

Contents

Contents	1
wBridge.dll.....	2
Installation.....	2
wBridge.dll location.....	2
wBridge.dll registration.....	3
Microsoft required kits.....	3
Mettler Toledo weighbridge cabling.....	4
Rinstrum and Sabre DWI weighbridge cabling	5
Dll weigh function call and arguments	5
Port.....	5
Weight.....	5
Tally	5
UOM	5
Places	5
The calling mechanism	6
Flag values	9
Hardware loopback mode	10
Rinstrum or Sabre DWIs.....	10
Mettler Toledo Weighbridge.....	10
Software loopback mode.....	10
Weigh function return codes	11
Default values	12
Auxiliary dll function calls.....	12
Serial port config.....	12
Rinstrum port	12
Sabre port.....	12
Mettler Toledo port.....	12
Default key entries	13
Common default values.....	13
Rinstrum port default values	13
Mettler Toledo port default values	13
testwb - a weighbridge diagnostic aid	14
Execution	14
Main window	15
Rinstrum strings window	16
Rinstrum command and response strings.....	16
Sabre command and response strings.....	17
Mettler Toledo command and response strings.....	17
Shering weighbridge simulator	18
Development environment.....	18
Pre-requisites.....	18
Appendix 1.....	19
Rinstrum RS-232 comms	19

wBridge.dll

WBridge.DLL is a DLL Written so that applications can use the weighbridges automatically. Its purpose is to define an interface to the Weighbridges. The dll supports the Mettler Toledo weighbridges and the Shering weighbridges equipped with **Rinstrum 420** or **Sabre 825** display weight indicators (DWIs).

There are two versions of the dll; one for a 64-bit platform, and the other for 32-bit platforms. Both have the same name but are distinguished by version number. Right-clicking the dll and examining the properties will show that the last field of the version number is either 86 (for 32-bit platforms) or 64 (for 64-bit platforms). The correct dll should be installed for the appropriate platform. On delivery, the dlls are packaged in zip kits identifying the target platform.

The dll's can be downloaded from the developer's web site at www.PetersEnterprise.Me.UK and will operate 100 times on evaluation. The dll is node-locked to a machine, so after purchase it will be necessary to contact the developer with the host name of the target machine; a license file will be returned which should then be copied to the same folder as the log file; by default this is **c:\ProgramData\wBridge**, but can be changed by editing appropriate registry setting. Careful use of **regedit.exe** may be required; alternatively use the **testwb.exe** diagnostic which can perform the registration for you.

Installation

wBridge.dll location

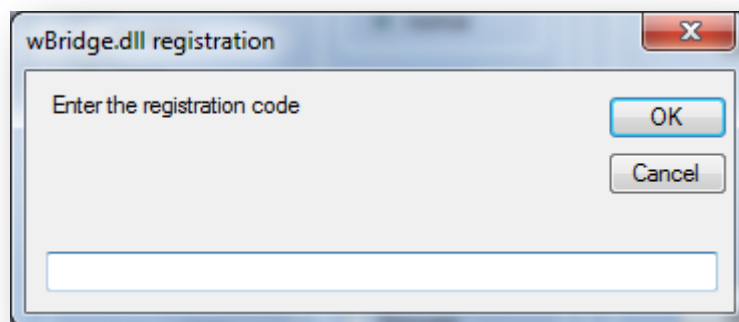
WBridge.DLL should be copied to the **\Windows\System32** folder for use, or should be in the PATH, or referenced directly; if using side-by-side techniques in Windows, the dll should be placed in the same folder as the executable that will reference it, along with a text file (which may be an empty file) with a name of <application.exe>.local (later versions of windows allow a **folder** with a name of <application.exe>.local that contains any required dlls).

If the operating system platform is a 64-bit environment, it will be necessary to copy the appropriate dll to **\Windows\SysWOW64** folder.

The port settings are pre-set within the DLL, but can be changed in the registry. By setting bit 2 (value=4) of the **flags** argument the current values for the port settings will be added to the registry where they can be edited if desired by using standard Windows tools (**regedit.exe**). Refer to the Windows API manual for details on the settings for the device control block (**DCB**).

wBridge.dll registration

The wBridge.dll will operate for 100 times on evaluation (that includes the simulation operation). The dll operates on a node-locked key, so it will be necessary to email the developer with the host name for the target machine; an appropriate key will be returned. The registration code can be entered by using the **testwb** diagnostic aid available as a separate download from <http://www.PetersEnterprise.Me.UK>. Click on **File>wBridge.dll** to see the dialog box as shown below; click on **Register** to enter the code provided.



