PRESENTATION

T6

Thursday, March 8, 2001
11:30 AM

DID YOUR TESTS PASS OR FAIL?
ANSWERING WITH AUTOMATION

Noel Nyman
Microsoft

International Conference On
Software Test Automation
March 5-8, 2001
San Jose, CA, USA

Did Your Tests Pass or Fail?

Using Self-Verifying Data with Hard-to-Automate Applications

Noel Nyman

Desktop Applications Automation Test Lead

Microsoft Windows Systems Group

Agenda

- Self-Verifying Data review
  - ✓ How to use SVD to tell that your tests pass
- Automating problem applications
  - ✓ Ideas on how to automate apps when your automation tools can't see parts of them
- Demos
- Resources for more information
- Questions

Self-Verifying Data

- Has codes embedded in the data that act as an oracle and can tell you if the data is…
  - ✓ **Legal** - is this data from the correct data set?
  - ✓ **Valid** - is this data that should be here?
  - ✓ **Correct type** – is this data the type we're looking

  - ✓ **From correct record** – data from the record we asked for?
  - ✓ **Accurate** - are characters missing, munged?
- ▪ SVD data is like a debug build of an app

Benefits of Using SVD

- No separate oracle needed
- Can be scaled to very large data sets
- Can be used with rich data not easily verified by humans
- Easy to verify with automated testing
- Adding additional test data usually doesn't require updating automated tests

SVD Example

First name from a rich data set…

First name: Ñr£¼üæxµ2ŝ¿

First name with embedded SVD codes…

Ñr£¼üKKQqkkqkq12345003277xxxæxµ2ŝ¿

SVD codes replaced with evocative tags…

Ñr£¼ü<Legal><Valid><Type><Record><CRC><Delim>æxµ2ŝ¿

Automating Problem Apps

- Forget capture/replay – if it works well for you, you don't have a "problem app"
- Wrap EVERYTHING!
- Dealing with Custom Controls
  - ✓ Get Management to require that all controls must be visible to your automation tool
  - ✓ Devs can add Windows messaging support to their own custom controls, but your tools must support Windows API calls

# Leverage Automation Strengths

-

  - ✓
  - ✓

- Don't lose bugs trying to force your automation tool to test everything (John Daly rule)
  - ✓ Use manual testing for some complete GUI test passes
- For browser hosted apps, automate only the browsers your tool supports
  - ✓ Sanity test all other supported browsers manually
 - ✓ Don't over test any browser…unless your group developed it

Custom Control Alternatives

-

-
    - ✓ Toolbar functions may be available on menus
- Accessibility features
    - ✓ "Easy to use" means easy to test
- App's macros, scripting
- Use Windows Clipboard to read data
- Backdoor to underlying data...ODBC, DAO

