Input Validation

3

◆ Overview

- Getting the Input

- Validating the Input

3

Using the InputBox$ Function

```
┌─────────────────────────────────────────────────┐
│ ▬          Input Box                             │
├─────────────────────────────────────────────────┤
│                                      ┌─────────┐ │
│   Enter your name.                   │   OK    │ │
│                                      └─────────┘ │
│                                      ┌─────────┐ │
│                                      │ Cancel  │ │
│                                      └─────────┘ │
│   ┌───────────────────────────────────────────┐ │
│   │ |                                         │ │
│   └───────────────────────────────────────────┘ │
└─────────────────────────────────────────────────┘
```
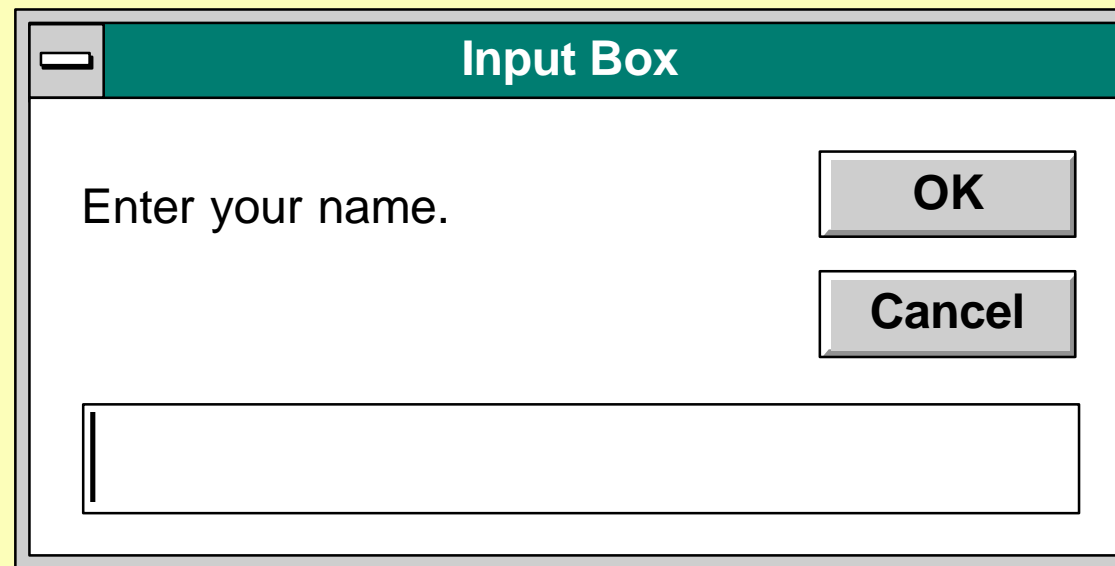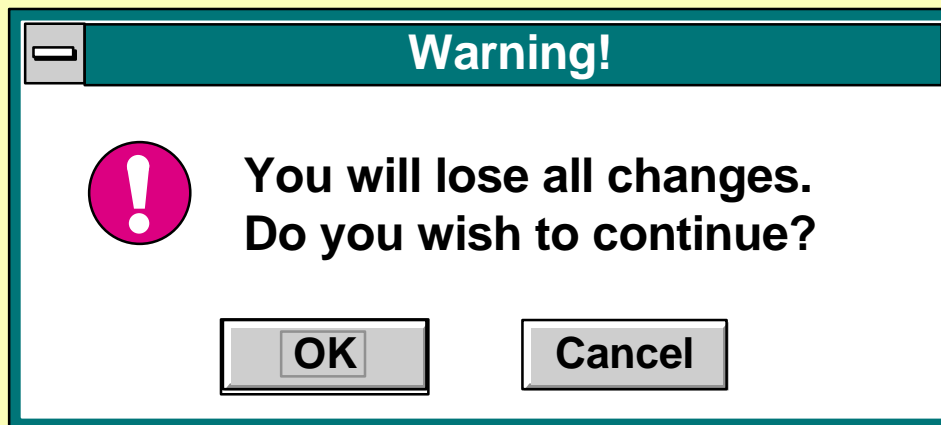
```
UserTypeIn = InputBox$ (MsgText, MsgTitle, Default)
```

3

Displaying Text with a Label or Message Box

Hello

```
Label1.Caption = "Hello"
```

Warning!

! You will lose all changes.
Do you wish to continue?

