

FieldTalk Modbus Master C++ and Visual Studio 2005

Application Note 202

Revision 1.0, August 2008

Installation	1
Adding the precompiled static Library to a Project	1
Add Code	3

This Application Note describes how to integrate the *FieldTalk*[™] Modbus library and driver into a Visual Studio 2005 C++ project.

Prerequisites

- Visual Studio 2005 Standard or Professional

Installation

1. Download and save the zip archive into a working directory.
2. Extract (unzip) the content of the archive.

The archive will create the following directory structure in your project directory:

```
My Projects
  fieldtalk
    doc
    src
    lib
    samples
```

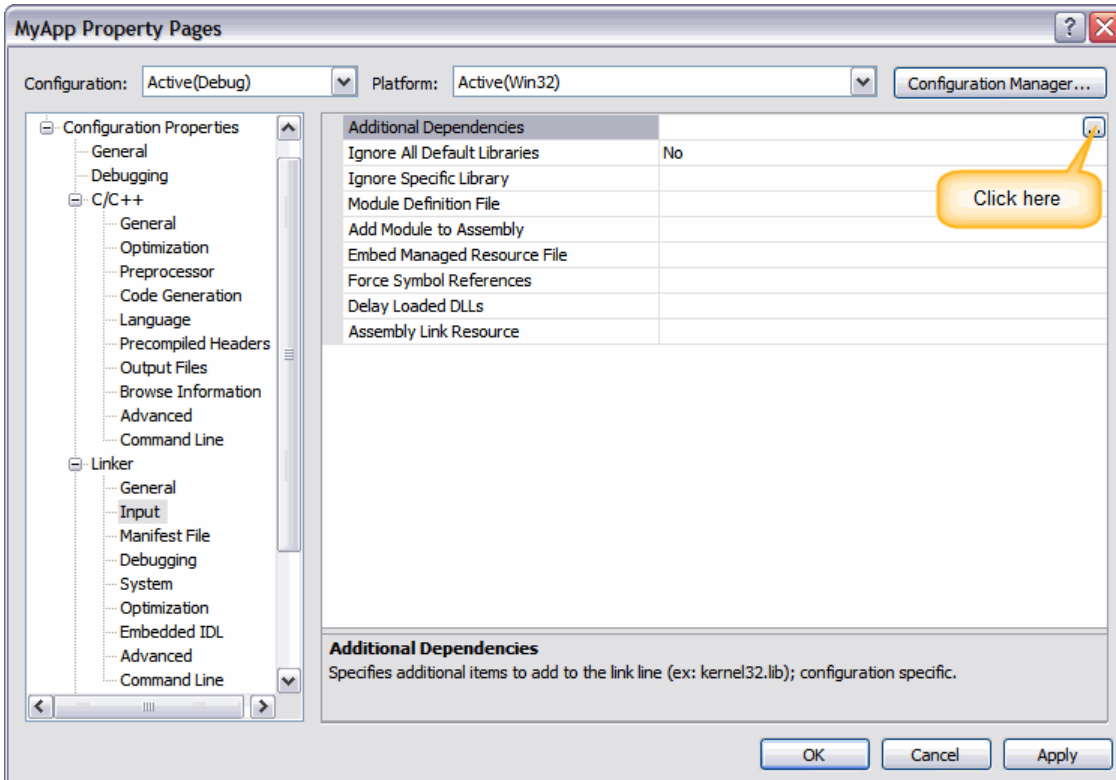
Adding the precompiled static Library to a Project

Launch Visual Studio 2005. Open an existing C++ project where you want to add Modbus functionality or create a new C++ Modbus project.

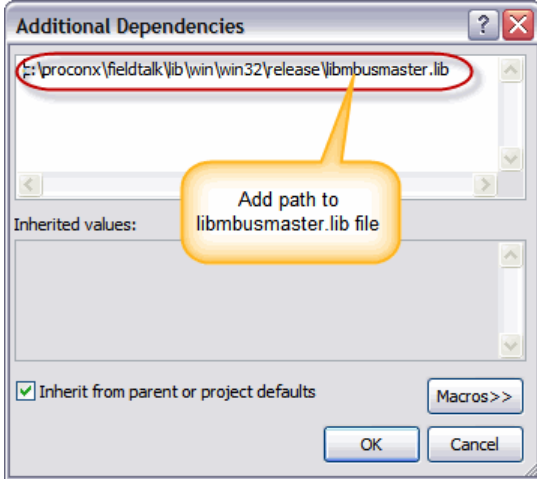
FieldTalk can be integrated into Win32, MFC or ATL Visual Studio project types.

Once the project has been opened, you have to add the static library `libbusmaster.lib` as a linker dependency and add the library include files to the compiler's list of include directories.

