

## Pushing the envelope with QuickTest Pro



Rob Anderson / Suresh Radhakrishnan  
SPAN Technologies  
www.spantechnologies.com  
973.287.8400



## Agenda

Objective: Innovation in Automation	
QTP: Scripts with a Vision	
Pushing the Envelope	
Q&A	



Copyright: SPAN Technologies 2006.

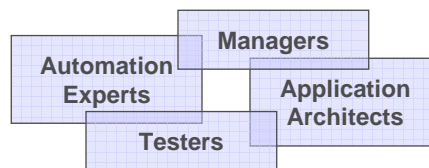


## Objective: Innovation in Automation

To promote innovative thinking in the use of Mercury tools

- Think outside the box and develop creative solutions
- Extend the capabilities of the tools
- Increase your ROI

Demographics



Copyright: SPAN Technologies 2006.



## QTP: Scripts with a Vision

### The Challenge

- Java based financial application for processing trades
- Based on third-party configurable software
- Customer uses the components provided by the vendor to assemble and configure various financial products
- Customer controls the statistical financial models underlying the products that are used to price the trades in the application
- When new versions of the application are created, customer wants to replicate existing trades from production in QA and verify that the trades are priced similarly
- They needed:
  - Automated approach to replicating production trades in QA and comparing trade prices
  - A framework that is applicable for future products that they may assemble, ones that do not exist yet



Copyright: SPAN Technologies 2006.



## QTP: Scripts with a Vision

### The Challenge

*Automated approach to replicate production trades in QA and comparing trade prices*

They have 70 different product types. We will need to create QTP scripts that will go into their production application, read data from existing trades that belong to any one of the 70 product types and recreate the trades in QA.

*A framework that is applicable for future products that they may assemble, ones that do not exist yet*

They plan on adding 5 – 10 new products each quarter. These products will be assembled from the same components provided by the vendor. They want the scripts to work for the existing 70 products and also any future products.



Copyright: SPAN Technologies 2006.



## QTP: Scripts with a Vision

### Solution

**Traditional approach would have resulted in:**

- One script for each product type  
(70 in total \* ~3 product subtypes = ~210 scripts)
- These scripts would be specific to each product, and would not work for future products that are unknown

**Our Approach:**

- Build automation for the components that make up these products
- Create a master script that discovers the components on the fly, and applies the appropriate handling events for each component
- This approach would work for the current products as well as all future products assembled from the same set of components



Copyright: SPAN Technologies 2006.



## QTP: Scripts with a Vision

### Solution

#### Traditional approach

- **70 Products \* ~3 Subtypes = ~210 Scripts**
  - (Each script ~500 lines of code)

#### Our Approach:

- **1 Script (~2000 lines of code)**
  - 70 Products with ~2000 lines of code.



Copyright: SPAN Technologies 2006.



## QTP: Scripts with a Vision

### Solution

- **Why parameterization would not work:**
  - Trade tables have different numbers of fields and different field types.
  - Each product has several different subtypes and differences would enumerate.

Product Type 1		Product Type 2	
Subtype	General	Comment	
Comment		Currency	USD
Currency	USD	Notional	100,000,000.00
Notional	100,000,000.00	Exercise Notice Date	12 Jun 2006
Effective Date	14 Jun 2006 Spot	Effective Date	14 Jun 2006
Maturity		Maturity	
Cancelable	YES	Option Position	We are Long Option
Call start delay		Opt Hold P/R	Pay Fixed
Exercise start date		Fixed Rate (%)	0.00000
Option Position	We are Long Option to Cancel	Model	Do LYOL
Opt Holder P/R	opt holder pays leg1	MF Model	MF Bermudan
Model Library	Optiko	Variance %	80.00
Model	Optiko BDT	Fixed Periods	S MFOL
Vol Model	IRMA	Fixed Use Daycount	Yes



Copyright: SPAN Technologies 2006.



## QTP: Scripts with a Vision

### Solution

Component	Type
Params Window	Popup
FD Fixed Rate (%)	Edit
Leg2 Is RangeAccrual	List
Floor Strike %	Edit
Auto Chooser	Popup;List
Digit Payout %	Edit
Include Strike	Static
.....	.....
.....	.....

1. A component dictionary was created that contains all the components that are used to assemble each financial product. The dictionary provided the component type information, which is used by the script to determine the sequence of events to perform on a component.
2. A Framework of function libraries to handle the components based on its type.

The script walks through the application, and for each object encountered on the screen, refers to the component dictionary to determine the actions to be performed on that object and calls the appropriate framework function.



Copyright: SPAN Technologies 2006.