OK   Cancel

```
Response = MsgBox (MsgText, MsgType, MsgTitle)
```

3

Controlling the Tab Order

TabIndex Values

Assigning Access Keys

Form1

Name:

Number:

City:

OK

Cancel

Text3.TabIndex = 5

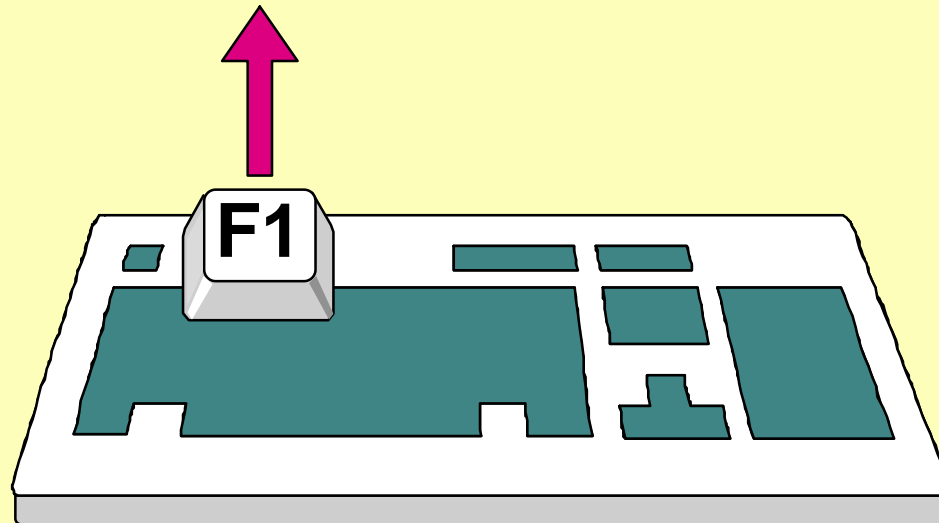Label1.TabIndex = 4

3

Using Properties to Restrict Data Entry

`.MaxLength = 5`

Form1

Name:

`.Default = TRUE`

OK

Number:

`.Cancel = TRUE`

Cancel

Password:

`.PassWordChar = "*"`

3
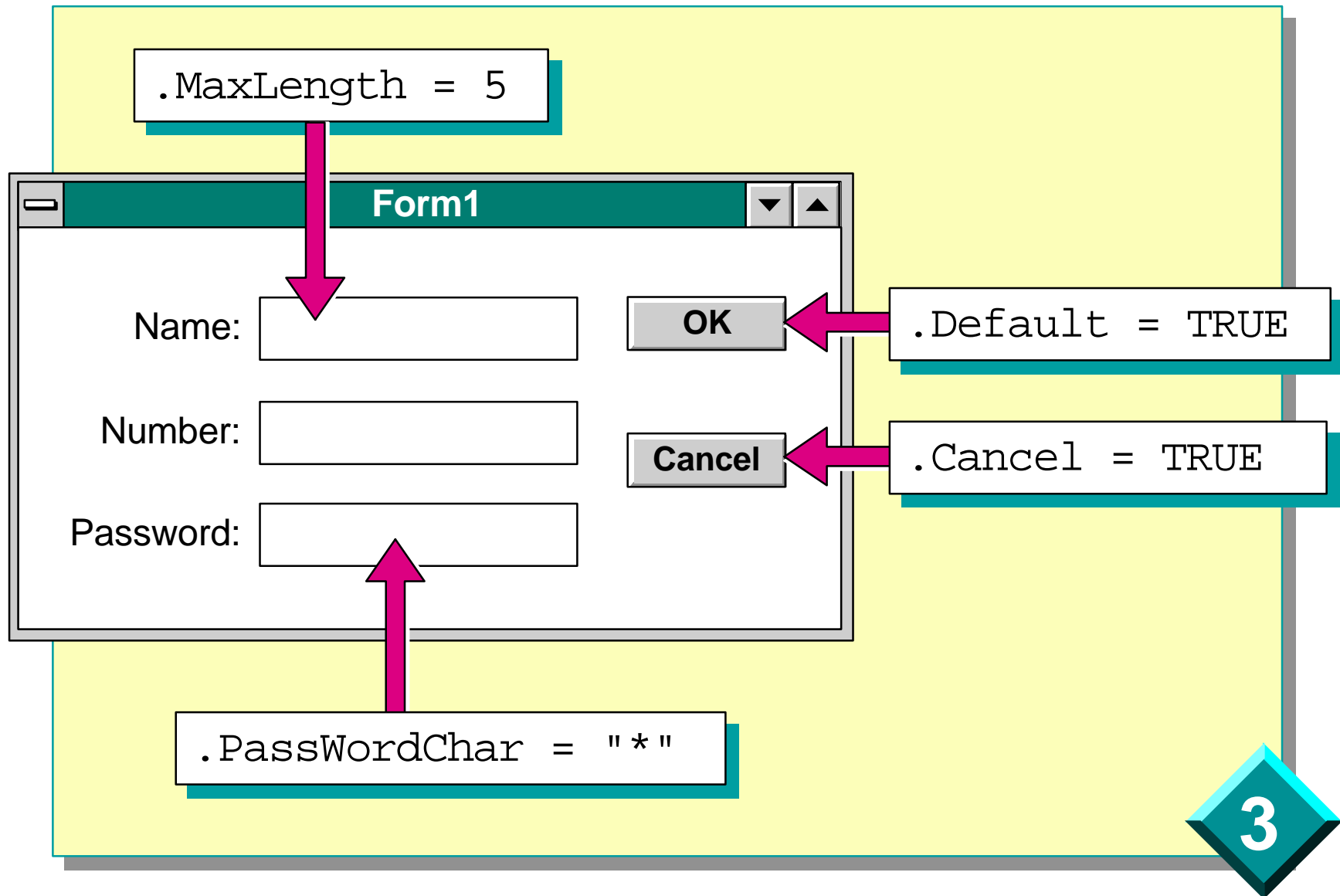
Using KeyPress to Restrict Keystrokes

```
Text1_KeyPress(KeyAscii...)
```
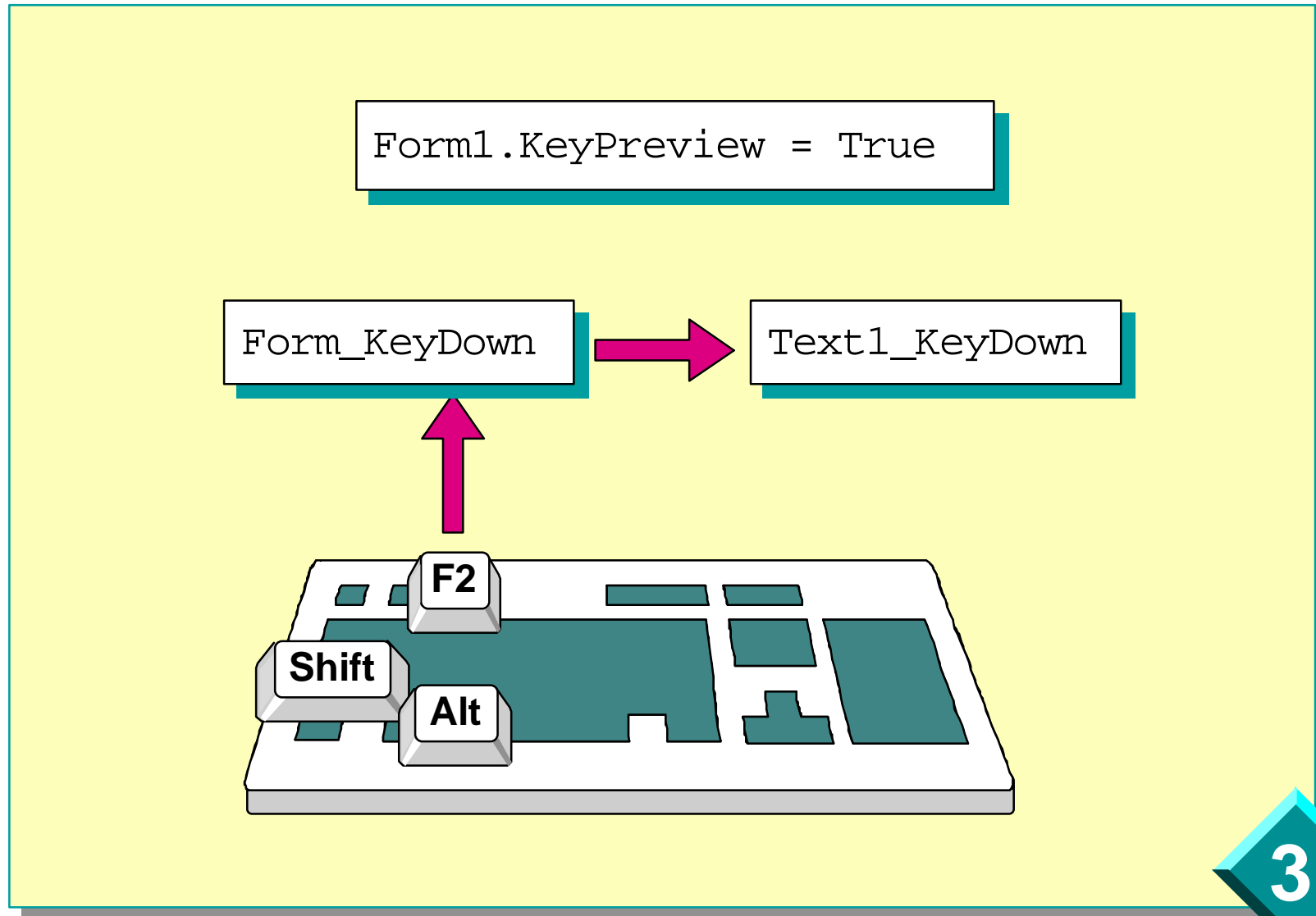
m

