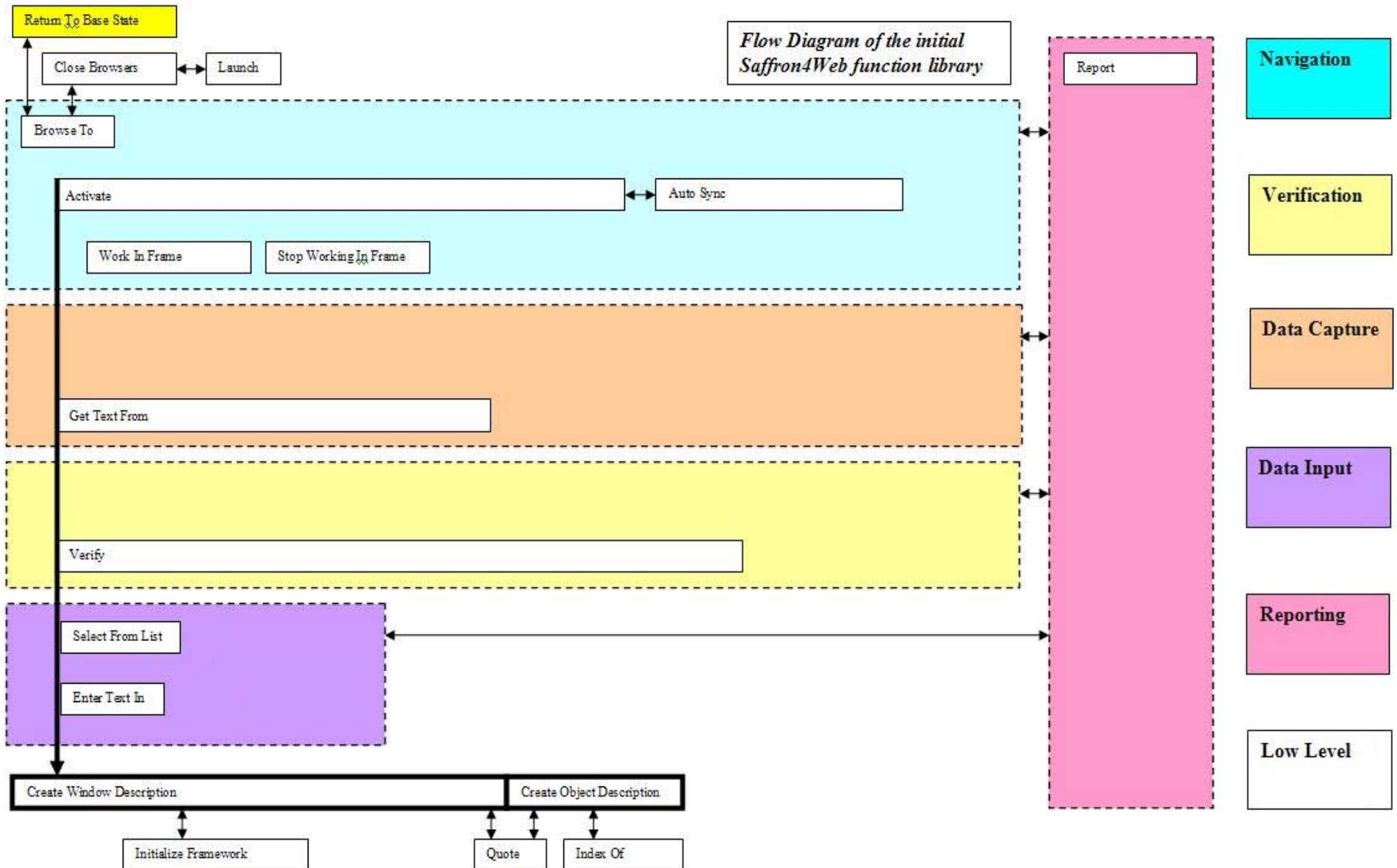
Adam Gensler's
Simple Automation Framework For Remarkably Obvious Notes

