VALUES HARD CODED

private void LoadMetrics()

{

ArrayList List = new ArrayList();

List<Metricas> lista = new List<Metricas>();

lista.Add(new Metricas() {id="0", nombre="AHT"});

lista.Add(new Metricas() {id="1", nombre="AR"});

lista.Add(new Metricas() {id="3", nombre="CCS"});

listMetric.DataSource = lista;

listMetric.DataTextField = "nombre";

listMetric.DataValueField = "id";

listMetric.DataBind();

}

USING CACHE

private void LoadClients()

{

DataSet ds2 = (DataSet)Cache["Clients"];

dt = ds2.Tables[0];

ddlClient.DataSource = dt;

}

DAO

<appSettings>

<add key="SqlConn" value="server=.\SQLEXPRESS;database=OS\_EVENTS;Integrated Security=SSPI"/>

</appSettings>

using System.Configuration;

string SqlConn = ConfigurationManager.AppSettings["SqlConn"];

System.Data.SqlClient.SqlConnection dbConnection = new System.Data.SqlClient.SqlConnection(SqlConn);

System.Data.SqlClient.SqlCommand dbCommand = new System.Data.SqlClient.SqlCommand();

string sql = "SELECT USERID\_ATTR FROM [OS\_EVENTS].[dbo].[OS\_CUSTOM\_EVENT] ";

dbCommand.CommandText = sql;

dbCommand.Connection = dbConnection;

System.Data.IDataReader dataReader = null;

dbConnection.Open();

dataReader = dbCommand.ExecuteReader(System.Data.CommandBehavior.CloseConnection);

dbUser.Items.Clear();

dbUser.Items.Add("All");

while (dataReader.Read()) dbUser.Items.Add(dataReader.GetValue(0).ToString());

dbConnection.Close();

DAO STORED PROCEDURE WITH SqlHelper

private void LoadProgramCode()

{

SqlParameter[] parameters = new SqlParameter[1];

parameters[0] = new SqlParameter("@Mapped", SqlDbType.Int);

if (CheckBox2.Checked)

parameters[0].Value = 1; //mapped

else

parameters[0].Value = 0; //unmapped

ddlProgram.DataSource = SqlHelper.ExecuteDataset(\_connStr, CommandType.StoredProcedure, "sp\_Dpt2LOB", parameters);

ddlProgramCode.DataTextField = "WFMProgramCode\_Name";

ddlProgramCode.DataValueField = "WFMProgramCode\_Id";

ddlProgramCode.DataBind();

ListItem li = new ListItem("--All--", "0");

ddlProgramCode.Items.Insert(0, li);

}

LINQ

private void LoadClients(DataTable dt)

{

var newSort2 = from row in dt.AsEnumerable()

where row.Field<object>("client\_id") != null

orderby row.Field<object>("client\_descr")

group row by new

{

ID = Convert.ToString(row.Field<object>("client\_id")),

nombre = row.Field<string>("client\_descr")

} into grp

select new

{

ids = Convert.ToString(grp.Key.ID),

nombres = grp.Key.nombre

};

ddlClient.DataSource = newSort2;

ddlClient.DataValueField = "ids";

ddlClient.DataTextField = "nombres";

ddlClient.DataBind();

ListItem li = new ListItem("Select One", "-1");

ddlClient.Items.Insert(0, li);

}

WEB METHOD

$(function() {

$("#ddlClient").focus();

$('#ddlClient').bind("change keyup", function () {

loadSites($(this).val());

});

});

// LOAD SITES

function Load\_Sites() {

var myDDL = document.getElementById('cplMianDisplay\_ddlClient');

var cliente = myDDL.value;

if (cliente == "-1") return;

// Si escogio cliente, borre la primera file que dice “Select One”, -1

if (myDDL[0].value == -1) {

myDDL.remove(0);

}

// HIDDEN FIELD STORE CLIENT

document.getElementById("HDClient").value = cliente;

PageMethods.Load\_Sitios(cliente, OnSuccess, OnFail);

}

function OnSuccess(response) {

var myDDL = document.getElementById('cplMianDisplay\_ddlSites');

myDDL.options.length = 0;

for (var i in response) {

var option = document.createElement('option');

option.text = response[i].names;

option.value = response[i].ids;

myDDL.add(option);

}

$("#cplMianDisplay\_listTeam").empty();

$("#cplMianDisplay\_listTeam").multiselect('refresh');

}

function OnFail(err) {

alert(err.get\_message());

}

**CODE BEHIND**

[System.Web.Services.WebMethod]

public static List<SupvListEntity> Load\_Sitios(string cliente)

{

System.Web.Caching.Cache cache;

cache = HttpContext.Current.Cache;

DataTable dt = (DataTable)cache["TeamLeaders"];

var newSort2 = from row in dt.AsEnumerable()

where row.Field<object>("agent\_site\_id") != null

&& row.Field<string>("client\_id") == cliente

orderby row.Field<object>("agent\_site\_descr")

group row by new

{

ID = Convert.ToString(row.Field<object>("agent\_site\_id")),

nombre = row.Field<string>("agent\_site\_descr")

} into grp

select new

{

ids = Convert.ToString(grp.Key.ID),

names = grp.Key.nombre

};

List<SupvListEntity> listado = new List<SupvListEntity>();

listado.Add(new SupvListEntity() { ids = "-1", names = "Select One" });

foreach (var sitios in newSort2)

{

listado.Add(new SupvListEntity() { ids = sitios.ids, names = sitios.names });

}

return listado;

//listTeam.Items.Clear();

}