M

M

Form 1

Name: M|

Text1

3
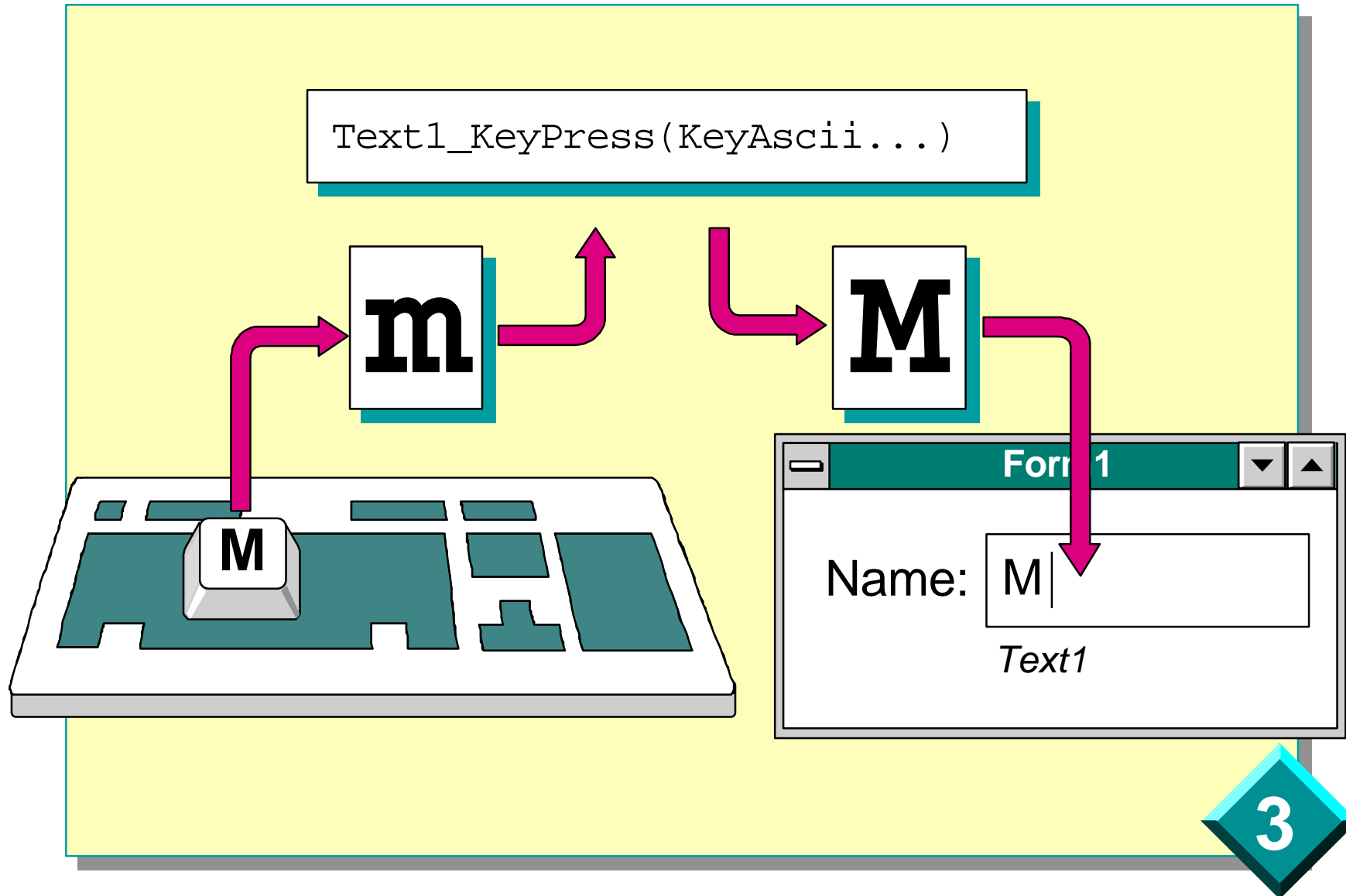
Using KeyUp and KeyDown to Trap Keystrokes

```
Text1_KeyDown(KeyCode...)     'Check Key
```