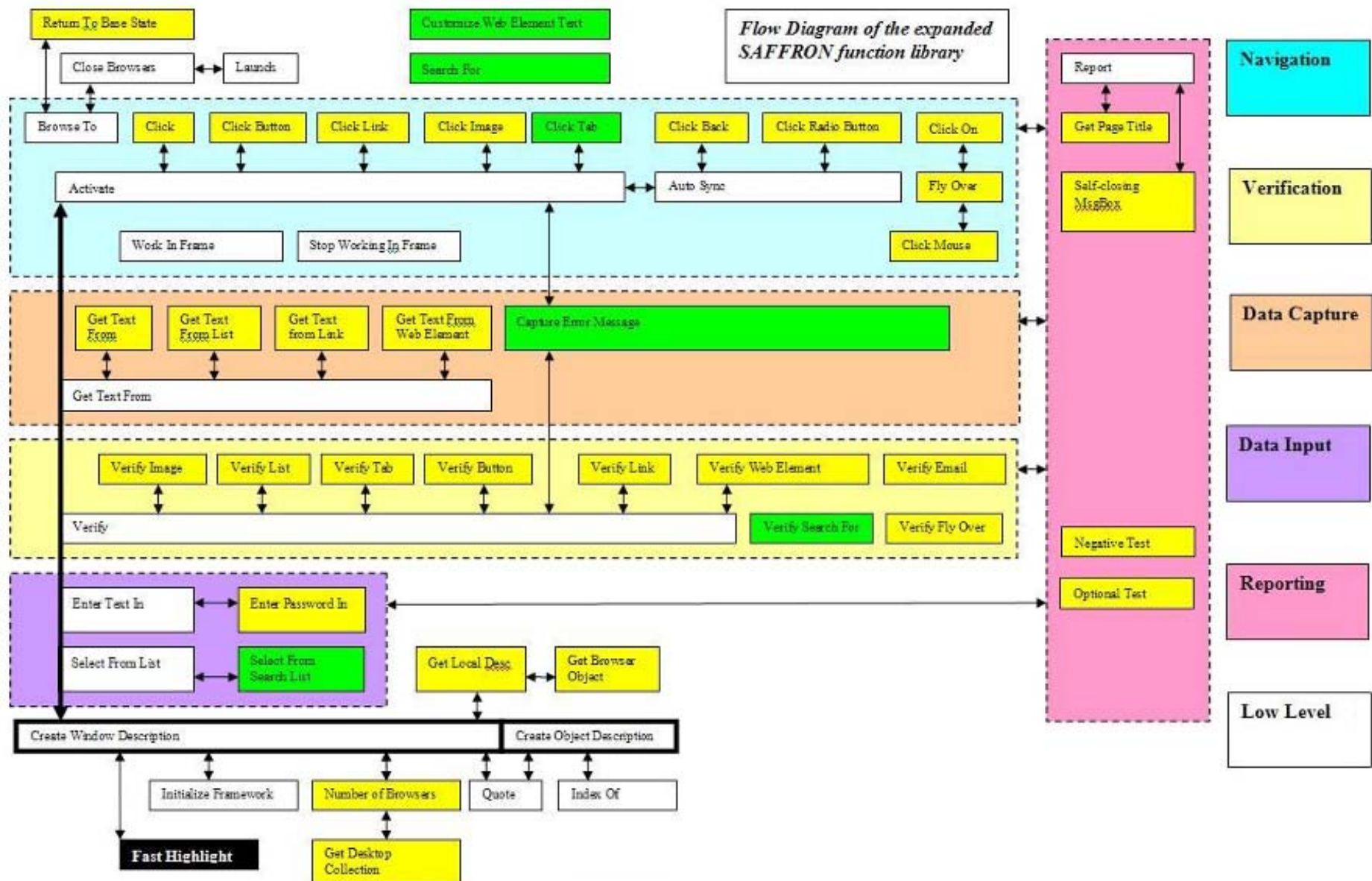
SAFFRON Framework



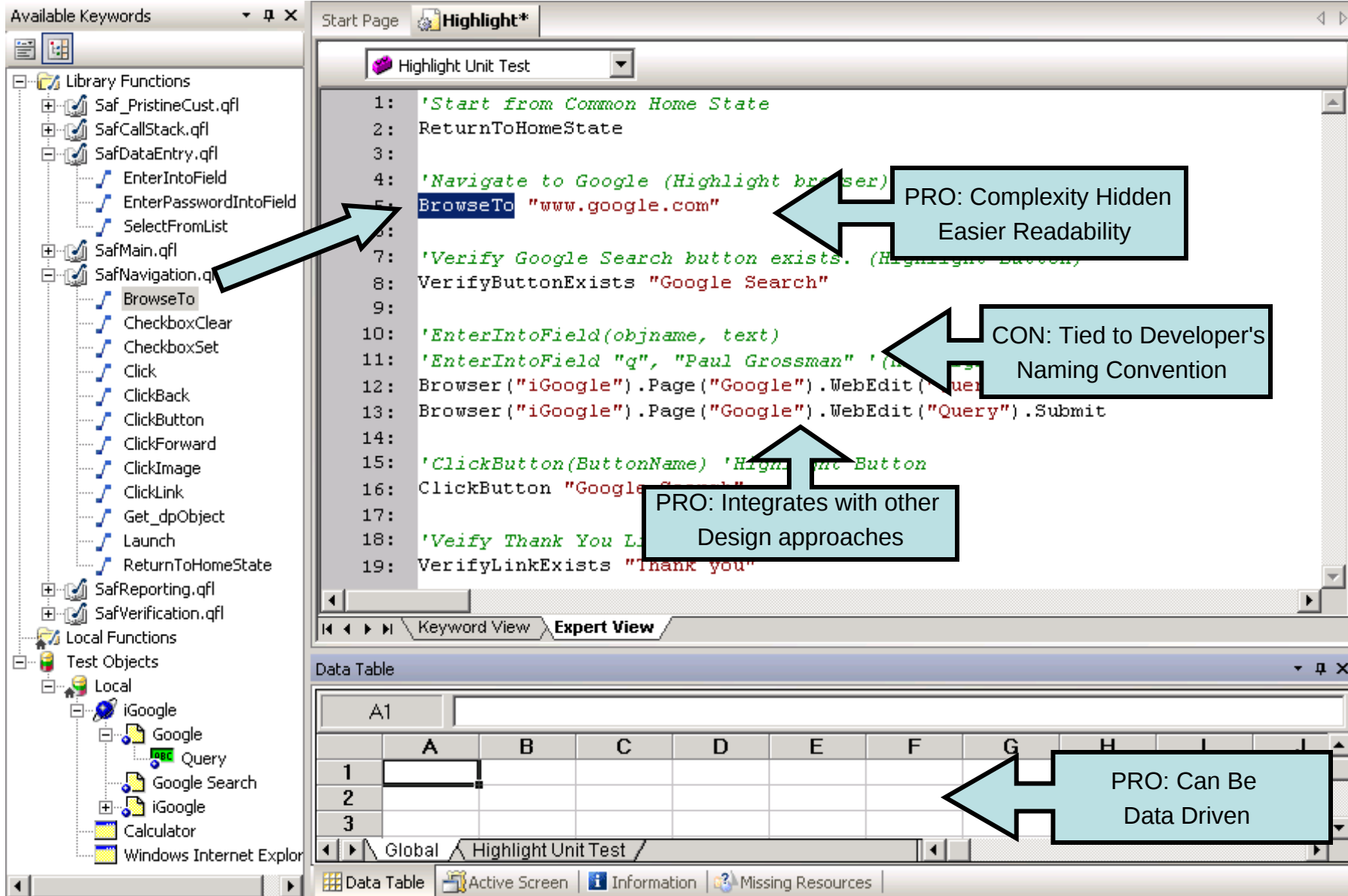
SAFFRON Framework

Flow Diagram of the initial Saffron4Web function library





SAFFRON Based Framework



The screenshot displays the SAFFRON Based Framework interface, which includes a left-hand pane for available keywords and test objects, a central area for writing test scripts, and a bottom section for a data table.

Available Keywords:

- Library Functions
 - Saf_PristineCust.qfl
 - SafCallStack.qfl
 - SafDataEntry.qfl
 - EnterIntoField
 - EnterPasswordIntoField
 - SelectFromList
 - SafMain.qfl
 - SafNavigation.qfl
 - BrowseTo
 - CheckboxClear
 - CheckboxSet
 - Click
 - ClickBack
 - ClickButton
 - ClickForward
 - ClickImage
 - ClickLink
 - Get_dpObject
 - Launch
 - ReturnToHomeState
 - SafReporting.qfl
 - SafVerification.qfl
- Local Functions
- Test Objects
 - Local
 - iGoogle
 - Google
 - Query
 - Google Search
 - iGoogle
 - Calculator
 - Windows Internet Explor

Test Script (Highlight Unit Test):

```

1: 'Start from Common Home State
2: ReturnToHomeState
3:
4: 'Navigate to Google (Highlight browser)
5: BrowseTo "www.google.com"
6:
7: 'Verify Google Search button exists. (Highlight browser)
8: VerifyButtonExists "Google Search"
9:
10: 'EnterIntoField(objname, text)
11: 'EnterIntoField "q", "Paul Grossman" ' (Highlight browser)
12: Browser ("iGoogle").Page ("Google").WebEdit ("Query").Click
13: Browser ("iGoogle").Page ("Google").WebEdit ("Query").Submit
14:
15: 'ClickButton(ButtonName) 'Highlight Button
16: ClickButton "Google Search"
17:
18: 'Verify Thank You Link
19: VerifyLinkExists "Thank you"
  
```

Data Table:

	A	B	C	D	E	F	G	H	I	J
1										
2										
3										

Annotations:

- PRO: Complexity Hidden Easier Readability:** Points to the `BrowseTo "www.google.com"` line in the script.
- CON: Tied to Developer's Naming Convention:** Points to the `'EnterIntoField "q", "Paul Grossman"` line in the script.
- PRO: Integrates with other Design approaches:** Points to the `ClickButton "Google Search"` line in the script.
- PRO: Can Be Data Driven:** Points to the Data Table.