Microsoft required kits

There is a known problem with applications that have been built using visual Studio 2005 or later, in that there are missing Microsoft dlls on a non-development machine. For a full description of the error refer to <http://support.microsoft.com/kb/948619> the redistribution kit required to resolve the issue will depend on the environment in which the dll was built; the kit included here is for Visual Studio 2005 (used to build versions 1, 2 and 3 of the 32-bit version of wBridge.dll). The necessary prerequisites for the later versions are included in the testwb - a weighbridge diagnostic aid application: -



Mettler Toledo weighbridge cabling

Where there are two weighbridges, two PCs can be cabled so that either PC can read from either weighbridge, giving an economic form of failover. The PCs will each need two COM ports for this infrastructure. Weighbridges are often positioned at the entrance and exit of a site (which may be very close together) and controlled from a single gatehouse, with one weighbridge assigned to the IN function and the other to the OUT function. If either weighbridge fails, the other weighbridge can be made to handle both functions. The schematic below is specific to the Mettler Toledo weighbridges.

For the Mettler Toledo weighbridges, it is expected that weighbridge1 (In Bridge) is cabled to COM1 and that weighbridge2 (Out Bridge) is cabled to COM2. The normal setup is that two PCs are arranged via automatic T-Switches and current loop converters to the appropriate bridges. See figure 1.

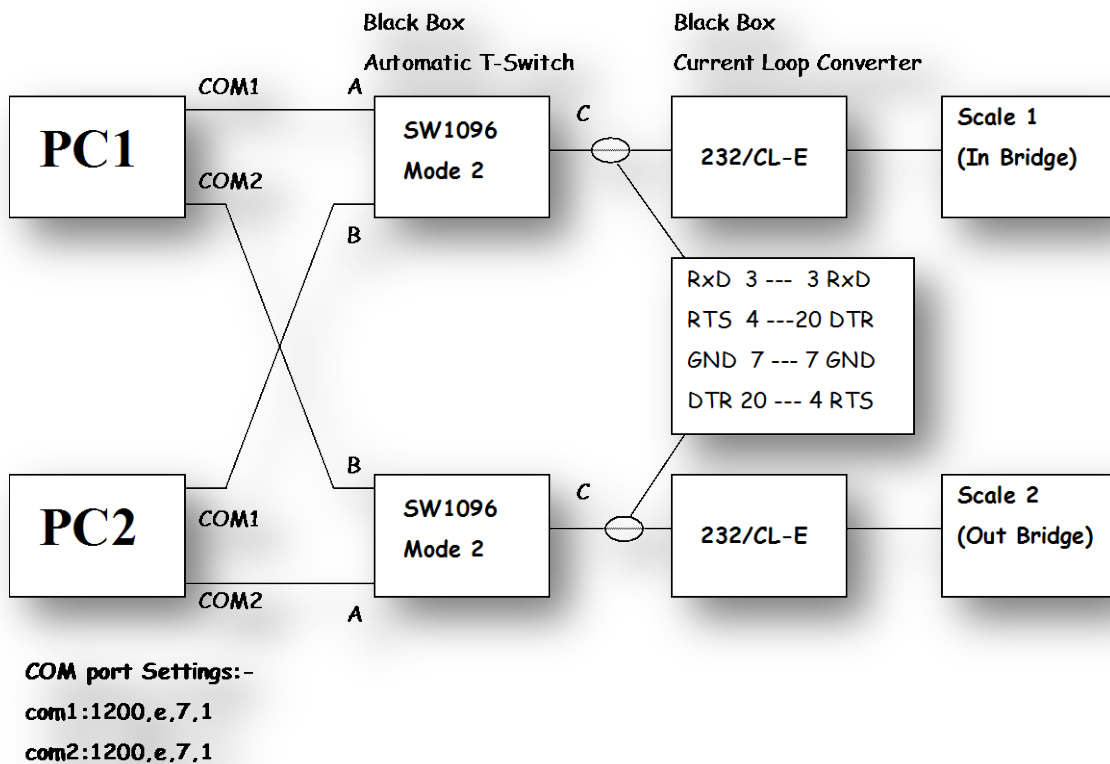


Figure 1, Weighbridge cabling

The PCs are cabled to the T-Switches in a manner that will cause least activation of the switch to prolong its life. I.E. COM1 on PC1 is cabled to port A of the T-Switch, whilst COM2 of PC2 is cabled to port A of the T-Switch.

