CacheTime60 60 minutes = 60 x 60 = 3600 it is in seconds

private void LoadRegion()

{

if (Cache["Regions"] == null)

{

DataSet ds = SqlHelper.ExecuteDataset(\_connStr, CommandType.StoredProcedure, "sp\_GetRegion");

//Cache.Insert("Regions", ds, null, DateTime.Now.AddMinutes(CacheTime), System.Web.Caching.Cache.NoSlidingExpiration);

Cache.Insert("Regions", ds, null, System.Web.Caching.Cache.NoAbsoluteExpiration,

new TimeSpan(0, CacheTime, 0));

}

ddlRegion.DataSource = (DataSet)Cache["Regions"];

ddlRegion.DataTextField = "Region\_Name";

ddlRegion.DataValueField = "Region\_ID";

ddlRegion.DataBind();

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

ddlRegion.Items.Insert(0, li);

}

using System.Xml.Linq;

if (Cache["DropDownData"] == null)

{

System.Xml.Linq.XDocument xDoc = new XDocument();

xDoc = XDocument.Load(Server.MapPath("~/Currencies.xml")));

// var qq = (from XX in xDoc.Descendants("Row")

// select XX).ToList();

var Queries = from XElem in xDoc.Descendants("Row")

select new

{

CourseID = XElem.Element("CourseID").Value,

Title = XElem.Element("Title").Value

};

**LoadQuery**(Queries);

Cache.Insert("Data1", Queries, new

System.Web.Caching.CacheDependency(Server.MapPath("~/Currencies.xml")));

}

else

{

var Queries2 = Cache["Data1"];

**LoadQuery**(Queries2);

}

}

public void **LoadQuery**(Object Queries)

{

DropDownList1.DataTextField = "Title";

DropDownList1.DataValueField = "CoursesID";

this.DropDownList1.DataSource = Queries;

this.DropDownList1.DataBind();

}

}

The requirements are:

-Store the number of active bugs in the cache.

-The value should remain in the cache when there are calls more often than every **15** seconds.

-The value should be removed from the cache after **60** seconds, which means at least 4 times can be accesed. The value will be definitely removed, even though it has been accessed several times.

If he cache is not accessed after 15 seconds it will not needed to get the dependency, it will be removed

Cache.Insert(“**Trigger**”, DateTime.Now,null, DateTime.Now.AddSeconds(**60**), Cache.NoSlidingExpiration);

CacheDependency **cd** = new CacheDependency(null, new string[] {“**Trigger**”]);

Cache.Insert(“ActiveBugs, result, **cd**, Cache.NoAbsoluteExpiration, TimeSpan.FromSeconds(**15**));

private void LoadRegion()

{

if (Cache["Regions"] == null)

{

DataSet ds = SqlHelper.ExecuteDataset(\_connStr, CommandType.StoredProcedure, "sp\_RefMap\_GetRegion");

Cache.Insert("Regions", ds, null, System.Web.Caching.Cache.NoAbsoluteExpiration, new TimeSpan(0, CacheTime60, 0));

}

ddlRegion.DataSource = (DataSet)Cache["Regions"];

ddlRegion.DataTextField = "Region\_Name";

ddlRegion.DataValueField = "Region\_ID";

ddlRegion.DataBind();

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

ddlRegion.Items.Insert(0, li);

}

private void LoadSite()

{

ddlSite.Items.Clear();

int RegionID = System.Convert.ToInt32(ddlRegion.SelectedValue);

if (Cache["SitesAll"] == null)

{

SqlParameter[] parameters = new SqlParameter[2];

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

parameters[0].Value = 0;

parameters[1] = new SqlParameter("@SubRegionId", SqlDbType.Int);

parameters[1].Value = 0;

DataSet ds = SqlHelper.ExecuteDataset(\_connStr, CommandType.StoredProcedure, "sp\_RefMap\_GetSites", parameters);

Cache.Insert("SitesAll", ds, null, DateTime.Now.AddMinutes(CacheTime60), System.Web.Caching.Cache.NoSlidingExpiration);

}

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

DataTable dt = ds2.Tables[0];

var query = from o in dt.AsEnumerable()

where (o.Field<int>("Region\_ID") == RegionID || RegionID == 0)

orderby o.Field<string>("Site\_Name")

select new

{

Site\_Name = o.Field<string>("Site\_Name"),

Site\_ID = o.Field<int>("Site\_ID")

};

ddlSite.DataSource = query;

ddlSite.DataTextField = "Site\_Name";

ddlSite.DataValueField = "Site\_ID";

ddlSite.DataBind();

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

ddlSite.Items.Insert(0, li);

}

private void LoadClient()

{

if ((Condition\_TR.Visible || ProgramCode\_TR.Visible) && ddlClient.Items.Count > 0) return;

ddlClient.Items.Clear();

int RegionID = 0;

int SiteID = 0;

if (!Condition\_TR.Visible && !ProgramCode\_TR.Visible)

{

try

{

RegionID = System.Convert.ToInt32(ddlRegion.SelectedValue);

SiteID = System.Convert.ToInt32(ddlSite.SelectedValue);

}

catch

{

}

}

DataTable dt = new DataTable();

if (CVPass.Visible)

{

if (Cache["ClientsCV"] == null)

{

SqlParameter[] parameters = new SqlParameter[2];

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

parameters[0].Value = 0;

parameters[1] = new SqlParameter("@Region\_ID", SqlDbType.Int);

parameters[1].Value = 0;

DataSet ds = SqlHelper.ExecuteDataset(\_connStr, CommandType.StoredProcedure, "sp\_GetClients", parameters);

Cache.Insert("ClientsCV", ds, null, DateTime.Now.AddMinutes(CacheTime30), System.Web.Caching.Cache.NoSlidingExpiration);

}

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

dt = ds2.Tables[0];

var query = from o in dt.AsEnumerable()

where (o.Field<int>("Region\_ID") == RegionID || RegionID == 0)

&& (o.Field<int>("Site\_ID") == SiteID || SiteID == 0)

orderby o.Field<string>("Client\_Name")

select new

{

Client\_Name = o.Field<string>("Client\_Name"),

ClientID = o.Field<int>("ClientId")

};

ddlClient.DataSource = query.Distinct();

}

else

{

if (Cache["Clients"] == null)

{

DataSet ds = SqlHelper.ExecuteDataset(\_connStr, CommandType.StoredProcedure, "sp\_RefMap\_GetClients");

Cache.Insert("Clients", ds, null, DateTime.Now.AddMinutes(CacheTime30), System.Web.Caching.Cache.NoSlidingExpiration);

}

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

dt = ds2.Tables[0];

ddlClient.DataSource = dt;

}

ddlClient.DataTextField = "Client\_Name";

ddlClient.DataValueField = "ClientId";

ddlClient.DataBind();

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

ddlClient.Items.Insert(0, li);

ddlProgram.Items.Clear();

ddlProgram.Items.Insert(0, li);

}