Dynamic Code Execution with Descriptive Programming

- Record & Playback

```
Window("PowerPoint").highlight
```

- Descriptive Programming & Regular Expressions

```
Window("Title:=.*PowerPoint.*").highlight
```

- EXECUTE statement

```
Execute "Window('Title:=.*PowerPoint.*').highlight"
```



Dynamic Code Execution

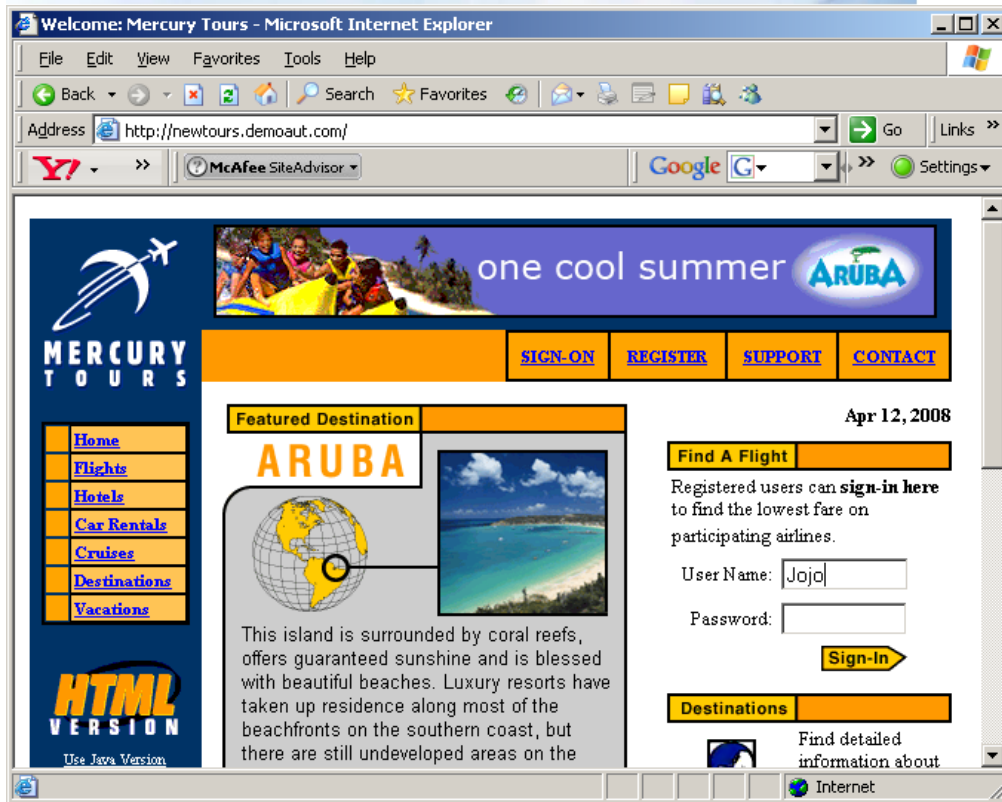
1. Standard QTP Code with Object Repository
2. Convert Page Object
3. Convert Other Objects
4. Convert to Execute

Browser("Mercury Tours").Page("Mercury Tours").WebEdit("userName").Highlight

Browser("Mercury Tours").Page("Index:=0").WebEdit("userName").Highlight

Browser("Title:=Mercury Tours").Page("Index:=0").WebEdit("name:=userName").Highlight

Execute "Browser("Title:=Mercury Tours").Page("Index:=0").WebEdit("name:=userName").Highlight"



Dynamic Code Execution

1. Standard QTP Code with Object Repository
2. Convert Page Object
3. Convert Other Objects
4. Convert to Execute
5. Parameterize Object & Browser Name, Object Class & Method

```
dpBrowser = ("Title:=Mercury Tours")." : sObjectClass = "WebEdit."
dpObject = ("name:=userName")         : sMethod = ".Highlight"
```

Execute "Browser" & dpBrowserName & sObjectClass & dpObjectName & sMethod

Execute "Set oObj = Browser " & dpBrowser & dpClass & dpObject

Execute "IF oObj.Exist THEN oObj" & dpMethod & dpArg



Set oObj = Browser ("Title:=Mercury Tours").WebEdit("Name:=userName")

IF oObj.Exist THEN oObj.Set "Steve"

Execute "Set oObj = Browser " & **dpBrowser** & **dpClass** & **dpObject**

Execute "IF oObj.Exist THEN oObj" & **dpMethod** & **dpArg**



Set oObj = Browser ("**Title:=Technisource**").**WebList**("Name:=SalaryRange")

IF oObj.Exist THEN oObj.**Select** "**50,000**"

Execute "Set oObj = Browser " & dpBrowser & dpClass & dpObject

Execute "IF oObj.Exist THEN oObj" & dpMethod & dpArg



Set oObj = Browser ("Title:=HP").Link("Name:=Return Home")

IF oObj.Exist THEN oObj.Select

Execute "Set oObj = Browser " & dpBrowser & dpClass & dpObject

Execute "IF oObj.Exist THEN oObj" & dpMethod & dpArg



Set oObj = Browser ("Title:=HP").Link("Name:=Return Home")

IF oObj.Exist THEN oObj.Click

Execute Command

- Pro/Con: Uses Dynamic Code Execution
- Con: Errors are "Quiet"
- Con: Sample Framework has no Object Verification

```
Select Case objtype
Case "Link"
    Execute localDesc & GenerateObjectDescription("Link","innertext:=" & text) & "Click"
    Report micPass, "Link Activation", "The Link " & Quote(text) & " was clicked."
Case "WebButton"
    Execute localDesc & GenerateObjectDescription("WebButton", "value:=" & text) & "Click"
    Report micPass, "WebButton Activation", "The WebButton " & Quote(text) & " was clicked."
End Select
End Function
```

DEMO

Framework Based Error Recovery

Tools of the Trade

Single Source Documentation
with Test Design Studio

- Single Source Documentation
 - Is the automation project code documented?
 - Is it documented well enough?
 - Is there a Help File that documents the project?
 - Is it up to date with the current project code?

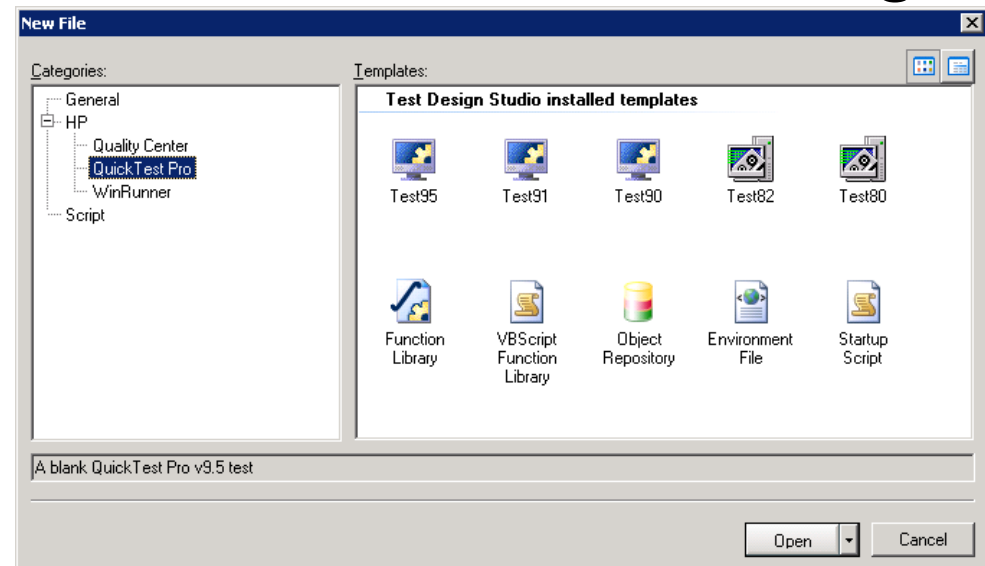
Tools of the Trade

- Detailed Documenter
 - Automatic XML Header Generation
 - Integration with MS Help Compiler
 - Centralized Documentation

