Module 7: Trapping Run-Time Errors

7

Overview

Where to Trap Errors

Writing Error Handlers
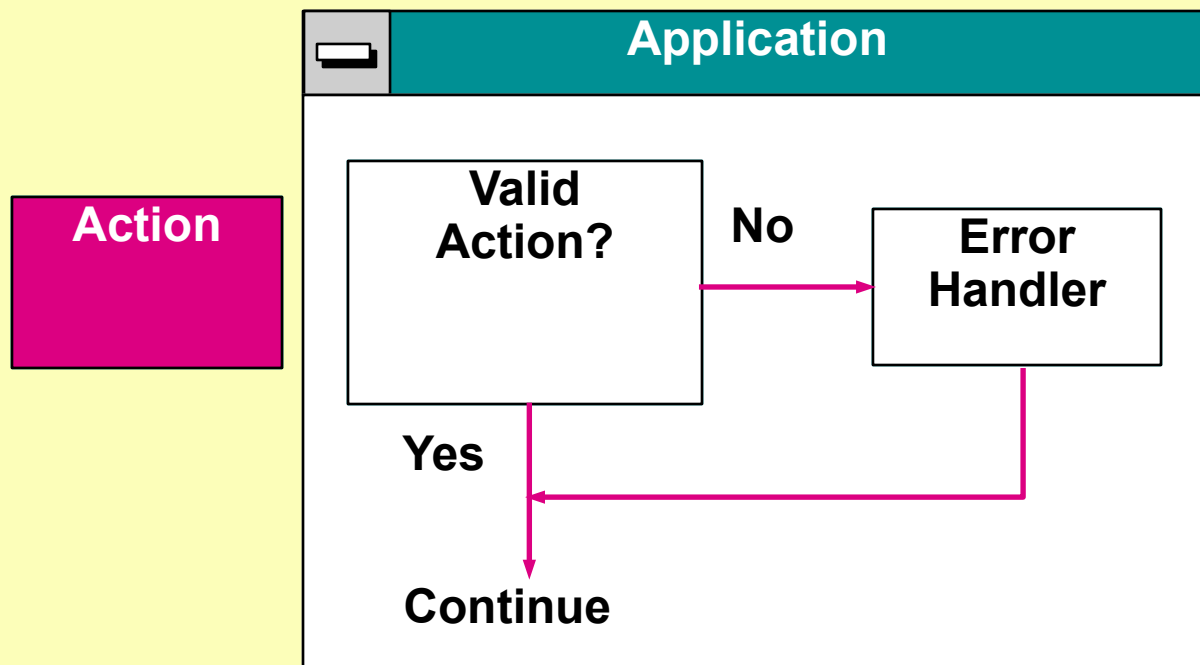
7

Trapping Errors Using Error Handlers

Error Handlers

Inform user of error actions
Request corrective action
Take corrective action

Action

Application

Valid Action?

