FANUC MTCONNECT README

October 24, 2016

This document describes operation and the upgrade to use fanuc fwlib64, but still uses the mtconnect agent core version 1.2. It was hoped that less moving parts will make it easier to debug.

Version info:

* VC++ 2015 community – "safe" and C++11 support.
* 1.2 and 1.3 MTConnect Agent
* Focas 64 bit – only 30 iseries and above. HSSB and Lan.
* NO boost – removed. Now C++11 bind and thread and group\_thread replacement. Makes compilation must simpler.

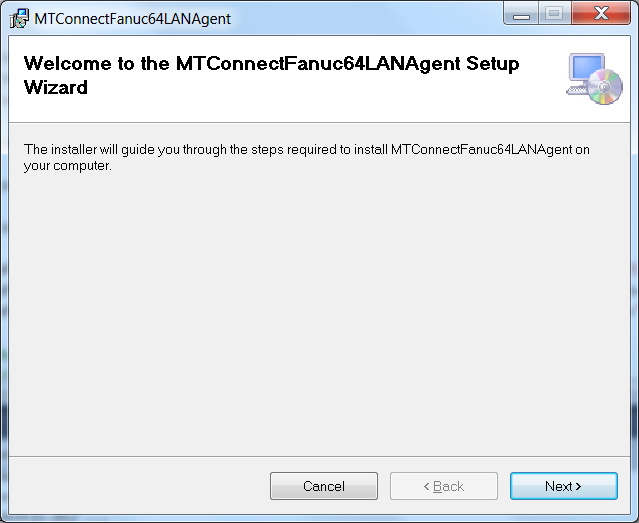
# Background

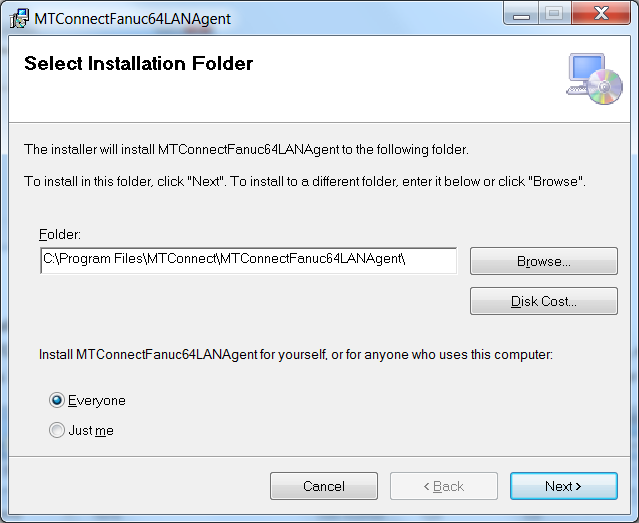
The MTConnect Focas Lan Agent can support connection to multiple Focas CNC devices.

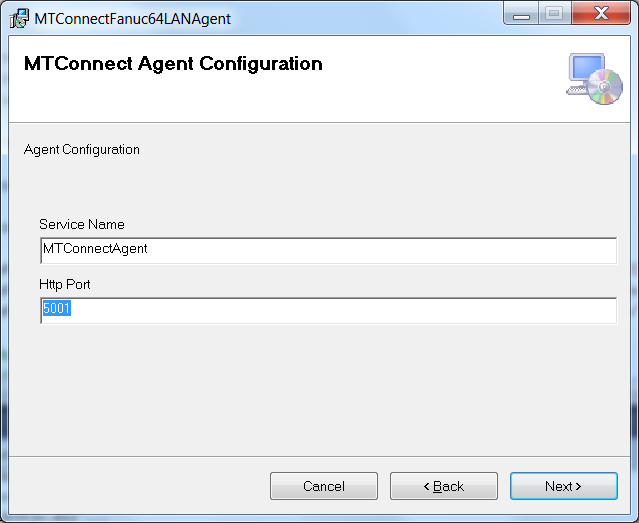
Even though the Agent and Adapter can support the older 15B and 15F Fanuc models, it is not handled as part of the msi distribution. In fact the Fanuc focas code has been compile to run on Window NT 3.5 using MSVC 5, but again, is not suggested.