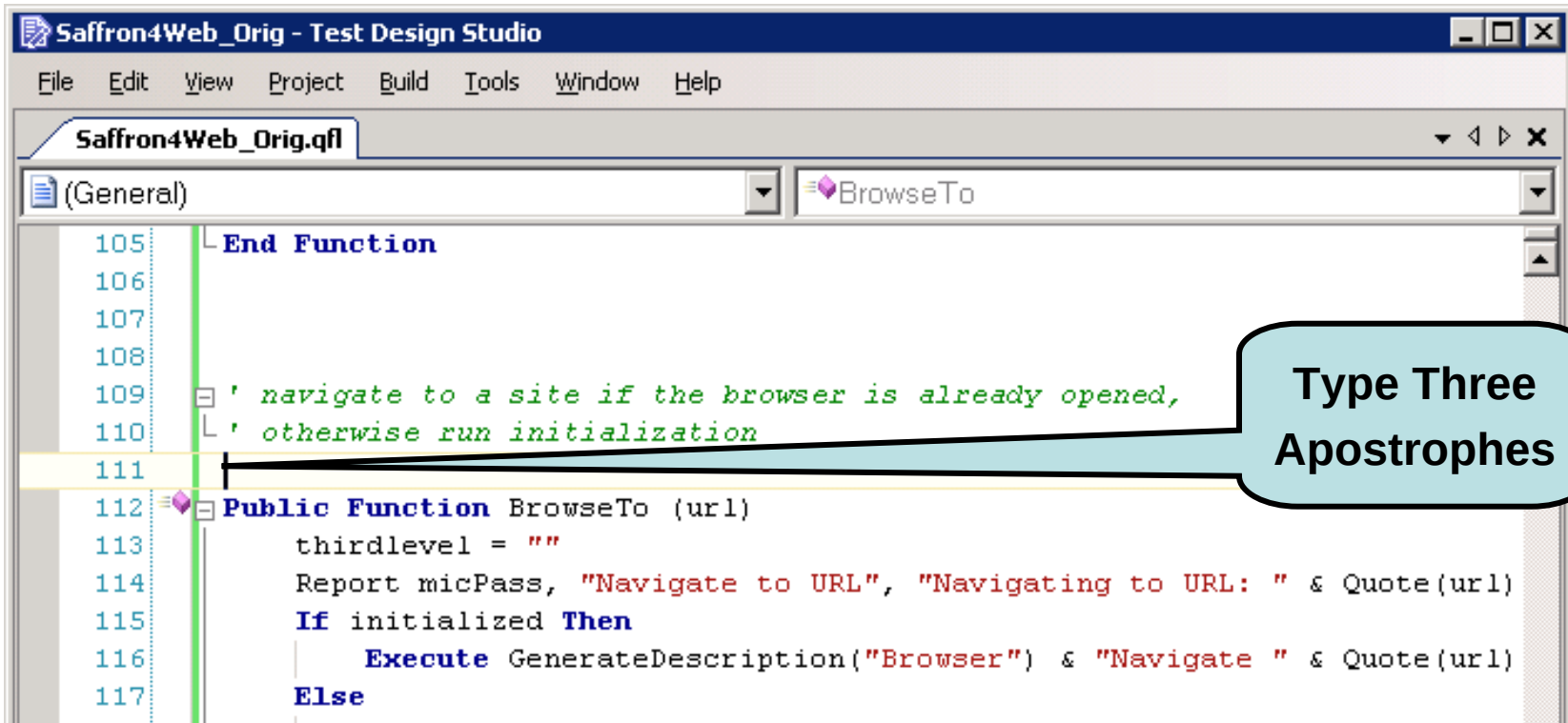


www.Patterson-Consulting.net

- Templates
 - QTP Startup Scripts
 - Quality Center Workflows
 - WinRunner
 - VBScript, JScript, WSH
 - XML, PHP, Perl,
 - 12 other files types



Tools of the Trade: TDS



Saffron4Web_Orig - Test Design Studio

File Edit View Project Build Tools Window Help

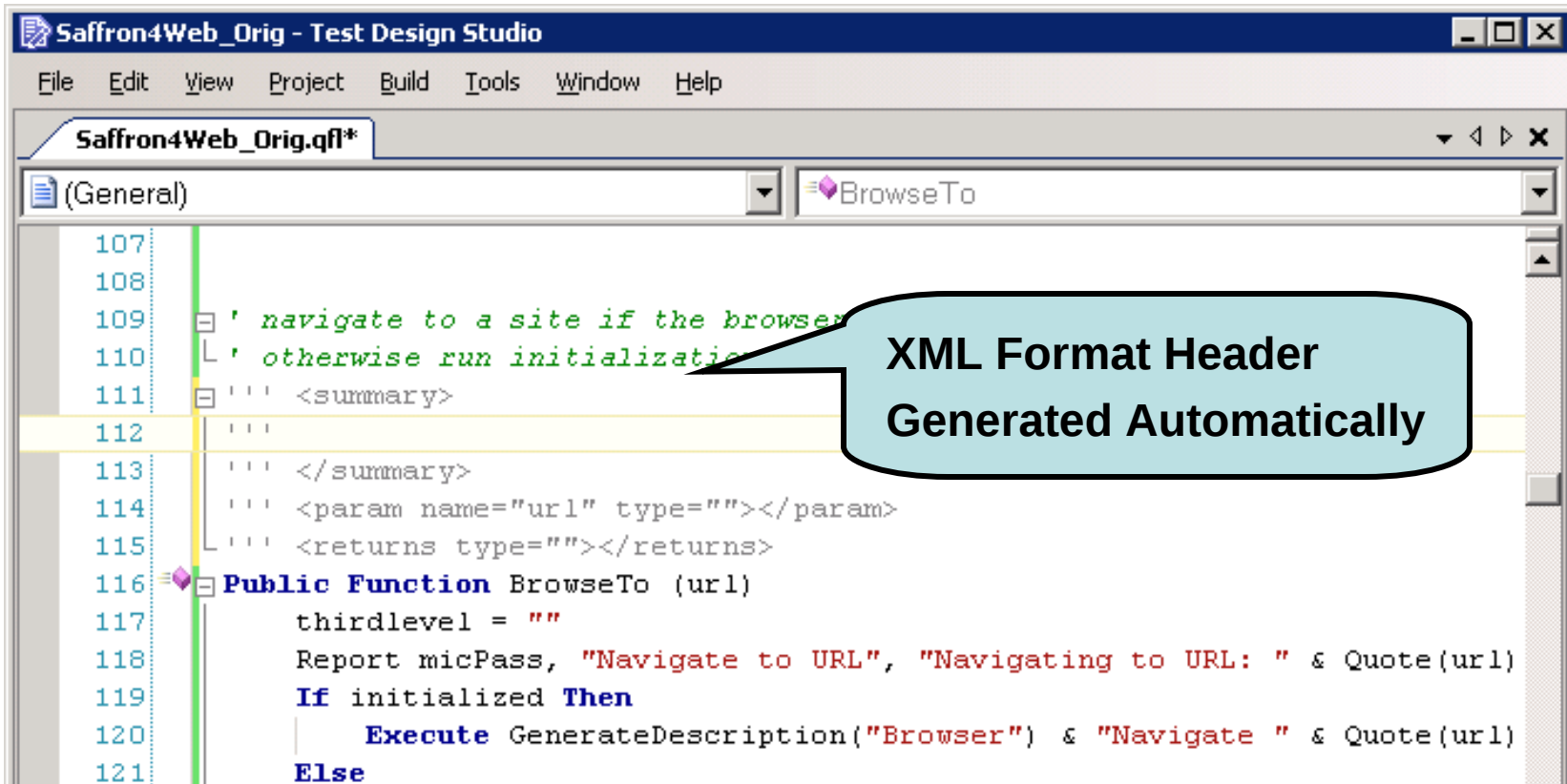
Saffron4Web_Orig.qfl

(General) BrowseTo

```
105 End Function
106
107
108
109 ' navigate to a site if the browser is already opened,
110 ' otherwise run initialization
111
112 Public Function BrowseTo (url)
113     thirdlevel = ""
114     Report micPass, "Navigate to URL", "Navigating to URL: " & Quote(url)
115     If initialized Then
116         Execute GenerateDescription("Browser") & "Navigate " & Quote(url)
117     Else
```

