

# **RMB/Win Extender**

*“A Rocky Mountain Basic for Windows  
TCP/ActiveX Extension”*

## Getting Started with Evaluation Package

Preliminary

- **RMB/Win Extender Evaluation Package Installation & Removal**
- **Evaluation Package Content**
- **Running Simple Example**
- **Running Test Example**
- **Running the two Examples at same time**
- **Using your Own Application**
- **Evaluation Release Limitations**

## •RMB/Win Extender Installation & Removal

The Evaluation Package is made of two Zipped (compressed) files. The final part of the names can be different due to versions update.

**RmbWextSetupV1\_2.zip**      contains the standard files of a Windows Setup package  
**RmbWextDocsV1\_2.zip**      contains the documentation and sample source files

### Installation

#### •If you have WinZip :

- double\_click **RmbWextSetupV1\_2.zip**
- then double\_click on " Setup.exe " in the WinZip Windows, to launch the Install process
- double\_click on **RmbWextDocsV1\_2.zip**
- extract to the directory where you installed RMBWinExtender

#### •If you don't have WinZip :

- you can download a Winzip Free Evaluation package from " [www.winzip.com](http://www.winzip.com) "
- the documentation can be viewed without Winzip, as it is also in an auto-extractible file **RmbWextDocsV1\_2.exe**
- double\_click **RmbWextDocsV1\_2.exe**
- extract to the directory of your choice, " C:\Program Files\RmbWinExtender " default

### Removal

Deinstallation is provided through standard Control Panel, Add/Remove Programs function.

After deinstallation, you can remove any remaining files in the directory where you have installed RmbWinExtender.

**•Evaluation Package Content**

## \RmbWinExtender Installation Directory

- \Doc
  - EULA.pdf
  - FAQ.pdf
  - Description.pdf
  - UserReferenceGuide.pdf
  - GettingStarted.pdf (preliminary)
- \Rmbw\_app
  - RmbWExt.dll
  - RmbWext.csb
  - RmbWext63.csb
  - RMBWinEtenderLog0.txt
  - RMBWinEtenderLog1.txt
  - Server.ini
  - SimpleServer.prg
  - TestServer.prg
- \VB\_app
  - Rmbwinextender.ini
  - SimpleClient.exe
  - TestClient.exe
- \XLS\_app
  - SimpleClient.xls (not available at this time)
- \CPP\_app
  - SimpleClient.exe (not available at this time)
- \Delphi\_app
  - SimpleClient.exe (not available at this time)
- \Vb\_source
  - \CommonClasses
    - ALGesCombo.cls
    - ALIniFil.cls
  - \SimpleClient
    - Form1.frm
    - Form1.frx
    - SimpleClient.vbp
    - SimpleClient.vbw
  - \TestClient
    - ALSockxTest.frm
    - ALSockxTest.frx
    - common.bas
    - Sequencer.cls
    - TestClient.vbp
    - TestClient.vbw

All files are installed in the directory you chose for RmbWinExtender, expected for :

MSSTKPRP.DLL, TABCTL32.OCX, MSCOMCTL.OCX, MSWINSCK.OCX

that are part of Visual Studio 6.0 redistributable files, installed in the Windows System Directory as shared files.

