

Blue Open Studio Recipes – MS Access, Grid Sample Project V2 Extended Technical Description

The purpose of this example project is to provide a simple way to manage recipes for small scale recipe projects.

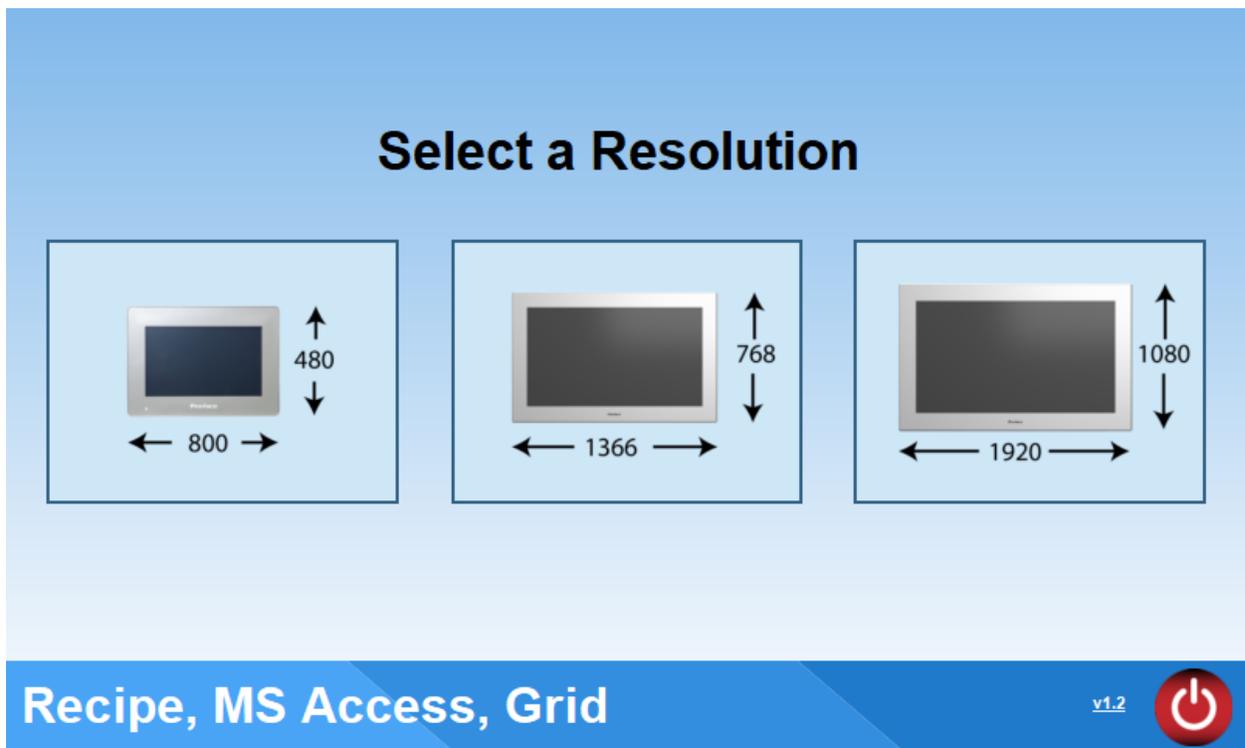
The Project uses an Access Database to contain the Recipes in a single table. You can edit the Recipes in Access or in the BOS Runtime project.

This document will show you how to use the project, how to integrate the project to your use, how the project works and how you can add ingredients to the built in recipes.

Section 1 -How to use project

Running project from local build time.

Opening Screen



Select a desired Resolution. This will check to make sure the current monitor can accept the selected resolution and then change screen to Recipe1_x where x equals selected resolution.

Recipe

Send and Receive

RecipeName
Recipe1
Recipe2
Recipe3

Filter:

Sort: **None** ▾

Send
Receive

Preview

Recipe Name	MyBoolean	MyInteger

PLC Values

MyBoolean:

MyInteger:

MyReal:

MyString:

Recipe, MS Access, Grid

v1.2

This is the Recipe Send and Receive Screen.
 This screen is for Sending and Receiving Recipes between the Access database and sudo-PLC variables.

When selecting a Recipe in the RecipeName Grid.

RecipeName
Recipe1
Recipe2
Recipe3

Will populate the Preview with the Recipe values in the database.

Recipe

Send and Receive

RecipeName
Recipe1
Recipe2
Recipe3

Filter:

Sort:



Preview

MyBoolean	MyInteger	MyReal
<input checked="" type="checkbox"/>	123	1.23

You can also filter and sort the RecipeName Grid. Clicking on the broom icon clears the filter and the ? gives a window showing the different wildcards possible for the filter. The Cycle button refreshes the grid with the filter applied.

