

# Trapping Run-Time Errors

11

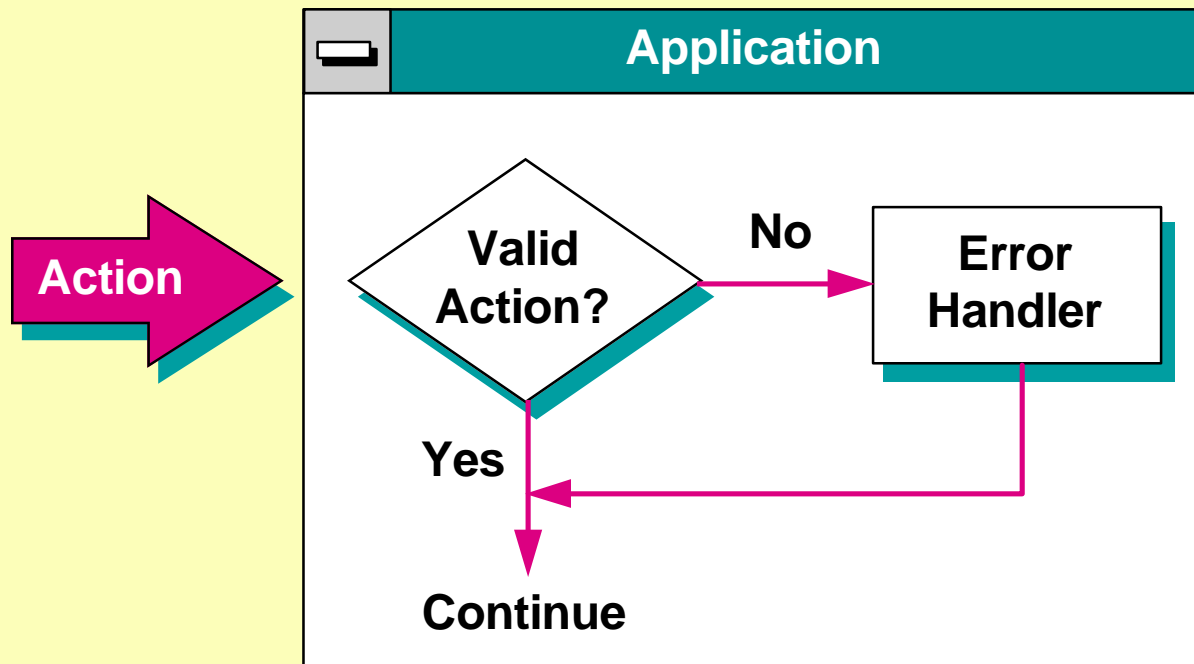
# ◆ Overview

- Where to Trap Errors
- Writing Error Handlers

# Trapping Errors Using Error Handlers

## Error Handlers

- Inform user of error actions
- Request corrective action
- Take corrective action



# Creating Error Handlers

- Enable the Error Trap
- Write the Error-Handling Procedure
- Exit the Error-Handling Procedure