Logically, then, for either PC, scale1 appears cabled to COM1, making it easy for software application configuration.

Note that the cable between the automatic T-Switch and the current loop converter is not a standard serial cable. This weighbridge is triggered electrically, and the RTS signal in the serial port handles this.

Rinstrum and Sabre DWI weighbridge cabling

For the Rinstrum DWIs (Display Weight Indicators), one PC is cabled directly to each DWI; there is no facility for the redundancy as provided using the Mettler Toledo weighbridges. Use a standard serial cable for connection to a Rinstrum or Sabre DWI. Shering do not recommend the use of USB to Serial devices; the PC should be equipped with a serial port.

Dll weigh function call and arguments

Version 4 or later of wBridge.dll has a new calling interface and is not compatible with earlier releases; a coding change will be required; extra arguments have been added to handle the Sabre DWI.

Port

This is an integer value and represents the port connected to the weighbridge; 1=COM1, 2=COM2, etc.

Weight

Notice that this argument is a floating point value (Sabre DWI can return a floating point value). Previously this was an integer. Declare this as `single` in your applications.

Tally

The Tally argument is a string and must be sized before calling the dll; a minimum of 20 spaces is required and not packed will null character values.

For Mettler Toledo weighbridges, the tally will be a string representation of a number (so it can be converted to a number using the VB function `CInt(tally)`).

For the Shering Rinstrum DWIs, the tally will be the concatenation of the DWI serial number (8 chars, zero-padded) and the `tally` number (9 chars, zero-padded); both are required for the weights and measures regulations. The serial number is the left-most part of the string (e.g. "sssssssstttttttt", where ssssssss represents the serial number and tttttttt represents the tally).

UOM

UoM is a string that returns the units of measure for the instrument; it will be either 1 or 2 characters (depending on the weighbridge). Allow up to 4 spaces; do not pack with null characters.

Places

This argument is an integer and returns the number of decimal places in the numeric weight value; it is only used in the Sabre DWI, but will return zero for other weighbridges. With the appropriate flag set, the weight can automatically be arranged to return the scaled value (as opposed to the raw value); the default is to return the raw value.

The calling mechanism

The calling mechanism for VBA code (e.g. Access 2000, Excel 2003) is now as follows:-

```

` Calling from VBA code on windows 7 64-bit - uses SysWOW64 folder
` VBA declaration in Excel 2007
Declare Function Weigh Lib "wBridge" Alias "weigh" _
    (ByVal port As Long, _
    ByRef Weight As Single, _
    ByVal Tally As String, _
    ByRef status As Long, _
    ByVal uom As String, _
    ByRef places As Long, _
    ByVal flags As Long _
    ) As Long

```

Figure 2, VBA code declaration

For use when developing in visual Studio 2005 (or later), the calling mechanism is different: -

```

` calling from applications developed in visual Studio 2005 (or later)...
Declare Function Weigh Lib "wBridge" Alias "weigh" _
    (ByVal port As Integer, _
    ByRef weight As Single, _
    ByVal tally As String, _
    ByRef status As Integer, _
    ByVal uom As String, _
    ByRef places As Integer, _
    ByVal flags As Integer _
    ) As Integer

```

Figure 3, Visual Studio declaration

Note that the definitions of the numeric arguments are different; the reason for the difference lies VB. In VB6 the size of an integer is 16 bits, whereas in Visual Studio 2005 or later the size of an integer is 32 bits. Since the C declaration of these arguments are `int`, in this function they are sized as 32-bit values.

The **Tally** string **must** be pre-defined before calling the DLL and must accommodate at least 20 characters. See the example VB code and the example VBA code in the following snippets.

Similarly the **uom** string must be pre-defined and must accommodate at least 4 characters.

The enclosed example Excel 2007 spreadsheet show a simple use of wBridge.dll ; it uses VB macros to work and therefore will require user acceptance to operate correctly; watch for the use of named ranges. No protection or security has been applied to this spreadsheet.



```
` Visual Studio 2005 application...
Declare Function Weigh Lib "wBridge" Alias "weigh" _
    (ByVal port As Integer, _
     ByRef weight As Single, _
     ByVal tally As String, _
     ByRef status As Integer, _
     ByVal uom As String, _
     ByRef places As Integer, _
     ByVal flags As Integer _
    ) As Integer

Dim portNum as Integer
Dim weight as Integer
Dim flags as Integer
Dim retVal as Integer
Dim tally as String
Dim status as Integer
Dim places as Integer
Dim flags as Integer
Dim uom as String

tally = Space(20);
uom = Space(4);

Try
    retVal = Weigh (portNum, weight, tally, status, uom, places, flags)
Catch
    MsgBox "Check arguments and presence of DLL"
End Try

If retVal < 0 Then MsgBox "Got an error"
If retVal >0 Then MsgBox "Got a warning"

If retVal = 0 Then
    'Continue with application...
End If
```

These lines are important to avoid the function attempting to write to a string that has not been established.