Recipe

Send and Receive

RecipeName
Recipe2

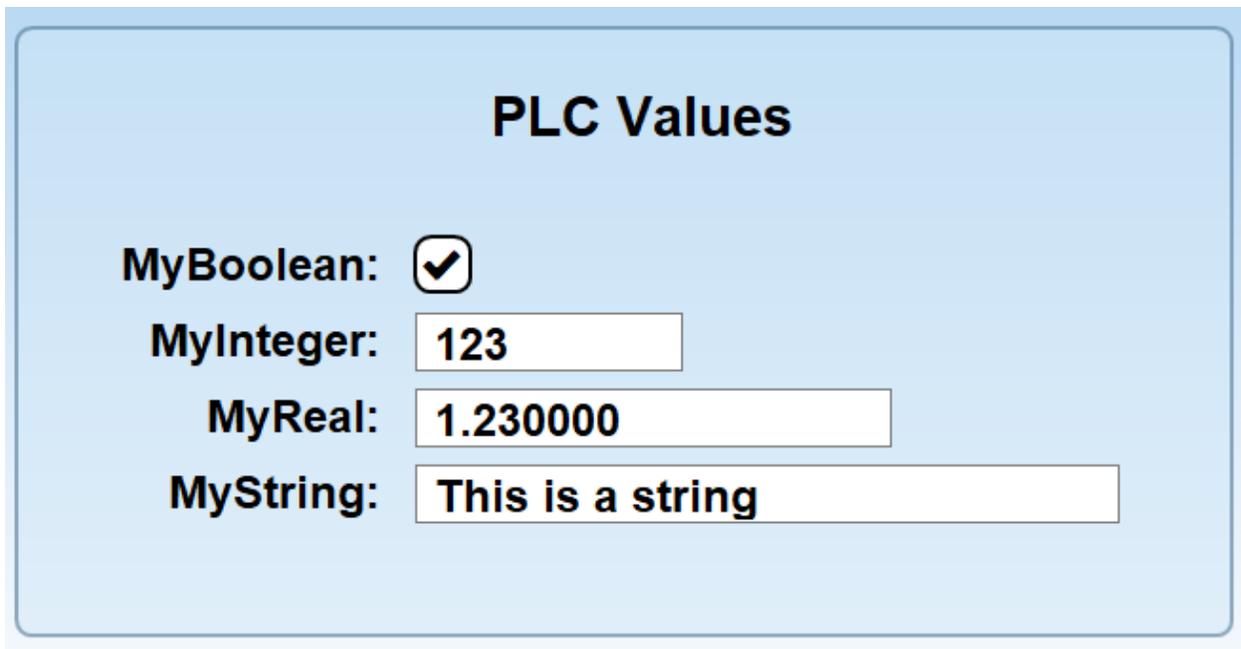
Filter:



Sort:



The PLC Values Section allow you to enter and view the sudo-PLC values.



Sending and Receiving Recipes to and from the sudo-PLC Values.

There are two objects on the screen.



Send uses the DBSelect Command to read up the values of the selected recipe from the database and transfers it to the sudo-PLC variables.

Receive uses the DBExecute Command to read the values of the sudo-PLC variables and writes them to the selected recipe in the database.

Title, Navigation and Version bar.



Navigates to a Recipe Edit Screen. RecipeEdit_x x is selected resolution.



Navigates back to the Resolution Selection Screen.



Exits the runtime viewer.

Recipe Edit Screen.

Recipe
Edit

  Filter:   Sort: **ASC**

Recipe Name	MyBoolean	MyInteger	MyReal	MyString
Recipe1	<input checked="" type="checkbox"/>	123	1.230	String1
Recipe2	<input type="checkbox"/>	456	4.560	String2
Recipe3	<input checked="" type="checkbox"/>	789	7.890	String3

Recipe Name:

Recipe, MS Access, Grid   v1.2 

This screen allows you to edit and add recipes to the recipe database.

When you enter new values in to the selected Recipe parameter it will change the value in the database file.

Recipe Name	MyBoolean	MyInteger	MyReal	MyString
Recipe1	<input checked="" type="checkbox"/>	0	1.230	String1
Recipe2	<input type="checkbox"/>	456	4.560	String2
Recipe3	<input checked="" type="checkbox"/>	789	7.890	String3
Test	<input checked="" type="checkbox"/>	444	4.440	ijklm

In the Create Recipe Area two options are available.

