
Tip: Creating Custom Dialog Enabling Mutually Exclusive Features

Tsung-Huang Hsieh
Consultant

Abstract

This article provides a quick highlight on SelectionTree control, and its limitations in the scope of MSI on Windows 2000/XP, and describes an alternative method to enable mutually exclusive features.

Background

In a Basic MSI project, SelectionTree control enables users to change the selection state of features in the Feature table. This control is created with the WC_TREEVIEW class with the following styles:

- WS_BORDER
- TVS_HASLINES
- TVS_HASBUTTONS
- TVS_LINESATROOT
- TVS_DISABLERAGDROP
- TVS_SHOWSELALWAYS
- WS_CHILD
- WS_TABSTOP
- WS_GROUP

The SelectionTree control will automatically publish the following control events,

- SelectionAction
- SelectionBrowse
- SelectionDescription
- SelectionNoItems
- SelectionPath
- SelectionPathOn
- SelectionSize

This control does not natively support mutually exclusive selection of features.

Alternative method to enable mutually exclusive feature selection

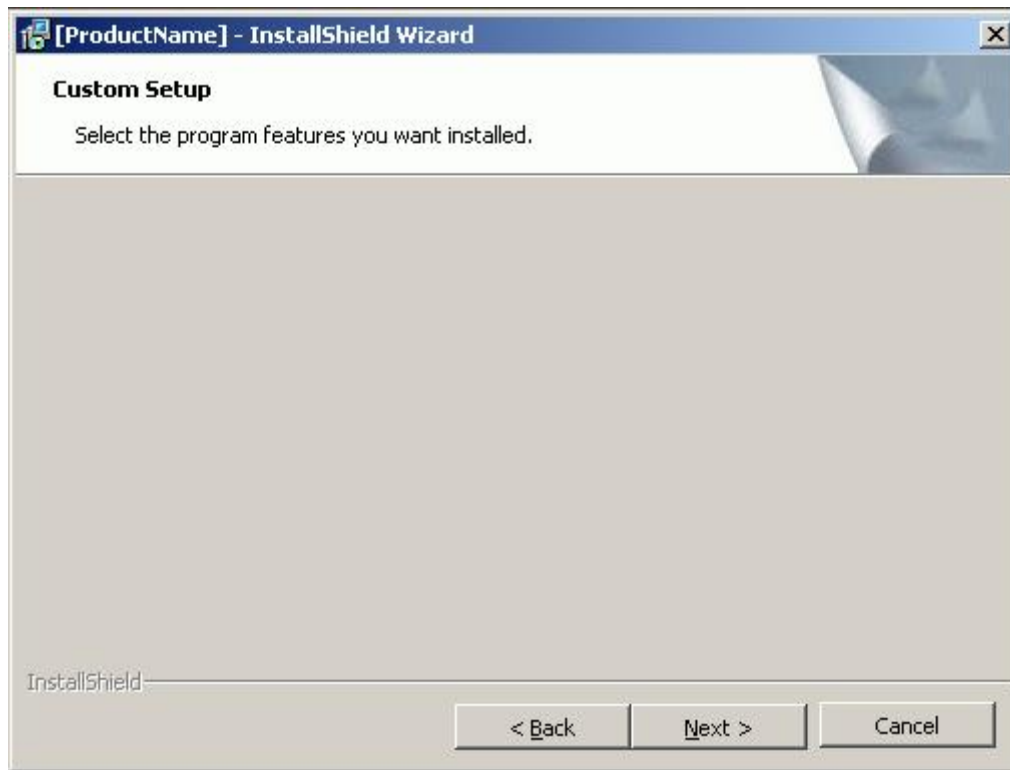
In a situation that only one (1) feature can be selected and installed from a set of given features, the SelectionTree control used in Developer standard dialog is constrained by technology. One alternative is to use a ListBox control to manipulate feature selection.

The steps to create a custom dialog with ListBox control populated with features in the project are as followed.