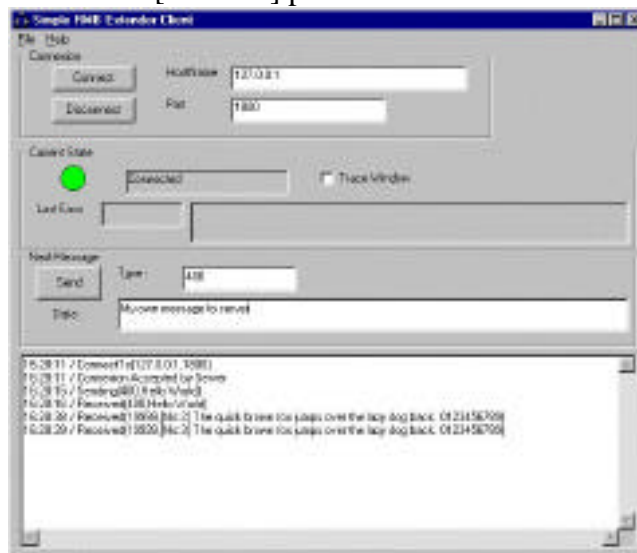
## •Running Simple Example

-Launch SimpleServer.prg (it's a 'stored' form of RmbWin program, the .prg extension would launch HP/HT Basic for Windows and start execution). An other way may be to first open HP/HT Basic for Windows, then load "SimpleServer.prg" and run. When it is running, SimpleServer will start listening automatically and wait for a client connection.

```

HP BASIC in Windows - SimpleServer.prg
Address: E28440 Release: 7.8
Function: SimpleServer:
-Load simple messages
-Display received messages
-Get echo received messages
[16:28:16] Received: Type=000, Data=C:\Hello World!
[16:28:16] Echoed: Type=000, Data=C:\Hello World!
[16:28:16] Sent: Type=1000, Data=C[1]He[2] The quick brown fox jumps over the lazy dog back: 8029A5470955
[16:28:16] Sent: Type=1000, Data=C[2]He[2] The quick brown fox jumps over the lazy dog back: 8029A5470955
[16:28:58] Port=1800, Connected, Echo OK, Msg_received=1, Msg_sent=0
  
```

-Then, launch SimpleClient.exe. On start, SimpleClient is in 'Disconnected' state, you must establish a connection with [Connect] push button.



-Once connected, you can send message to server (echoed by default) and observe message coming from server.

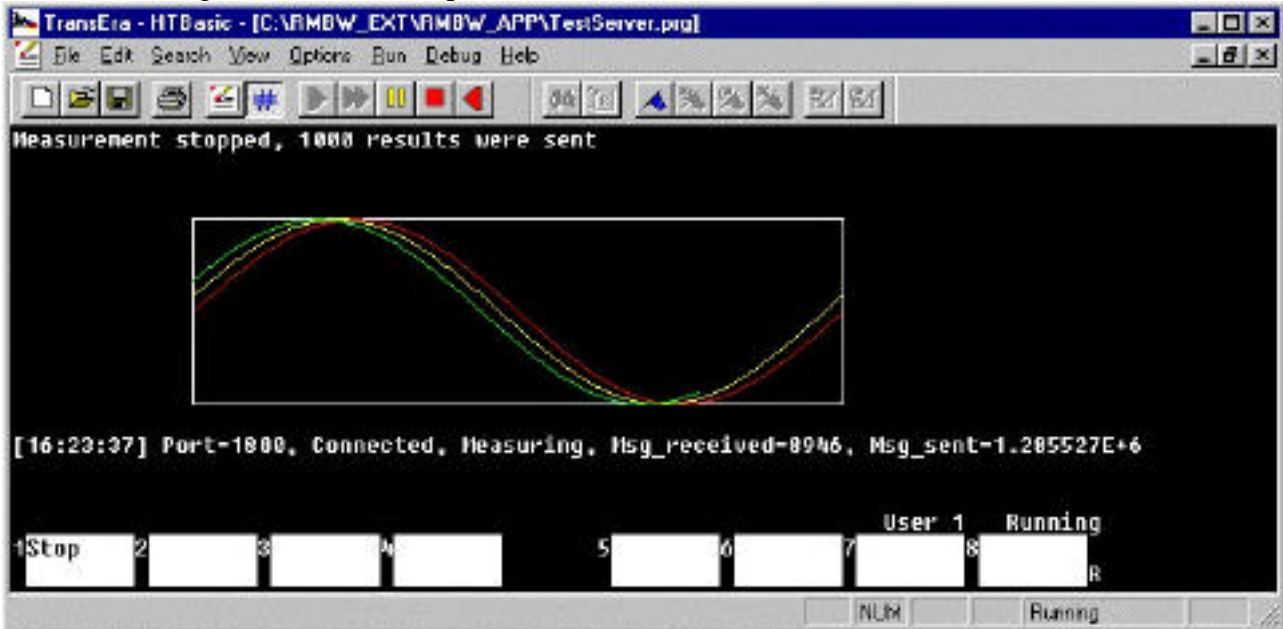
-You may Start/Stop the server using RmbWin function keys, as well as Connect/Disconnect the client with push buttons.

-By default, SimpleClient is configured to deal with SimpleServer on same machine (HostName 127.0.0.1) through port number 1800. As SimpleServer.prg offers connection on port number 1800, all that would run with default configuration.