Recipe Name	MyBoolean	MyInteger	MyReal	MyString
Recipe1	<input checked="" type="checkbox"/>	0	1.230	String1
Recipe2	<input type="checkbox"/>	456	4.560	String2
Recipe3	<input checked="" type="checkbox"/>	789	7.890	String3
Test	<input checked="" type="checkbox"/>	444	4.440	ijklm

Create New Recipe

▼

Recipe Name:

Create

Create New Recipe

▼

Duplicate Recipe

Create New Recipe will create a new recipe with the name entered and zero values.

Recipe Name	MyBoolean	MyInteger	MyReal	MyString
Recipe1	<input checked="" type="checkbox"/>	0	1.230	String1
Recipe2	<input type="checkbox"/>	456	4.560	String2
Recipe3	<input checked="" type="checkbox"/>	789	7.890	String3
Test	<input checked="" type="checkbox"/>	444	4.440	ijklm
Recipe4	<input type="checkbox"/>	0	0.000	

Create New Recipe

▼

Recipe Name:

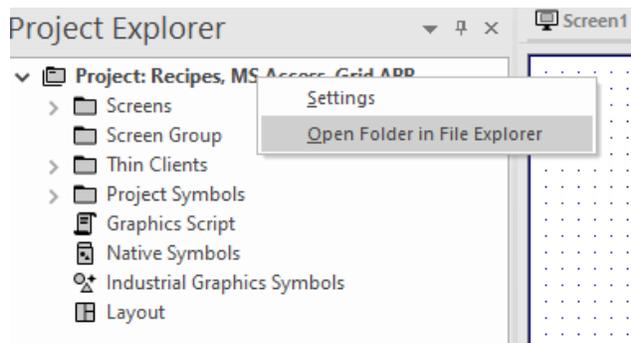
Create

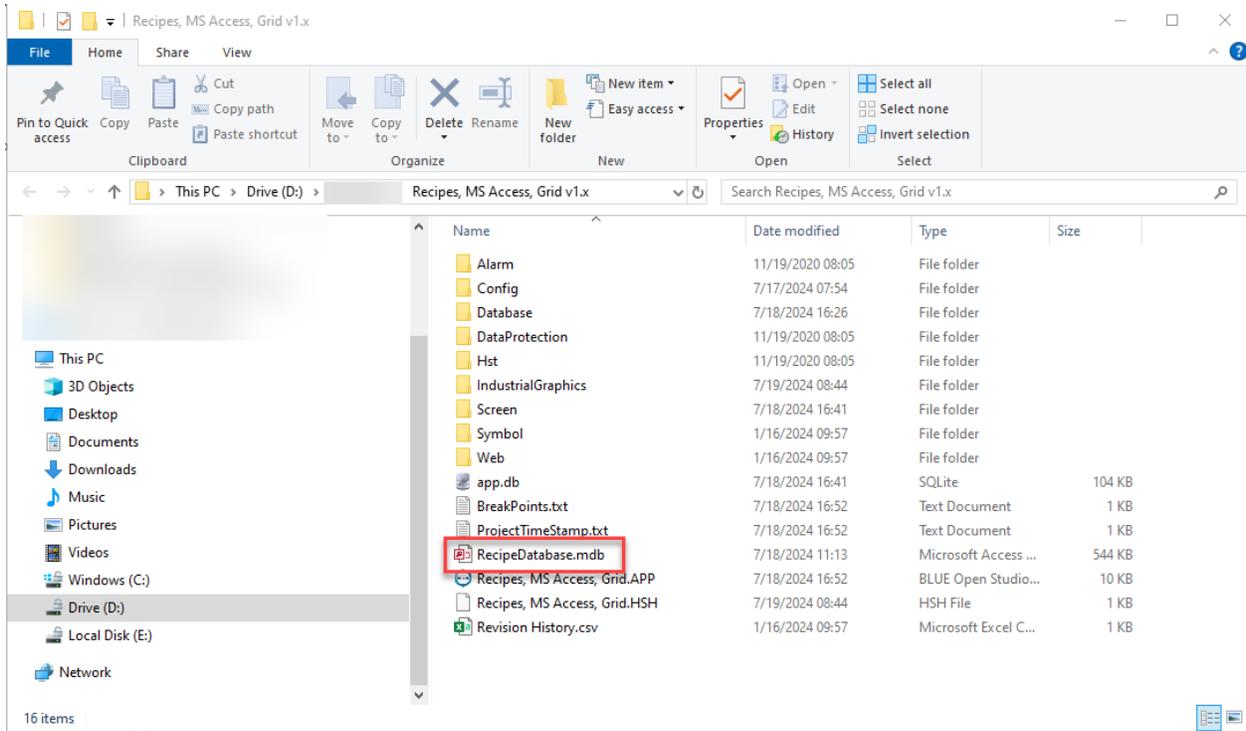
Duplicate Recipe creates a new recipe with the name entered and the value duplicated from the selected recipe in the grid.