Type Three
Apostrophes

Tools of the Trade: TDS



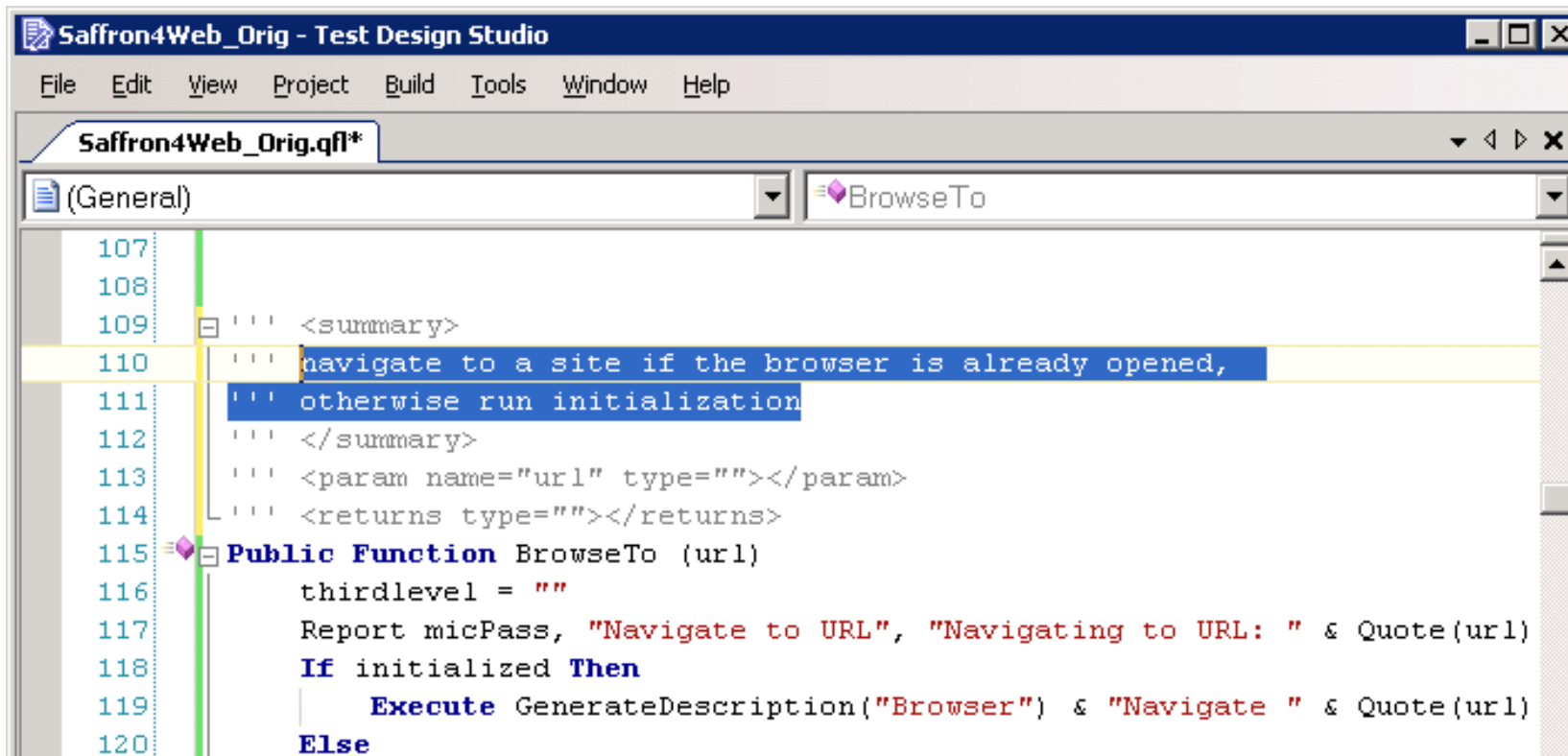
The screenshot shows the Test Design Studio (TDS) interface. The title bar reads "Saffron4Web_Orig - Test Design Studio". The menu bar includes File, Edit, View, Project, Build, Tools, Window, and Help. The file name "Saffron4Web_Orig.qfl*" is displayed in the top bar. The main editor shows the code for the "BrowseTo" function. The code is in XML format, with a callout bubble pointing to the header section.

```
107
108
109  ' navigate to a site if the browser
110  ' otherwise run initialization
111  ' <summary>
112  '
113  ' </summary>
114  ' <param name="url" type=""></param>
115  ' <returns type=""></returns>
116  Public Function BrowseTo (url)
117      thirdlevel = ""
118      Report micPass, "Navigate to URL", "Navigating to URL: " & Quote(url)
119      If initialized Then
120          Execute GenerateDescription("Browser") & "Navigate " & Quote(url)
121      Else
```