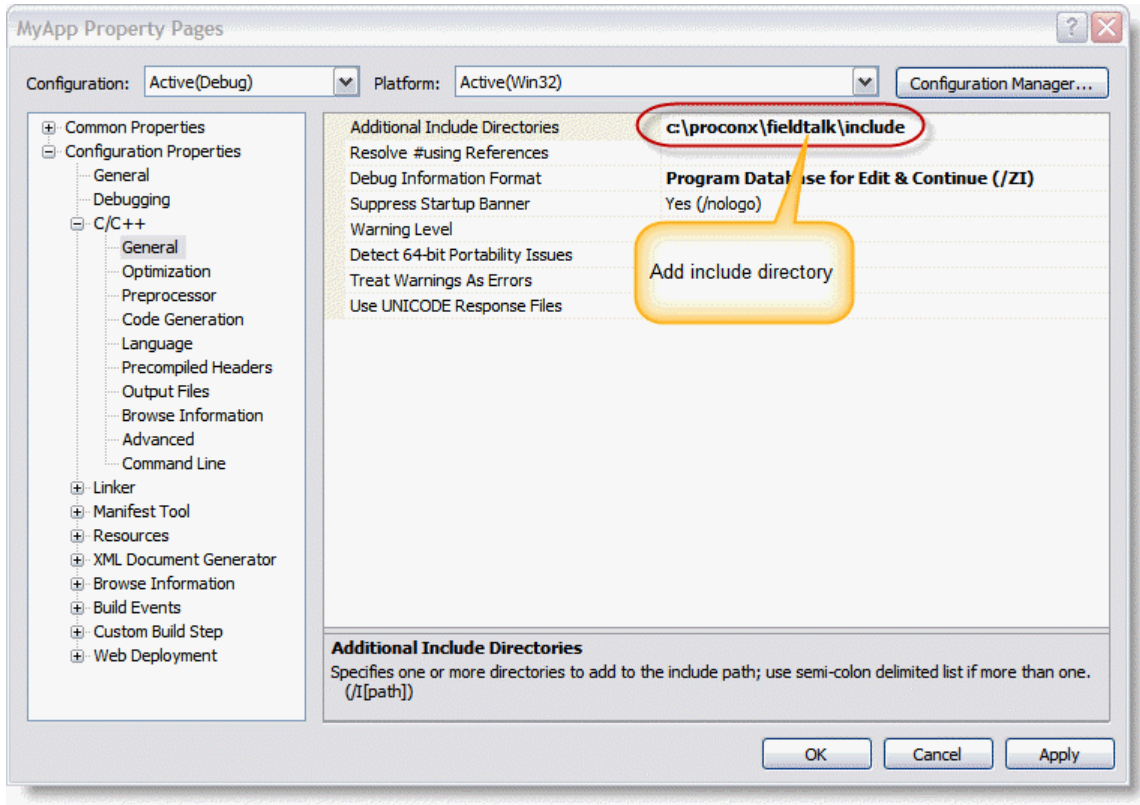
To do this, from the Property Pages dialog, expand the Configuration Properties node, then the Linker node, and select Input. Click on the ... button to open the Additional Dependency dialog.



Type in the path to the location of the libbusmaster.lib library file and click OK.



To reference the header files of the static library, you must modify the include directories path. To do this, from the Property Pages dialog, expand the Configuration Properties node, then the C/C++ node, and select General. Next to Additional Include Directories, type in the path to the location of the fieldtalk/include directory.



Click OK to close the Property Pages dialog.

Add Code

Add to the C++ module where you want to perform the Modbus communication one of the following *FieldTalk* header files:

```
#include "MbusTcpMasterProtocol.hpp"
```

In your communication routine, add the following code to open the Modbus protocol and perform Modbus requests:

```
MbusTcpMasterProtocol mbusProtocol;
int result;
short dataArr[10];

result = mbusProtocol.openProtocol(hostName);
while (threadIsRunning)
{
    result = mbusProtocol.readMultipleRegisters(1, 100, dataArr,
                                                sizeof(dataArr) / 2);
}
```

proconX and *FieldTalk* are trademarks of *proconX* Pty Ltd. All other product and brand names mentioned in this document may be trademarks or registered trademarks of their respective owners.

Disclaimer

proconX Pty Ltd makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in the Terms and Conditions located on the Company's Website. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of *proconX* are granted by the Company in connection with the sale of *proconX* products, expressly or by implication. *proconX* products are not authorized for use as critical components in life support devices or systems.

Support & product feedback

We provide an electronic support and feedback system for our *FieldTalk* products. It can be accessed through the following web link:

<http://www.modbusdriver.com/support>

Your feedback and comments are always welcome. It helps improving this product.

Contact

For further information about this document please contact us at:

proconX Pty Ltd
PO Box 791
Sumner QLD 4074
Australia
Website: <http://www.proconx.com>