Recipe Name	MyBoolean	MyInteger	MyReal	MyString
Recipe1	<input checked="" type="checkbox"/>	0	1.230	String1
Recipe2	<input type="checkbox"/>	456	4.560	String2
Recipe3	<input checked="" type="checkbox"/>	789	7.890	String3
Test	<input checked="" type="checkbox"/>	444	4.440	ijklm
Recipe4	<input type="checkbox"/>	0	0.000	
Test2	<input checked="" type="checkbox"/>	444	4.440	ijklm

Section 2 -Using Access to add Recipes and modify Recipe Values.

This project includes a MS Access Database file. The file(s) are within the BOS Project Folder. Right click on the project name in the BOS build time environment to access the project folder directly.





In Access open the Recipes Table to add and edit Recipes.

The screenshot shows the Microsoft Access interface. On the left, the 'All Access Objects' pane is open, showing 'Tables' with 'ParameterTable' and 'Recipes' listed. The 'Recipes' table is selected. The main window displays the data table for 'Recipes':

ID	RecipeName	MyBoolean	MyInteger	MyReal	MyString	Click to Add
1	Recipe1	1	0	1.23	String1	
2	Recipe2	0	456	4.56	String2	
3	Recipe3	1	789	7.89	String3	
40	Test	1	444	4.44	ijklm	
41	Recipe4	0	0	0		
43	Test2	1	444	4.44	ijklm	
*	(New)	0	0	0		

From the Recipe Edit Screen in the runtime project you can see the new values entered into the database table by clicking the refresh button.

The screenshot shows the 'Recipe Edit' screen. At the top left, there is a refresh button (circular arrow icon) highlighted with a red box. To its right is a trash icon, a filter input field, a help icon, and a sort dropdown menu set to 'None'. Below this is a table with the following data:

Recipe Name	MyBoolean	MyInteger	MyReal	MyString
Recipe1	<input checked="" type="checkbox"/>	6	1.230	String1
Recipe2	<input type="checkbox"/>	456	4.560	String2

Section 3 - To use this project and build your own project around it, in the communication driver sheet simply add the "PLC" tags to the sheet and associate them with the correct addresses.

9	Selected_Recipe	0	Integer	Used to select
10	PLC_MyBoolean	0	Boolean	PLC tag
11	PLC_MyInteger	0	Integer	PLC tag
12	PLC_MyReal	0	Real	PLC tag
13	PLC_MyString	0	String	PLC tag

Example for Modbus.

Project Tags MOTCP - MAIN DRIVER SHEET x

Description:
MAIN DRIVER SHEET

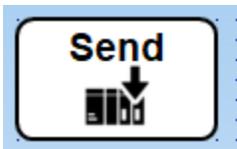
Disable:

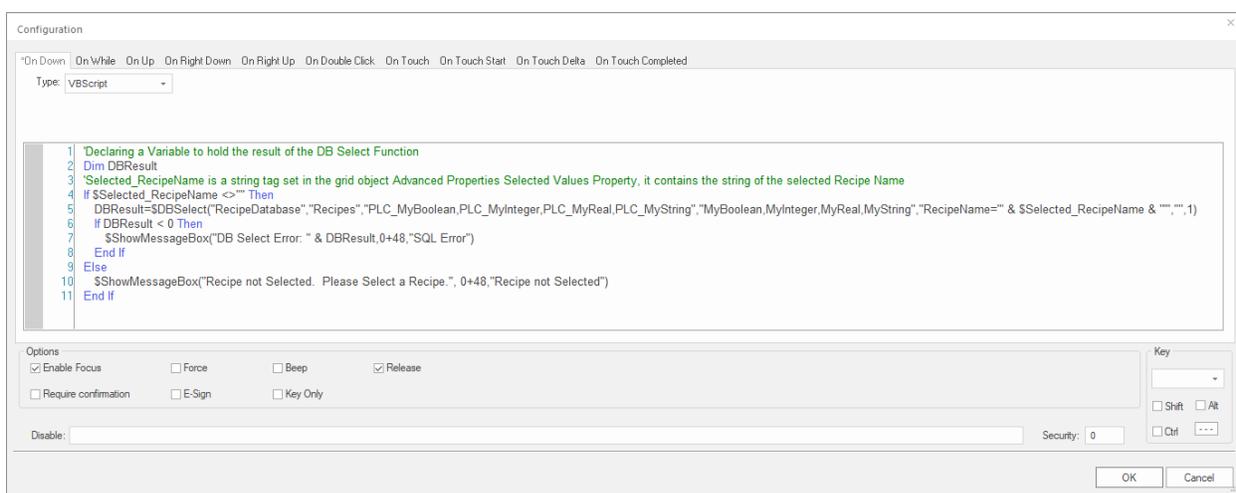
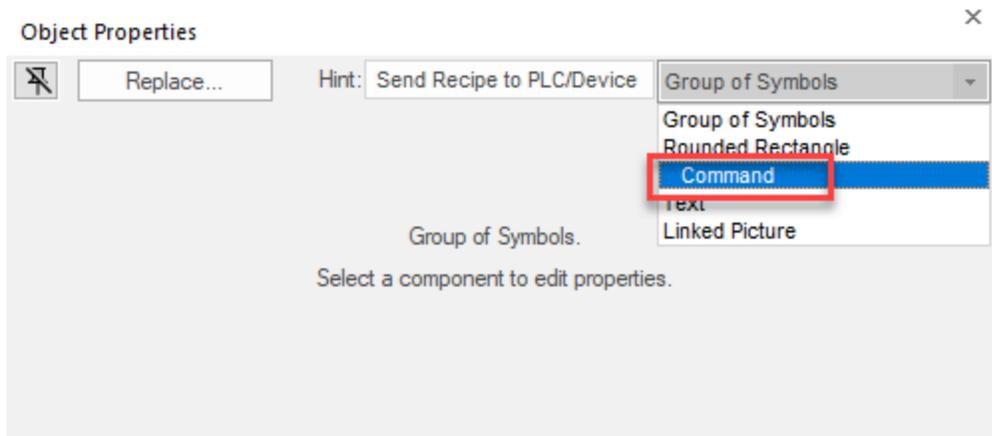
Read Completed: Read Status: Min: Max:

Write Completed: Write Status: Min: Max:

	Tag Name	Station	I/O Address	Action	Scan	Div	Add
	Filter text	Filter text	Filter text	(All)	(All)	Filter text	Filter text
1	PLC_MyBoolean	127.0.0.1:502:1	0X:1	Read+Write	Always		
2	PLC_MyInteger	127.0.0.1:502:1	4X:1	Read+Write	Always		
3	PLC_MyReal	127.0.0.1:502:1	4X:2	Read+Write	Always		
4	PLC_MyString	127.0.0.1:502:1	4X:4	Read+Write	Always		
*				Read+Write	Always		
*				Read+Write	Always		
*				Read+Write	Always		
*				Read+Write	Always		
*				Read+Write	Always		

Section 4 -How the Recipe Write and Recipe Read Functions.





Code:

```
'Declaring a Variable to hold the result of the DB Select Function
Dim DBResult
'Selected_RecipeName is a string tag set in the grid object Advanced Properties Selected Values
Property, it contains the string of the selected Recipe Name
If $Selected_RecipeName <>"" Then
    DBResult=$DBSelect("RecipeDatabase","Recipes","PLC_MyBoolean,PLC_MyInteger,PLC_MyR
eal,PLC_MyString","MyBoolean,MyInteger,MyReal,MyString","RecipeName=" & $Selected_RecipeName
& """, "", 1)
    If DBResult < 0 Then
        $ShowMessageBox("DB Select Error: " & DBResult,0+48,"SQL Error")
    End If
Else
    $ShowMessageBox("Recipe not Selected. Please Select a Recipe.", 0+48,"Recipe not
Selected")
End If
```

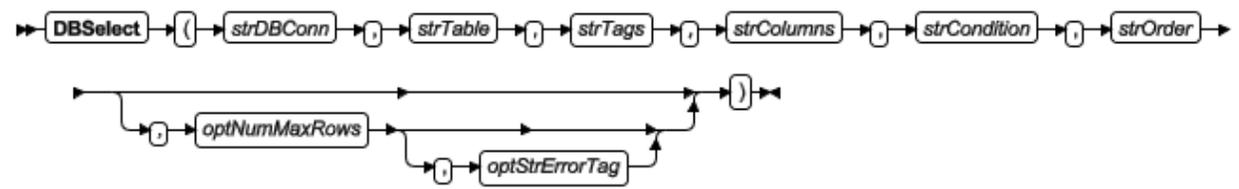
DBSelect Function

DBSelect is a built-in function that selects a result set from an external database (equivalent to a SQL SELECT statement), maps the columns of the result set to array tags in your project, and then copies the values from the result set to the array tags.

