

Quick Search for: in language:



[Code/Articles](#) >> | [Newest/Best](#) >> | [Community](#) >> | [Jobs](#) >> | [Other](#) >> | [Goto](#) >> |

Categories	Search	Newest Code	Coding Contest	Code of the Day	Ask a Pro	Jobs	Upload

[.Net Stats](#)

Code: 548,080. lines

Jobs: 655. postings

[How to support the site](#)

[Sponsored](#) by:

You are in::

[Login](#)

POPULATING NODES OF A TREEVIEW ON DEMAND

[Print](#)

[Email](#)



Submitted on: 9/7/2006 8:20:18 AM

[\(About the author\)](#)

By: Erika Stival

Level: Beginner

User Rating: By 2 Users

Compatibility: C#, ASP.NET

Users have accessed this article 5136 times.

Latest postings for .Net.

[ListView: Get Small & Large File Icons](#)

By Marc Foumberg on 2/28

[\(Screen Shot\)](#)

[Puzzle8 using A* algorithm](#)

Click [here](#) to [put this ticker on your site!](#)



Add this ticker to your desktop!

This code allows to avoid the problem of a large amount of data to display in a TreeView. Infact, using the population on demand property of this control you can choose to fill only parts of it (fill a node only when the user expands it).

Daily Code Email

To join the '[Code of the Day](#)' Mailing List [click here!](#)

Help us improve the site!

[Take the Planet Source Code survey today!](#)

```

[-] ParentCategory1
  [-] Category1
    [+ Subcategory1
      [-] Subcategory2
        [+ Expand Subcategory1
      [+ Subcategory3
    [-] Category2
      Subcategory1
  [+ ParentCategory3
  [+ ParentCategory4
  [+ ParentCategory2
  
```

Terms of Agreement:

By using this article, you agree to the following terms...

- 1) You may use this article in your own programs (and may compile it into a program and distribute it in compiled format for languages that allow it) freely and with no charge.
- 2) You MAY NOT redistribute this article (for example to a web site) without written permission from the original author. Failure to do so is a violation of copyright laws.
- 3) You may link to this article from another website, but ONLY if it is not wrapped in a frame.
- 4) You will abide by any additional copyright restrictions which the author may have placed in the article or article's description.

POPULATING NODES OF A TREEVIEW ON DEMAND:

Set the PopulateOnDemand property of the TreeView to true for any TreeNode that you want to fill in at the last minute.

When the users expand the branch, the TreeView will fire a TreeNode populate event, which you can use to add the next level of nodes.

Here is the code to do this.

All you need is:

A treeview control (id of the control: TreeView1)

A SqlDatabase with one table: TCategories(CategoryID, Name, ParentCategoryID); the table is related to itself

(CategoryID is connected to ParentCategoryID) to form a tree made by levels of parents and children categories.

