



## Generated Bookmap

Publication ID:

# Course Overview

---

**Unit 1:** Generated Learning Group

# Unit 1: Generated Learning Group

## Lesson 1: Generated Learning Object

# Lesson 1: Generated Learning Object

## Lesson Objectives

After completing this lesson, you will be able to:

- Extend workbook display options with functions and Microsoft Excel VBA

# Lesson Agenda

## Learn About:

- Functions and VBA
- Macros
- Analysis Functions

## Watch:

- How to Create a List and Use Microsoft Excel VBA to Transfer the Value to a Prompt

## Do:

- Create a Dynamic Drop-down List Using Microsoft Excel VBA
- Create a Button to Transfer Selections Using Microsoft Excel VBA
- Using Microsoft Excel VBA with Check Boxes for Filtering (Optional)

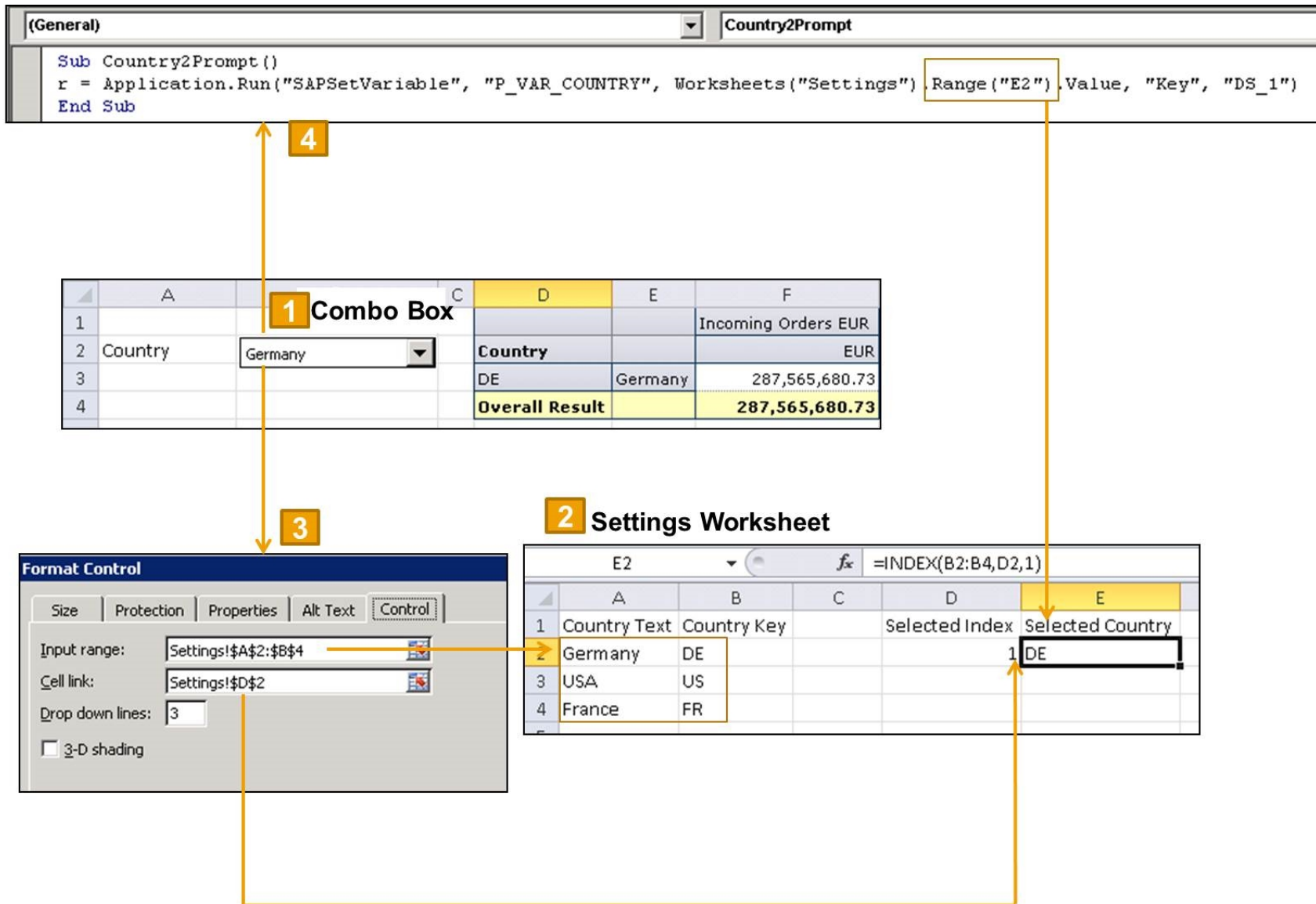
## API Methods in Analysis

- **SAPAddMessage:** Define messages and add them to the standard message dialog
- **SAPCallMemberSelector:** Call the standard input help (member selector) from the worksheet
- **SAPExecuteCommand:** Use this API method as a generic command to execute commands such as hide/show ribbon components, refresh, and so on.
- **SAPExecutePlanningFunction:** Execute Integrated Planning functions (See the BOAN20 class)
- **SAPExecutePlanningSequence:** Execute Integrated Planning sequences
- **SAPGetCellInfo:** Define a command to get information on a crosstab cell such as dimensions and selections
- **SAPGetProperty:** Read properties of a data source or a workbook such as whether the data source is input ready, for example