Figure 4, Visual Studio2005 sample code

```

' VBA declaration in Excel 2007
' tested on windows 7 64-bit with wBridge.dll in \Windows\SysWOW64 folder

Declare Function Weigh Lib "wBridge" Alias "weigh" _
    (ByVal port As Long, _
    ByRef Weight As Single, _
    ByVal Tally As String, _
    ByRef status As Long, _
    ByVal uom As String, _
    ByRef places As Long, _
    ByVal flags As Long _
    ) As Long

Public Sub perform_weigh()

    Dim retVal As Long
    Dim portNum As Long
    Dim Weight As Single
    Dim Tally As String
    Dim status As Long
    Dim uom As String
    Dim places As Long
    Dim flags As Long

    portNum = Range("port").Value
    flags = Range("flags").Value

    Tally = Space(20)
    uom = Space(4)

    ' this code uses named ranges...
    Range("result").ClearContents
    Range("weight").ClearContents
    Range("tally").ClearContents
    Range("status").ClearContents
    Range("uom").ClearContents
    Range("places").ClearContents

    retVal = Weigh(portNum, Weight, Tally, status, uom, places, flags)

    Range("result").Formula = retVal
    Range("weight").Formula = Weight
    Range("tally").Formula = "" & Tally ' make sure this is seen as a string...
    Range("status").Formula = status
    Range("uom").Formula = uom
    Range("places").Formula = places

End Sub

```

These lines are important to avoid the function attempting to write to a string that has not been established.

Figure 5, VBA sample code

Where **portNum** is the COM port number to read from (1=COM1), **weight** is a floating point variable to receive the weighbridge value; UoM will also be returned for Rinstrum or Sabre DWI's), **tally** is a string with a minimum length of 20 characters to receive the tally, which consists of a 8-digit decimal serial number (zero-padded) concatenated with a nine-digit decimal number (also zero-padded). **flags** is usually used as a test and diagnostic mechanism that can accept many values, some of which may be required to distinguish the type of weighbridge that is being read. The **flags** form a mask so that many items can be held within the one value. Some items are mutually exclusive (such as reading display weight and requesting firmware version); in such a case, the function will return an error.

- The flag setting (value=&h8) is required to read from a Rinstrum DWI;
- The flag setting (value = &h10) is required to read from a Sabre DWI.

Flag values greater than 8 are specific to either the Rinstum or Sabre DWIs.

Flag values

Flag values and their meanings are defined in the table below: -

Value	Meaning
&h1	Hardware loopback. This simulates a weighbridge response; the computer does not need to be connected to a weighbridge, but this can be used to test both the port and the cable provided a loopback adaptor is available and used.
&h2	Software loopback. This also simulates a weighbridge but does not require a loopback adaptor because the loopback is performed within the DLL; however, a COM port is required, accessed and checked. No data is sent to or received from the COM port.
&h4	Request the DLL to save port settings to the registry.
&h8	Weighbridge is a Rinstrum DWI (display weight indicator). Necessary because the requests and responses are entirely different from the Mettler Toledo or Sabre devices.
&h10	Weighbridge is Sabre DWI . Necessary because the requests and responses are entirely different from the Mettler Toledo or rinstrum device.
&h20	Read the display weight. This value is not valid under the weights and measures regulations. No tally reading is available under this setting.
&h40	Send set-zero calibrate command to Rinstrum DWI
&h80	Set DWI reading to "in motion" (in loopback mode only). Test purposes only, this will check that the DLL and the diagnostic application can recognise the returned value. The diagnostic application will report this as a warning, but the value should be treated as unreliable and the user should perform a re-read.
&h100	Read DWI firmware value. This value will be returned in the tally field; the weight will be zero. Rinstrum only.
&h200	Dump the Tx and Rx buffers to log file for analysis. The communication strings will be dumped to a log file for analysis.
&h400	Read DWI version number. This value is return in the tally field; the weight will be zero. In loopback mode the value returned will be the version number of the DLL.
&h800	Read DWI units of measure. This value is returned in the UOM field; the weight will be zero. Rinstrum only.
&h1000	Return DWI serial number. This value is automatically returned as part of the trade weight details, but in this case only the serial number is returned in the tally field; the weight will be zero.
&h2000	Return the system status. This is returned as an 8-char hex string in the tally field. It will contain flags regarding stable , in motion , underload , overload , zero , centre-zero . The weight will be zero. The significant bits are: - <ul style="list-style-type: none"> • &H00000400 = Zero. • &H00000800 = Centre of Zero • &H00001000 = Motion • &H00010000 = Underload • &H00020000 = Overload Rinstrum only.
&h4000	In loopback mode only, this is used to set the overload status as returned in the trade weight response; for test purposes only.
&h8000	In loopback mode only, this is used to set the underload status as returned in the trade weight response; for test purposes only.
&h10000	In loopback mode only, this is used to set the centre-zero status as returned in the trade weight response; for test purposes only.
&h20000	Return the device type. Sabre DWI only.
&h40000	Display the return weight. Sabre DWI only.
&h80000	Return the non-trade weight. Sabre DWI only.