F1

3

A Form-Level Keyboard Handler

```
Form1.KeyPreview = True
```

```
Form_KeyDown
```
→
```
Text1_KeyDown
```

F2

Shift

Alt

3

Using ENTER Key to Tab Between Controls


```
Form_KeyPress
'If ENTER key pressed
    'Check TabIndex
    'Set new TabIndex
    'Set focus
```

Form1

0 Name: 1

2 Number: 3

3

The Masked Edit Control

Form1	▼ ▲

Name: [   ] ⬅ `.Mask = &&&&&`

State: [   ] ⬅ `.Mask = ??`

Number: [   ] ⬅ `.Mask = ##`

Soc-Sec-No: [   ] ⬅ `.Mask = ###-##-####`

[   ] ⬅ `lblError.Caption`
`➥ = "Error Message"`

[**OK**]   [**Cancel**]

3

The ValidationError Event

- **ValidationError Event**

```
MaskedEdit1_ValidationError
(InvalidText as String, StartPosition as Integer)
```

- Analyze InvalidText

- Analyze StartPosition

3

String Functions

Function	Purpose
LTrim$	Deletes left trailing spaces
RTrim$	Deletes right trailing spaces
Trim$	Deletes left and right trailing spaces
Mid$	Returns a string that is part of another string
Left$	Returns a string consisting of the n leftmost characters of a string
Right$	Returns a string consisting of the n rightmost characters of a string

3

String Functions *(continued)*

Function	Purpose
Chr$	Converts ASCII to single character
Val	Converts String to Numeric
Format$	Converts Numeric to Formatted String as specified
Str$	Converts Numeric to String
Instr	Searches Target String for matching Search String; Returns Position
Hex$/Oct$	Returns a string consisting of the converted numerical value

3

Enabling the OK Button

```
Sub ValidateOKButton
    'Verify Controls
```

Form1

Name:

Number:

OK   Cancel

```
mskName_Change
    ValidateOKButton mskName
```
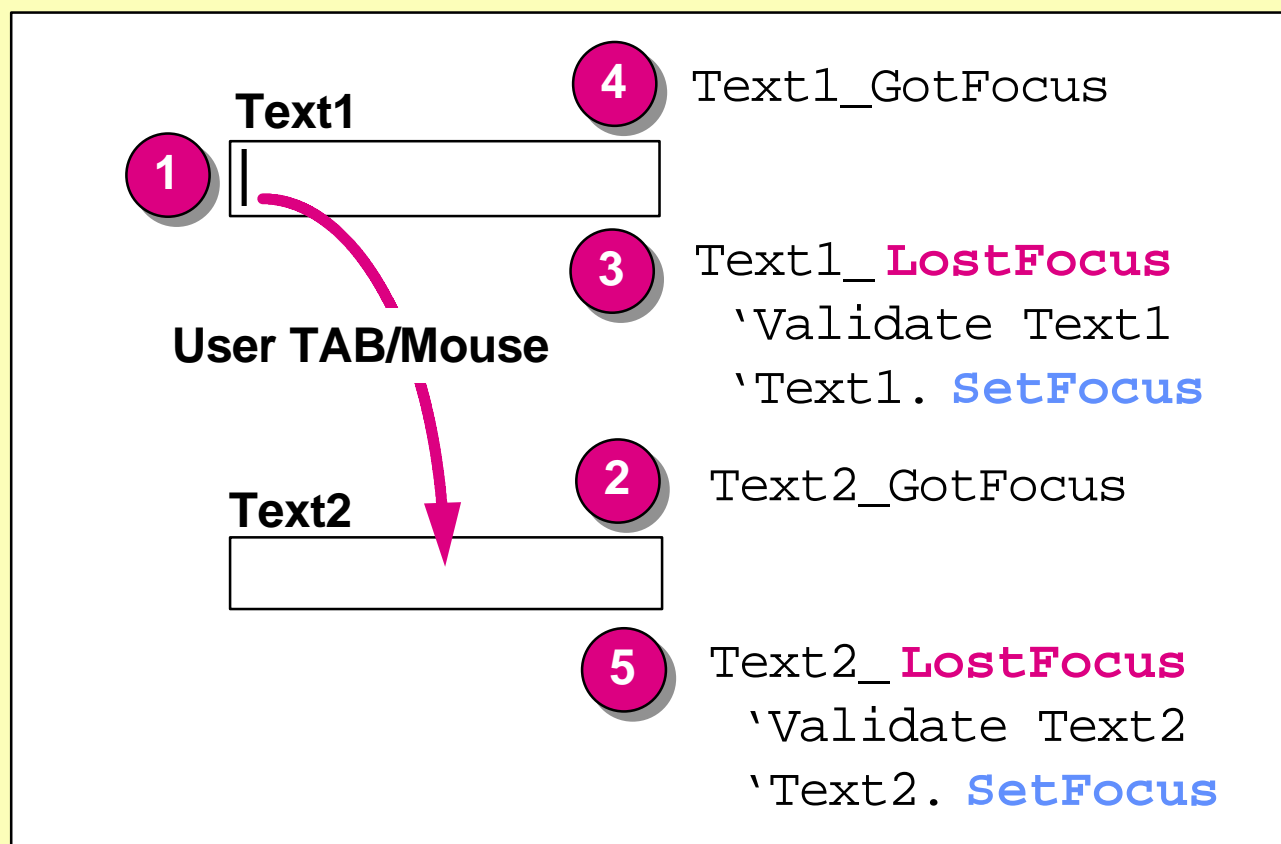
```
mskNumber_Change
    ValidateOKButton mskNumber
```
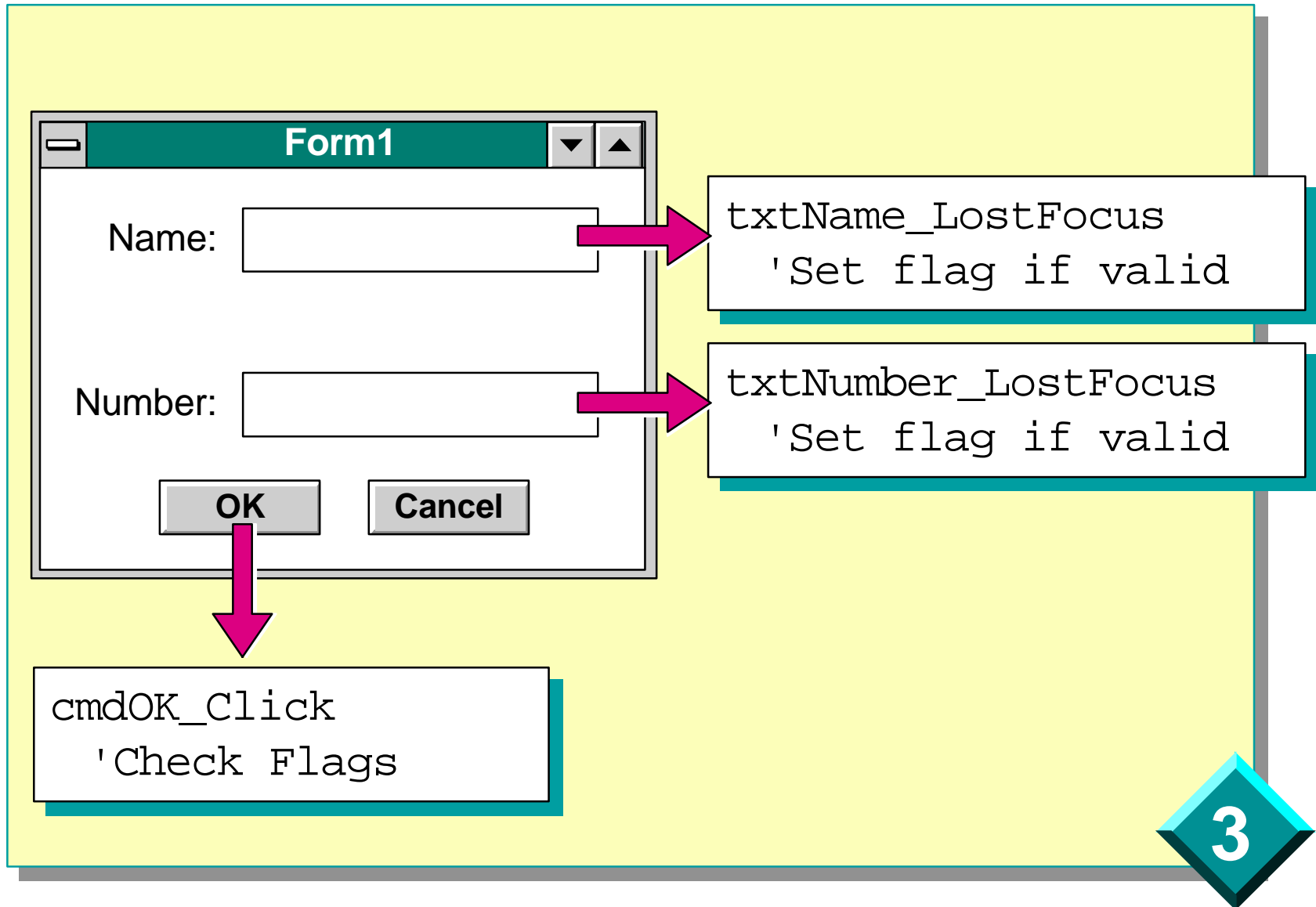
Disable at Design Time