Function	Group	Execution	String Exp.	Windows	HMI Runtime	Thin Clients	Mobile Access
DBSelect	Database/ERP	Synchronous	No	Supported	Not supported	Supported	Supported

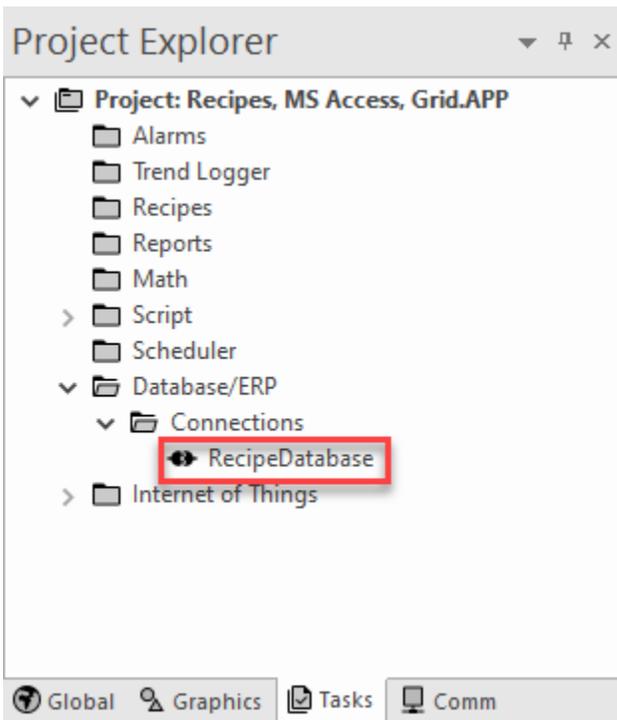
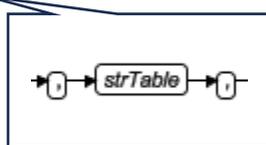
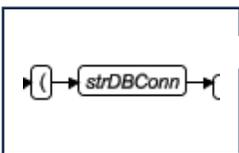
Syntax

```
DBSelect(strDBConn, strTable, strTags, strColumns, strCondition, strOrder, optNumMaxRows, optStrErrorTag)
```



Expanded Function

\$DBSelect("RecipeDatabase", "Recipes",



ID	RecipeName
1	Recipe1
2	Recipe2
3	Recipe3
40	Test
41	Recipe4
43	Test2
*	(New)



"PLC_MyBoolean,PLC_MyInteger,PLC_MyReal,PLC_MyString"

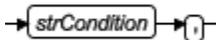


,"MyBoolean,MyInteger,MyReal,MyString"

strCondition

A statement specifying which rows in the database table or view to select. This is equivalent to the SQL WHERE clause and must follow the same syntax.

To select all the rows in the table or view, specify an empty string ("") for this parameter.



'RecipeName is Column (FROM) Selected_RecipeName is (WHERE) Variable holding selected row