Hardware loopback mode

In hardware loopback mode, the dll will simulate weighbridge responses. This aid allows a serial line to be tested by attaching a loopback adaptor either at the port on the PC or at the remote end of the cable to test the cable's integrity.

Rinstrum or Sabre DWIs

For the Rinstrum or Sabre DWI, a request is transmitted via the selected serial port; the dll will listen for the string to be echoed, and then generate a simulated weighbridge response which will be returned to the user.

Mettler Toledo Weighbridge

For the Mettler Toledo weighbridge, a suitable response is generated and then transmitted via the serial port; the dll then listens for this to be echoed back and returns the result to the user.

Software loopback mode

In software loopback mode, the dll will simulate weighbridge responses; a serial port is required, because there are checks made on the validity of the port even though no actual activity is made via the port.

In addition, for the Rinstrum or Sabre simulation, trade weights that are greater than 40000 kg will return an overload status, and trade weights that are less than 50 kg will return a status of underload.

Weigh function return codes

The function returns a value of zero if it's successful. Errors are indicated by negative values and warnings by positive values. A summary of the return values is given in the following table.

Value	Meaning
0	Success
-1	Error - port number is out of range (1-255 is acceptable)
-2	Error opening port (non-existent or already in use?)
-3	Error from clrDTR windows function
-4	Error from clrRTS windows function
-5	Error from getDCB windows function
-6	Error from setDTR windows function
-7	Error from purgeComm windows function
-8	Error attempting to toggle DTR signal on COM port
-9	Error attempting to write to port
-10	Error attempting to read from port
-11	Error attempting to close port
-12	Error reading port – insufficient characters reads.
-13	Error in interpreting string – no CR seen in string
-14	Error in interpreting string – invalid sumcheck value
-15	Error setting RTS signal on COM port
-16	Error from clrComm windows function
-17	Error from setTimeouts windows function
-18	Error from setDCB windows function
-19	Error from buildDCB windows function
-20	Error from clrBreak windows function
-21	Error attempting to claim port (already in use?)
-22	Error - Mutually exclusive flags set
23	Warning - the weighbridge reports in motion
24	Warning - the weighbridge reports over load
25	Warning - the weighbridge reports under load
-26	Error - unable to read weighbridge load state (possible corruption)
-27	Error - unable to read weighbridge state (possible corruption)
-28	Error - set-zero calibration failed
-29	Error - no STX in string (required for lrc sumcheck routine)
-30	Error - tally buffer is too small (at least 20 characters required)
-31	Error - buffer is excessively large (possibly a coding error)
-32	Error - unable to create registry settings
-33	Error - string overflow
-34	Error - invalid string
35	Warning - evaluation grace period
-36	Error - evaluation expired
-37	Error - unregistered wBridge.dll
38	Warning - evaluation copy of wBridge.dll
-39	Error result failure
-40	Error – invalid license file
-41	Error – failed to open license file

Default values

Initialisation data is stored in the registry. An attempt is made to first create the key in the **HKEY_LOCAL_MACHINE** hive; if this fails (possibly by running the application on Windows NT which has increased system security), the key is created in the **HKEY_CURRENT_USER** hive (e.g **HKEY_CURRENT_USER\Software\wBridge**). If it becomes necessary to write to **HKEY_LOCAL_MACHINE** hive on an NT platform, the application will be required to run with elevated access rights (run the application as an administrator).

Many entries are port-specific (baud rate, parity, etc), so the keys created have the port number appended to them; **retries**, **dump** and **log** are not port-specific.

