

1. Change presentation

* Highlight Row withouth losing the previous background color
* Set a ToolTip in each row to remain the user what he may do

protected void GridView1\_RowCreated(object sender, GridViewRowEventArgs e)

{

Panel1.Visible = true;

if (e.Row.RowType == DataControlRowType.DataRow)

{

e.Row.Attributes.Add("**onmouseover**", "this.previous\_color=this.style.backgroundColor;this.style.backgroundColor='#C1DAD7'");

e.Row.Attributes.Add("**onmouseout**","this.style.backgroundColor=this.previous\_color;");

e.Row.**ToolTip** = "Click Row to Start Update";

}

}

1. Retrieve Data from GridView

Now what we want is to retrieve the values from the row so we can update the data

protected void GVMetrics\_RowDataBound(object sender, GridViewRowEventArgs e)

{

if (e.Row.RowType == DataControlRowType.DataRow)

{

string clients = e.Row.Cells[2].Text;

string metrics = e.Row.Cells[3].Text;

string values = e.Row.Cells[4].Text;

string goals = e.Row.Cells[5].Text;

string rowClick="LoadDataToUpdate('"+clients+"','"+metrics+"','"+values+"','"+goals+"'); return false; ";

e.Row.Attributes.Add("onclick", rowClick);

}

}

An attribute is created for each row, which invokes a function in javascript that has been created in the Header of the HTML page

1. Set Data in Labels and TextBoxes

<script type="text/javascript">

function LoadDataToUpdate(clients, metrics, values, goals) {

document.getElementById("lbClient").innerText = clients;

document.getElementById("lbMetric").innerText = metrics;

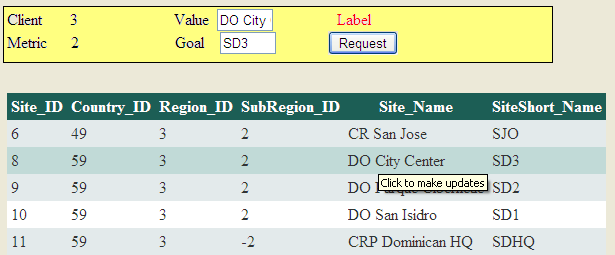
document.getElementById("txValue").innerText = values;

document.getElementById("txGoal").innerText = goals;

}

</script>

And updates 2 labels and 2 textboxes. lbClient, lbMetric, txValue, txGoal



1. Set validations

The button can be set with a ClientScript so we can avoid also the post back

<asp:Button ID="Button1" runat="server" OnClientClick='checkValues(); return false;' Text="Request" />

It calls a function to validate the data entered in the textboxes

function checkValues() {

var valor = document.getElementById("txValue").value;

var goal = document.getElementById("txGoal").value;

var valor2 = parseFloat(valor);

var goal2 = parseFloat(goal);

if (valor.length == 0) {

document.getElementById("lbCheck").innerText = "Type a value";

return false;

}

if (goal.length == 0) {

document.getElementById("lbCheck").innerText = "Type a goal";

return false;

}

PageMethods.InputAuthorization(valor2, goal2);

alert("Request for Authorization Submitted");

}

1. Submit Update

If the validation is correct the javascript calls a process in the code behind with

PageMethods.InputAuthorization(valor2, goal2);

But to allow the PageMethod you have to include a ScriptManager in the page

<body>

<form id="form1" runat="server">

<asp:ScriptManager ID="ScriptManager1" runat="server" EnablePageMethods="true"/>

The code that invokes a process in the code-behind is InputAuthorization and should be

* Static
* Web Method

[System.Web.Services.WebMethod]

public static void InputAuthorization(double valor, double goal)

{

string connection =

System.Configuration.ConfigurationManager.ConnectionStrings["Conn1"].ConnectionString;

SqlConnection conn = new SqlConnection(connection);

string sql = ( SQL COMAND, INSERT, UPDATE, DELETE , ETC)

SqlCommand cmd = new SqlCommand(sql, conn);

cmd.ExecuteNonQuery();