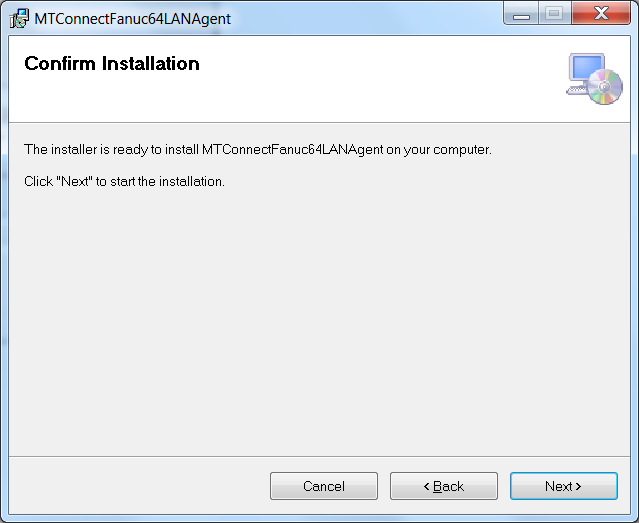
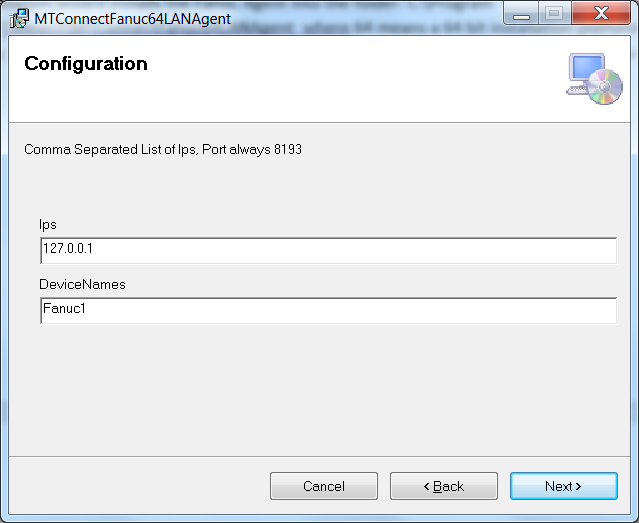
# Installation

Double click the msi install script:

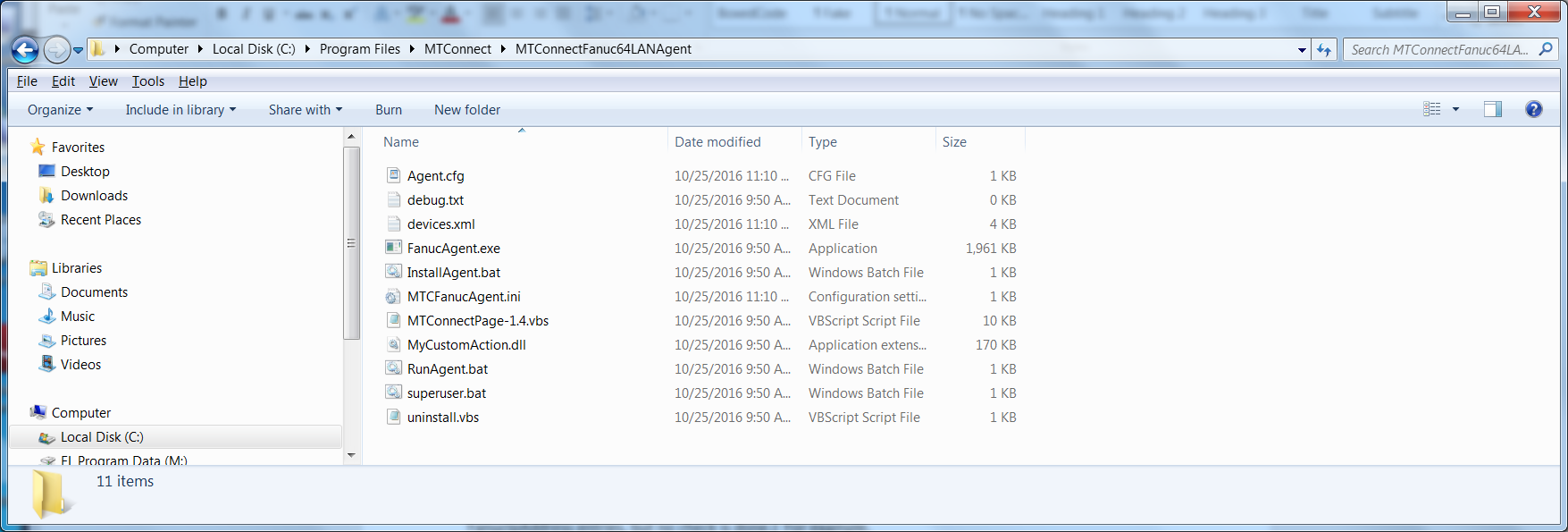








The installation wizard installs the Fanuc Agent into the folder: C:\Program Files\MTConnect\MTConnectFanuc64LANAgent where 64 means a 64 bit installation platform, e.g., windows 7. The MTConnect Agent executable must be paired with the correct platform (32 or 64 bit).



In the folder, the vb script MTConnectPage-1.4.vbs can be used to verify that the log file Agent is working. MTConnectPage-1.4.vbs reads the data from the agent e.g., <http://127.0.0.1:5000> and then formats the data. (assuming you have configured the Agent port to 5000 in the install wizard).

# Configuration

## Configuring MTConnect Fanuc Devices

MTCFanucAgent.ini file is the ini file to handle configuration.

Configuration of the Devices.xml file is done with the "MTCFanucAgent.ini" file in the installation folder. The Fanuc Agent looks in the [CONFIG] section at the “FanucDevices” tag to see how many devices with accompanying sections are defined. (The device names must be unique and match the count of the FanucIpAddress entries, but no check is done.) For example,

[CONFIG]

FanucDevices =Fanuc1

FanucIpAddress= 169.254.22.252

specifies a device Fanuc1.

A complete MTCFanucAgent.ini file is shown below:

[CONFIG]

NewDevicesFile=OLD

DebugLevel=0

#xyzabc

MachineToolConfig = xyz

##########FOCAS######################

# IP address of Fanuc CNC

FanucIpAddress=136.241.9.81

FanucDevices=Fanuc1

# TIme delay between updates focas updates

FocasDelay=1000

#HSSB,LAN

Protocol=LAN

# Port number

FanucPort=8193

# Fanuc CNC version - ignored for now

FanucVersion=iSeries

#FanucVersion=15i

#FanucVersion=15M

# Automatically detect Focas version - ignored

AutoVersionDetect=0

##########AGENT######################

#Agent Port Number

HttpPort=5000

The MTConnect Fanuc Agent will read the ini file: MTCFanucAgent.ini. In that you can configure a LAN agent to gather data from multiple devices, AND the agent will oeprate propely even if one or more of the Fanuc devices is not operational. Many of the parameter tags in the ini file are not used. The file is located in the exe folder, and must have the same "title" as the executable, i.e., MTCFanucAgent. The following tags are important:

* The tag DebugLevel specifies the logging level of information from 0..5, that is written to the file "debug.txt", found in the Installation folder.
* The tag FocasDelay specifies the delay in milliseconds between Fanuc device updates. For example, 1000 specifies one second (1000 milliseconds.)
* The tag HttpPort specifies the Webpage port that the client will use to ccess the MTConnect Agent. For example, when HttpPort=5000 the MTConnect Agent is accessed via <http://127.0.0.1:5000/current> to retrieve the latest data. Note the colon 5000 specfies the http port, if the port is 80 it can be omitted, but port 80 WILL NOT WORK IN Windows 7! Also note this tag is also found in Agent.cfg file, and once installed only the Agent.cfg http port is monitored by the Agent for http servicing.
* The tag FanucPort is hardwired to port 8193. For LAN Focas implementations the remote connection to Focas is based on the ip and socket port.
* The tag FanucDevices specifies the names of the "Devices" in the Devices.xml file and the names that will be displayed when acccessing the MTConnect data. Note, the number of FanucDevices specifed must match the number of FanucIpAddress ip addresses or the Agent will fail and write a message to debug.txt.