(the case with two related tables as TCategories(Name,ID) and

TProducts(Name,ID), where TCategory contains the parents
and TProducts the children is slightly different)

```
using System;
using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class _Default : System.Web.UI.Page
{
    //variables useful for the code in the P // ageLoad
    public System.Data.DataSet DataSetCategories;
    public System.Data.SqlClient.SqlConnection ObjConn;
    public System.Data.SqlClient.SqlCommand SqlCommandTreeView;
    public System.Data.SqlClient.SqlDataAdapter myAdapter;
    public System.String StrSQL;
    //variables useful for the code in the T //
    reeView1_TreeNodePopulate1 event
    public System.Data.DataSet DataSetCategories1;
    public System.Data.SqlClient.SqlConnection ObjConn1;
    public System.Data.SqlClient.SqlCommand SqlCommandTreeView1;
    public System.Data.SqlClient.SqlDataAdapter myAdapter1;
    public System.String StrSQL1;
    //variables useful to create the dataarea // der
    public System.Data.SqlClient.SqlConnection ObjConn2;
    public System.Data.SqlClient.SqlCommand SqlCommandTreeView2;
    public System.String StrSQL2;
    public System.Data.SqlClient.SqlDataReader SqlDataReader2;
```

```
protected void Page_Load(object sender, EventArgs e)
{
//with this code I populate the TreeView // control with the first
series of nodes ( // depth=0)
if (!Page.IsPostBack)
{

//connection
this.ObjConn = new System.Data.SqlClient.SqlConnection();
this.ObjConn.ConnectionString = " "; //use your connection string
here
//command
this.SqlCommandTreeView = new
System.Data.SqlClient.SqlCommand();
this.SqlCommandTreeView.Connection = this.ObjConn;
//sql
//you have to make a query of this kind< // br> this.StrSQL = "
SELECT CategoryID, Name, ParentCategoryID From TCategories
WHERE (ParentCategoryID = 1)";
this.SqlCommandTreeView.CommandText = this.StrSQL;
//dataadapter
this.myAdapter = new System.Data.SqlClient.SqlDataAdapter();
this.myAdapter.SelectCommand = this.SqlCommandTreeView;
//dataset
this.DataSetCategories = new DataSet();
this.DataSetCategories.Tables.Add("TCategories");
this.myAdapter.Fill(this.DataSetCategories.Tables["TCategories"]);

//create a datatable dtCategories
DataTable dtCategories =
this.DataSetCategories.Tables["TCategories"];
//create a datarow rows
DataRow[] rows = dtCategories.Select("ParentCategoryID=" + 1);
```

```
foreach (DataRow singlerow in dtCategories.Rows) //for each
singlerow in the collection datarow rows...
{
//..you have to assigned a name and a va // lue
TreeNode nodeCategory = new
TreeNode(singlerow["Name"].ToString(),
singlerow["CategoryID"].ToString());
//..you have to choose populate on deman // d true, because
singlerow certainly has // children
nodeCategory.PopulateOnDemand=true;
//..you have to collapse the node, at fi // rst
nodeCategory.Collapse();
//.w1.Nodes.Add(nodeCategory);

}

this.DataBind();

}

}

protected void TreeView1_TreeNodePopulate(object sender,
TreeNodeEventArgs e)
{

//now, you have to fire TreeView1_TreeNodePopulate event and fill
the TreeView with the second level of nodes (depth=1)
//connection
this.ObjConn1 = new System.Data.SqlClient.SqlConnection();
this.ObjConn1.ConnectionString = " "; <%--//use your connection
string here
//command
this.SqlCommandTreeView1 = new
```

```
System.Data.SqlClient.SqlCommand();
this.SqlCommandTreeView1.Connection = this.ObjConn1;
//sql
//you have to make a query like this (e. // Node.Value is the id of
the "selected" n // ode)
this.StrSQL1 = " SELECT CategoryID, Name, ParentCategoryID From
TCategories WHERE (ParentCategoryID = "+e.Node.Value+")";
this.SqlCommandTreeView1.CommandText = this.StrSQL1;
//dataadapter
this.myAdapter1 = new System.Data.SqlClient.SqlDataAdapter();
this.myAdapter1.SelectCommand = this.SqlCommandTreeView1;
//dataset
this.DataSetCategories1 = new DataSet();
this.DataSetCategories1.Tables.Add("TCategories");
this.myAdapter1.Fill(this.DataSetCategories1.Tables["TCategories"]);
```

```
//create a datatable dtCategories1
DataTable dtCategories1 =
this.DataSetCategories1.Tables["TCategories"];
//create a datarow rows
```

