

Creating an ISAPI based application server

Using kbmMW v. 4.xx and SQLite

From start of kbmMW it has been 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 are also minor drawbacks doing this instead of having a standalone application server. One of them is the larger 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 where a standalone application server is an exe.

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

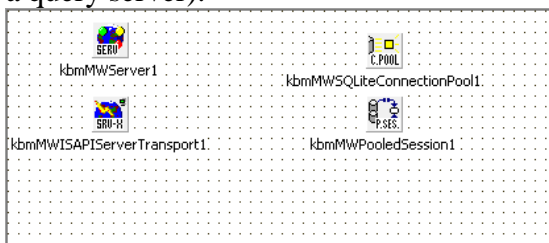
In XE2, add VCL to the unit scope names for Win32 applications. Without it, it will fail to compile. Save the project. In this whitepaper, we have chosen the name for the project (and thus the server DLL) to be kbmMWISAPIServer. The name is important when you setup your clients transport.

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

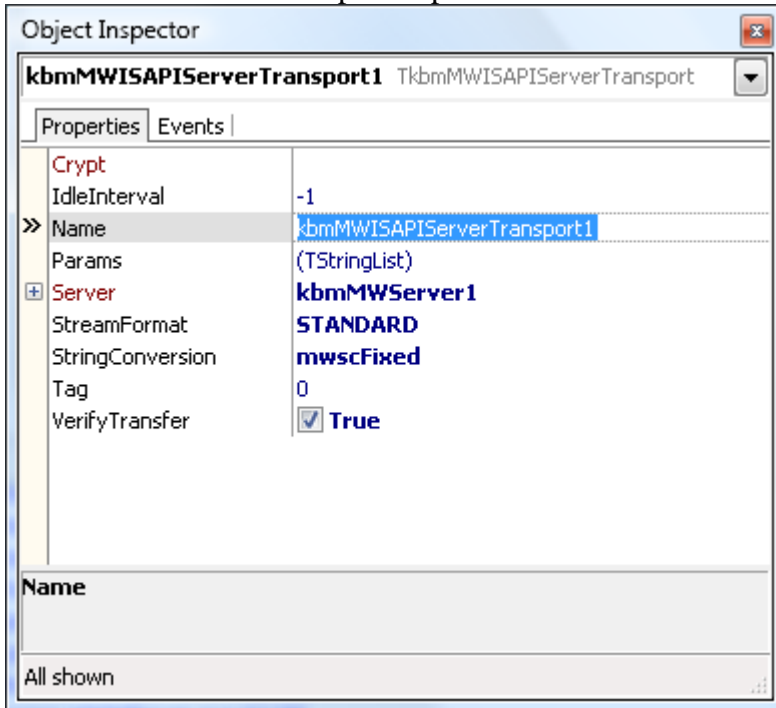
Add a TkbmMWServer to the datamodule.

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.

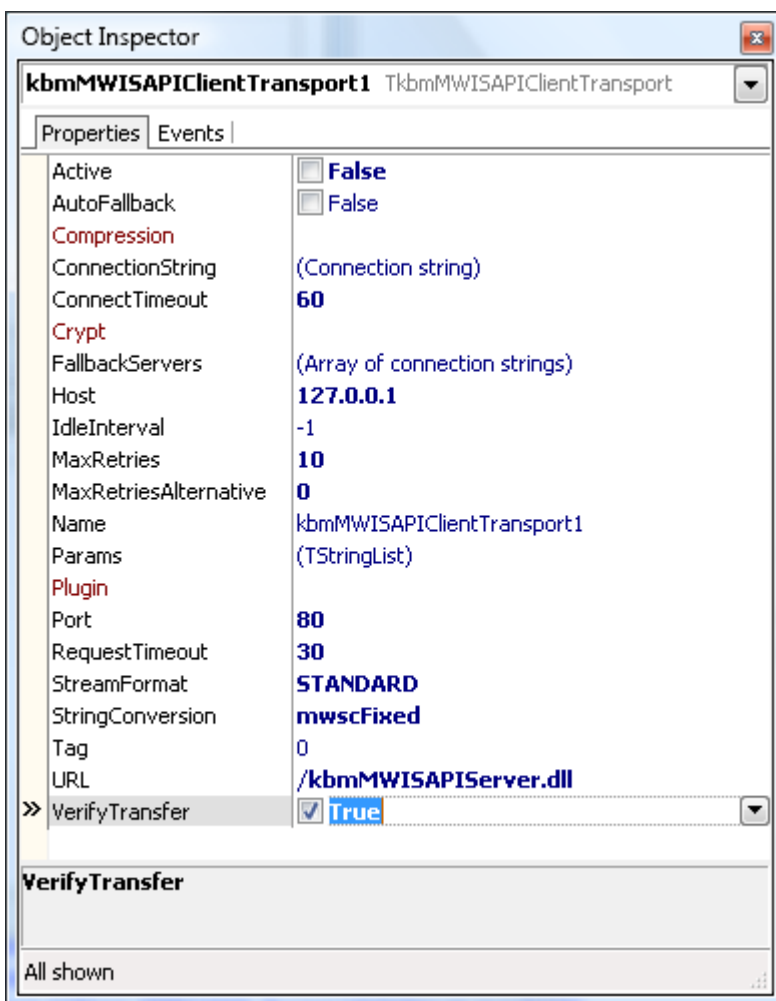
Then you will have an ISAPI dll which you can move into a directory the web server can see.

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 `TkbnMWTCPIPIndyClientTransport` or another transport, use the `TkbnMWISEAPIClientTransport`.

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). You will also want to set the `MaxRetries` to something greater than 0, since some web servers may disconnect the client regularly. Having a value different than 0, makes the client transport trying to reconnect the maximum number of times before giving up, and throwing an exception.



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.