

Getting Started With the Buddy System



Before you can use The Buddy System, you will need to tell the VB Add-In Manager to add it to the Add-Ins menu and select the appropriate Buddy:

1. From the VB IDE, select **Add-Ins|Add-In Manager...**
2. Click to check the box next to **The Buddy System** and click **OK**.
3. For the RAD-Copy Buddy, open the form or class to which you want to RAD-Copy. For the Classy Form Buddy, open the class for which you want to create an interface form.
4. From the VB IDE, select **Add-Ins|The Buddy System...** and select the appropriate Buddy.
5. If you select the RAD-Copy Buddy, you will see the **Open Source Form or Class** dialog. Select the Form or Class module containing the controls and code that you want to RAD-copy.

Additional Information:

[What is a RAD-Copy?](#)

[RAD-Copying Code](#)

[RAD-Copying Controls](#)

[Creating and Customizing Templates](#)

[Using the Classy Form Buddy](#)

[About The RAD-Copy Buddy](#)

Using the Classy Form Buddy

To generate an interface form for an existing class module, perform the following:

1. Use the arrows to select all properties and methods that you want the Classy Form Buddy to include on the form.
2. Click **Generate**.
3. Save the form. You will also be prompted to add the form to the current project.

Note: If a property or method of your class does not appear in the appropriate list, it either returns or takes as an argument an unsupported data type, or is a property with a property procedure that takes arguments. To implement controls for these methods and properties, you will need to create a wrapper function that takes and returns only basic data types.

What is a RAD-Copy ?

A RAD-Copy is any of the following:

1. Copying a Control from one form to another, and bringing all Procedures and Variables whose names contain the control's name to the target form. For example, copying btnPushMe would also copy the entire btnPushMe_Click procedure and a variable name btnPushMeCount in the declarations section.
2. Renaming a Control at copy-time, which renames all references to the control in the target form. With the Big Buddies, a RAD-Copy will also copy all child controls (applies to Menus, Frames, Picture Boxes and some OCX controls), as well as all of their variables and procedures. Create your own reusable Custom Controls completely within Visual Basic
3. For Class Modules, a RAD-Copy inserts statements from an existing procedure into a target procedure, at the exact locations you specify, replacing strings of text that you mark with values that you specify at copy-time

RAD-Copying Code

You can copy code from an existing Form or Class Module using the methods below. For more information on giving your Buddy a work-out with advanced formatting and re-use options, see [Creating Templates](#).

Merging Code Into an Existing Procedure

1. In the **Replace Tags With Values** grid, click on tags to enter the values that you want each occurrence of the tag to be replaced with in the target procedure.
2. In the **Merge Into Procedure** combo box, select the procedure that you want the code copied into.
3. In the **RAD-Copy This Object** combo box, select the procedure containing the code that you want to copy.
4. Drag the selected procedure onto the target procedure.
5. You should see the inserted code in the **Merge Into Procedure** listing.
6. When you have finished RAD-Copying, click **Save and Return** to save your changes. You can also click **Save As** to save the updated module with a different file name.

Inserting a Procedure into a Module

1. In the **Replace Tags With Values** grid, click on tags to enter the values that you want each occurrence of the tag to be replaced with in the target procedure.
2. In the **RAD-Copy This Object** combo box, select the procedure that you want to copy.
3. Drag the selected procedure onto the **Copy To Module** frame.
4. You should see the inserted procedure and its code in the **Merge Into**

Procedure listing and the module listing.

5. When you have finished RAD-Copying, click **Save and Return** to save your changes. You can also click **Save As** to save the updated module with a different file name.

RAD-Copying Controls

You can copy any control from an existing form (the Little Buddy version does not support nested controls, such as frames and menus or control arrays). When you copy a control, your Buddy will copy all procedures and variable declarations that begin with controls name.

Inserting Controls into a Module

1. In the **Replace Tags With Values** grid, click on tags to enter the values that you want each occurrence of the tag to be replaced with in the target procedure.
2. In the **RAD-Copy This Object** box, select the control that you want to copy.
3. Drag the selected control onto the **Copy To Module** frame.
4. You should see the control name in the module listing.
5. When you have finished RAD-Copying, click **Save and Return** to save your changes. You can also click **Save As** to save the updated module with a different file name.

Creating and Customizing Templates

Your Buddy will work happily with any procedures and controls that you have developed without modification. However, a few simple instructions can instruct your Buddy to place code anywhere in a procedure. You can also create your own Replace Tags that mark strings of text for your Buddy to replace with new values at the time you RAD-Copy. Use these tools to create generic code Templates that can be reused in many different procedures.