# Enabling the Error Trap

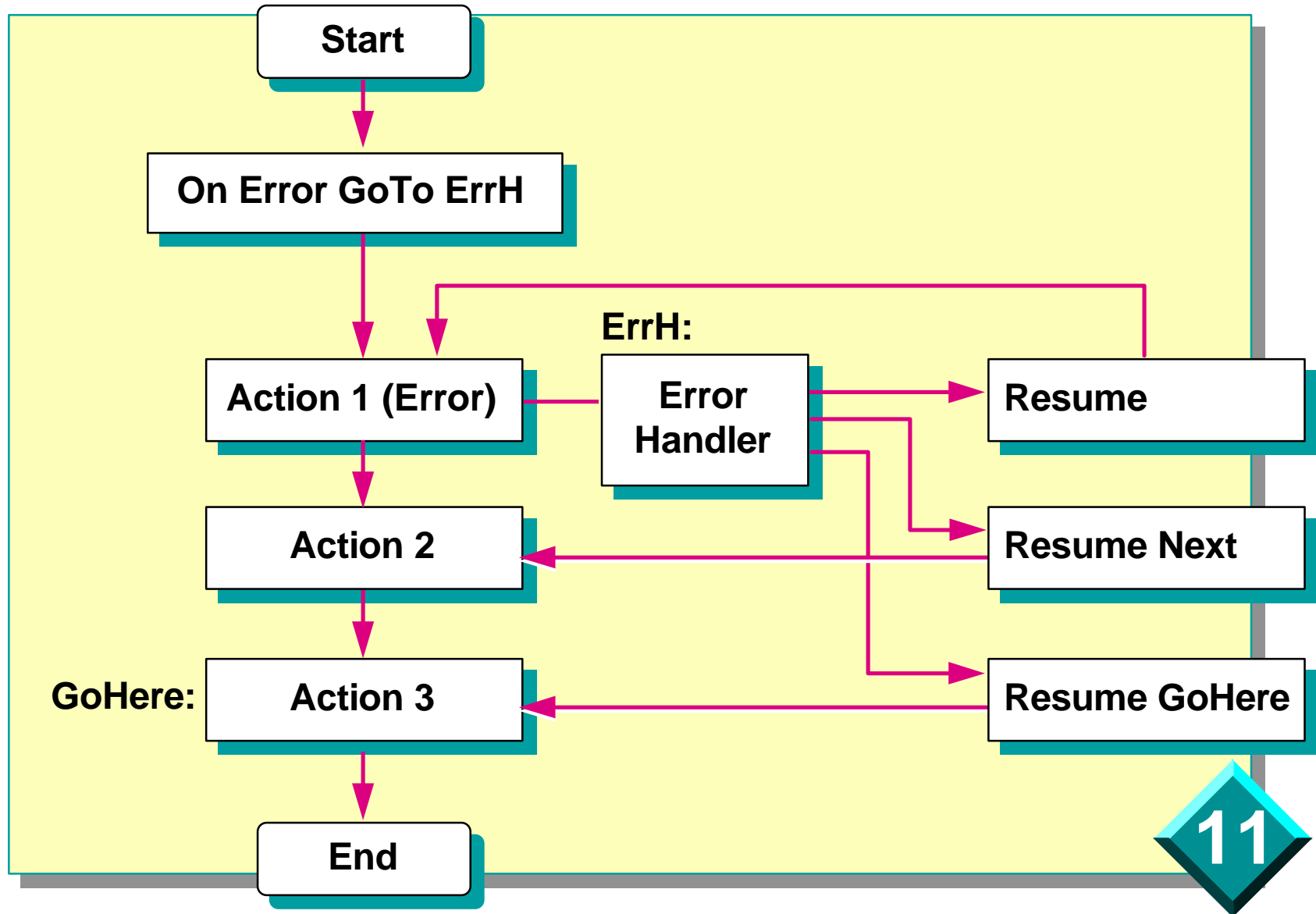
```
On Error GoTo ErrorTrap
.
.
.
Exit Sub
ErrorTrap:
    'Deal with error
End Sub
```

