

## **kbmMW C# client library**

The kbmMW C# client library contains native C# based classes that allow a developer to access a kbmMW application server, in a similar way as using TkbmMWSimpleClient with a traditional request/response transport like TkbmMWTCP IndyClientTransport.

The library is compatible with both .Net Standard Framework and .Net Compact Framework, and thus allow targeting PDA's and other types of devices where these frameworks exists.

This document shows how to compile and use the library using Microsofts C# command line compiler, but there is nothing to hinder using an IDE like VS.Net or C#Builder or similar.

### ***Prerequisites***

- a kbmMW application server running somewhere to try out your client application.
- Standard Framework .Net SDK v. 1.1.
- Optionally Compact Framework .Net SDK v. 1.0 SP1 or newer.
- Optionally some .Net capable IDE like VS.Net (required for CF compilation), C#Builder etc.
- The kbmMW C# client library.

The .Net SDK's can be downloaded for free from this place:

<http://msdn.microsoft.com/netframework/downloads/updates/default.aspx>

### ***Whats in the package***

The kbmMW C# client library package contains the following:

- kbmMWClient\kbmMWClient.cs and kbmMWClient\AssemblyInfo.cs which combined are the API itself
- Demo\Form1.cs, Demo\Form1.resx and Demo\AssemblyInfo.cs which combined are a small demo application using the kbmMW C# client library.



## ***Installation***

- 1) If the .Net SDK is not installed, install it first. Let the installation modify your environment variables if requested.
- 2) If you want to compile for Compact Framework, make sure that the .Net CF SDK is installed. Let the installation modify your environment variables if requested.
- 3) Extract the files from the kbmMW C# client library package to a new empty directory. eg. C:\Program Files\kbmMW\CSCClient
- 4) Open a DOS prompt to build the kbmMWClient.dll assembly:
  - a. If to compile for Standard .Net, execute **Compile.bat**
  - b. If to compile for Compact Framework .Net, execute **CompileCF.bat**

After installation, kbmMWClient.dll will have been built. If you have build errors stating missing libraries etc. then your environment variable Path, LIB and INCLUDE have not been set properly during installation of the MS .Net SDK(s).

To fix it, locate **sdkvars.bat** which usually is to be found in **C:\Programmer\Microsoft.NET\SDK\v1.1\Bin** and run it within the DOS prompt. This will set the proper environment variables.

The kbmMWClient.dll assembly can then be installed or referenced from all .Net development environments, and must be copied to end user machines when deploying an application relying on the kbmMW C# client library.



## ***Using the library***

The library contains several classes which generally are modelled according to the same structure as the main Delphi kbmMW client classes. The most important ones are:

- TkbmMWObject
- TkbmMWTCPTransport
- TkbmMWSimpleClient
- TkbmMWGlobal

A basic setup requires an instance of TkbmMWTCPTransport and TkbmMWSimpleClient. Eg.

```
using kbmMWClient;

public SomeMethod() {
    object o;
    int id;

    // Define our instances.
    TkbmMWTCPTransport tr = new TkbmMWTCPTransport();
    TkbmMWSimpleClient sc = new TkbmMWSimpleClient();

    // Hook the transport to the simpleclient.
    sc.Transport:=tr;

    // Setup the connection parameters.
    tr.Host='192.168.1.22';
    tr.Port=3000;

    // Connect to application server.
    tr.Connect();

    // Make a request.
    try {
        o=sc.SendRequest('INVENTORY','','LIST',nil);

        // Check status of request.
        lStatusText.Text=sc.StatusText;

        // Get StateID if any.
        id=sc.StateID;
        lStateID.Text=id.ToString();

        // Show response.
        mResponse.Text=o.ToString();
    }
    catch (Exception ex) {
        lStatusText.Text=ex.Message;
    }

    // Disconnect.
    Tr.Disconnect();
}
```

This sample connects to an application server, requests an inventory list and disconnects.



## TkbnMWTCPTransport

### Properties

**string Host** – Set or get the hostname/IP address of the app. server to connect to.  
**int Port** – Set or get the port number of the app. server to connect to.  
**int ConnectTimeout** – Number of seconds to wait before timing out during a connection.  
**int RequestTimeout** – Number of seconds to wait before timing a request out.

### Methods

**void Connect()** – Connect to the server given by Host and Port.  
**void Connect(string AHost, int APort)** – Connect to the specified AHost/APort.  
**void Disconnect()** – Disconnect from the application server.

## TkbnMWGlobal

### Methods

**public static Hashtable kbnMWRegisteredObjects**

A list of registered object classes which can be used for transfer between client and application server.

**public static void kbnMWRegisterObject(String AName, System.Type AObjectType)**

Register a new object class for transfer between client and application server.  
The registration overrides the default name of the class.

**public static void kbnMWRegisterObject(System.Type AObjectType)**

Register a new object class for transfer between client and application server.

**public static System.Type kbnMWGetObjectTypes(String ATypeName)**

Return an object class for a given class name.



## TkbmMWSimpleClient

### Properties

**public string UserName**  
**public string Password**  
**public string Token**

Specifies the username/password/Token to use while communicating with the application server.

**public string Location**

Specifies the optional location of the client.

**public int StateID**

Get or set the StateID for a stateful service.

**public TkbmMWCustomTransport Transport**

Set to point on a transport instance.

**public string StatusText**  
**public int StatusCode**

Get the status text/code of the last request.

**public const string Version = "1.00b"**

Returns the current version of the kbmMW C# client.

**public MemoryStream ResultStream**  
**public MemoryStream RequestStream**

Write to or read from these streams to stream binary or other arbitrary data to and from the application server.

### Methods

**public bool IsOK()**

Returns true if last request was a success.

**public bool IsWarning()**

Returns true if last request returned a warning.



**public bool IsError()**

Returns true if last request ended with an error.

**public object SendRequest(string AServiceName, string AServiceVersion, string AFunction, object[] AArgs)**

The main method to use to access the application server. Used very similarly to the normal SendRequest method in Delphi.

### ***Preparing an application server***

The application server must have at least one TCPIP request/response transport defined and hooked up to the TkbmMWServer component.

The TCPIP transport must set StreamFormat=STANDARD, StringConversion=fixed and VerifyTransfer=true for the kbmMW C# client to be able to communicate with it.

Kim Madsen  
Components4Developers