

CE95940 – Introduction to Visual BASIC

- Review
- Working with the Clipboard
- DDE -- Dynamic Data Exchange

Review common questions and problems from last week

Clipboard

- Standard component of Windows
- Temporary/transient storage
- Different types of data
- Clipboard viewer

2

used primarily to cut/paste between applications

applications must be made "clipboard aware"

data types: e.g. VB supports text, bitmap, palette, link; others include RTF, spreadsheets

each application chooses what formats will be usable

one datum of each type can coexist e.g. one bitmap, one text, one link

placing a new one of a type replaces existing one

typical used with Edit menu: cut (puts onto CB and removes), copy (just put onto CB), paste (copy from CB to whatever)

demo clipboard viewer -- cut/paste word/excel

The VB Clipboard

- Invisible object
- No properties or events
- Methods:
 - Clear
 - GetFormat, GetData, GetText
 - SetData, SetText

3

“Built-in” like Screen, printer, debug

Only interaction is methods (observe that this could just as easily be a set of procedures, but use object paradigm)

Clear:

GetFormat: arg is type-identifier, returns true/false if CB contains that type

GetData: returns something that is compatible with Picture property of Image or PictureBox: i.e. assign result to picture property to display it.

GetText: two kinds of text: simple or DDE link information. use an arg to specify which

SetData, SetText: place information into clipboard. pass argument to indicate what kind of data (text=text or DDE link; data=bitmap, metafile, DIB, palette)

Simple demo: (picture on CB with alt-prtscrn)

```
form1.picture = clipboard.GetData()
```

other demo: pgm 7.1: decl section for consts (note 7.1 doesn't clear;

picture still there

An “Edit” menu

- Edit: cut, copy and paste to/from clipboard
- Typically work on selected text

standard part of many applications
aside: for selected text

Text selection

- Text and Combo controls
- Select portion of text with mouse
- Properties:
 - SelStart
 - SelLength
 - SelText
- Read/write: can be used to change contents

5

SelStart: position of first selected char, 0 means to left of first char
if nothing selected, indicates insertion point (position of vertical bar)

SelLength is number of characters selected, 0 means nothing selected

SelText is actual text that is selected, null string is nothing

Can write to these properties to do selection under program control.
Assigning to seltext replaces selected text with string that is being
assigned. To delete selected text, assign null string

demo 7.2: see text1.change

demo 7.3: see replace.click; note also annoyance about focus shifting

An “Edit” menu

- Use text selection together with Clipboard methods
- Example
- Problem: too specific

6

show example 7.4 (copy, paste, cut)

Note, default format for settext is TEXT (as opposed to DDE_LINK)

Not general enough, only works on “text1”

Use property of Screen called ActiveControl -- refers to control on form that has focus

(using focus in this manner requires menus, since they don't get focus)

ActiveControl

- Property of Screen built-in object
- Typical use: `Screen.ActiveControl.Text`
- Can declare object variables:

```
dim target as control  
set target = Screen.ActiveControl  
target.text = "this is text"
```
- Also `Screen.ActiveForm`

7

use to write code that refers to whatever control has focus
can use object variables if need be -- can be used as procedure arguments, too (similar to last week with form object variables)

`Screen.ActiveForm` for use in multi-form applications and especially MDI applications; see examples in on-line help for `activecontrol` and `activeform`

`screen.activeform.activecontrol.text` is valid

demo 7\5 (copy, paste, cut)

Clipboard data format

- Clipboard supports various data formats
- How to generalize code?
- Use “TypeOf”:
if TypeOf screen.ActiveControl Is TextBox then
 clipboard.SetText screen.ActiveControl.Text
elseif TypeOf screen.ActiveControl Is PictureBox then
 clipboard.SetData screen.ActiveControl.Image
end if
- Example

8

We’ve been dealing with text, but images can also be manipulated

Need to be able to figure out which methods to use eg SetText or SetData. Or: cut/copy/paste are context-sensitive (they know what they’re doing)

Cut and copy need to distinguish between Set*; Paste needs to choose between Get*

show 7.6; (copy/paste/cut) note that it only handles a couple of cases:

text applies to text, combo and list

bitmaps applies to picturebox and image control

Note: we do a paste (eg) without knowing if there is valid data. eg if picture is active, we do a getdata, but we don’t know if there is anything available. Use GetFormat to find out. Enable/disable menu items depending on whether or not valid data is present. Look at edit-menu.

