

ASP.net with kbmMWSimpleClient

Using kbmMW with ASP.net pages gives you a lot of speed advantages over the direct connection methods standard with the D2005 Development environment. Using kbmMWSimpleClient give you all the advantages and benefits of multi tier development that you experience in your win32 applications.

Requirements:

You need to install version 2.5x on your computer following the installation instructions from kbm for D2005 dot.net. To ensure you are installed properly create a new winform project and compile it with both a kbmMemtable and a kbmMWSimpleClient control dropped on the form. If you can compile the app then your installation is correct. If not carefully review the pathing instructions in the installation instruction file provided with your kbmMW and kbmMemtable installation.

NOTE: .Net Winform and WebForm applications are very picky about pathing, most problems you will experience with D2005 will be pathing issues.

Installing the demo

Extract the demo project on your computer and then copy the directory (kbmDotNetclient) to the web root directory of IIS (f.ex wwwroot). Next open the IIS control panel – right click on the directory copied to the root and select properties. In the virtual directories tab under Application Settings click on create, next click on Apply. Go to your Documents tab and make sure that default.aspx is listed in your default documents. The demo app should now be running as an ASP.net application and access should be [http://localhost/ kbmDotNetclient](http://localhost/kbmDotNetclient).

FYI:

Adding References:

When building ASP.net WebForms you must add references to all resources you want to use in your project. When using kbmMW you need to add kbmMWRunD2005dn.dll and kbmMemD2005dnRun.dll to the references of your WebApplication Project.

Uses clause:

Just like win32 applications you must add your design resources to the **USES** clause of your form. Some typical ones that cover most kbmMW client projects are:

System.Collections, System.ComponentModel, Variants, System.Web.Security, System.Data, System.Drawing, System.Web, System.Web.SessionState, System.Web.UI, System.Security, System.Web.UI.WebControls, System.Web.UI.HtmlControls, kbmMWClient, kbmMWCustomConnectionPool, kbmMWClientDataset, Borland.Vcl.SysUtils, kbmMemTable,



kbmMWStreamFormat, kbmMWBinaryStreamFormat, Borland.Vcl.Db, kbmMWCCustomDataset, kbmMWGlobal, kbmMWCCustomTransport, kbmMWTCPiPIndyClientTransport;

Declarations

Setting up a kbmclient is now the same as programming with kbmMW and kbmMemtable as in the win32 environment. What works best for me is:

Declare the types

type

```
TDefaultForm = class(System.Web.UI.Page)
//add these items after form is assigned
  kbmPool: TkbmMWClientConnectionPool;
  kbmTransport: TkbmMWTCPiPIndyClientTransport;
  kbmQuery: TkbmMWClientQuery;
  kbmStreamFormat: TkbmMWBinaryStreamFormat;
  kbmSession: TkbmMW PooledSession;
  kbmClient: TkbmMW SimpleClient;
```

Create the kbmControls

procedure TDefaultForm.Page_Load(sender: System.Object; e: System.EventArgs);

begin

```
// TODO: Put user code to initialize the page here
  kbmPool := TkbmMWClientConnectionPool.Create(nil);
  kbmTransport := TkbmMWTCPiPIndyClientTransport.Create(nil);
  kbmClient := TkbmMW SimpleClient.Create(nil);
  kbmStreamFormat := TkbmMWBinaryStreamFormat.Create(nil);
  kbmSession := TkbmMW PooledSession.Create(nil);
  kbmQuery := TkbmMWClientQuery.Create(nil);
  with kbmTransport do begin
    host := '127.0.0.1';
    Port := 3000;
    RequestTimeout := 60;
    ConnectTimeout := 30;
    StreamFormat := 'STANDARD';
    StringConversion := mwscFixed;
    VerifyTransfer := True;
  end;
  kbmPool.Transport := kbmTransport;
  kbmClient.Transport := kbmTransport;
  kbmSession.SessionName := DEMO;
  kbmSession.ConnectionPool := kbmPool;
  with kbmQuery do begin
    SessionName := kbmSession.SessionName;
    TransportStreamFormat := kbmStreamFormat;
```



```
QueryService := 'KBMMW_QUERY';  
QueryServiceVersion := 'kbmMW_1.0';  
end;  
end;
```

Use the kbmMWSimpleClient all you want at this point (more on this in a moment)

Freeing resources

I just feel better freeing things up so, in the Page Unload procedure free the controls. Remember if you free the controls before you unload the page you will have to recreate them!

```
procedure TDefaultForm.TDefaultForm_Unload(sender: System.Object; e:  
System.EventArgs);  
begin  
    kbmPool.Free;  
    kbmTransport.Free;  
    kbmClient.Free;  
    kbmStreamFormat.Free;  
    kbmSession.Free;  
    kbmQuery.Free;  
end;
```

Binding data controls in ASP.net

For those new to ASP.net here are a couple import tips.

There are 2 very important things to remember in asp.net that differ from the win32 environment.

1. you must bind data to the webcontrols by calling the controls 'DataBind' method for example Datagrid1.Databind;
2. objects in ASP.Net all need a namespace, for example

```
var  
dt : DataTable;  
begin  
    dt := DataTable.Create('kbmMT'); //dot.net wants a namespace
```

Until future updates come available it is necessary to add your kbmQuery information to a Dataview component. Then you bind your controls to the Dataview component. (located in Data component on the D2005 Tool Palette)

EXAMPLE:

```
procedure TDefaultForm.GetInfo(bTable,bField,bSearchStr:string);  
var
```

```
dt : DataTable;
dr : DataRow;
i : integer;
x : Integer;
begin
    //dotnet wants a namespace
    dt := DataTable.Create('kbmMT');
    // the query
    with kbmQuery do begin
        with query do begin
            Clear;
            if Trim(bSearchStr) > " then
                Add('SELECT * FROM '+bTable+' WHERE UPPER('+bField+') LIKE
                    '+QuotedStr('%'+UpperCase(bSearchStr)+'%'))
            else
                Add('SELECT * FROM '+bTable);

            end;
            TableName := bTable;
            KeyFields := 'ID';
            FetchDefinitions;
            Open;
            First;
            end;

    //assign the results to a Dataview component
    Dataview1.Table := dt;

    //add the results to the Dataview component
    with dt do begin
        BeginLoadData;
        for i := 0 to kbmQuery.FieldDefs.Count - 1 do
            begin
                Columns.Add(kbmQuery.FieldDefs.Items[i].Name, TypeOf(string));
            end;
            i := 0;
            while not kbmquery.Eof do
                begin
                    dr := dt.NewRow;
                    for x := 0 to kbmQuery.FieldDefs.Count - 1 do begin
                        dr[x] := kbmQuery.Fields[x].AsString;
                    end;
                    Rows.InsertAt(dr, i);
                    i := i + 1;
                end;
            end;
        end;
```



```
        kbmquery.next;  
    end;  
    EndLoadData;  
end;
```

```
//add the data to the Grid  
DataGrid1.DataSource := DataView1.Table;  
DataGrid1.DataKeyField := 'EventNo';  
DataGrid1.DataBind;
```

Hope this information is helpful! I try to provide more as I figure out D2005 .net myself ☺

Enjoy

Steve Winsett