No → **Error Handler**

Yes

Continue

7

Creating Error Handlers

Enable the Error Trap

Write the Error-Handling Procedure

Exit the Error-Handling Procedure

7

Enabling the Error Trap

```
On Error GoTo ErrorTrap
    .
    .
    .

Exit Sub
ErrorTrap:
    'Deal with error
End Sub
```
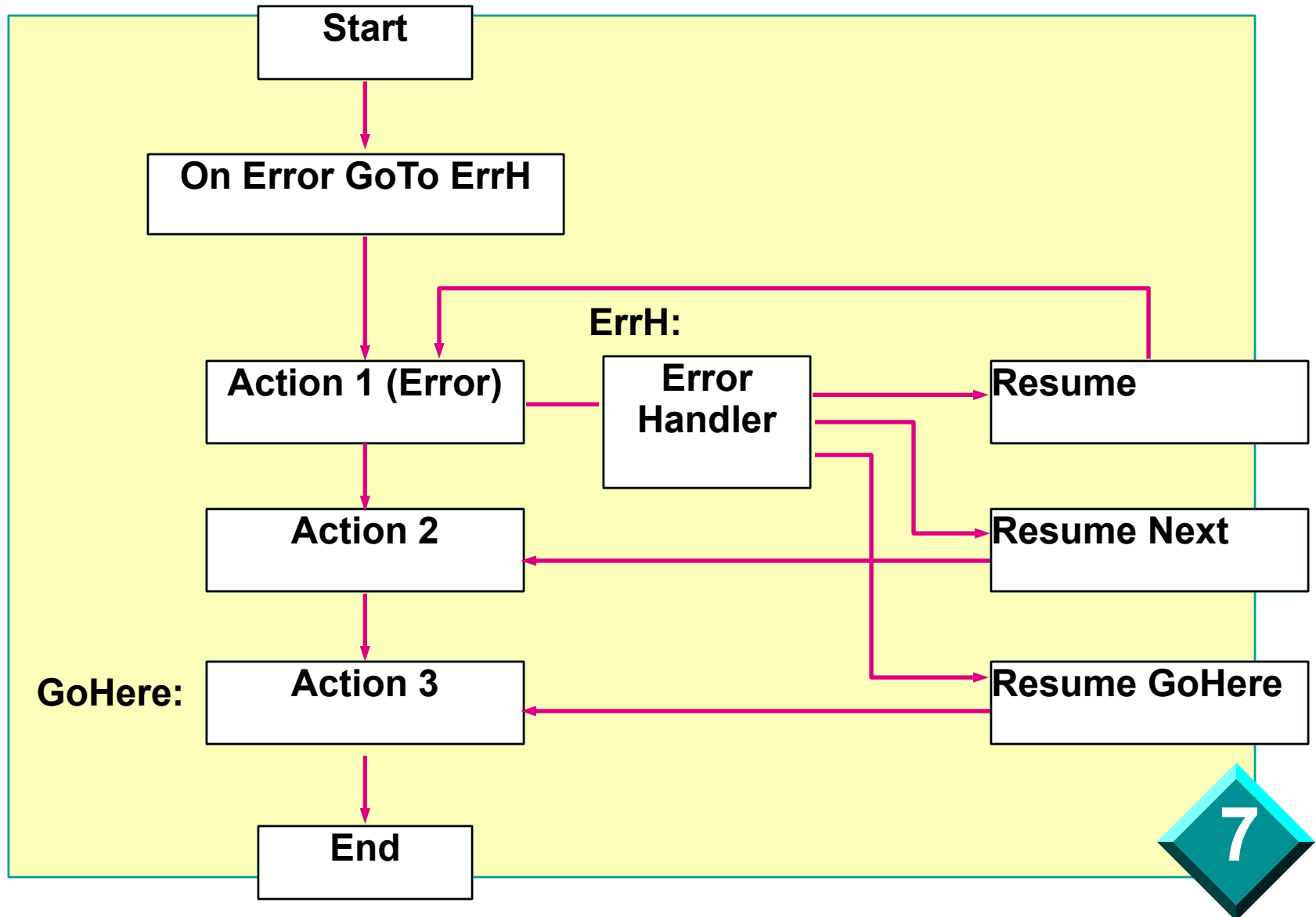
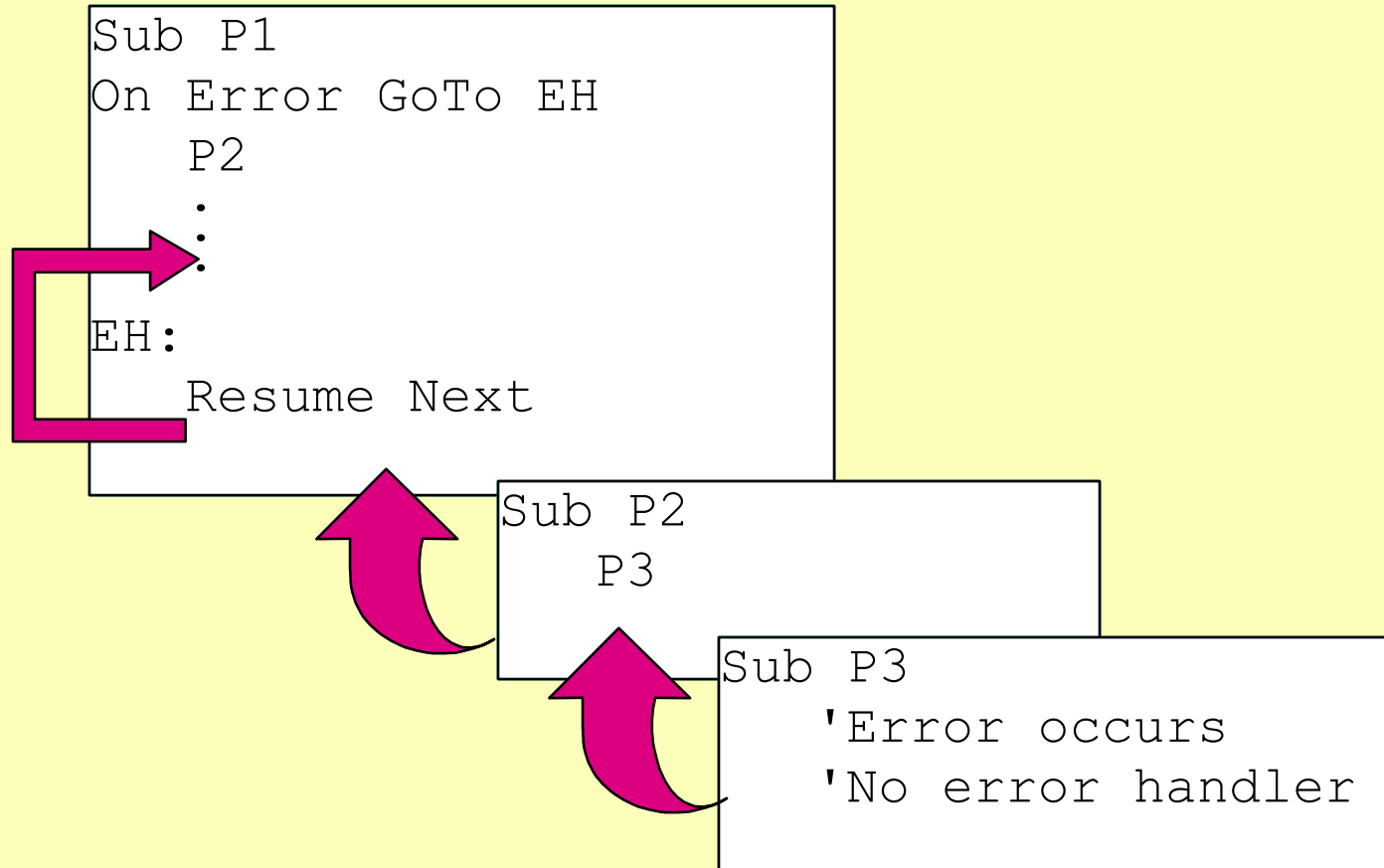*Procedure*