## QTP: Scripts with a Vision

### Solution

The scripts operate in one of two modes

1. Extract Mode: To extract data from existing trades in production
2. Replicate Mode: Replicate these trades in QA environment and compare the pricing to production pricing

Depending on the mode, the scripts dynamically decide on the steps to be executed based on the information available in the component dictionary. This approach also helped in handling dynamic fields within a product, since each product in itself can have hundreds of variations due to different combination of fields.

Any new products introduced in future will be handled by the scripts automatically since the universal set of components is already included in the component dictionary.



Copyright: SPAN Technologies 2006.



## QTP: Scripts with a Vision

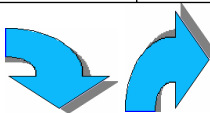
### Solution

#### Production

Instrument	AUD/USD
Payout type	Standard
Payout currency	USD
Exotic style	No touch
Buy/Sell	Sell
Nominal	280,600 USD
Option expiration	19 Jun 2006
Cutoff	
Delivery	19 Jun 2006
Payment time	
Barrier	0.6746
Barrier two	0.8446
Barrier quot	AUD-USD
Barrier style	American
Window start date	19 Jun 2006
Window end date	19 Jun 2006
Premium	NaN

**Extract:** Data is extracted from production and stored in an Excel file.

Component	Type
Payout currency	List
Delivery	Edit
Nominal	Edit;List



	A	B	C
1	Field	FieldValue1	FieldValue2
2	Instrument	AUD/USD	
3	Payout type	Standard	
4	Payout currency	USD	
5	Exotic style	No touch	
6	Buy/Sell	Sell	
7	Nominal	280,600	USD
8	Option expiration	29 May 2006	
9	Cutoff		
10	Delivery	06 Jun 2006	
11	Payment time	Payment time	
12	Barrier	0.6746	
13	Barrier two	0.8446	
14	Barrier quot	AUD-USD	
15	Barrier style	American	
16	Window start date	31 May 2005	
17	Window end date	29 May 2006	
18	Premium	99.91%	

#### QA

Instrument	AUD/USD
Payout type	Standard
Payout currency	USD
Exotic style	No touch
Buy/Sell	Sell
Nominal	280,600 USD
Option expiration	19 Jun 2006
Cutoff	
Delivery	19 Jun 2006
Payment time	
Barrier	0.6746
Barrier two	0.8446
Barrier quot	AUD-USD
Barrier style	American
Window start date	19 Jun 2006
Window end date	19 Jun 2006
Premium	NaN

**Replicate:** The trade is replicated / re-generated in the QA version of the application.



Copyright: SPAN Technologies 2006.



## QTP: Scripts with a Vision

### Solution

- Sample Code for showing different actions that are performed on objects within the application.

Select Case field\_type

Case "popup"

```
JavaWindow("App").JavaTable("Table").SelectCell i,j+1
JavaWindow("App").JavaTable("Table").ActivateCell i,j+1
ReplicatePopupWindowInfo TradeID, FieldName, objMainValue1Dict
```

Case "edit", "list"

```
JavaWindow("App").JavaTable("Table").SelectCell i,j+1
JavaWindow("App").JavaTable("Table").SetCellData i, j+1, sFieldValue
JavaWindow("App").JavaTable("Table").SelectCell i,j+1
```

Case "poplist"

```
JavaWindow("App").JavaTable("Table").SelectCell i,j+1
JavaWindow("App").JavaTable("Table").Type " "
iPopupTable.ActivateCell split(cellLoc, ";")(0), split(cellLoc, ";")(1)
```



Copyright: SPAN Technologies 2006.



## QTP: Scripts with a Vision

### The Outcome

The scripts were able to:

1. Successfully handle all existing products in the application
2. Handle new products (previously unseen) automatically based on the component dictionary. No additional scripting needed!

When new products were developed using net new components, all we had to do was add these components to the dictionary and object repository if necessary. Since the framework was already in place, these new products were handled automatically.



Copyright: SPAN Technologies 2006.



## Pushing the Envelope

### Harness the power of Mercury

Mercury Tools are very powerful and flexible, so always ask if there is room to build a better mousetrap.

Examples:

- Building frameworks using function libraries.
- Using Data Dictionaries to make the script extendable.
- Using Excel Sheets for input, processing, and output to the script.



Copyright: SPAN Technologies 2006.



## Do you need to push the Envelope?

Don't look for RESOURCES  
Look for SOLUTIONS !!

For creative solutions that meet your needs, contact  
SPAN Technologies.  
973.287.8400

[www.spantechnologies.com](http://www.spantechnologies.com)



Copyright: SPAN Technologies 2006.



## Q&A

For a copy of this presentation,  
visit [www.spantechnologies.com](http://www.spantechnologies.com)