}

1. Change Grid View Cells

If you have updated the data you may change the grid view without a postback, you make a scroll and look for the row which has been highlighted. But in this case we will scroll each cell, and the values “I” is the number of cells, and the value “j” es the value of the cell we are looking for matching, which has the UNIQUE KEY.

function checkValues() {

var i =0;

var j = 0;

$("#GVMetrics td").each(function () {

var Celda = $(this).html();

i++;

if (Celda == ValueKey) {

$(this).html('ERASE');

j = i;

}

});

}

But in this case the key is made by 2 values Clients and Metrics

function checkValues() {

var client = document.getElementById("lbClient").value;

var metric = document.getElementById("lbMetric").value;

var I =0;

var j = 0;

var cliente\_Found = false;

$("#GVMetrics td").each(function () {

var Celda = $(this).html();

i++;

if (Celda == metric && cliente\_Found) {

$(this).html('ERASE');

j = i; **// Row where key is has been found**

}

cliente\_Found=false;

if (Celda == client) cliente\_Found = true;

});

}

Now that you have found the row where to update you can update the data in columns 4, 5

function checkValues() {

var client = document.getElementById("lbClient").value;

var metric = document.getElementById("lbMetric").value;

var valor = document.getElementById("txValue").value;

var goal = document.getElementById("txGoal").value;

var I =0;

var j = 0;

var k = 0;

var cliente\_Found = false;

var row\_Found = false;

$("#GVMetrics td").each(function () {

var Celda = $(this).html();

i++;

if (Celda == metric && cliente\_Found) {

j = i; **// Row where key is has been found**

row\_Found = true;

}

cliente\_Found=false;

if (Celda == client) cliente\_Found = true;

if (row\_Found) k++;

if (k==4) $(this).html('Update');

if (k==5) $(this).html(valor);

if (k==6) $(this).html(goal);

});

}

1. Deleting Rows

If you have deleted the data you can remove it from the grid without making a postback, first look for the row that has been highlighted

We know the row in the grid, it is j = j / 6 +1; because the grid has 6 fields, and before fading the row, turn it into green.

function BtnHideRow() {

**// Check which row**

$("#GVMetrics td").each(function () {

var Celda = $(this).html();

i++;

if (Celda == metric && cliente\_Found) {

j = i; **// Row where key is has been found**

}

cliente\_Found=false;

if (Celda == client) cliente\_Found = true;

});

**// Fade row**

i = 0;

j = j / 6 + 1;

$("#GVMetrics tr").each(function () {

i++;

if (i == j) {

$(this).css("background-color", "green");

$(this).fadeOut(1000, function () {

$(this).remove();

});

}

});

}

1. Final Scripts

<script src="Scripts/jquery-1.4.1.min.js" type="text/javascript"></script>

<script type="text/javascript">

function LoadDataToUpdate(clients, metrics, values, goals) {

// $("#Panel1").show();

document.getElementById("lbClient").innerText = clients;

document.getElementById("lbMetric").innerText = metrics;

document.getElementById("txValue").innerText = values;

document.getElementById("txGoal").innerText = goals;

}

function checkValues() {

var valor = document.getElementById("txValue").value;

var goal = document.getElementById("txGoal").value;

var valor2 = parseFloat(valor);

var goal2 = parseFloat(goal);

if (valor.length == 0) {

document.getElementById("lbCheck").innerText = "Type a value";

return false;

}

if (goal.length == 0) {

document.getElementById("lbCheck").innerText = "Type a goal";

return false;

}

var siteID = document.getElementById("dbSite").value;

var clientID = document.getElementById("lbClient").innerText;

var Metric = document.getElementById("lbMetric").innerText;

var i = 0;

var j = 0;

var found = false;

$("#GVMetrics td").each(function () {

var celda = $(this).html();

i++;

if (celda == Metric && found) {

$(this).html('SUBMITTED');

j = i;

}

found = false;

if (celda == clientID && j==0) found = true;