*Procedure*

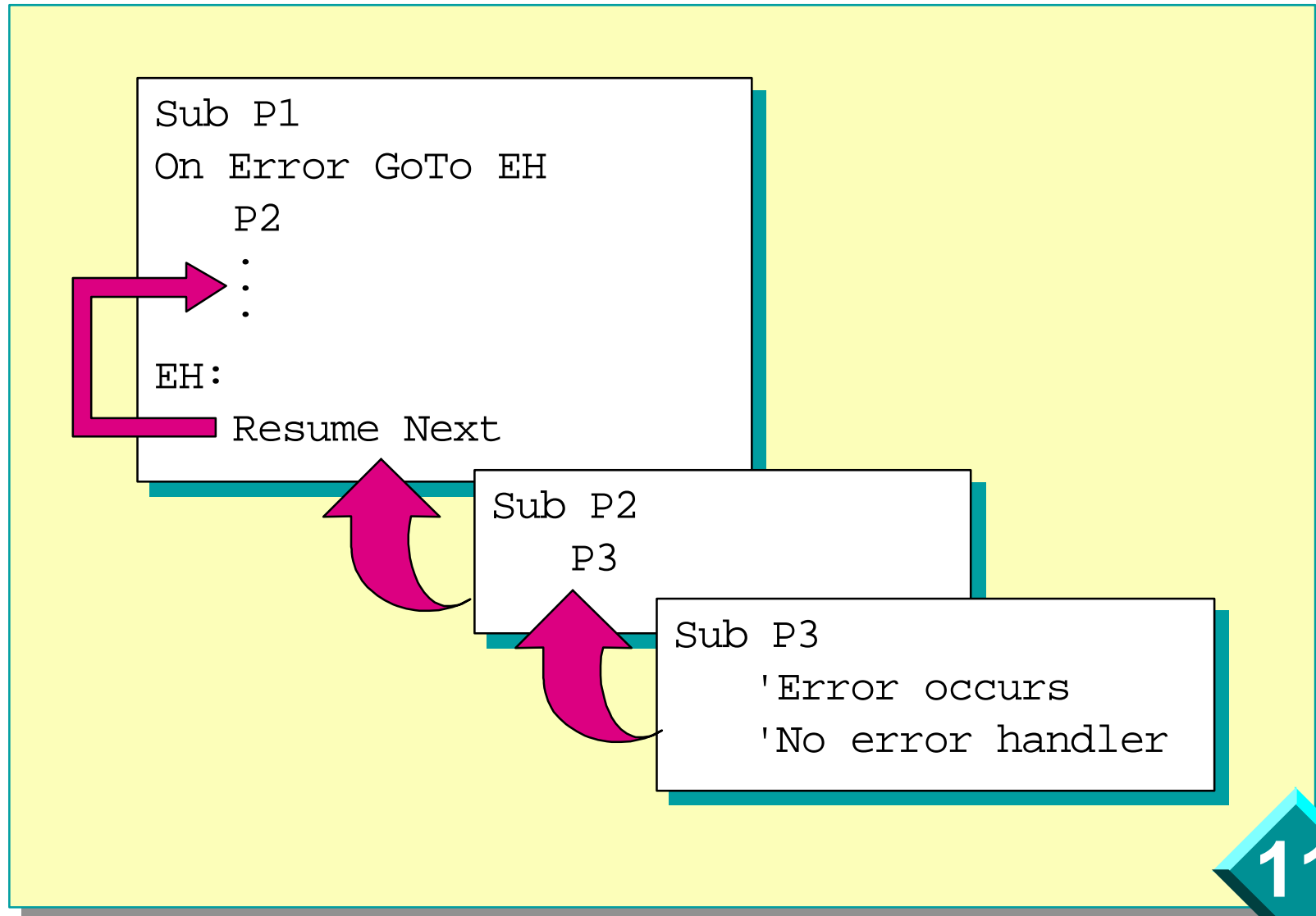
# Writing the Error Handler

- Err - Error Number
- Error\$ - Error Message

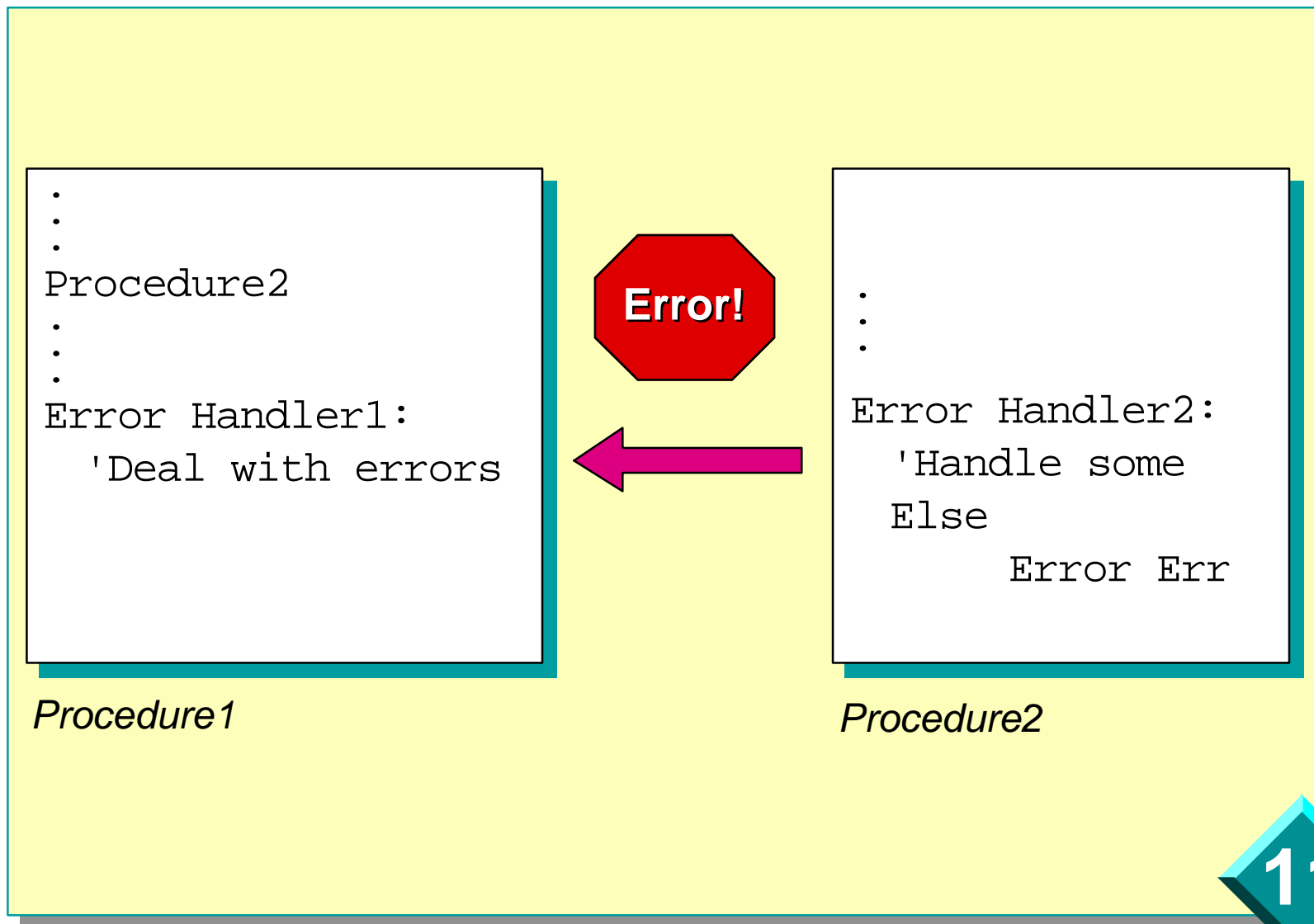
# Exiting the Error Handler



# Errors Automatically Passed to Calling Procedure