XML Format Header
Generated Automatically

- Automatic Header Generation in XML

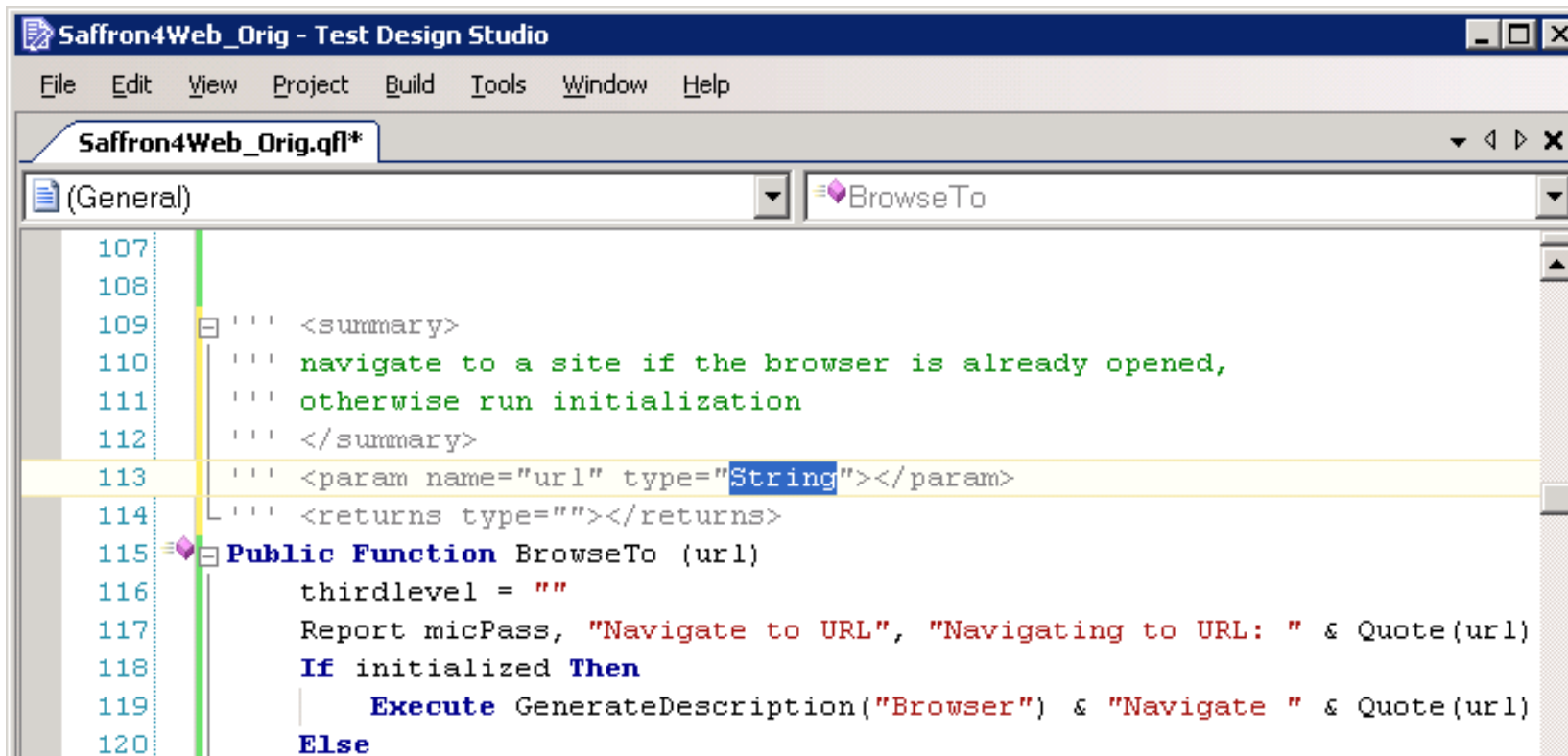
Tools of the Trade: TDS



```
107
108
109  ''' <summary>
110  ''' navigate to a site if the browser is already opened,
111  ''' otherwise run initialization
112  ''' </summary>
113  ''' <param name="url" type=""></param>
114  ''' <returns type=""></returns>
115  Public Function BrowseTo (url)
116      thirdlevel = ""
117      Report micPass, "Navigate to URL", "Navigating to URL: " & Quote(url)
118      If initialized Then
119          Execute GenerateDescription("Browser") & "Navigate " & Quote(url)
120      Else
```

- Automatic Header Generation in XML

Tools of the Trade: TDS



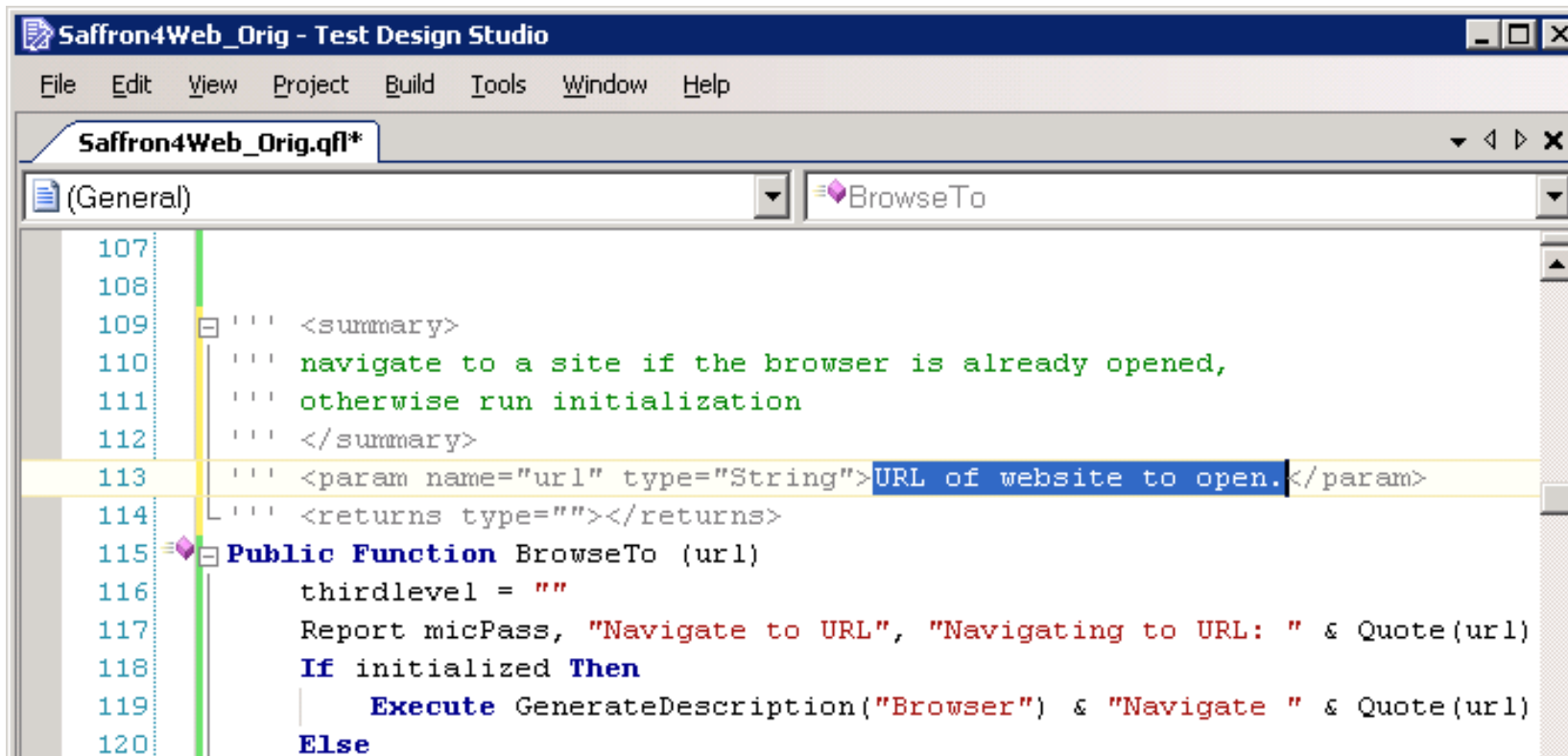
```
Saffron4Web_Orig - Test Design Studio
File Edit View Project Build Tools Window Help

Saffron4Web_Orig.qfl*
(General) BrowseTo

107
108
109 ''' <summary>
110 ''' navigate to a site if the browser is already opened,
111 ''' otherwise run initialization
112 ''' </summary>
113 ''' <param name="url" type="String"></param>
114 ''' <returns type=""></returns>
115 Public Function BrowseTo (url)
116     thirdlevel = ""
117     Report micPass, "Navigate to URL", "Navigating to URL: " & Quote(url)
118     If initialized Then
119         Execute GenerateDescription("Browser") & "Navigate " & Quote(url)
120     Else
```