- Specific automation tool features
    - ✓ Visual Test's OCX control, OLE automation
- Intelligent x/y clicking
    - ✓ Reference x/y coordinates to the smallest possible window
    - ✓ Use percentages instead of absolutes (Cursor Locator tool, see Resources slide)
    - ✓ Use pixel colors to locate target areas
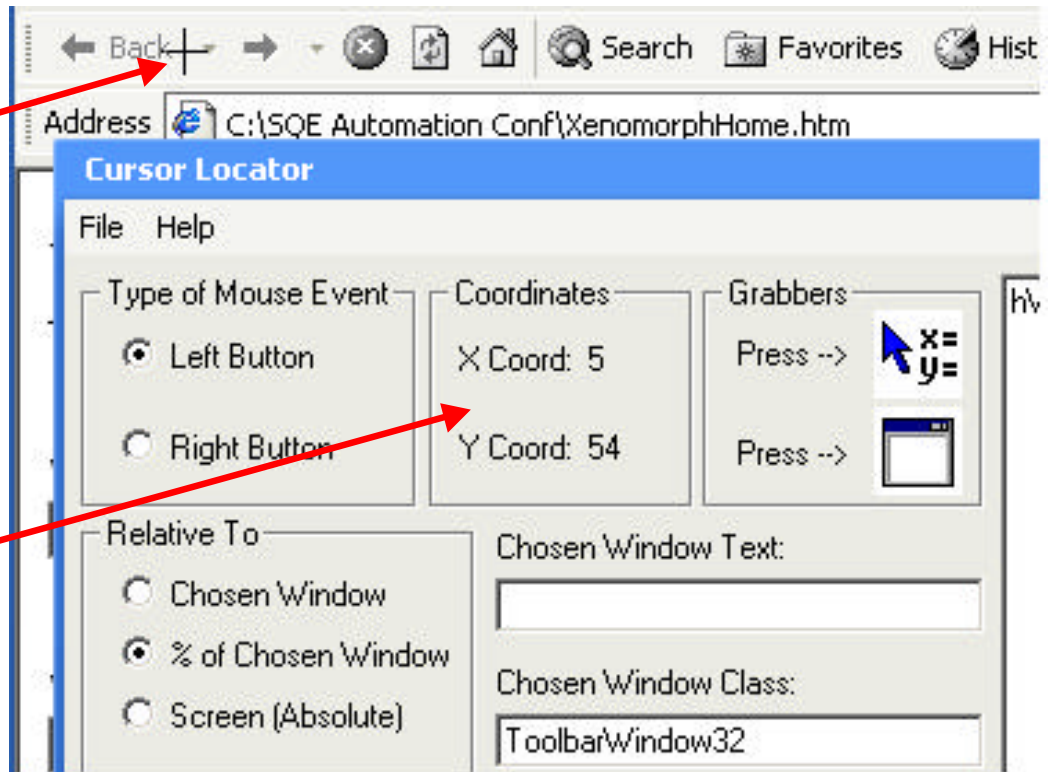
Cursor Locator
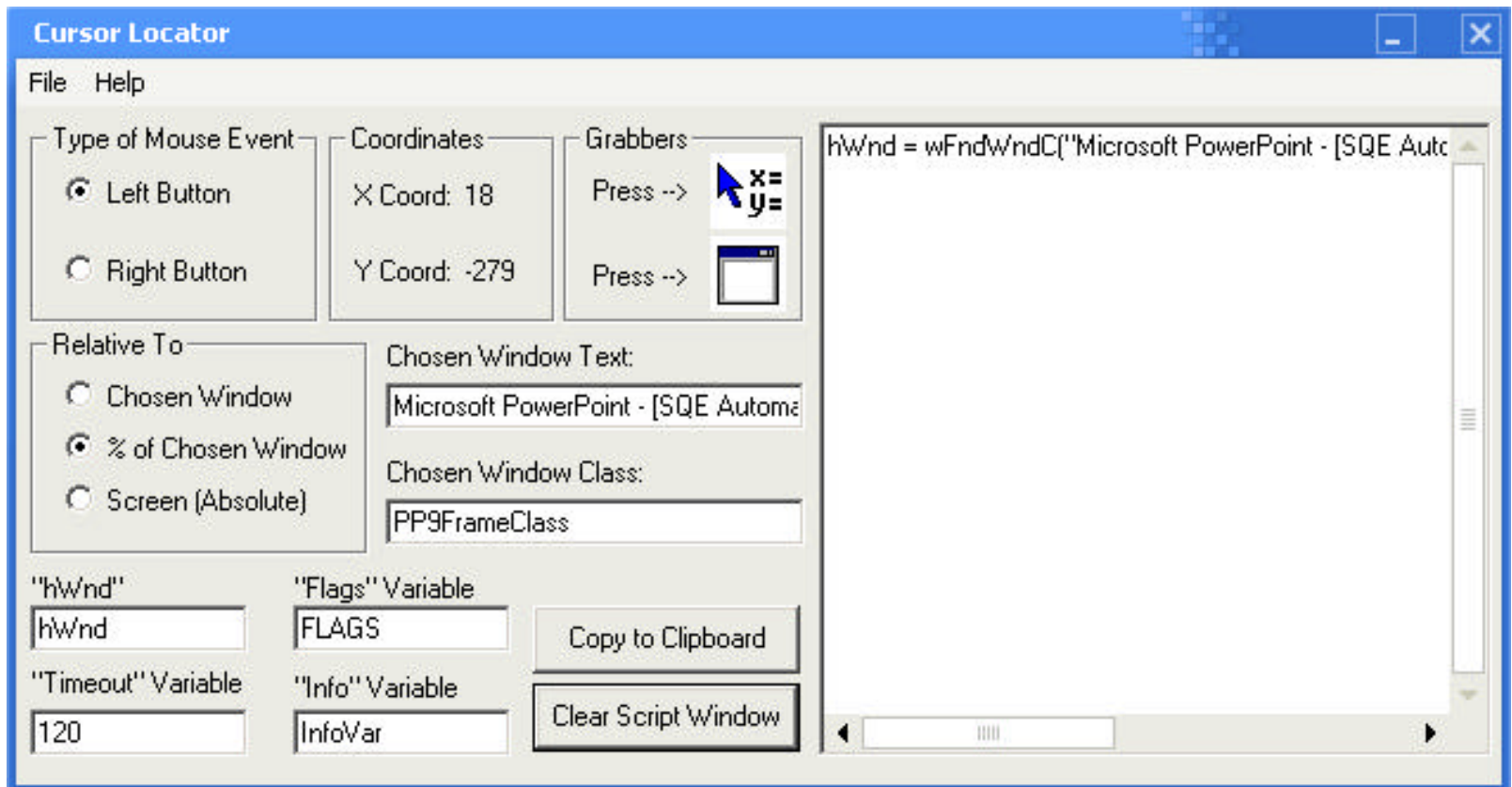
Cursor Locator – Choosing Target Window
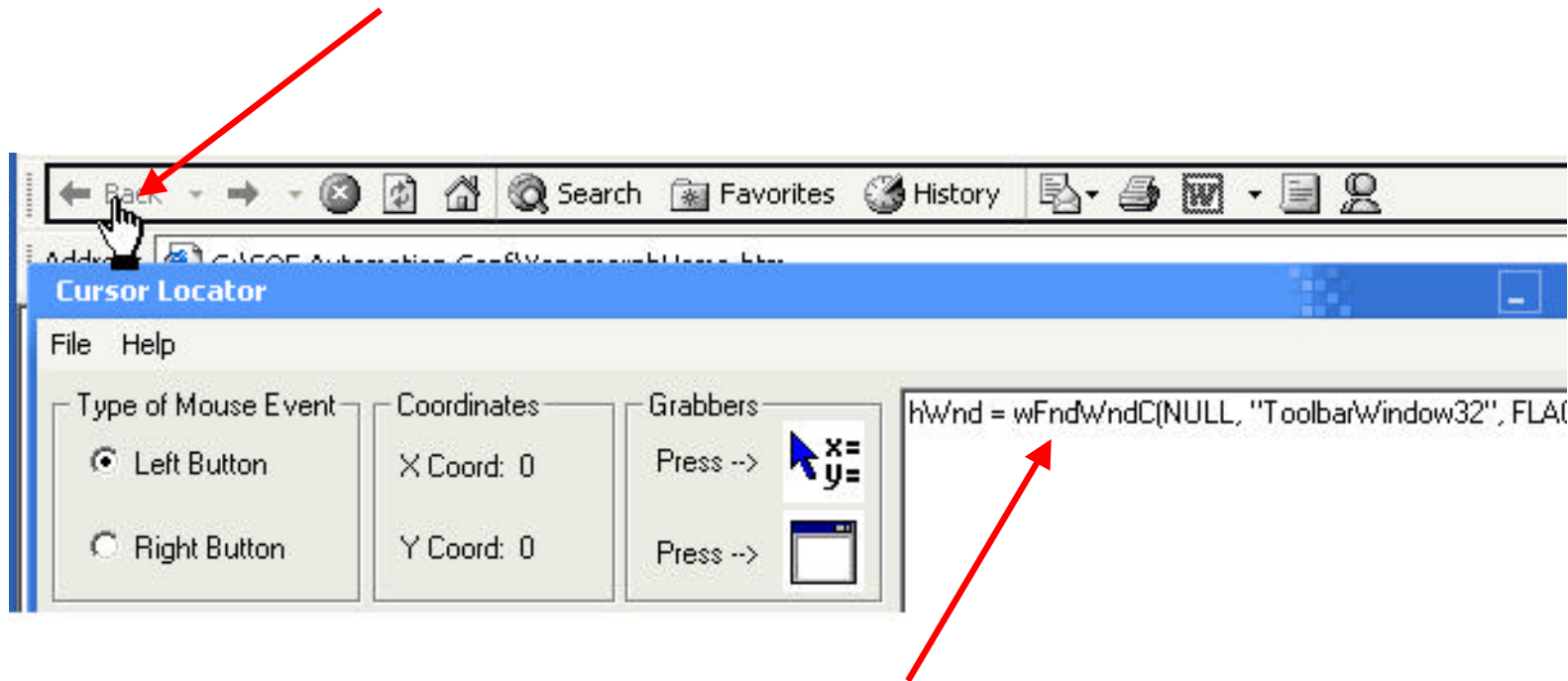
Coordinate grabber hovering over a location in the target window