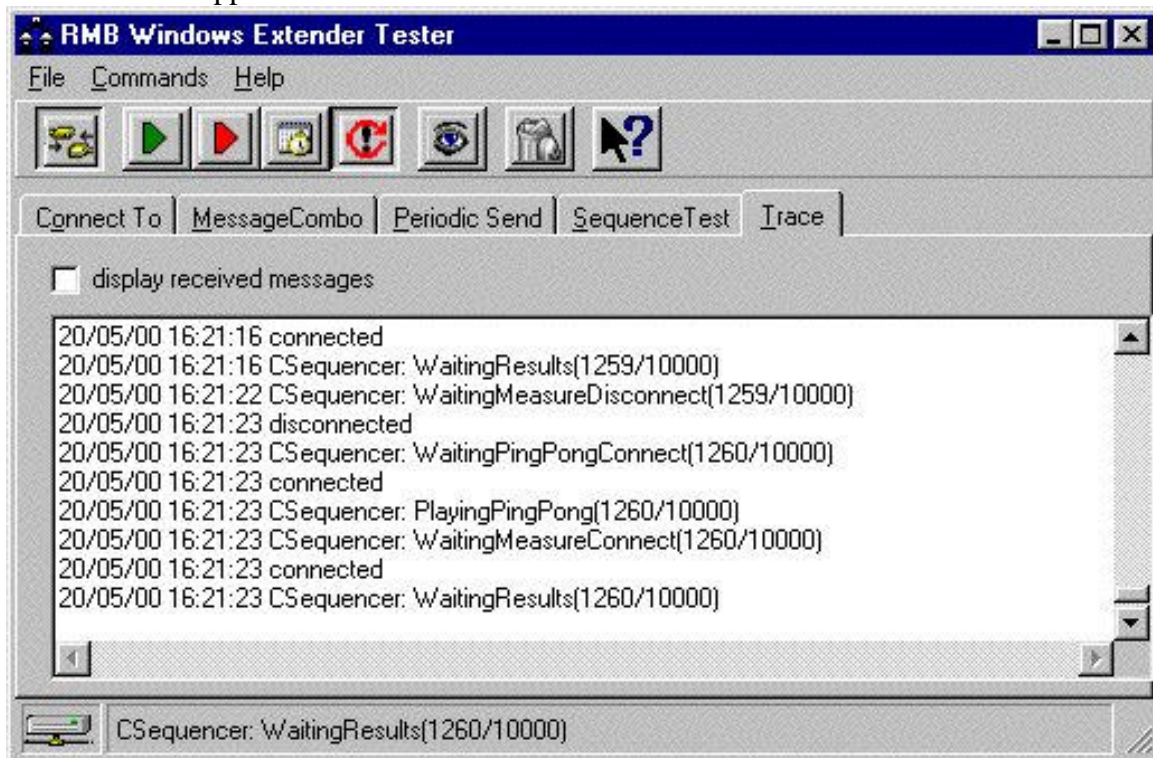
-You may then try to change default configuration, dealing through network, ...

•Running Test Example

-As for Simple Example, first launch the server side TestServer.prg. This RmbWin application will place itself in listen state using the initialization file Server.ini. You will have nothing to do as it will respond to client side.



-After TestClient.exe is launched, you can configure the client side with five panels and control the application with buttons in a task bar :





Connect To

### -Connection button & Connect To panel

Host server name or IP address may be entered in HostName area, use 127.0.0.1 to deal with the same machine.

You can choose Port number to use, it may be the same as the one used by TestServer.prg (1800 by default).

Using the Connect button, you can toggle the connection state, current connection status is reported by an icon at left bottom of the window. Trace panel will indicate TCP report from your Connect/Disconnect try.



MessageCombo

### -Send Button & MessageCombo panel

Message combo enable to select a message from the existing list and to enter a new one. Message must be typed as : TypeNumber,String. Used message for TestServer.prg are :

100,AnyCharString	Ask server to shutdown (Stop RmbWin)
400,AnyCharString	Ask server to return the same string, with 401 type
200,AnyCharString	Ask server to start measurement
201,AnyCharString	Ask server to stop measurement
202,SomeText,100	Ask Server to return 100 points with couple X,Y for each measurement
202,SomeText,nnn	Ask Server to return nnn points with couple X,Y for each measurement

When Send button is used, selected message is sent to the server, to be sent the connection must be established.

