
Technical Note

An Efficient Combo Box Containing Millions of Items

Author: R&D Department

Publication date: January 7, 2007



© 2007 Consyst SQL Inc. All rights reserved.

An Efficient Combo Box Containing Millions of Items

Overview

Populating a .NET Framework **ComboBox** control with a large number of items takes a long time and causes a significant performance decrease of your application. The solution to this problem is to use a virtual combo box. A virtual combo box does not load all the items in its list: the items are kept in a separate array. It only calls for an item when it needs to be shown. The Microsoft® .NET Framework does not provide a virtual combo box control in its class library, so we developed one and included it in the REP++ Toolkit.

This article shows how you can use the virtual combo box control of the REP++ Toolkit.

Note: code samples are given in C#.

Using the REP++ virtual combo box

The REP++ virtual combo box, or **VComboBox** control, allows the developer to display efficiently a large number¹ of list items. This control requires the total number of items, specified by setting the **ItemCount** property of the control. Each time an item needs to be shown, the virtual combo box raises the **GetItemText** event. The user should catch that event and return the item defined by its index in the list.

The following example shows how you can use the virtual combo box to display a large number of items. The items are stored in an **ArrayList** object and are returned to the virtual combo box each time an item needs to be presented in the list.

1. Create a Windows® Application.
2. Add a **RepPPInfo** component to Form1 and set the REP++ connection parameters.
3. Add a REP++ virtual combo box (**RepPP.Toolkit.Window.VComboBox**) to Form1.
4. Add three members to Form1: A REP++ **Application** object, an **ArrayList** object to store the items to be displayed in the virtual combo box, and an integer defining the number of items.

```
/// <summary>
/// Rep++ application object
/// </summary>
private RepPP.Application          m_app;
/// <summary>
/// Array list to keep the combobox items
/// </summary>
private System.Collections.ArrayList m_arrItems;
/// <summary>
/// Constant to define the count of the items
/// </summary>
private const int                  ITEMS_COUNT = 100000;
```

¹ The number of items is limited only by the largest integer that can be represented on your computer.

Technical Note

5. Modify the constructor of Form1 to initialize the REP++ **Application** object, to save the combo box items in the **ArrayList** object, and to initialize the virtual combo box control.

Notice that you have to set the **ItemCount** property of the combo box to indicate the total number of items in the combo box list. You also have to catch the **GetItemText** event, which is raised each time a new item needs to be shown.

```
/// <summary>
/// Constructor
/// </summary>
public Form1() {
    int iIndex;

    InitializeComponent();
    // Create and initialize Rep++ application object
    m_app = RepPP.Application.CreateFromRes();

    // Prepare and save the combobox items in an arraylist
    m_arrItems = new System.Collections.ArrayList(ITEMS_COUNT);
    for (iIndex = 0; iIndex < ITEMS_COUNT; iIndex++) {
        m_arrItems.Add("Item " + (iIndex + 1).ToString());
    }
    // Initialize the virtual combobox
    vComboBox1.GetItemText += new
        RepPP.Toolkit.Window.VComboBox.ItemTextEventHandler(vComboBox1_GetItemText);
    vComboBox1.ItemCount = ITEMS_COUNT;
}
```

6. Modify the **Dispose** method of Form1 to delete resources.

```
protected override void Dispose(bool disposing) {
    if (disposing) {
        if (components != null) {
            components.Dispose();
        }
        if (m_app != null) {
            m_app.Dispose();
            m_app = null;
        }
    }
    base.Dispose(disposing);
}
```

7. When you receive a **GetItemText** event, you have to retrieve an item from the **ArrayList** object and return it.

```
/// <summary>
/// vComboBox1_GetItemText: Handle the GetItemText event
/// </summary>
/// <param name="sender"> Event sender</param>
/// <param name="e"> Event arguments</param>
void vComboBox1_GetItemText(object sender, RepPP.Toolkit.Window.VComboBox.ItemTextArg e) {
    e.Text = m_arrItems[e.Index].ToString();
}
```

8. Run your program. Notice the performance of the virtual combo box.

You can also increase the number of items by changing the value of the constant ITEMS_COUNT in the code and verify the performance of your program.

Technical Note

