Understanding Event Driven Programming

Daniel Appleman
Desaware
(408) 377-4770, CIS: 70303,2252

For beginners

Demos in file vbtevnt.zip

Roadmap

Programming the “Old fashioned” way.

The architecture of event driven software.

Top down design - coding sideways?

Event driven user interface:

- Power to the user = pain to the programmer?

Testing.

Q&A

Characteristics of non-event driven code

Code requests input.

Deterministic execution flow.

Defines which events are allowed.

Characteristics of event driven code

Input requests code.

Execution flows based on events.

Defines which events are prohibited.

Designing event driven software

Setup via events (resize / load / paint).
Living with Multitasking.
Use of DoEvents.
Rearchitect your algorithms.

Impacts on Design

- **Classic approach.**
  - Top down design.
  - Bottom up coding.
  - Lots of code to see results.
- **Event driven approach.**
  - Top down design - yes!
  - Coding sideways.

User Interface Code

Anything can happen,

(and usually does).

Enabling UI objects.

Validation primer.

Testing

Complex as compared to non-event driven.
Testing in real environment.
Incremental Testing.
Event drivers (MS-Test).
Tracing.

Conclusions

Changing thought patterns.
Rich programming environment.

Q&A

There is no such thing as a stupid question