

Creating an ISAPI based application server

Starting with kbmMW v. 0.93a its very easy to create an application server which can run under IIS or any other caching ISAPI compliant web server.

A good reason why to put an application server under the control of a web server is for example that you will not have to worry about firewalls. If the customer can access your web server, they can access your application server if they are allowed to.

There is also some minor drawbacks doing this compared to having a standalone application server. There are a bigger overhead due to the extra tier (the web server). This results in typically 10-15ms longer response times compared to a standalone application server.

Ok, lets get to work...

An ISAPI application server is a DLL compared to a standalone application server which is an exe.

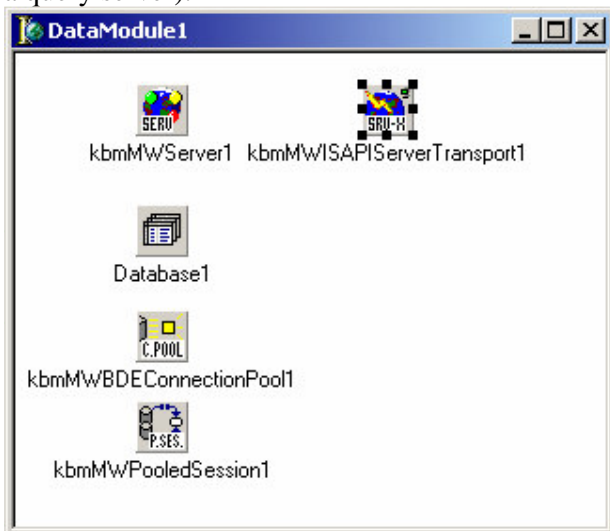
Thus create a new DLL project, File->New->DLL Wizard

Add a new TDatamodule which will hold the TkbmMWServer. File->New->TDatamodule

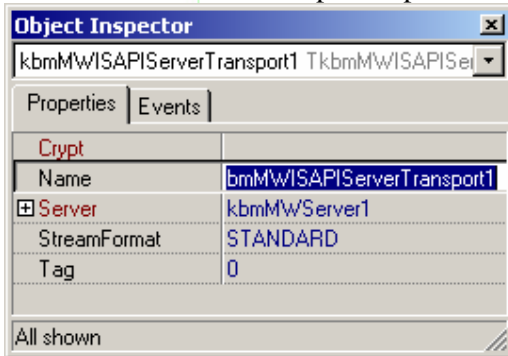
Add a TkbmMWServer to the datamodule.

Add a TkbmMWISAPIServerTransport to the datamodule.

Add other database/connectionpool components as needed (refer to the document Using kbmMW as a query server).



Set the ISAPIServerTransport to point on the kbmMWServer:



Select a StreamFormat (STANDARD is fine here).

Add whatever services you may already have to the project.
Register them for the kbmMWServer in the OnCreate event of the TDatamodule.

What is very important is to add an initialization section to the TDatamodule on which TkbmMWServer is placed:

```
initialization  
  DataModule1 := TDataModule1.Create(nil);
```

Save and compile.

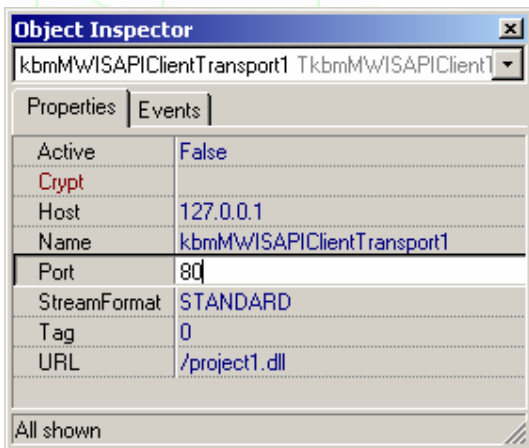
Then you will have an ISAPI dll which you can move into a directory the web server can see.

Creating a client

Follow the same procedures as normal in creating a client. See f.ex. the document 'Using kbmMW as a query server'.

Instead of using `TkbmMWTCPIPIndyClientTransport` or another transport, use the `TkbmMWISAPIClientTransport`.

Set its `Host` and `Port` properties to the IP-address/name of the web server and the `Port` number its serving requests on (typically 80).



Make sure the `StreamFormat` is set the same as on the application server.

This is all what's needed to change on a client to let it contact the ISAPI based application server.

Kim Madsen.