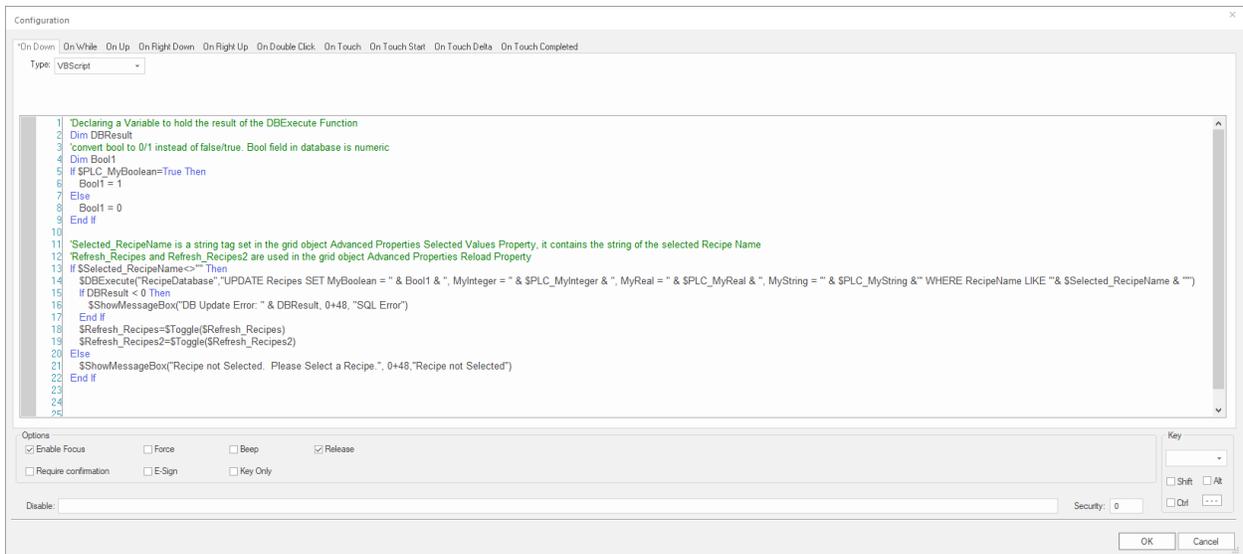
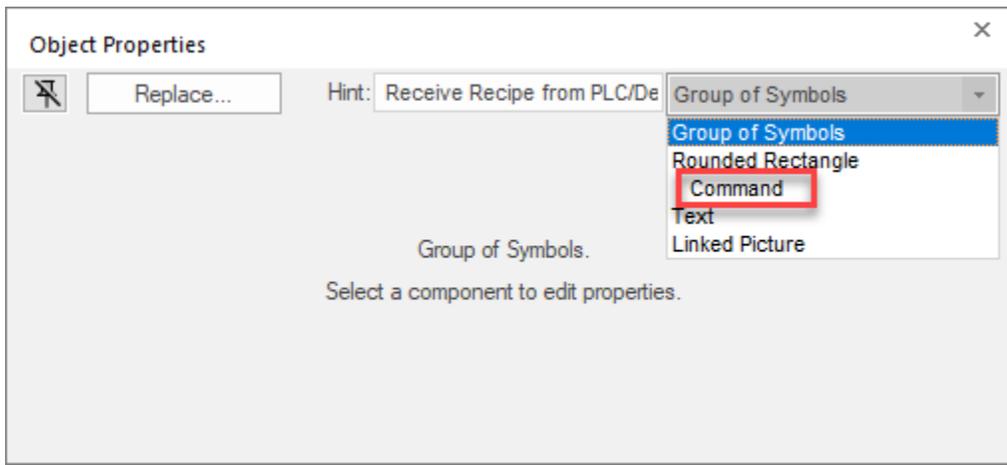
"RecipeName="" & \$Selected_RecipeName & """,



"" ,



1)



Code:

Declaring a Variable to hold the result of the DBExecute Function

```
Dim DBResult
```

'convert bool to 0/1 instead of false/true. Bool field in database is numeric

```
Dim Bool1
```

```
If $PLC_MyBoolean=True Then
```

```
    Bool1 = 1
```

```
Else
```

```
    Bool1 = 0
```

```
End If
```

'Selected_RecipeName is a string tag set in the grid object Advanced Properties Selected Values Property, it contains the string of the selected Recipe Name

'Refresh_Recipes and Refresh_Recipes2 are used in the grid object Advanced Properties Reload Property

```
If $Selected_RecipeName<>"" Then
```

```
    $DBExecute("RecipeDatabase","UPDATE Recipes SET MyBoolean = " & Bool1 & ", MyInteger = " & $PLC_MyInteger & ", MyReal = " & $PLC_MyReal & ", MyString = " & $PLC_MyString & " WHERE RecipeName LIKE "" & $Selected_RecipeName & """)
```

```
    If DBResult < 0 Then
```

```
        $ShowMessageBox("DB Update Error: " & DBResult, 0+48, "SQL Error")
```

```
    End If
```

```
    $Refresh_Recipes=$Toggle($Refresh_Recipes)
```

```
    $Refresh_Recipes2=$Toggle($Refresh_Recipes2)
```

```
Else
```

```
    $ShowMessageBox("Recipe not Selected. Please Select a Recipe.", 0+48,"Recipe not Selected")
```

```
End If
```

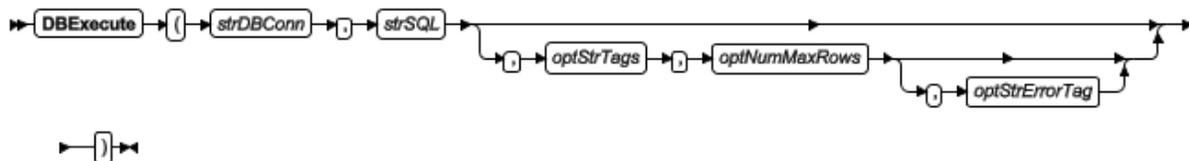
DBExecute Function

DBExecute is a built-in scripting function that executes a custom SQL statement on an external database. If the statement is a query (e.g., SELECT), the database values are copied to specified array tags.