- Create an interior dialog
- Add ListBox control to dialog
- Populate ListBox control
- Define feature selection

Step 1: Create an interior dialog as shown below

Figure 1: Blank Interior Dialog

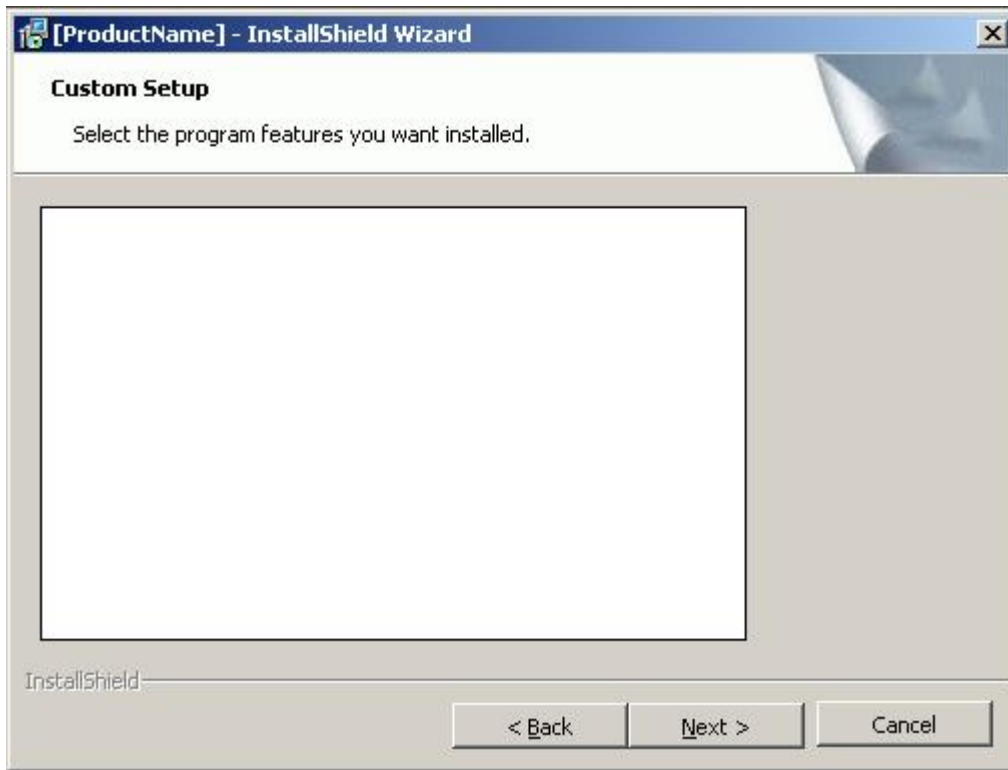


Step 2: Add ListBox control to dialog

Figure 2: ListBox control button



Figure 3: Custom Feature Selection Dialog



Step 3: Populate ListBox control

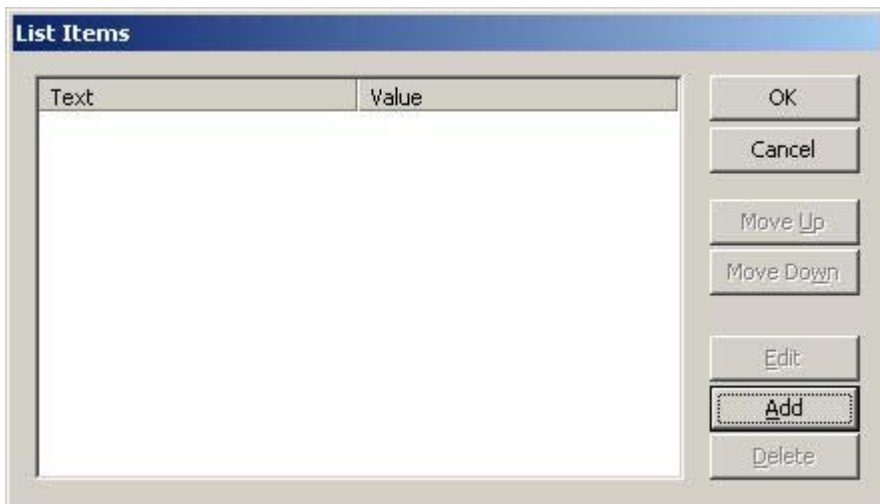
Figure 4 shows the complete list of attributes available to ListBox control. We should familiarize ourselves with, at least, the following property names, Property and Items.