});

var num\_columns = 6; // Numero de columnas en grid

j = Math.round(j / num\_columns) + 2;

i = 0;

$("#GVMetrics tr").each(function () {

i++;

if (i == j) {

$(this).css("background-color", "00CCCC");

// $(this).fadeOut(1000, function () {

// $(this).remove();

// });

}

});

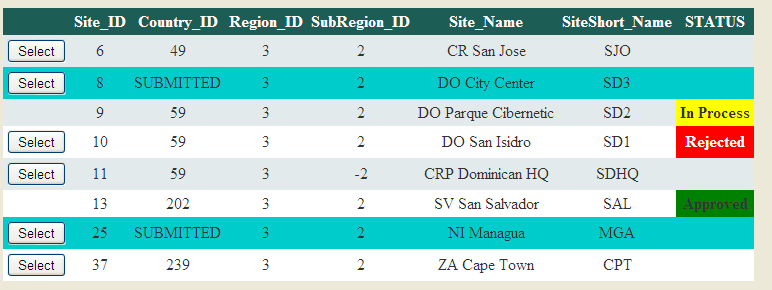
PageMethods.InputAuthorization(( SiteID, ClientID, Metric, valor2, goal2);

alert("Request for Authorization Submitted");

}

</script>

1. Improving Presentation



protected void GVMetrics\_RowDataBound(object sender, GridViewRowEventArgs e)

{

int flag\_cell = 7;

//===============

if (e.Row.RowType == DataControlRowType.DataRow)

{

string clients = e.Row.Cells[1].Text;

string metrics = e.Row.Cells[2].Text;

string values = e.Row.Cells[4].Text;

string goals = e.Row.Cells[5].Text;

string flag = e.Row.Cells[flag\_cell].Text;

string rowClick = " LoadDataToUpdate('" + clients + "','" + metrics + "','" + values + "','" + goals + "'); return false; return false;";

int ss = e.Row.RowIndex;

if (flag == "1")

{

// in process

e.Row.Cells[flag\_cell].Text = "In Process";

e.Row.ToolTip = "Request in Process";

rowClick = "alert('Request in Process'); return false;";

e.Row.Cells[flag\_cell].BackColor = System.Drawing.Color.Yellow;

e.Row.Cells[flag\_cell].Font.Bold = true;

e.Row.Cells[0].Text = "";

}

else

if (flag == "2")

{

//rejected

e.Row.Cells[flag\_cell].Text = "Rejected";

e.Row.ToolTip = "Request Rejected";

e.Row.Cells[flag\_cell].BackColor = System.Drawing.Color.Red;

e.Row.Cells[flag\_cell].ForeColor = System.Drawing.Color.White;

e.Row.Cells[flag\_cell].Font.Bold = true;

// rowclick

}

else

if (flag == "3")

{

// approved

e.Row.Cells[flag\_cell].Text = "Approved";

e.Row.ToolTip = "Request Approved";

rowClick = "alert('Request Approved'); return false;";

e.Row.Cells[flag\_cell].BackColor = System.Drawing.Color.Green;

e.Row.Cells[flag\_cell].Font.Bold = true;

e.Row.Cells[0].Text = "";

}

else

{

e.Row.Attributes.Add("onmouseover", "this.previous\_color=this.style.backgroundColor; if (this.style.backgroundColor!='00CCCC') { this.previous\_color=this.style.backgroundColor;this.style.backgroundColor='#C1DAD7'} ");

e.Row.Attributes.Add("onmouseout", " if (this.style.backgroundColor!='00CCCC') this.style.backgroundColor=this.previous\_color;");

e.Row.ToolTip = "Click to make updates";

// rowclick

}

e.Row.Attributes.Add("onclick", rowClick);

}

}

Button to select

<asp:TemplateField>

<ItemTemplate>

<asp:Button runat="server" ID="SelectButton"

Text ="Select"

OnClientClick="return false;" />

</ItemTemplate>

</asp:TemplateField>