VBA ERROR HANDLING - COMPLETE GUIDE

1. What is Error Handling in VBA?

When VBA runs into a problem (like dividing by zero, wrong file path, or missing sheet), it throws an error and usually **stops execution**.

With Error Handling, we control what happens when an error occurs (instead of letting VBA crash).

2. Error Handling Keywords

■ On Error Resume Next

Ignores the error and continues with the next line.

- Use when the error is not critical
- Execution continues regardless of errors

■ On Error GoTo 0

Turns OFF error handling.

- Any new error will stop the code normally
- Returns to default VBA error behavior

■ On Error GoTo Label

When an error happens → VBA jumps to the line with that **Label**.

- Best for structured error handling
- Provides full control over error response

3. VBA Error Handling Examples

Example 1: Without Error Handling

```
Sub NoErrorHandling()
    Dim x As Integer
    Dim y As Integer
    Dim z As Integer

    x = 10
    y = 0
    z = x / y    ' → Division by zero → VBA will stop with runtime error
    MsgBox z
End Sub
```

Result: VBA shows error and stops execution immediately.

Example 2: Using On Error Resume Next

```
Sub ResumeNextExample()
    Dim x As Integer
    Dim y As Integer
    Dim z As Integer

    x = 10
    y = 0

    On Error Resume Next    ' Ignore the error
    z = x / y               ' Error ignored, z = 0
    On Error GoTo 0         ' Turn off error ignoring

    MsgBox "Result is " & z
End Sub
```

Result: Code continues running. The error is silently ignored, and z becomes 0.

Example 3: Using On Error GoTo Label

```
Sub GoToLabelExample()
    Dim x As Integer
    Dim y As Integer
    Dim z As Integer

    On Error GoTo ErrHandler

    x = 10
    y = 0
    z = x / y    ' Error happens here

    MsgBox "Result is " & z
    Exit Sub

ErrHandler:
    MsgBox "Oops! Error number " & Err.Number & " : " & Err.Description
End Sub
```

Explanation:

- When error happens → jumps to ErrHandler:
- Displays a friendly message instead of crashing
- Err.Number = error code (like 11 for divide by zero)
- Err.Description = human-readable description

Example 4: File Handling Example

```
Sub ErrorHandlingFile()
    Dim filePath As String
    filePath = "C:\NotExistFolder\myfile.xlsx"

    On Error GoTo ErrHandler

    Workbooks.Open filePath

    MsgBox "File opened successfully!"
    Exit Sub
```

```
ErrHandler:
    MsgBox "Error " & Err.Number & ": " & Err.Description
End Sub
```

Result: If file doesn't exist, shows error message instead of crashing. User gets clear feedback about what went wrong.

4. Error Object (Err) Properties

Property	Description	Example
Err.Number	Error code number	11 (Division by zero)
Err.Description	Text description of error	"Division by zero"
Err.Source	Name of object that caused error	"VBAProject"
Err.Clear	Clears all error properties	Err.Clear

5. Best Practices for Error Handling

1. **Always use structured error handling** (On Error GoTo Label)
2. **Avoid leaving On Error Resume Next** without turning it off with On Error GoTo 0
3. **Use Err.Clear** if you want to reset error values after handling
4. **In big projects** → create a global error handler for logging errors
5. **Always use Exit Sub** before error handler to prevent accidental execution
6. **Provide meaningful error messages** to users instead of technical jargon

6. Quick Reference Table

Statement	Purpose
On Error Resume Next	Ignore error, move on
On Error GoTo 0	Turn OFF error handling
On Error GoTo Label	Jump to error handling block
Err.Number	Get error code number
Err.Description	Get error message text
Err.Clear	Reset error object
Exit Sub	Exit procedure before error handler

7. Common VBA Error Numbers

Error Number	Description
6	Overflow
7	Out of memory
9	Subscript out of range
11	Division by zero
13	Type mismatch
53	File not found
91	Object variable not set
1004	Application-defined or object-defined error

Important: *Error handling is essential for creating robust VBA applications. Always anticipate potential errors and handle them gracefully to improve user experience.*