Figure 4: ListBox Control Attributes

Property	Value
(Name)	ListBox1
Base Text Style	
Cancel	False
Code Page	Database
Context Help	
Control Identifier	11
Default	False
Enabled	True
Height	163
Indirect Property	False
Items	
Left	10
Left Scrollbar	False
Other Windows Styles	0
Property	FEATURESRLIST
Property Is Integer	False
Right-Aligned	False
Right-to-Left	False
Sorted	False
Sunken	False
Tab Index	3
Tab Stop	True
Text	
Text Style	
Tooltip	
Top	57
Visible	True
Width	265

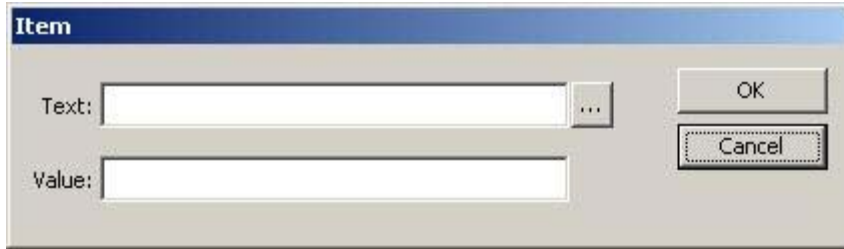
Select and click on property name Items in Figure 4 to bring up dialog (Figure 5).

Figure 5: Add ListBox Item Dialog



Click on Add button bring up dialog (Figure 6) to add individual feature into ListBox control. String entered for Value is used internally by MSI, and will not be visible to users at runtime. String entered for Text will be displayed to users. Value and Text do not have to have the same string. If Text is left blank, string entered for Value will be used for Text as well.