Auxiliary dll function calls

There are some other functions within wBridge.dll which may be useful to developers: -

```

Declare Function lrc Lib "wBridge" Alias "lrc" _
    (ByVal buffer As String, _
    ByVal buflen As Integer _
    ) As Integer

Declare Function crc16 Lib "wBridge" Alias "crc16" _
    (ByVal buffer As String, _
    ByVal buflen As Integer _
    ) As Integer

Declare Function crc32 Lib "wBridge" Alias "crc32" _
    (ByVal buffer As String, _
    ByVal buflen As Integer _
    ) As Integer

```

Figure 6, auxilliary function calls

These routines are standard sumcheck functions often used in serial communications.

LRC is the longitudinal redundancy checksum algorithm, and the CRC routines are cyclic redundancy checksum algorithms. The LRC routine forms part of the Mettler Toledo string.

Serial port config

Rinstrum port

The Rinstrum port default configuration is 9600 baud, 8-bit, no parity, and 1 stop bit.

Sabre port

The Rinstrum port default configuration is 9600 baud, 8-bit, no parity, and 1 stop bit.

Mettler Toledo port

The Mettler Toledo port default configuration is 1200 baud, 7-bit, even parity, and 2 stop bits.

Default key entries

The default values can be found in the registry after performing a "dump settings". This will allow changes to be made by editing the registry using standard Windows tools (regedit.exe). The values will be found under the key `\Software\wBridge` in the `HKEY_LOCAL_MACHINE` or `HKEY_CURRENT_USER` hive (depends on windows security). The values listed below show the settings for the Rinstrum weighbridge: -

Common default values

Dump=0
log=c:\ProgramData\wBridge\wBridge.log
licFile=c:\ProgramData\wBridge\wBridge.lic
Retries=3

Rinstrum port default values

Baud=9600
Data=8
Stop=1
Parity=none
Rinstrum=1
TO=on
ReadTotalTimeoutConstant=500
ReadTotalTimeoutMultiplier=500
WriteTotalTimeoutMultiplier=100
ReadIntervalTimeout=10
WriteTotalTimeoutConstant=100

Mettler Toledo port default values

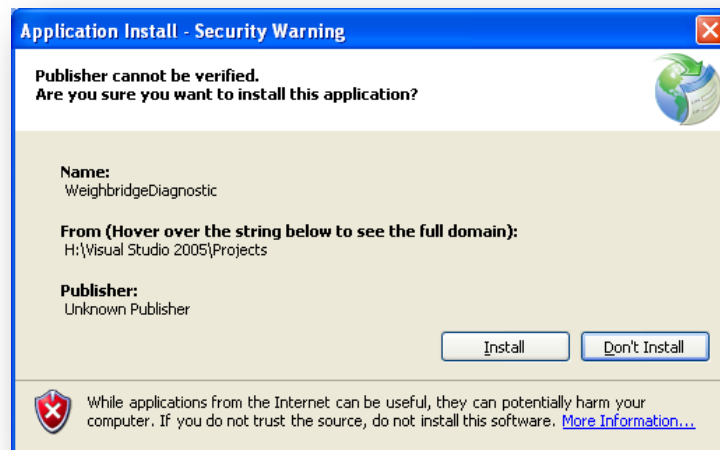
Baud=1200
Data=7
Stop=2
Parity=no
msPause=500
msPulseLen=100
Digital=1
TO=on
ReadTotalTimeoutConstant=500
ReadTotalTimeoutMultiplier=500
WriteTotalTimeoutMultiplier=100
ReadIntervalTimeout=10
WriteTotalTimeoutConstant=100

testwb - a weighbridge diagnostic aid

Accompanying this wBridge.dll is a simple diagnostic aid which can be used to test the functions of the DLL. It comes as an installation kit; install by double-clicking setup.exe. Installation is very quick, followed by the application starting.

The installation kit can be downloaded from <http://www.PetersEnterprise.Me.UK> it is not included in this document because of its size (20MB).

The application has been compiled on Windows 7 premium using Visual Studio 2010 and targeted for **x86**). The application has been tested on Windows XP 32-bit and Windows 7 64-bit platforms. The diagnostic is installed by running setup.exe (as any normal Windows installation). When the Application install security warning appears, click **Install**. After the installation, the application executes immediately.



Execution

Select the required port to which the weighbridge is connected; if the computer only has a single COM port this will be pre-selected. If the computer does not have a COM port, a message to this effect will be displayed; since this application requires a COM port no further progress can be made.