- Automatic Header Generation in XML

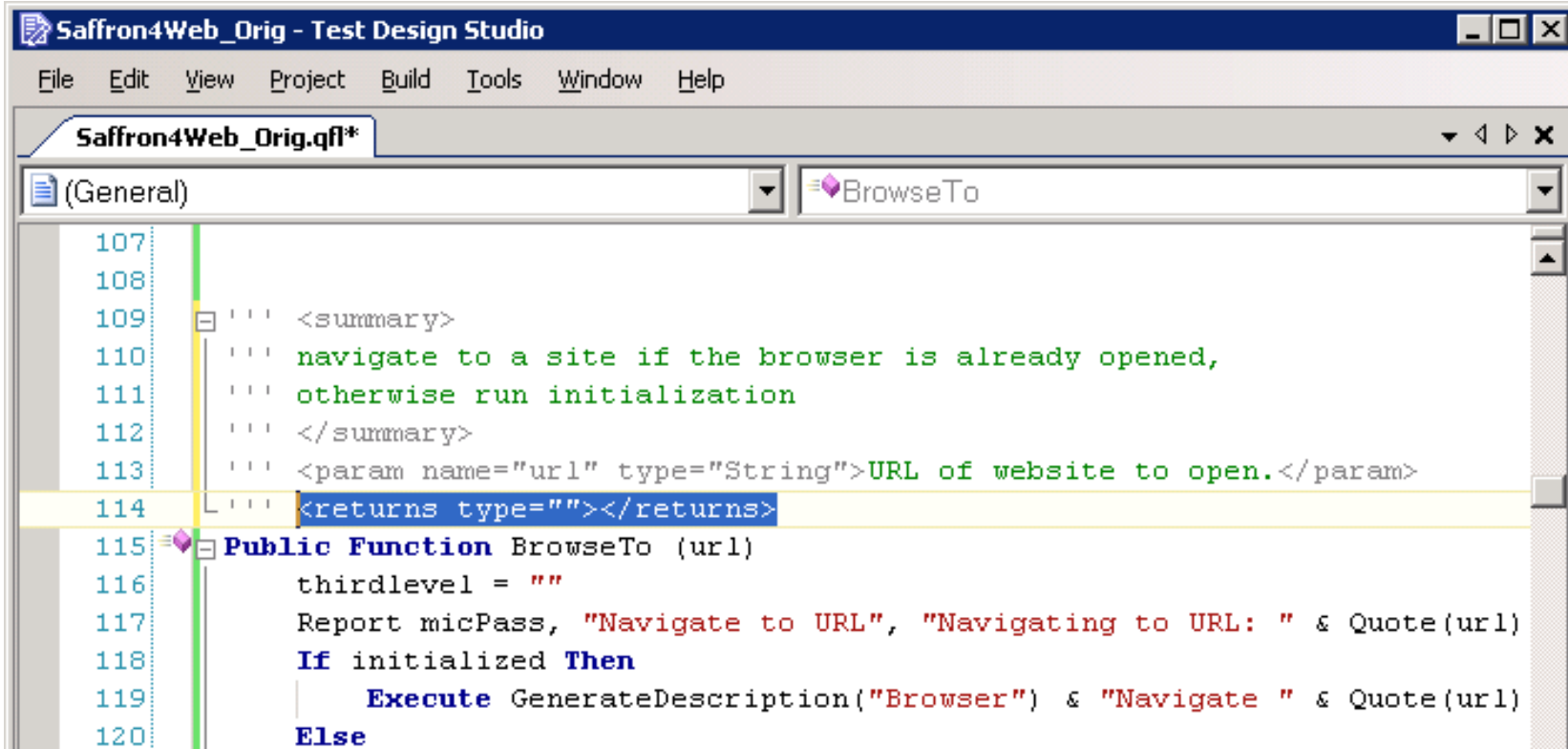
Tools of the Trade: TDS



The screenshot shows the Test Design Studio (TDS) interface for a test case named 'Saffron4Web_Orig.qfl*'. The 'General' tab is selected, and the 'BrowseTo' function is highlighted. The function is defined as a public function with a parameter 'url' of type 'String'. The function body includes a summary, a parameter description, and a series of steps to navigate to a URL.

```
107
108
109  ''' <summary>
110  ''' navigate to a site if the browser is already opened,
111  ''' otherwise run initialization
112  ''' </summary>
113  ''' <param name="url" type="String">URL of website to open.</param>
114  ''' <returns type=""></returns>
115  Public Function BrowseTo (url)
116      thirdlevel = ""
117      Report micPass, "Navigate to URL", "Navigating to URL: " & Quote(url)
118      If initialized Then
119          Execute GenerateDescription("Browser") & "Navigate " & Quote(url)
120      Else
```

- Automatic Header Generation in XML



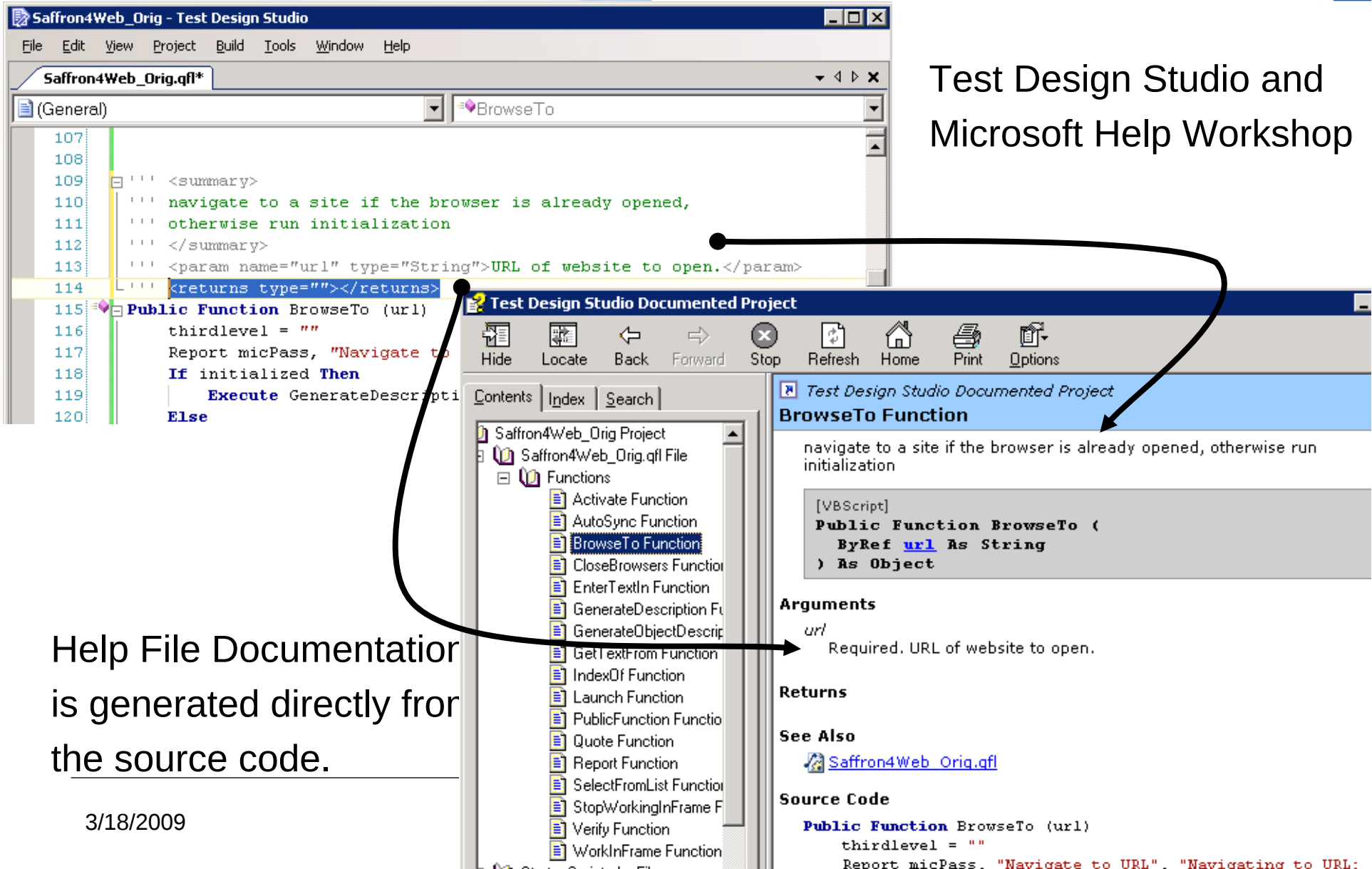
The screenshot shows the Test Design Studio interface for a project named 'Saffron4Web_Orig'. The main window displays the 'Saffron4Web_Orig.qfl*' file. The left pane shows the '(General)' tab. The right pane shows the 'BrowseTo' function. The code is as follows:

```
107  
108  
109     ''' <summary>  
110     ''' navigate to a site if the browser is already opened,  
111     ''' otherwise run initialization  
112     ''' </summary>  
113     ''' <param name="url" type="String">URL of website to open.</param>  
114     ''' <returns type=""></returns>  
115     Public Function BrowseTo (url)  
116         thirdlevel = ""  
117         Report micPass, "Navigate to URL", "Navigating to URL: " & Quote(url)  
118         If initialized Then  
119             Execute GenerateDescription("Browser") & "Navigate " & Quote(url)  
120         Else
```