Coordinates as percentage of width and height of target window

Example #1- Xenomorph

Xenomorph Products customers use a Web page to locate dealers
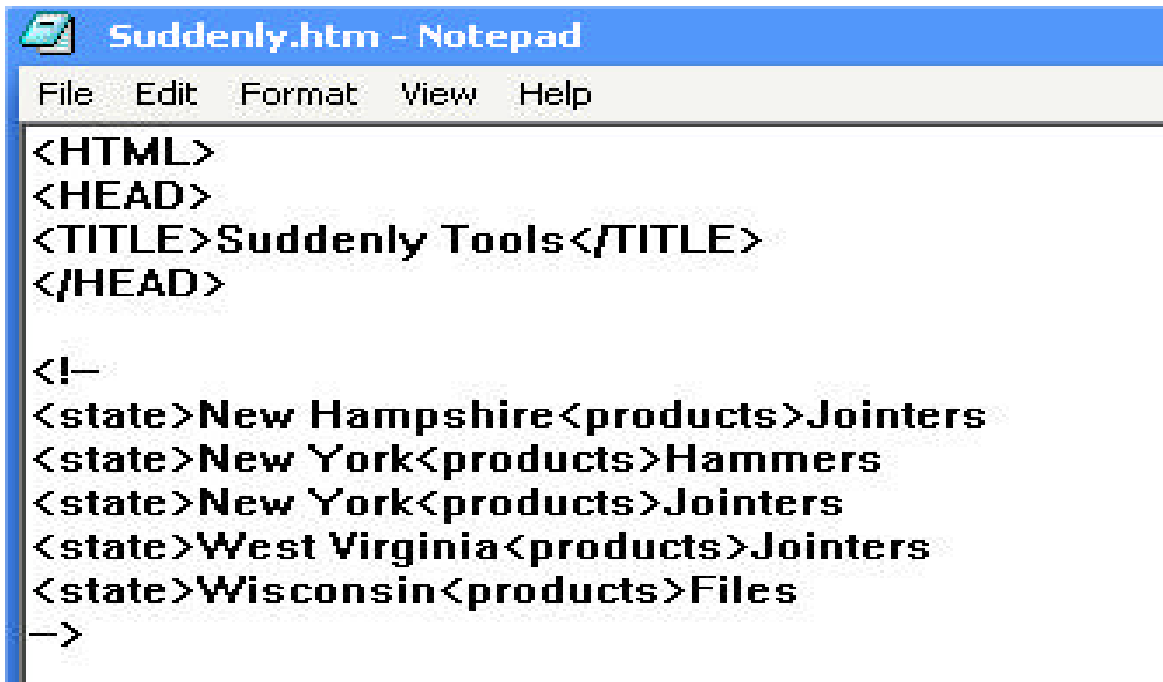
Xenomorph – Using Traditional Oracle

-                 customers in New York state is located in Baldwin Place NY

-                 customers in New York state is in Crowder OK!