7

Writing the Error Handler

Err - Error Number

Error$ - Error Message
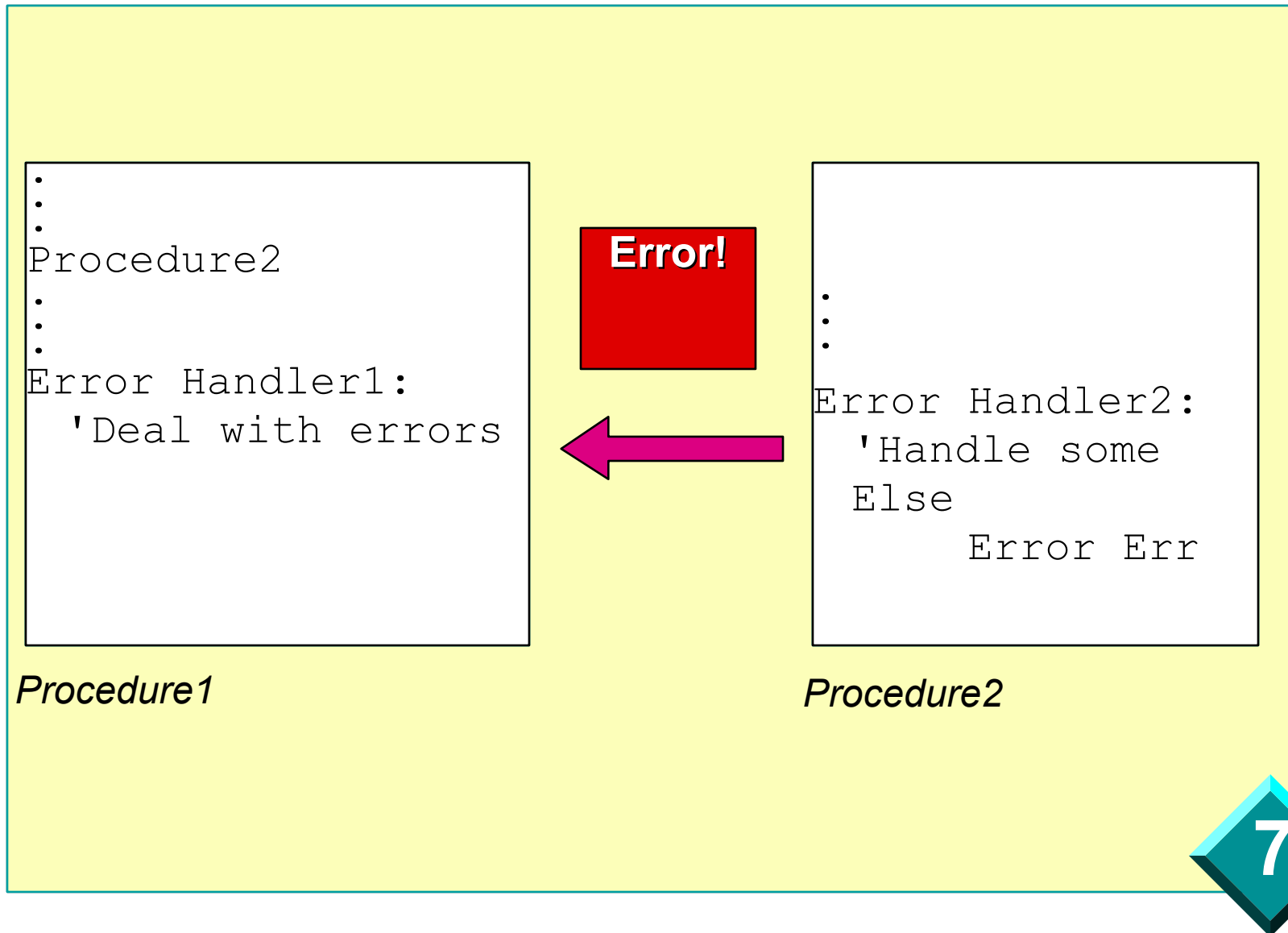
7

Exiting the Error Handler

Errors Automatically Passed to Calling Procedure

```
Sub P1
On Error GoTo EH
    P2
    .
    .
    .

EH:
    Resume Next
```

```
Sub P2
    P3
```

```
Sub P3
    'Error occurs
    'No error handler
```

7

Passing the Error to the Calling Procedure

```
.
.
.
Procedure2
.
.
.
Error Handler1:
  'Deal with errors
```