3

Competing for Focus

■ **Do Not Use SetFocus in LostFocus Event**

Text1

(1)

User TAB/Mouse

Text2

(4) `Text1_GotFocus`

(3) `Text1_`**`LostFocus`**
   `'Validate Text1`
   `'Text1.`**`SetFocus`**

(2) `Text2_GotFocus`

(5) `Text2_`**`LostFocus`**
   `'Validate Text2`
   `'Text2.`**`SetFocus`**

3

Validate Controls

Form1

Name:

Number:

OK   Cancel

txtName_LostFocus
  'Set flag if valid

txtNumber_LostFocus
  'Set flag if valid

cmdOK_Click
  'Check Flags

3

Detecting Form Focus

- A Form Receives a LostFocus Event Only If There Are No Controls on the Form

- A Form Receives an Activate or Deactivate Event

3

Input Validation Hierarchy

- **Keystroke Validation**

  - Masked Edit Control - Mask Setting; ValidationError

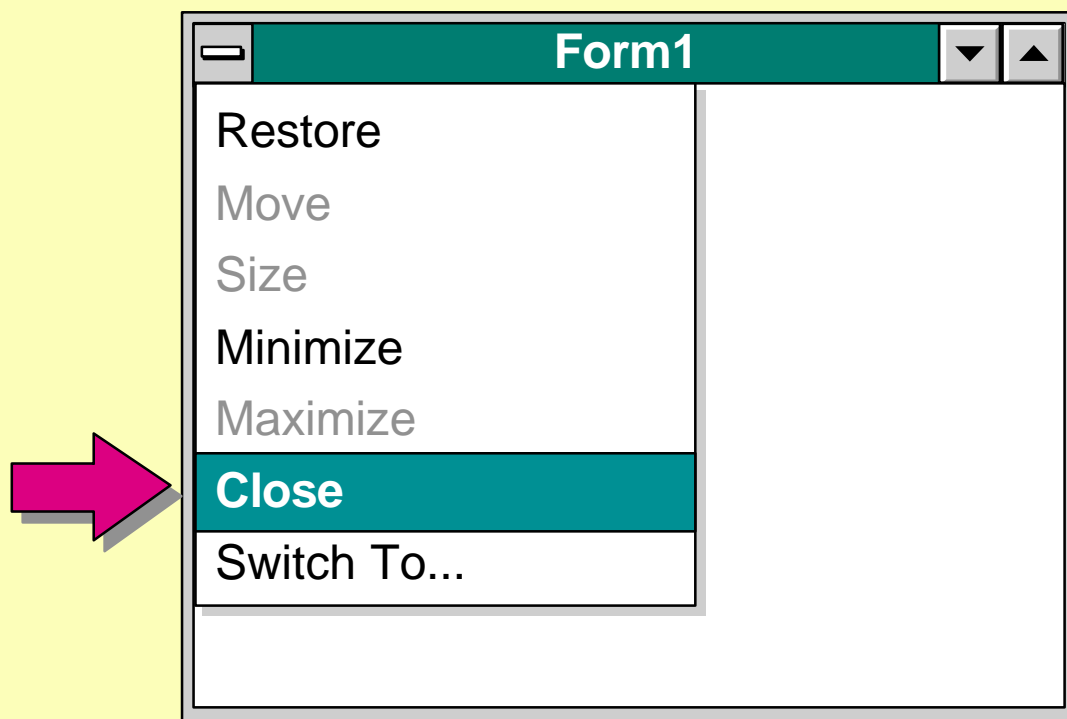 - KeyPress, KeyUp, KeyDown, Change Events