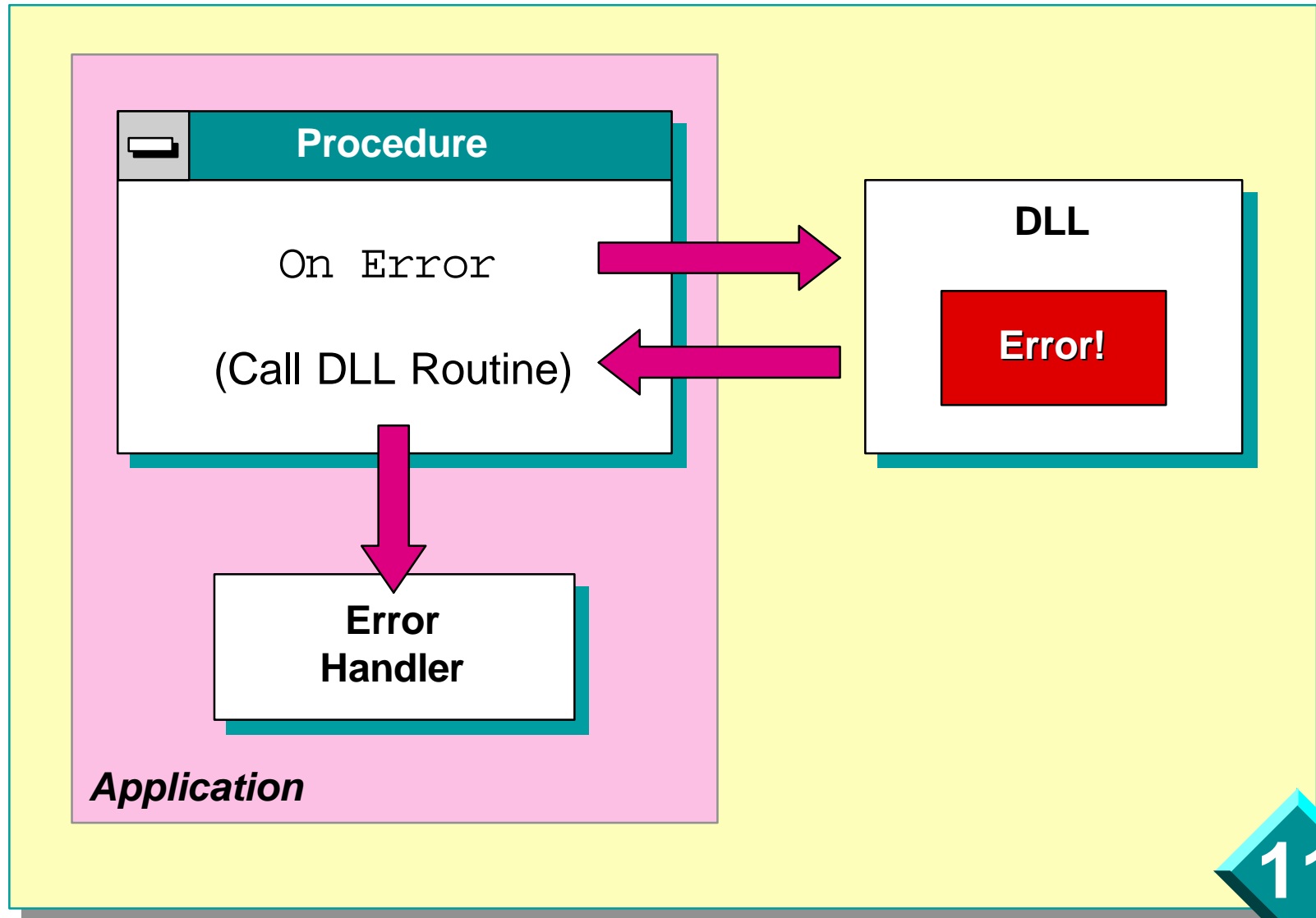
# Passing the Error to the Calling Procedure



# Generating An Error

- **Error** *number*
- Use Error Handling Mechanism For Your Purposes
- Don't Duplicate Existing System Error Numbers

# Trapping Errors in External Procedures



# 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

# Centralizing Error Handling

- Centralize the Error Handling
- Consider Using Error Err
- Include Last-Resort Error Handler

# Summary

- Where to Trap Errors
- Writing Error Handlers