The fields in Agent.cfg are described here (and are set at installation): <https://pmcoltrane.wordpress.com/2013/05/13/set-up-an-mtconnect-agent-in-three-or-so-steps/>

Note, this Fanuc implmentationis a specialization of the typical MTConnect Agent implementation!

# Add devices after installation

Use the following steps to modify MTCFanucAgent.ini in C:\Program Files\MTConnect\FanucAgentx64 folder:

1. Stop Fanuc agent, edit config.ini file, add new configuration:

[CONFIG]

Config=NEW

1. Add new devices under [GLOBALS] section tag “MTConnectDevice” (spaces are stripped out)

FanucIpAddress=136.241.9.81, 36.241.9.82

FanucDevices=Fanuc1, Fanuc2

1. Retart Fanuc agent, the agent will detect a new configuration, and then write a new Devices.xml file to add the new devices.
2. If it works MTCFanucAgent.ini tag should say: "Config=UPDATED” if a problem tag will say: “Config=ERROR”

# Focas MTConnect Issues

## Focas LAN Port Number

8193 is the Focas TCP port that the agent communicates through the Fanuc Adapter that connects to the Fanuc Focas library. Port 8193 must not be firewalled on the CNC/HMI side.

This 8193 number is a Fanuc # so it is hardwired!

Confirm Focas Installed

## **Confirm FOCAS is installed**

**How can i check to see if i Fanuc FOCAS?**

In order to see if you have Fanuc FOCAS available on your CNC, is to first check if you have Ethernet. To do that follow these steps:

1. Press the [SYSTEM] hard key.
2. Press [>] approximately 5 times until you see "ETHPRM"
3. Press {ETHPRM} soft key, followed by {OPRT}
4. This will display the ethernet connections available.
5. Select a connection (normally EMBEDD or BOARD)
6. Press the [DOWN] hard key. (page 2)
7. This should display FOCAS / ETHERNET

FOCAS is an optional function.  If it is installed, the setting screen will be available.  The screen is located under the system hard key; followed by the continuous menu soft key several times.

system

[continuous menu softkey](http://www.ellisware.com/wordpress/wp-content/uploads/2014/05/continuous-menu-softkey.png)

Keep scrolling through the available softkeys until the [Embed Port] soft key is displayed.  All of the available ethernet interfaces will be shown.

[ethernet softkeys](http://www.ellisware.com/wordpress/wp-content/uploads/2014/05/ethernet-softkeys.png)

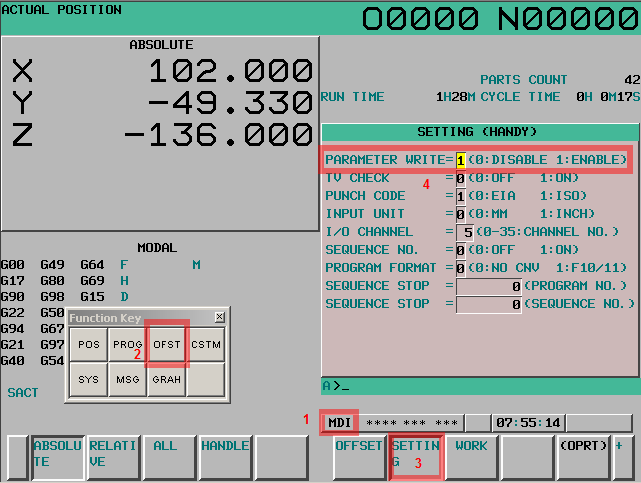
Of the three listed here, [Ether Board] is the best choice.   It is an add on hardware board for Ethernet communication.  Press the [Ether Board] soft key.  If the [FOCAS2] softkey is now displayed, FOCAS is installed !!!  Plug your Ethernet cable into the add on boards RJ45 plug.