example 7.7

DDE -- Dynamic Data Exchange

- Move data directly between applications
- Update is (usually) automatic
- Source: where the data comes from
- Destination: where the data goes
- Destination initiates a *conversation*, requesting data from the source
- Not supported by all applications

9

exchange data without having to use the clipboard, normally automated
other terminology (client and server)

use source and destination (ie.e relative to movement of data), use
conversation analogy to described data transfer

demo a DDE link between Word and Excel

DDE conversations

- Initiated by destination
- Specify name of source and topic of conversation
- Exchange items of information
- Can create both source and destination with VB

10

destinations can carry on several conversations simultaneously
no restrictions, can be a source in one conversation, destination in another
name & topic identify the conversation

Sources

- DDE source is application-name
- DDE topic is form's LinkTopic property
- Conversation items are names of text, label or picturebox controls

11

demo dde\source: look at linkmode and linktopic, note names of textboxes

note: create exe for use by destination

Destinations

- Destination of conversation is any text, label or picturebox control
- Destination properties:
 - LinkMode: none, automatic, manual, notify
 - LinkTopic: "*application|topic*" of source
 - LinkItem: "*control-name*" of control on source

12

mode indicates type of conversation

topic is name of application, for VB name of executable or project;
followed by or-bar; followed by LinkTopic property of the form that is to
be the source

item is name of control on the form specified by the topic

demo dde\dest

note: shell function to start program

demo dde2;src & dest; discuss manual vs automatic

Notify links

- Source application notifies destination of pending change
- Destination control LinkMode is “notify”
- Destination control gets LinkNotify event
- Destination control LinkRequest method

13

may not want automatic updates, want to know when changes pending
source creates event automatically - no programming required
when a notify event occurs, invoke LinkRequest to get new value
use a button, enabled/disabled as appropriate

demo program 3

note: linkNotify event of textdest

updatenow button click

form-load

Updating Picture links

- Automatic can be time-consuming
- Use LinkSend method of PictureBox to create LinkNotify event in destination

14

pictures can be big and updates can be time-consuming

on source: use linksend method of picture to create a linknotify event in the destination (dest must still use linkrequest to get data)

allows source to control when destination will try to get new data

demo 3a: source is freehand drawing demo; it must always use linksend to advise of changes (regardless of dest'n mode)

Other DDE events

- LinkOpen, LinkClose, LinkError
- LinkOpen for source allows rejection

15

mostly useful for debugging and program-tracking, eg source linkclose could terminate program.

But, source can reject open request (eg limit number of conversations)

DDE command interface

- Send/receive commands (vs data)
- Destination sends command with LinkExecute method
- Source receives command via LinkExecute event
- Command syntax and semantics are program-defined

16

distinguish between data and commands

establish conversation as usual (omit item; typically not relevant)

arg to linkexecute method is command to be executed

arg to linkexecute event is command & boolean that sends acknowledgement back (false means ok, true means not ok)

demo program 4: source linkexecute event, dest send-command button

Reversed dataflow

- Send data from destination to source
- LinkPoke method of destination control

17

can be used as a reverse channel for data

must be careful -- poke will cause an update which may update destination again (if auto)

better to use commands to do updates? or at least a separate control (if it's a control-type purpose)

demo pgm 5: send-back control; data that is send is contents of control (no choice)

Creating links dynamically

- Implementing “Paste Link”
- Use Clipboard to pass link information (clipboard format CF_LINK)
- Windows convention for link information: *application|topic!item*

18

basic idea is to use clipboard to communicate link information
standard format for such information, clipboard reserves special data
class for holding it

Copy & Paste Link

- “Copy” places link information into clipboard along with raw data
- “Paste Link”
 - fetch link information (if available)
 - separate into LinkTopic and LinkItem strings
 - set destination control properties
- VB Programmer’s Guide, Chapter 20

19

source is where copy occurs

dest is where paste link occurs

handle details like: valid control type

good examples in ref (including error handling)

Non-DDE applications

- Want to be able to control applications that do not support DDE
- Send keystrokes: SendKeys
- Sends to active application
- AppActivate: active an application

20

want to be able to do stuff to non-dde apps.

can use SendKeys to simulate typing

sends to active application, so need to switch focus: AppActivate

AppActivate uses title, which is unreliable sometimes

If more than one running, get one at random

Sendkeys can send special codes for escapes, alts etc.; see sendkeys help for details

demo program 8: note alt-space to activate sys-menu -- need to know the Windows keyboard alternates to use sendkeys!