You can try any of messages understood by TestServer.prg and have a look on what happens. If you choose to send some unknown message type, the TestServer will only display last received one without any other action.



### -BadSend Button

This button can be used to have a look of what happens if a bad message try to be sent from client side. There are different 'bad messages', before to try be sure the connection is established. To have a better look on what happens, you can use trace dialog on client side and view LogFile on server side.



Periodic Send

### -PeriodicSend button & Periodic Send panel

This function enables to send messages with a controlled wait between each. The controlled wait depends upon system availability and is never less than required one, however it gives a good idea on own things may run with any other application. For example, if you require a zero wait state between messages, RmbWin Extender will ask WinSock layer as long as required messages number isn't sent, system will buffer waiting TCP/IP is able to proceed.

You can tell the application to increment message type or to stay with the same one.

Before to use PeriodicSend button, be sure the connection is established.



SequenceTest

### -SequenceTest button & Sequence Test panel

Test sequencing is dealing with TestServer application, for each sequence :

Client try to connect to server

Client send Ping message (type 400) and on each send wait for Pong reply (type 401)

Client disconnect, then client redo a connection

Client initialize a measurement burst (type 202), indicating number of wanted results

Client start a measurement (type 200)

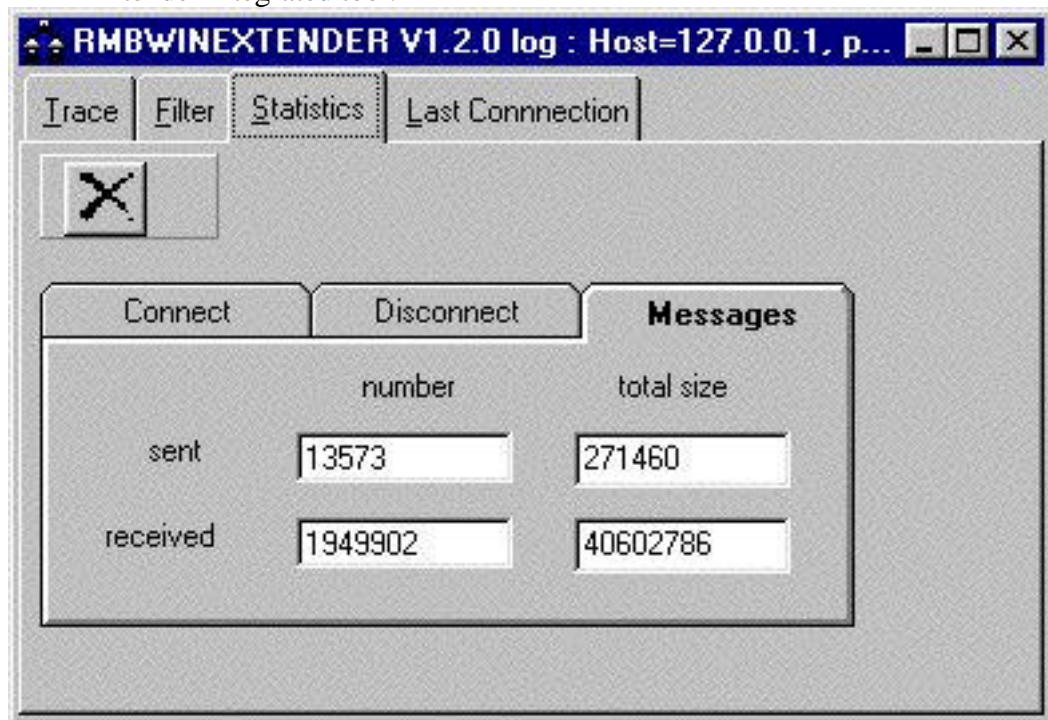
Client wait for results number asked coming from server (type 500)

If anything go wrong during TestSequence, client indicates the problem and stop sequencing.



#### -TraceDialog button

This button enable the ActiveX component to activate/deactivate the trace window. Trace window isn't a part of TestClient application but an RmbWin Extender integrated tool.



Trace

#### -Trace panel & ClearTrace button

Application local trace panel enable you to watch TestClient requests and reports, as an option you can choose to display or not received messages from server. Be aware that displaying received messages can slow down as there may be some high number of messages to print.