  - ✓

  - ✓ Separate oracle requires constant maintenance, easy to make mistakes

Xenomorph – Using SVD

Customer state and product SVD codes embedded in non-printing comments on each dealer's Web

Xenomorph – Automating Verification Using SVD

- Select state and product name, click button to jump to dealer's page
- View Source on dealer's page
- Copy source from Notepad to the Windows Clipboard
- Extract non-printing comment
- Parse for selected state and product name

Example #2 - LargeRich

- LargeRich uses Filters in Excel to show selected employee records from a large set of rich data
- Not-Quite LargeRich (below) shows the same filters with a small number of records and "not-

	A	B	C	D	E	F	G
1	**Not-Quite LargeRich Filter Demonstration**						
2							
3							
4	**Salary**	**Salary**	**Div**	**Dept**	**Group**		
5	< 975			2	A		
6							
7							
8	**First Name**	**Last Name**	**Emp#**	**Salary**	**Div**	**Dept**	**Group**
9	Aamar	Macon	12345	$481.00	North	1	A
10	Beavis	Nilam	12346	$350.00	South	2	A
11	Camden	October	12347	$275.00	North	1	B
12	Daley	Platon	12348	$1,000.00	East	3	B
13	Edvin	Quade	12349	$963.00	West	4	A
14	Fargo	Raybert	12350	$1,107.00	South	6	C

Demo

16

LargeRich - Using SVD

Sample LargeRich employee record showing typical data, SVD codes and an added "Codestuff" field

```
First name: íòÁÑæŠnQQQKQkkkq0000154304xxxxU,,'1
Last name:  üVÝ¬ØJ%éQQQKQkkqk0000154304xxxx›Æü4
Emp Number: 154304
Salary:     3191
Division:   8ñîo**Ù‹øiÑ
Department: IÎ¡=Ø•ù¼¸
Group:      fýÊÏ&÷åT
Codestuff:  QQQKQqqkkkkkqíòÁÑæŠnU,,'1xxxxkkqküVÝ¬ØJ%é›Æü4xxxx
            qqqk154304xxxxkqkk3191xxxx
            qkkk8ñîo**Ù‹øiÑxxxxkkqqIÎ¡=Ø•ù¼¸xxxx
            kqqkfýÊÏ&÷åTxxxx
```

LargeRich - Unique Data Field

-

-

```
First name: iòÁÑæŠnQQQKQkkkq0000154304xxxxU,,'1
Codestuff:  QQQKQqqkkkkkqiòÁÑæŠnU,,'1xxxxkkqküVÝ¬§J%é›Æü4xxxx
            qqqk154304xxxxkqkk3191xxxx
```

-

- QQQ

- Valid data SVD code: KQ
- First name SVD code: kkkq
- Record number: 154304
- Record number SVD code: qqqk

LargeRich - Shared Data

-
-

```
Division:     8ñîo★★Ù‹øiÑ
Codestuff:    QQQKQqqkkkkkqiòÁÑæŠnU,,'1xooodkkqküVÝ¬ØJ%é›Æü4xoox
              qqqk154304xooxkqkk3191xooxx
              gkkk8ñîo★★Ù‹øiÑxooxkkqqIÎ¡=Ø•ù¼¸xoox
```

-
- QQQ
- KQ
-
-
-

LargeRich – Automating Verification Using SVD

- Use macros to trigger filters and select

- Choose filter parameters, enter in cells
- Trigger filter and select data
- Copy data to the Windows Clipboard
- Parse each data line
  - ✓ Verify displayed unique parameter data is

  - ✓ Verify displayed shared parameter data against Codestuff field

Resources

- SVD
  - ✓ "Self Verifying Data - Testing Without an Oracle," paper presented at STAR East '99
- Cursor Locator tool
  - ✓ noeln@microsoft.com
- Sample code from this presentation
  - ✓ noeln@microsoft.com
- Visual Test information
  - ✓ http://www.rational.com

Noel Nyman

Noel Nyman has worked in software product development and testing for over twenty years on an eclectic project mix including embedded controllers, shrink-wrap applications, and operating systems. As the lead for the Microsoft Windows NT 4.0 32-bit Applications Test team, he pioneered the use of "dumb" monkey test tools to increase operating system reliability. One of those tools is featured in the "Visual Test 6 Bible" (Tom Arnold, IDG books). Noel created the test plans and frameworks used for the Certified for Windows 2000 logo program. He worked with James Bach to create the Windows Exploratory Testing procedure used by the Microsoft Windows Application Experience Test teams to develop test case outlines for over 1500 applications.

Noel is a regular participant in the Los Altos Workshop on Software Testing and the Austin Workshop on Automated Testing, and he's an occasional contributor to *Software Testing & Quality Engineering* magazine.