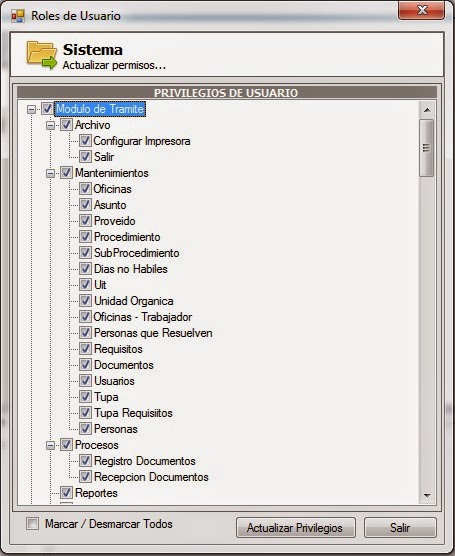
Hola buenas, aquí les traigo un código para llenar un treview en c#, y activar y desactivar sus nodos, de una tabla control de acceso de usuarios con la siguiente estructura:  
Tabla usuarios:  
id\_usuario:  
nombre

Tabla Menu  
id\_menu: Id unico  
idpadre: id del Padre del Menu  
descripción: Nombre del Formulario  
posición: posición del menu

Tabla: Menu\_Usuarios  
id: Id Unico  
id\_usuario  
id\_menu  
acceso: Activado o desactivado

[](https://lh3.googleusercontent.com/-yKqahHYn_zU/VUaAiQXLq0I/AAAAAAAAAD4/m7TseM-eXzc/s1600/treview.jpg)

using

System;

using

System.Collections.Generic;

using

System.ComponentModel;

using

System.Data;

using

System.Drawing;

using

System.Linq;

using

System.Text;

using

System.Threading.Tasks;

using

System.Windows.Forms;

using

CNegocio;

namespace

CPresentacion

{

public partial class frmUsuariosAcceso : Plantillas.frmPlantilla

    {

Usuarios\_MimenuCN objUsuarios\_MimenuCN = new Usuarios\_MimenuCN();

private DataTable dtTv;

public frmUsuariosAcceso()

        {

            InitializeComponent();

        }

private void frmUsuariosAcceso\_Load(object sender, EventArgs e)

        {

            dtTv =new DataTable();

//objUsuarios\_MimenuCN.GetUsuarios\_mimenuXIdUsuario(\_Key);

            dtTv = objUsuarios\_MimenuCN.GetUsuarios\_mimenuXIdUsuario(1);

            treeView1.CheckBoxes =true;

foreach (DataRow row in dtTv.Rows)

            {

TreeNode NodoPadre = new TreeNode();

    NodoPadre.Name = row["id\_menu"].ToString();

    NodoPadre.Text = row["descripcion"].ToString();

    NodoPadre.Tag = row["id\_menu"].ToString();

if (string.IsNullOrEmpty(row["id\_padre"].ToString()))

                {

                    treeView1.Nodes.Add(NodoPadre);

                    NodoPadre.Checked = (bool)row["acceso"];

                }

                CargarNodosHijos(NodoPadre.Tag.ToString(), NodoPadre);

            }

            treeView1.ExpandAll();

//Expandir todos los nodos

            treeView1.SelectedNode = treeView1.Nodes[0];

//seleccionar el primer nodo

        }

private void CargarNodosHijos(string indicePadre, TreeNode nodePadre)

        {

foreach (DataRow item in dtTv.Select("id\_padre = " + indicePadre, "posicion asc"))

                {

TreeNode nodoHijo = new TreeNode();

                    nodoHijo.Name = item[

"id\_menu"].ToString();

                    nodoHijo.Text = item[

"descripcion"].ToString();

                    nodoHijo.Tag = item[

"id\_menu"].ToString();

if (dtTv.Select("id\_padre =" + item["id\_menu"]).Length == 0)

                    {

//si no tiene hijos

                        nodePadre.Nodes.Add(nodoHijo);

//nodePadre.Nodes.Add(nodoHijo.Tag.ToString(), item["descripcion"].ToString());

                        nodoHijo.Checked = (

bool)item["acceso"];

                    }

else

                    {

//si tiene hijos

                        nodePadre.Nodes.Add(nodoHijo);

                        nodoHijo.Checked = (

bool)item["acceso"];

                        CargarNodosHijos(nodoHijo.Tag.ToString(), nodoHijo);

                    }

            }

        }

private void btnSalir\_Click(object sender, EventArgs e)

        {

this.Close();