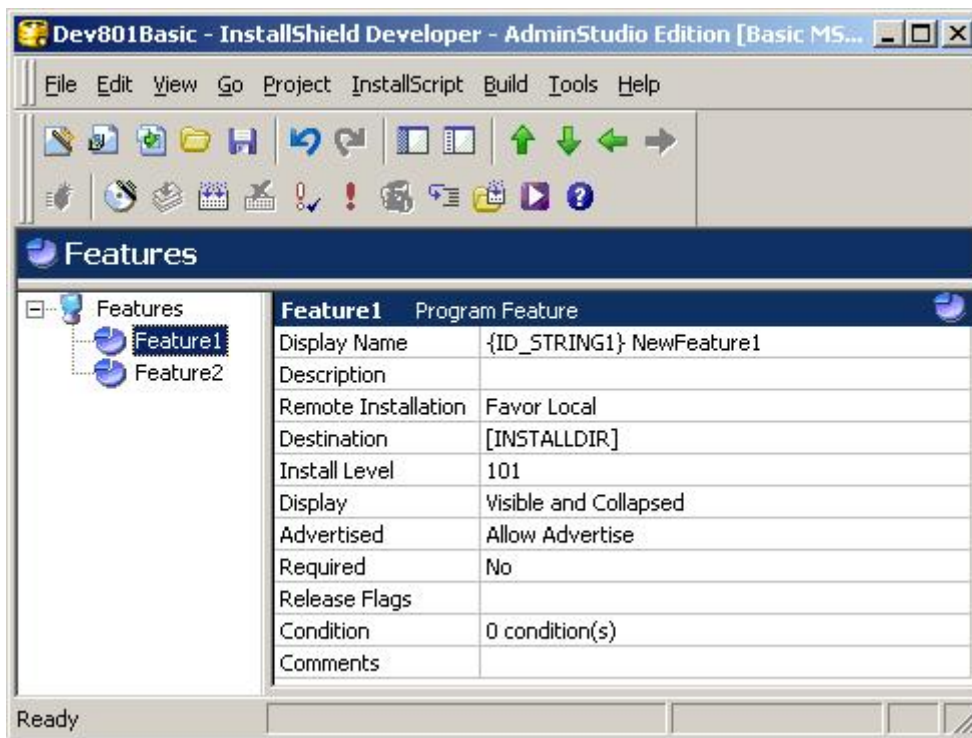
Figure 6: Add Items



Step 4: Define feature selection

Given a set of features as shown in Figure 7, set Install Level of all features to a number higher than the value of INSTALLLEVEL property. The default INSTALLLEVEL is 100. By doing that, it is to initialize all features to be not selected for installation by default.

Figure 7: Existing Features



Add AddLocal events to Next button as in Figure 8. It will select a feature for installation based on user's selection. You can add other necessary events per your installation requirements. AddLocal event takes a string argument, and does not resolve the string. When a list item is selected, the property of ListBox control will contain the string of Value column in Figure 5.

Figure 8: AddLocal Events on Next button

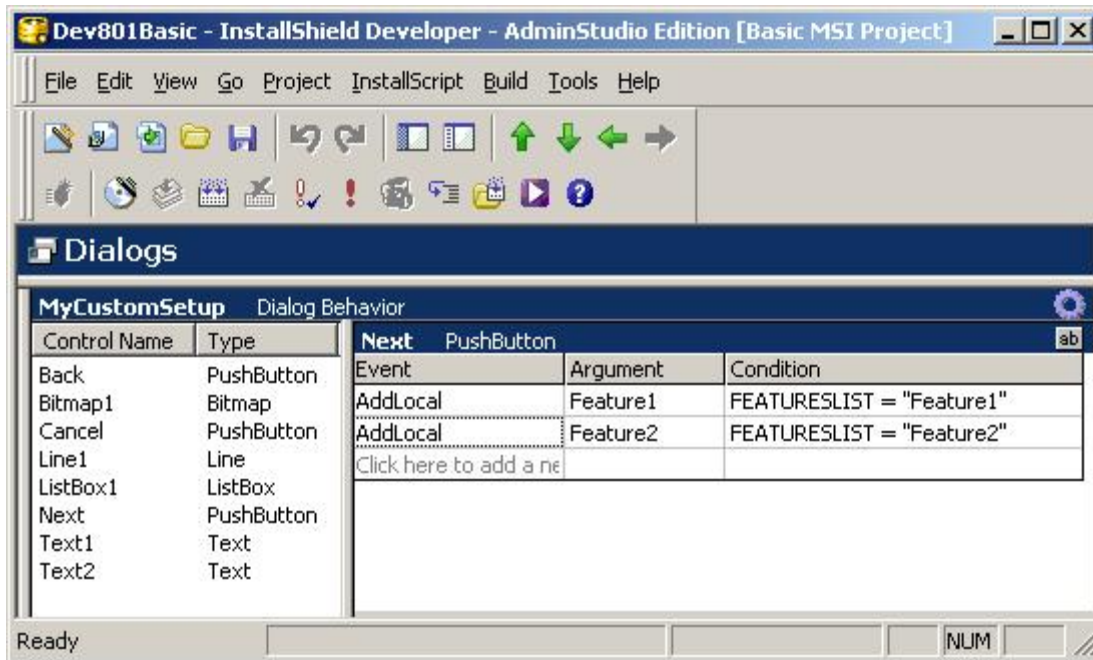


Figure 9 shows how the dialog will appear at runtime. To set a default selection, you can assign a value to the property of ListBox control in Property Manager in IDE.

Figure 9: Feature Selection Dialog at Runtime