Procedure1

Error!

```
.
.
.
Error Handler2:
  'Handle some
  Else
     Error Err
```
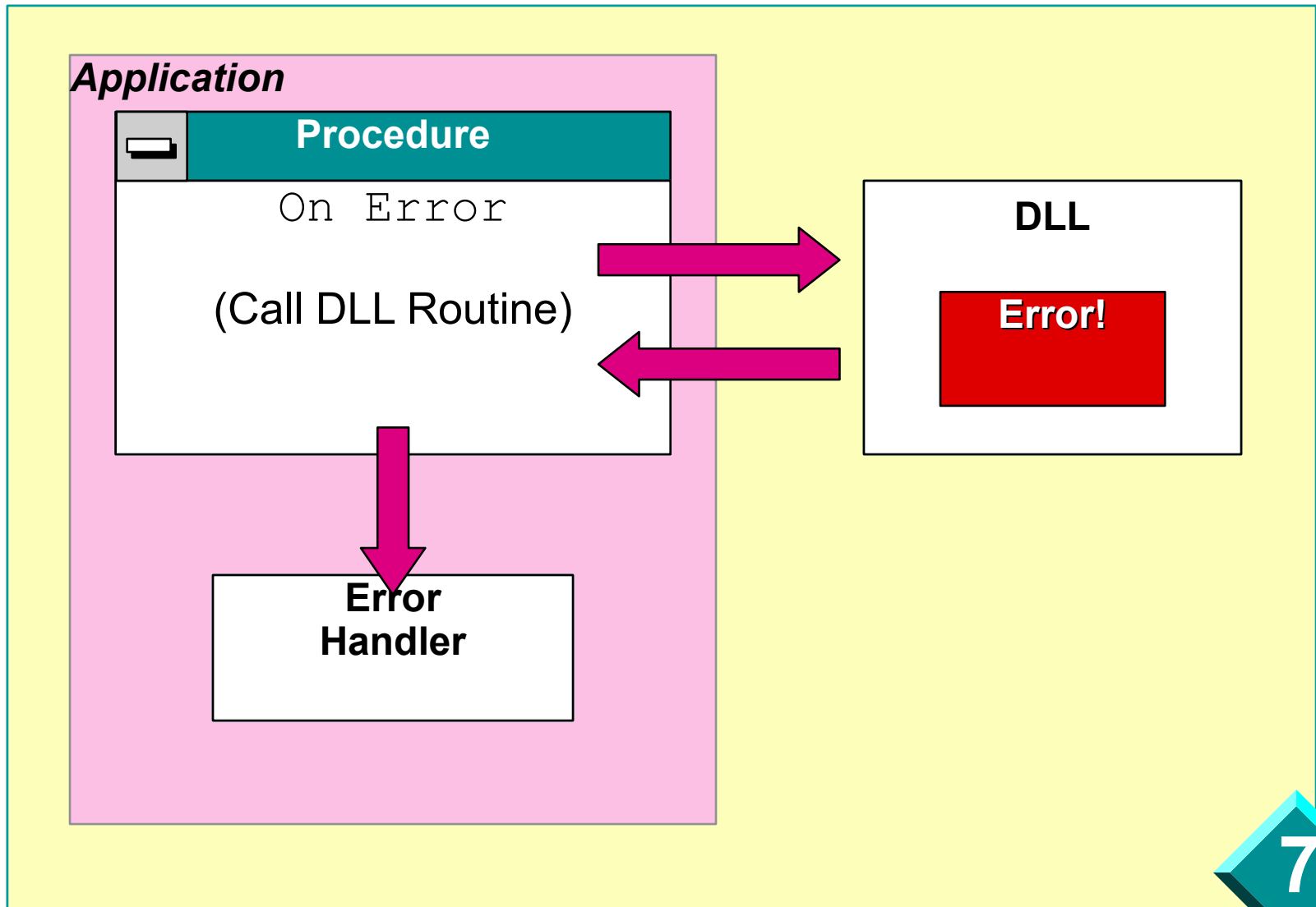
Procedure2

7

Generating An Error

Error *number*

Use Error Handling Mechanism For Your Purposes

Don't Duplicate Existing System Error Numbers

7

Trapping Errors in External Procedures

Application

Procedure

On Error

(Call DLL Routine)

DLL

Error!

Error Handler

7

Scope of Errors

Error Cleared at End Sub or End Function

Error can be quashed within Sub or Function scope:

On Error GoTo 0 - Resets Error Mechanism

7

Centralizing Error Handling

Centralize the Error Handling

Consider Using Error Err

Include Last-Resort Error Handler

7

Summary

Where to Trap Errors

Writing Error Handlers

7