Set other options as necessary (in particular the weighbridge - Rinstrum, Sabre or Mettler Toledo); some options are only enabled under certain settings to avoid mutually exclusive settings). If there is no actual weighbridge, check the loopback box to simulate a weighbridge. A hardware loopback will require a physical loopback connector on the serial port, because requests are sent out over the cable and need to be returned to test the communications. A software loopback still requires a physical port, though no data is transmitted via the port. Both loopback options will make the dll simulate a weighbridge, useful for developers during code development.

Clicking the **weigh** button should display a weight and tally in the appropriate text boxes. If the selected weighbridge is Rinstrum or Sabre, other items from the DWI can be returned (such as version number); this detail is returned in the Tally field with the weight blanked.

If a message indicating a dll error is displayed, confirm that the dll appropriate for the platform in use is installed; also check the calling definition. There is a difference in the headers of 32-bit dlls and 64-bit dlls which renders them incompatible.

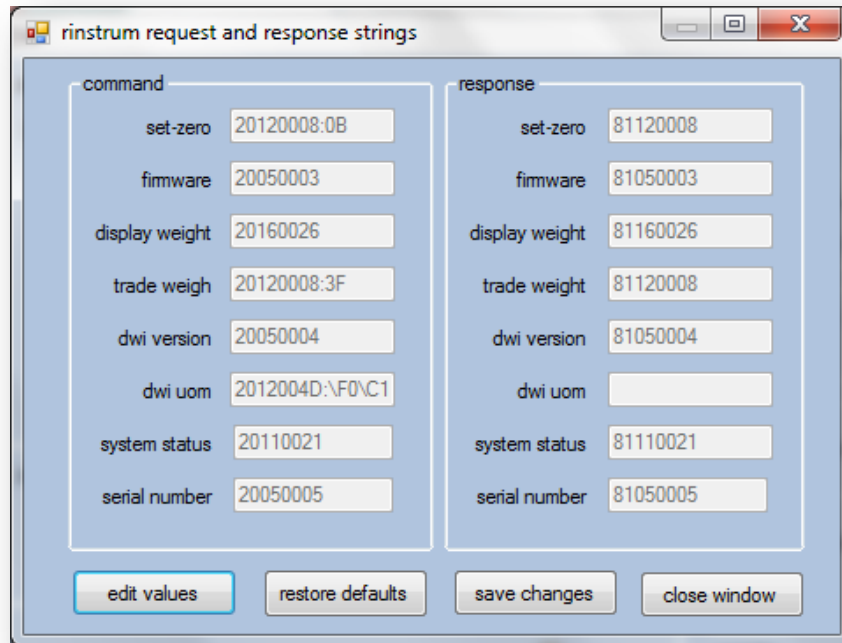
Errors are reported in the file `c:\ProgramData\wBridge\wBridge.log` (this may vary - check the log file location shown in the application window). This file may need occasional truncation, though generally, following the axiom that "no news is good news", the file will be largely empty unless there are errors. More diagnostic messages are thrown into this file if the *hex dump* checkbox is checked.

Main window

Changes made to any settings on this form are not activated by the dll unless the **save settings** button is clicked; settings will automatically be saved between sessions.

Rinstrum strings window

These values are not likely to change unless a new version of the Rinstrum DWI requires new command strings. This screen can, however, be used to restore values in the registry.



Rinstrum command and response strings

Command strings need to be terminated with a semi-colon, response strings are terminated with CRLF.

Command	Command string	Response string
Firmware	20050003	81050003:K4xx
Version	20050004	81050004:vn.n(string)
Serial number	20050005	81050005:nnnnnnnn (space-padded decimal value)
Set-zero	20120008:0B	81120008:0000(fixed response)
Display weight	20110026 20160026 20050026	81110026:xxxxxxx (value in hex) 81160026:nnnn (value in decimal) 81050026: nnnn kg G (7-char literal string)
Trade weight	20120008:3F	81120008:0000CRLFnnnnnnnnkgssssssnnnnnnnnSI s=serial number with leading zeroes S=stable, M=in motion I=in range, O=over load, U =under load
units of measure	2012004D:\F0\C1	xxx (3-char string only)
status	20110021	81110021:xxxxxxx (value in hex) 00020000 = Overload 00010000 = Underload 00001000 = Motion 00000800 = Centre of Zero 00000400 = Zero.

Sabre command and response strings

The Sabre command and response strings handled by this dll are shown in the form below. For further information refer to the Shering PDF document Sabre825.pdf.