ClearTrace button erase the content of trace panel

**•Running the two Examples at same time**

You may try to run more than one client-server application at the same time. You have just to don't use twice the same TCP port on the same machine.

The simpler method is to run at the same time on the same machine :

SimpleServer *with* SimpleClient  
*and*  
TestServer *with* TestClient

If you want to do so, before to launch TestServer, place a port value number different of 1800 in Server.ini file (1800 is the value used by SimpleServer, written inside the RmbWin program). Before to connect with TestClient, change port number in ConnectTo panel according to the value used in Server.ini. Then, you can run each application at the same time exactly in the same manner as when they were alone.

You may also experiment different method, for example with two machines running on a network you can reverse the order and have :

<u>Machine X</u>		<u>Machine Y</u>
TestServer	<=>	TestClient
TestClient	<=>	TestServer

Only remember to don't use twice the same port on same machine.

You may also want to have different LogFile for each server running on same machine. That's true for SimpleServer and TestServer, because they are in two different directories. If you run more than one time the same server application, you can use separate directories or just separate directories for LogFile (see § 3.3 Logging Services in User Reference Manual).

...

**•Using your Own Application**

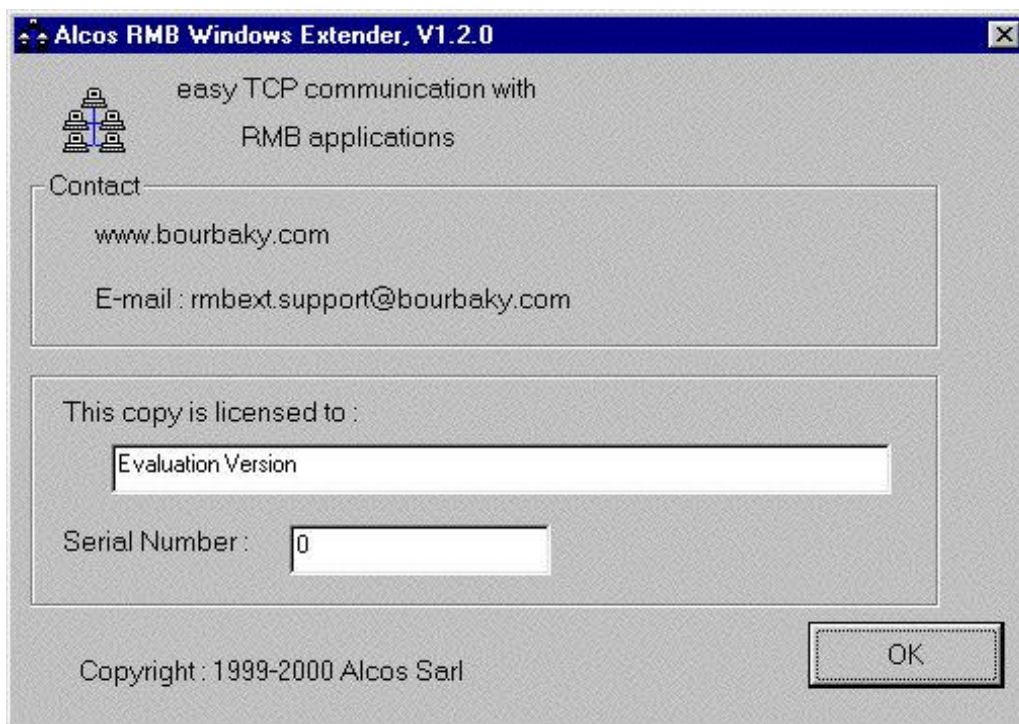
RmbWin Extender evaluation package isn't just provided samples, it contains the full product (in evaluation release) and documentation. Using ideas from inside samples source code, you can experiment your own application on server or client side, and even on the both side from your own software.

For Server in RmbWin, you can use evaluation release from Bourbaky web site (Rev 6.3 or Rev 7.1).

For Client, if you have Excel (97 or higher) you can build some application in Visual Basic for Application.

**•Evaluation Release Limitations**

After thirty days, the evaluation version ceases to execute the Connection requests. At each client launch, the evaluation release pops up a validation windows.



Once you get your regular Serial Number and type it inside the required place this windows will not pop up at launch time and Connection requests are usable.