- **SAPLogon:** Trigger a logon to a system for a specified data source

## API Methods in Analysis

- **SAPMoveDimension:** Define the position of a dimension in the crosstab, swap the dimension with another one or position a dimension relative to another one.
- **SAPSetFilter:** Define which members of a dimension should be filtered.
- **SAPSetRefreshBehaviour:** Use different API methods one after the other without the result set being refreshed and without the message dialog being shown
- **SAPSetVariable:** Define values for input-ready BW variables (prompts). If you want to set multiple variables, you can use the `PauseVariableSubmit` command with the `SAPExecuteCommand` method.

# Figure 1: How to Use the SAPSetVariable API





## Figure 2: How to Use the SAPSetRefreshBehaviour API

```

(General) Transfer_Selection
Sub Transfer_Selection()
r = Application.Run("SAPSetRefreshBehaviour", "Off")
r = Application.Run("SAPSetVariable", "P_VAR_COUNTRY", Worksheets("Settings").Range("E2").Value, "Key", "DS_1")
r = Application.Run("SAPSetFilter", "DS_1", "ODIVISION", Worksheets("Settings").Range("E8").Value, "Key")
r = Application.Run("SAPSetRefreshBehaviour", "On")
End Sub
    
```

4

	A	B	C	D	E	F
1						Incoming Orders EUR
2	Country	USA	Country			EUR
3			US	United States		17,486,150.98
4	Division	High Tech	Overall Result			17,486,150.98
5						
6		Transfer Selections				
7						

1

### 2 Settings Worksheet

E8 =INDEX(B8:B10,D8,1)

	A	B	C	D	E
7	Division Text	Division Key	Selected Index	Selected Division	
8	Pumps	01	2	07	
9	High Tech	07			
10	Service	08			