Syntax

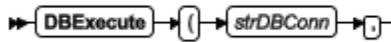
Syntax

```
DBExecute (strDBConn, strSQL, optStrTags, optNumMaxRows, optStrErrorTag)
```

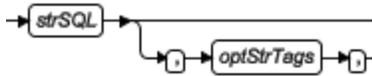


Expanded Function

\$DBExecute("RecipeDatabase",



"UPDATE Recipes SET MyBoolean = " & Bool1 & ", MyInteger = " & \$PLC_MyInteger & ", MyReal = " & \$PLC_MyReal & ", MyString = "" & \$PLC_MyString &"" WHERE RecipeName LIKE ""& \$Selected_RecipeName & "



UPDATE SQL Statement = Modify Records in a Table

Section 5 -Adding Ingredients / Variables to the Recipes

In the Access Table View and the Recipes table open add additional columns (ingredients).

Datasheet View:

ID	RecipeName	MyBoolean	MyInteger	MyReal	MyString	MyBoolean2	MyInteger2	MyReal2	MyString2	Click to Add
1	Recipe1	1	123	1.23	String1	456				
2	Recipe2	0	456	4.56	String2	789				
3	Recipe3	1	789	7.89	String3	589				
40	NewRecipeTom	0	110	1.2	String 4	369				
*	(New)	0	0	0		0	0	0		

Design View (right-click on Tab)

Field Name	Data Type
ID	AutoNumber
RecipeName	Long Text
MyBoolean	Number
MyInteger	Number
MyReal	Number
MyString	Short Text
MyBoolean2	Number
MyInteger2	Number
MyReal2	Number
MyString2	Short Text

Save the table.

Add variables to Project.

	Name	Array Size	Type
	PLC*2	Filter text	(All)
20	PLC_MyBool2	0	Boolean
18	PLC_MyInteger2	0	Integer
21	PLC_MyReal2	0	Real
22	PLC_MyString2	0	String
*			Integer

Need to add the correct Field Names and Variable Names to the DB commands, paying close attention to the alignment of the Fields versus Variable and proper use of quotes and comma separators.

Completed commands with additional ingredients.

Send – DBSelect

Declaring a Variable to hold the result of the DB Select Function

Dim DBResult

'Selected_RecipeName is a string tag set in the grid object Advanced Properties Selected Values Property, it contains the string of the selected Recipe Name

If \$Selected_RecipeName <>"" Then

DBResult=\$DBSelect("RecipeDatabase","Recipes","PLC_MyBoolean,PLC_MyInteger,PLC_MyReal,PLC_MyString,PLC_MyBool2,PLC_MyInteger2,PLC_MyReal2,PLC_MyString2","MyBoolean,MyInteger,MyReal,MyString,MyBoolean2,MyInteger2,MyReal2,MyString2","RecipeName=" & \$Selected_RecipeName & """,",",1)

If DBResult < 0 Then

\$ShowMessageBox("DB Select Error: " & DBResult,0+48,"SQL Error")

End If

Else

\$ShowMessageBox("Recipe not Selected. Please Select a Recipe.", 0+48,"Recipe not Selected")

End If

Receive – DBExecute

Declaring a Variable to hold the result of the DBExecute Function

Dim DBResult

'convert bool to 0/1 instead of false/true. Bool field in database is numeric

Dim Bool1, Bool2

If \$PLC_MyBoolean=True Then

Bool1 = 1

Else

Bool1 = 0

End If

```
If $PLC_MyBool2=True Then
```

```
    Bool2 = 1
```

```
Else
```

```
    Bool2 = 0
```

```
End If
```

'Selected_RecipeName is a string tag set in the grid object Advanced Properties Selected Values Property, it contains the string of the selected Recipe Name

'Refresh_Recipes and Refresh_Recipes2 are used in the grid object Advanced Properties Reload Property

```
If $Selected_RecipeName<>"" Then
```

```
    $DBExecute("RecipeDatabase","UPDATE Recipes SET MyBoolean = " & Bool1 & ", MyBoolean2 = " & Bool2 & ", MyInteger = " & $PLC_MyInteger & ", MyInteger2 = " & $PLC_MyInteger2 & ", MyReal = " & $PLC_MyReal & ", MyReal2 = " & $PLC_MyReal2 & ", MyString = " & $PLC_MyString & " ', MyString2 = " & $PLC_MyString2 & " WHERE RecipeName LIKE ""& $Selected_RecipeName & """)
```

```
    If DBResult < 0 Then
```

```
        $ShowMessageBox("DB Update Error: " & DBResult, 0+48, "SQL Error")
```

```
    End If
```

```
    $Refresh_Recipes=$Toggle($Refresh_Recipes)
```

```
    $Refresh_Recipes2=$Toggle($Refresh_Recipes2)
```

```
Else
```

```
    $ShowMessageBox("Recipe not Selected. Please Select a Recipe.", 0+48,"Recipe not Selected")
```

```
End If
```