WARNING.: the TreeView1_SelectedNodeChanged event fires after
TreeView1_TreeNodePopulate1 event
so TreeView1.SelectedNode.Value can't be use here
instead, you have to use, with almost the same meaning,
e.Node.Value

```
DataRow[] rows1 = dtCategories1.Select("ParentCategoryID=" +
Int32.Parse(e.Node.Value));
```

```
foreach (DataRow singlerow1 in dtCategories1.Rows) //for each
singlerow1 in the collection datarow rows1...
{
//..you have to assigned a name and a va // lue
```

```
TreeNode nodeCategory1 = new
TreeNode(singlerow1["Name"].ToString(),
singlerow1["CategoryID"].ToString());

//..you have to add children to the selected node of the TreeView
e.Node.ChildNodes.Add(nodeCategory1);

//now, you have to create a DataReader < // br> //if the ChildNodes
//have children, you have to choose populate on demand true ag //
//ain
//connection
this.ObjConn2= new System.Data.SqlClient.SqlConnection();
this.ObjConn2.ConnectionString = " "; //use your connection string
//open connection
this.ObjConn2.Open();
//command
this.SqlCommandTreeView2 = new
System.Data.SqlClient.SqlCommand();
this.SqlCommandTreeView2.Connection = this.ObjConn2;
//sql
//with this sql, I want to know if the single row has children
this.StrSQL2 = " SELECT CategoryID, Name, ParentCategoryID From
TCategories WHERE (ParentCategoryID = " +
singlerow1["CategoryID"] + ")";
//command
this.SqlCommandTreeView2.CommandText = this.StrSQL2;
this.SqlCommandTreeView2.CommandType = CommandType.Text;
//datareader
this.SqlDataReader2 =
this.SqlCommandTreeView2.ExecuteReader();
//if the datareader has rows and can be read...
if (this.SqlDataReader2.HasRows == true)
```

```
{
if (this.SqlDataReader2.Read())
{

//...the ChildNodes have children, so populate on demand true
nodeCategory1.PopulateOnDemand = true;
// at first, the node has to be collapse // d
nodeCategory1.Collapse();
}

}
//close the connection
this.ObjConn2.Close();

}

this.DataBind();

}

}
```

[Other 1 submission\(s\) by this author](#)

Report Bad Submission

Use this form to notify us if this entry should be deleted (i.e contains no code, is a virus, etc.).

This submission should be removed because:

[Report it!](#)

Your Vote!

What do you think of this article(in the Beginner category)?

(The article with your highest vote will win this month's [coding contest!](#))


Excellent Good Average Below Average Poor

[Rate It!](#)

[See](#)


[Voting Log](#)

Other User Comments

 9/6/2006 5:49:24 PM: [Gláucio](#)

nice code!

(If this comment was disrespectful, please [report it.](#))

 10/26/2006 5:39:29 AM: [Kumar Manoj](#)

very good and use full and guide line for any user.

(If this comment was disrespectful, please [report it.](#))

Add Your Feedback!

Note:Not only will your feedback be posted, but an email will be sent to the code's author from the email account you registered on the site, so you can correspond directly.

NOTICE: The author of this article has been kind enough to share it with you. If you have a criticism, please state it politely or it will be deleted.

For feedback not related to this particular article, please click [here](#).

To post feedback, first please [login](#).

[Categories](#) | [Articles and Tutorials](#) | [Advanced Search](#) | [Recommended Reading](#) | [Upload](#) | [Newest Code](#) | [Code of the Month](#) | [Code of the Day](#) | [All Time Hall of Fame](#) | [Coding Contest](#) | [Search for a job](#) | [Post a Job](#) | [Ask a Pro Discussion Forum](#) | [Live Chat](#) | [Feedback](#) | [Customize](#) | [.Net Home](#) | [Site Home](#) | [Other Sites](#) | [Open Letter from Moderators](#) | [About the Site](#) | [Feedback](#) | [Link to the Site](#) | [Awards](#) | [Advertising](#) | [Privacy](#)

Copyright© 1997-2008 by [Exhedra Solutions, Inc.](#) All Rights Reserved. By using this site you agree to its [Terms and Conditions](#). Planet Source Code (tm) and the phrase "Dream It. Code It" (tm) are trademarks of Exhedra Solutions, Inc.