3

**Format Control**

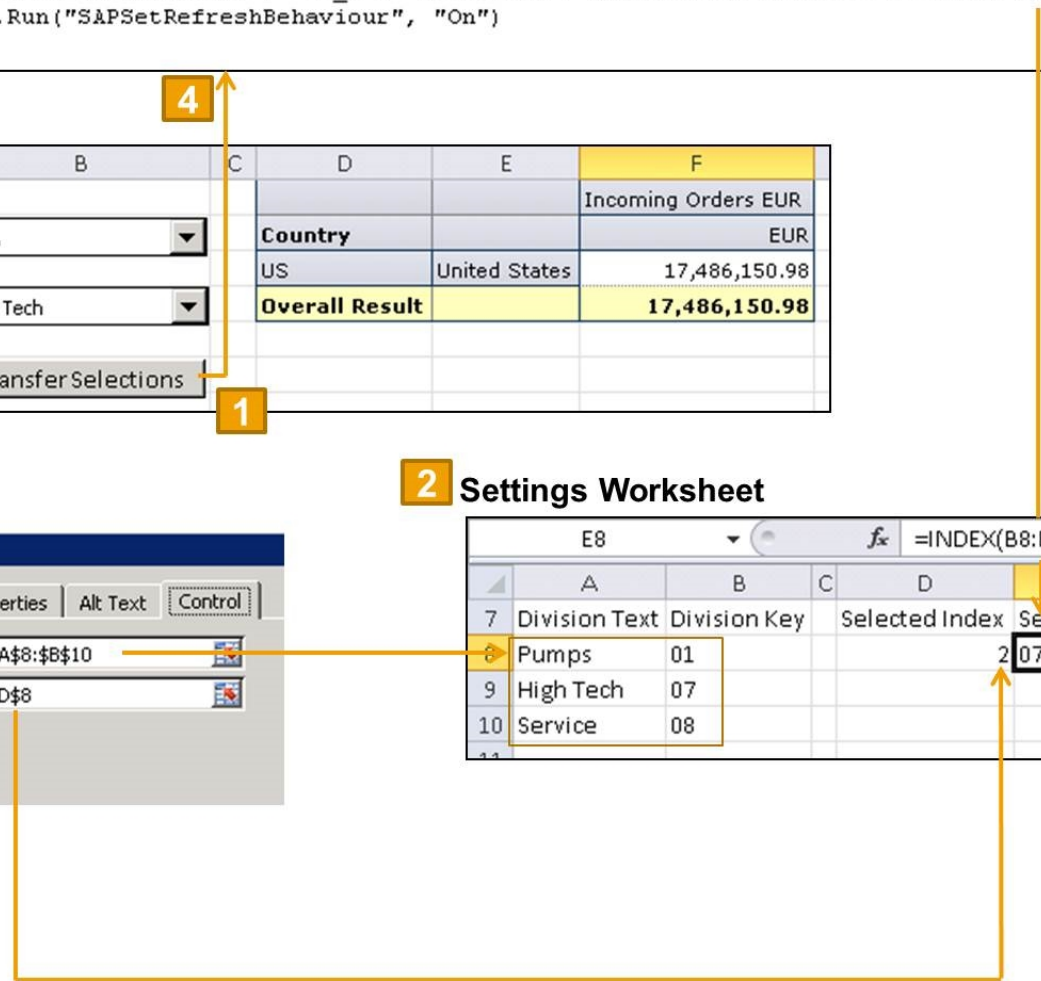
Size Protection Properties Alt Text Control

Input range: Settings!\$A\$8:\$B\$10

Cell link: Settings!\$D\$8

Drop down lines: 3

3-D shading



(General) Create\_Filter

```

Sub Create_Filter()
Dim Act_true As Integer
Dim Number_true As Integer
Dim Filter_Value As String
Number_true = Worksheets("Settings").Range("C9").Value
For i = 4 To 7
  If Worksheets("Settings").Cells(i, 3).Value = "True" Then
    Act_true = Act_true + 1
    Filter_Value = Filter_Value & Worksheets("Settings").Cells(i, 2).Value
    If Act_true < Number_true Then Filter_Value = Filter_Value & ";"
  End If
Next
ret = Application.Run("SAPSetFilter", "DS_1", "OSALESORG", Filter_Value, "Input_String")
End Sub
        
```

**4**

**Format Control**

Colors and Lines | Size | Protection

Value

Unchecked **3**

Checked

Mixed

Cell link: Settings!\$C\$4

**2** Settings Worksheet

	A	B	C	D	E	F
3		<b>Sales Org Key</b>	<b>Selected</b>			
4	Frankfurt	1000	TRUE			
5	Denver	2400	TRUE			
6	Rotterdm	2500	FALSE			
7	Milan	3020	FALSE			
8						
9						<b>2</b>
10						

	A	B	C	D	E	F
1	<input checked="" type="checkbox"/> Frankfurt <b>1</b>					Sales Volume EUR
2	<input checked="" type="checkbox"/> Denver			<b>Sales Organization</b>		EUR
3	<input type="checkbox"/> Rotterdm		1000	Germany Frankfurt		14,217,648.73
4	<input type="checkbox"/> Milan		2400	Italy, Milan		10,785,700.00
5			<b>Overall Result</b>			<b>25,003,348.73</b>
6						
7						



# Summary

You should now be able to:

- Extend workbook display options with functions and Microsoft Excel VBA