        }

private void treeView1\_AfterCheck(object sender, TreeViewEventArgs e)

        {

            treeView1.AfterCheck -= treeView1\_AfterCheck;

            check\_nodos\_padre(e.Node);

            check\_nodos\_hijos(e.Node);

            treeView1.AfterCheck += treeView1\_AfterCheck;

        }

private void check\_nodos\_padre(TreeNode pNode)

        {

TreeNode NodoPadre = pNode.Parent;

if (NodoPadre != null) //si el nodo padre no es nulo

            {

if (pNode.Checked == true) // si el nodo esta activado

                {

if (pNode.Parent != null) // si tiene padre

                    {

if (pNode.Checked != NodoPadre.Checked) //si el padre esta activado

                        {

                            NodoPadre.Checked = pNode.Checked;

//check\_nodos\_padre(NodoPadre);

                        }

                    }

                }

else

                {

var cont = 0; //para acumular cuantos nodos hijos estan activos o desactivados

foreach (TreeNode node in NodoPadre.Nodes) //Recorrer todos los nodos del mismo nivel

                    {

cont = cont + ((node.Checked == pNode.Checked) ? 1 : 0);

                    }

if (cont == NodoPadre.Nodes.Count) //si el numero total de nodos hijos estan activos o desactivados

                        NodoPadre.Checked = pNode.Checked;

//establece el mismo estado al padre

                }

                treeView1.SelectedNode = NodoPadre;

if (NodoPadre.Parent != null) // si el nodo padre no es nulo

                {

                    check\_nodos\_padre(NodoPadre.Parent);

// realizamos la recursividad

                }

if (NodoPadre != null)

                {

                    NodoPadre = NodoPadre.Parent;

                }

            }

            treeView1.SelectedNode = pNode;

        }

private void check\_nodos\_hijos(TreeNode pNode)

        {

if (pNode.Nodes.Count > 0)

            {

for (int i = 0; i < pNode.Nodes.Count; i++)

                {

                    treeView1.SelectedNode = pNode;

                    pNode.Nodes[i].Checked = pNode.Checked;

if (pNode.Nodes[i].Nodes.Count > 0) //el nodo tiene subnodos

                    {

                        check\_nodos\_hijos(pNode.Nodes[i]);

                    }

                }

            }

            treeView1.SelectedNode = pNode;

        }

private void btnActualizar\_Click(object sender, EventArgs e)

        {

int resul;

try

            {

foreach (TreeNode Node in treeView1.Nodes)

                {

                    resul = 0;

                    resul = objUsuarios\_MimenuCN.UpdateUsuarios\_Mimenu(

Convert.ToInt32(1), Convert.ToInt32(Node.Tag), (bool)Node.Checked);

if (resul == 0)

                    {

break;

                    }

                    RecorrerNodos(Node);

                }

            }

catch (Exception ex)

            {

throw;

            }

MessageBox.Show("Los datos han sido grabados correctamente.", VariablesGlobales.Titulo, MessageBoxButtons.OK, MessageBoxIcon.Information, MessageBoxDefaultButton.Button1);

this.Close();

        }

private void RecorrerNodos(TreeNode treeNode)

        {

int resul;

try

            {

foreach (TreeNode tn in treeNode.Nodes)

                {

                    resul = 0;

                    resul = objUsuarios\_MimenuCN.UpdateUsuarios\_Mimenu(

Convert.ToInt32(1), Convert.ToInt32(tn.Tag), (bool)tn.Checked);

if (resul == 0)

                    {

break;

                    }

                    RecorrerNodos(tn);

                }

            }

catch (Exception ex)

            {

throw;

            }

        }

private void checkBox1\_CheckedChanged(object sender, EventArgs e)

        {

try

            {

foreach (TreeNode Node in treeView1.Nodes)

                {

//if (checkBox1.Checked == true)

//    checkBox1.Checked = true;

//else

//    checkBox1.Checked = false;

                    Node.Checked = (checkBox1.Checked ==

true) ? true : false;

                    RecorrerNodos(Node);

                }

            }

catch (Exception ex)

            {

throw;

            }

            treeView1.SelectedNode = treeView1.Nodes[0];

        }

private void ActivarDesactivarNodos(TreeNode treeNode)

        {

try

            {

foreach (TreeNode tn in treeNode.Nodes)

                {

                    tn.Checked = (checkBox1.Checked ==

true) ? true : false;

                    RecorrerNodos(tn);

                }

            }

catch (Exception ex)

            {

throw;

            }

        }

    }

}