- **Input Field Validation**

  - MaxLength Property

  - ValidationError Event; LostFocus Event

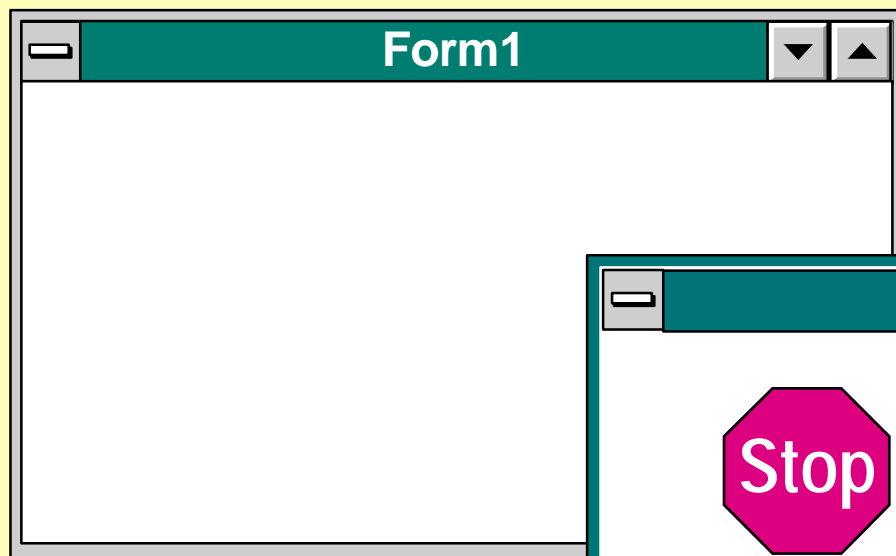 - String Analysis

- **Ensemble Validation**

  - Flags Array

3

Using QueryUnload

```
Sub Form_QueryUnload (Cancel As Integer,
➥ UnloadMode As Integer)
    If UnloadMode = FORM_CONTROLMENU Then...
```
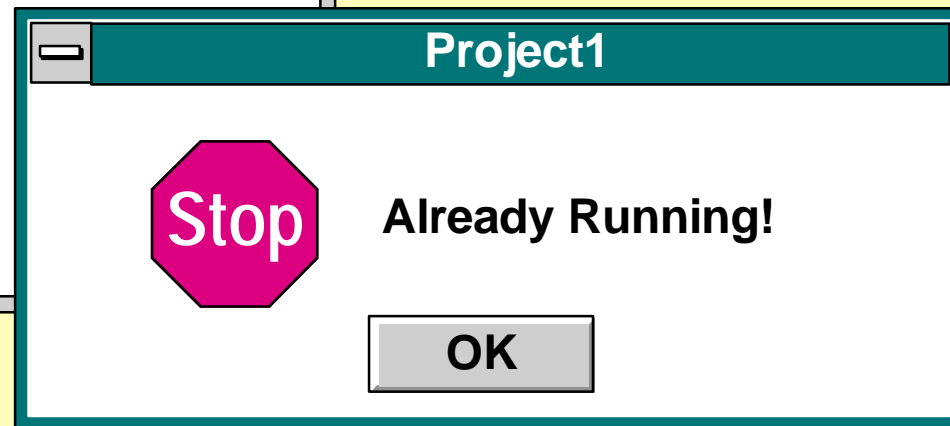
Form1
Restore
Move
Size
Minimize
Maximize
Close
Switch To...

3

Checking the App Object

```
Sub Form_Load ()
    If App.PrevInstance...
```

Form1

MYPROG.EXE

Project1

Stop **Already Running!**

OK

MYPROG.EXE

3

Summary

- **Getting the Input**

- **Validating the Input**

3