- Automatic Header Generation in XML

Tools of the Trade: TDS

Test Design Studio and
Microsoft Help Workshop



The screenshot displays the Test Design Studio interface with the 'Saffron4Web_Orig.qfl' file open. The code editor shows a summary and a 'BrowseTo' function. A black arrow points from the function definition in the code to the 'BrowseTo Function' entry in the 'Test Design Studio Documented Project' pane. Another black arrow points from the 'BrowseTo Function' entry to the 'BrowseTo Function' help topic in the Microsoft Help Workshop pane. The help topic displays the function's summary, arguments, returns, and source code.

Help File Documentation
is generated directly from
the source code.

3/18/2009

Saffron4Web_Orig - Test Design Studio

File Edit View Project Build Tools Window Help

Saffron4Web_Orig.qfl*

(General) BrowseTo

```
107
108
109  <summary>
110    navigate to a site if the browser is already opened,
111    otherwise run initialization
112  </summary>
113  <param name="url" type="String">URL of website to open.</param>
114  <returns type=""></returns>
115  Public Function BrowseTo (url)
116    thirdlevel = ""
117    Report micPass, "Navigate to
118    If initialized Then
119      Execute GenerateDescripti
120    Else
```

Test Design Studio Documented Project

Hide Locate Back Forward Stop Refresh Home Print Options

Contents Index Search

- Saffron4Web_Orig Project
 - Saffron4Web_Orig.qfl File
 - Functions
 - Activate Function
 - AutoSync Function
 - BrowseTo Function
 - CloseBrowsers Function
 - EnterTextIn Function
 - GenerateDescription F
 - GenerateObjectDescrip
 - GetTextFrom Function
 - IndexOf Function
 - Launch Function
 - PublicFunction Functio
 - Quote Function
 - Report Function
 - SelectFromList Function
 - StopWorkingInFrame F
 - Verify Function
 - WorkInFrame Function

Test Design Studio Documented Project

BrowseTo Function

navigate to a site if the browser is already opened, otherwise run initialization

[VBScript]

```
Public Function BrowseTo (  
  ByRef url As String  
) As Object
```

Arguments

url
Required. URL of website to open.

Returns

See Also

[Saffron4Web_Orig.qfl](#)

Source Code

```
Public Function BrowseTo (url)  
  thirdlevel = ""  
  Report micPass. "Navigate to URL". "Navigating to URL:"
```

Three Online Resources

Top 5 Web Resources

WilsonMar.com



Test Automation Frameworks

This page compares and describes "data driven" and "keyword" driven automated tests using open source libraries and WinRunner from Mercury Interactive — the hardest won (but easily forgotten) insights not obvious from available documentation.

 "Hey, we've got a problem here."

Topics this page:

- ▼ Why Frameworks?
- ▼ Driver Frameworks
- ▼ Infrastructures
- ▼ Driver Keywords
- ▼ Summary
- ▼ Flow Diagram
- ▼ SAFS/STAF2 Install
- ▼ WRSafs Install
- ▼ WRSafs Customization
- ▼ Keywords & Functions
- ▼ Fittesse/Abbot
- ▼ STAF3 Install
- ▼ Related Topics
- ▼ Your comments???

Site Map  About this site   

☐ Web ☒ wilsonmar.com



Top 5 Web Resources

SoftwareInquisition.com



June 29th, 2008

WebExt Performance Demo

Will Roden | [No Comments »](#)

One of the benefits of using Web Extensibility Add-ins is that you get a performance gain in running tests vs using the old WebElement method. For a demonstration, I wrote a basic WebExt add-in that only has one class defined: Div. Div is any object whose tagName is "DIV", so it is the equivalent of

Top 5 Web Resources

AdvancedQTP.com

AdvancedQTP Forum

[RETURN TO ADVANCEDQTP](#) [HOME](#) [SEARCH](#) [PROFILE](#) [MY MESSAGES \[1\]](#) [LOGOUT](#)

Hello Paul Grossman

you have [63 messages](#) [1 is new](#).

[Show unread posts since last visit.](#)

[Show new replies to your posts.](#)

[Advanced search](#)

7201 Posts in 1770 Topics- by 10975 Members - Latest Member: [Snafu64](#)

you are here: AdvancedQTP Forum

- AdvancedQTP Forums			
	General QTP Q&A QTP Problem Solving Community <i>Moderators: Paul Grossman, Stefan Thelenius, Meir Bar-Tal</i>	4253 Posts 1084 Topics	Last post by Meir Bar-Tal in Re: Getting Save-Cancel ... on Today at 06:49:15 AM
	Web Q&A Here you can get answers to all QTP web related issues <i>Moderators: Anshoo Arora, Stefan Thelenius</i>	1159 Posts 258 Topics	Last post by Kyatham in Re: QTP takes more time ... on Today at 04:38:37 AM

Conclusion

- **Successful Frameworks**
 - Upfront Planning
 - Object Verification
 - Modular Design
 - Robust Execution
 - Use Multiple Tools



Questions?

Paul M. Grossman
Senior Staff Consultant

paulgrossman@technisource.com



References

The following service marks & registered marks are used in this presentation:

- **Capability Maturity Model®**
- **CMM®**
- **CMM IntegrationSM**
- **CMMI®**
- **CMMI, CMM, & Capability Maturity Model are registered in the U.S. Patent & Trademark Office.**
- **CMM Integration is a service mark of Carnegie Mellon University.**
- **eVTPSM is a service mark of Spherion.**