See Also:

[Locating Code Withing A Procedure](#)

[Creating Replace Tags](#)

Locating Code Within a Procedure

When you instruct your Buddy to copy code to a new procedure, the following rules apply:

- Code from the top of the Source Procedure to the first blank line encountered will be inserted at the top of the Target Procedure. To insert a blank line in this section, use the single-quote (') to create a comment line.
- Labels without a line number comment will be inserted immediately after the top section.
- To instruct your Buddy to insert a label after a specific line in the procedure, place a comment on the line using the single-quote ('), followed immediately by a number. If you supply a negative number comment, your buddy will insert the label at the end of the procedure. If the number comment is greater than the number of lines in the Target Procedure, your buddy will insert enough blank lines to position the label at the line number you specify.
- Code immediately following a label, to the first blank line encountered will be inserted immediately after the label.
- All remaining code will be inserted at the end of the Target Procedure.

Example:

```
Public Sub ErrorHandler()  
On Error Goto ErrHandler:  
'This Will Go at Top  
  
ErrHandler:      '10  
  MsgBox "Problem!"  
  exit sub  
  
'This Line Goes At The Bottom
```

RAD-Copies To:

```
Public Sub MyProc()  
On Error Goto ErrHandler:  
'This Will Go at Top  
  
Dim Any as Variant  
For Any = 1 to 99  
  MsgBox "Hi"  
Next Any  
  
Exit Sub  
ErrHandler:      '10  
MsgBox "Problem!"  
exit sub  
  
'This Line Goes At The Bottom
```

Creating Replace Tags

You can mark strings of code to be replaced by your Buddy at the time you RAD-Copy. You change the value that these flagged strings will be changed into in the Replace Tags grid. If you do not specify a new value, the original Replace Tag will be inserted.

To Create a Replace Tag, just surround the text that you will want to replace with double braces in the format **{{text}}**. The string that you want to replace only needs to be bracketed once in the module to be interpreted as a Replace Tag. However, ALL occurrences of the string, bracketed or not, will be replaced when you RAD-Copy.

Your Buddy will recognize five system Tags and update them as follows:

{{ProcName}}	becomes the name of the destination procedure.
{{ModuleName}}	becomes the name of the destination module.
{{Date}}	becomes current date in DD-MMM-YY format.
{{Time}}	becomes current time in HH:MM:SS format.
{{Sub}}	becomes the procedure type (Sub, Function or Property) of the destination procedure.

Note: In a Procedure Declaration, only the ModuleName tag will be translated.

Replace tags can be used in any procedure declaration or statement. The VB Editor will flag the brackets as invalid some cases, however. You may want to create a dummy tag in a comment to allow the text to be replaced without using the invalid characters in actual code.

All Control Names automatically become Replace Tags. Use this feature to create reusable controls whose event procedures are copied with all references to the name replaced.

Example (source btnPrev):

```
Private Sub btnPrev_Click()  
-----  
` Function {{ProcName}}  
` Inserted at {{Date}} {{Time}}  
` Into {{ModuleName}}  
-----  
{{RecSet}}.MovePrevious  
  
If {{RecSet}}.AbsolutePosition = 0 Then  
    btnPrev.Enabled = False  
Else  
    btnPrev.Enabled = True  
End If  
Exit {{Sub}}
```

Might RAD-Copy To (as btnDB1Prev):

```
Private Sub btnDB1Prev_Click()  
-----  
` Function btnDB1Move_Click  
` Inserted at 19-AUG-1995 22:10:51  
` Into Project1  
-----  
rsDB1.MovePrevious  
  
If rsDB1.AbsolutePosition = 0 Then  
    btnDB1Prev.Enabled = False  
Else  
    btnDB1Prev.Enabled = True  
End If  
Exit Sub
```



The Buddy System

For more information, suggestions for future versions, or to be notified when the Big Buddies release of this product is available, visit the Official Buddy Blue Sky Page:
<http://www3.interaccess.com/budyhome/>

Copyright

The Buddy System Copyright (c) 1996 by Erik Krueger. All Rights Reserved.

Visual Basic 4.0 is a registered trademark of Microsoft Corporation.

Permission to use and distribute this software and its documentation in non-commercial purposes and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both the copyright notice and this permission notice appear in supporting documentation.

DISCLAIMER

THE BUDDY SYSTEM IS SUPPLIED AS IS. THE AUTHOR DISCLAIMS ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THE WARRANTIES OF MERCHANTABILITY AND OF FITNESS FOR ANY PURPOSE. THE AUTHOR ASSUMES NO LIABILITY FOR DAMAGES, DIRECT OR CONSEQUENTIAL, WHICH MAY RESULT FROM THE USE OF THE BUDDY SYSTEM.