Mettler Toledo command and response strings

There is no command string to trigger the Mettler Toledo weighbridge; rather it is an electrical signal. This is arranged by toggling the **RTS** signal in the serial port (hence the non-standard cable to the weighbridge). The response is a string sequence of the format: -

```
<XON><STX>ssNNNNNsCCsNNNNNN<CR>x<LF>
```

Where

- **s** = white space
- **N** = decimal digit
- **C** = Character
- **x** = Irc sumcheck character

Shering weighbridge simulator

Shering have developed a Rinstrum weighbridge simulator allowing a full development environment to be built. This tool creates a virtual serial port which can be used by the wBridge.dll and the diagnostic application described here. See the Shering documentation on the simulator for full details.

Development environment

Pre-requisites

To develop with the Shering simulator, the following need to be installed: -

- Serial-Tools Runtime. It is unlikely that the Virtual Serial Port runtime will install on a 64-bit Windows-7 machine; if it does or if it can be made to do so, then it provides a 32-bit interface so needs to be called from an application compiled for x86
- Visual Studio .Net runtime version 4
- WeightSim.exe (with SerialNET200.dll in the same folder), supplied by Shering
- Visual Studio 2005 (or later)

Both the **wBridge.dll** and the **testwb.exe** applications have been developed as 32-bit applications in visual Studio 2010 on a Windows 7 64-bit platform.

The 64-bit dll was developed in Visual Studio 2010 on windows 7 64-bit platform.

Appendix 1

Rinstrum RS-232 comms

These diagrams offer details regarding the M4201 communications device in the display weight indicator. This device is for attaching the printer; the computer attaches to the DWI using the built-in port which has similar wiring but not the same configuration options.



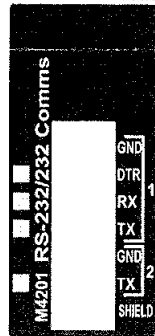
**M4201
RS-232/232 Comms**

Module Description:

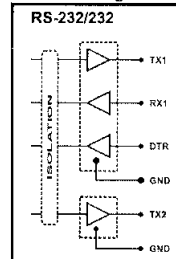
General	1 x isolated duplex RS-232 with DTR 1 x isolated TX-only RS-232
Isolation	>500V from indicator to module
Power Supply	Internal
Baud rate ¹	2400, 4800, 9600, 19200, 38400, 57600
Parity bit ¹	None, Even, Odd
Data bits ¹	7, 8
Stop Bits ¹	1, 2
DTR ¹	OFF, ON

Note 1: Instrument may not support all options.

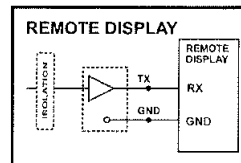
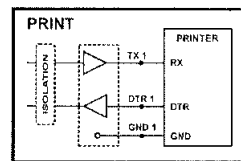
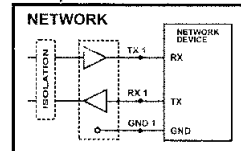
Module Connections



Interface Diagram:



Example Use:



004M-609-110

M4201 SETUP

The indicator should be switched OFF before installing or removing this device.

Modules are configured using the indicator setup. Setup can be done via the indicator keypad or the viewer program. An overview of setup is given below; see the indicator Quick Start or Reference manuals for detailed setup information.

The indicator provides configuration checking capabilities for the M4201. This is found in the H.WARE setup menu. See the indicator Quick Start or Reference manuals.

K402 SOFTWARE SETUP

This module is configured as serial port 2 on the indicator.

Network

(i) Setup. . . . SER.NET. . . . (Refer to quick start manual for settings)

Automatic Output

(i) Setup. . . . SER.AUT (Refer to quick start manual for settings)

Print

(i) Setup. . . . PRINT. (Refer to quick start manual for settings)

Port Configuration (i.e. Baud, etc)

(i) Setup. . . . H.WARE. . . . SER2.HW. . . . (Refer to quick start manual for settings)

M4201 TROUBLESHOOTING

The M4201 has 4 annunciators which help with troubleshooting. These annunciators are shown as yellow on the Module Connections figure above.

Problem	Possible solutions
Receive not working. M4201 receive annunciator not lit.	Check wiring.
Receive not working. M4201 receive annunciator lit.	Check indicator setup (H.WARE:ALLOC).
Transmit not working. M4201 transmit annunciator not lit.	Check indicator setup (H.WARE:ALLOC). When printing: Check DTR annunciator and indicator setup.
Transmit not working. M4201 transmit annunciator lit.	Check wiring.

004M-609-110