[focas2 softkey](http://www.ellisware.com/wordpress/wp-content/uploads/2014/05/focas2-softkey.png)

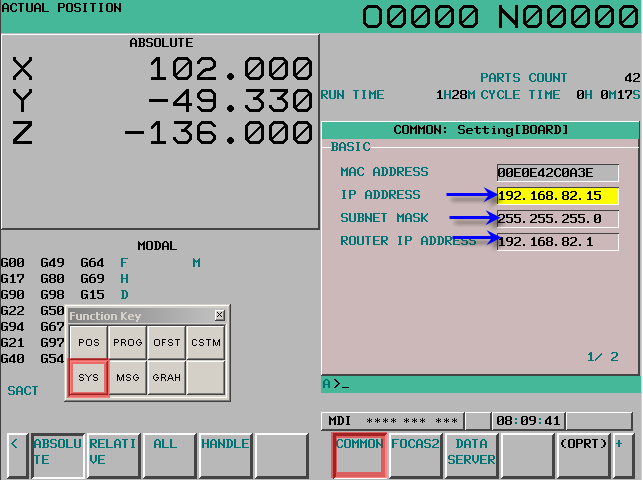
If you didn’t find the [FOCAS2] softkey, then check to see if it exists under the [Embed Port] softkey.  If it does, then connect your Ethernet cable into the RJ45 plug directly on the mainboard of the FS0iD contol.

**Setting FOCAS2**

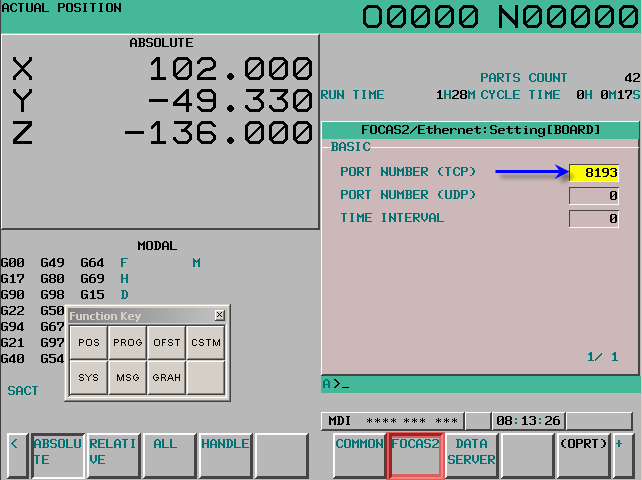
In order to set the FOCAS settings, we first need to be in MDI mode.  Parameter Write Enable (PWE) is then enabled under the <OFS/SET> hard key [SETTING] softkey.

[](http://www.ellisware.com/wordpress/wp-content/uploads/2014/05/set-PWE.png)

Once PWE is set the IP Address of the control is set by <SYSTEM>, [Ethe Board], [COMMON].  The IP address will be a fixed address assigned by the network administrator.

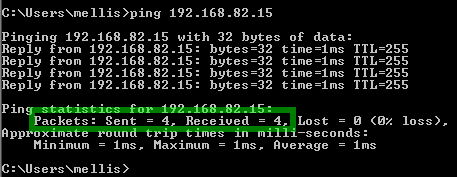
[](http://www.ellisware.com/wordpress/wp-content/uploads/2014/05/IPAddress-Setting.png)

Finally, the TCP Port number is set under the [FOCAS2] settings.  This port is typically 8193, however, any valid TCP port number can be used.

[](http://www.ellisware.com/wordpress/wp-content/uploads/2014/05/TCPPort-setting.png)

**Confirmation**

If everything is set correctly the FANUC FS0iD CNC should be reachable from any computer on the same network.  In Microsoft Windows, start a new command prompt and ping the CNC.

[](http://www.ellisware.com/wordpress/wp-content/uploads/2014/05/pingsuccess.png)

You are now ready to use FANUC FOCAS to read and write information to the CNC.  Although the FOCAS drivers and libraries is a programming interface for C/C++ or Visual Basic, there are many available options that don’t require programming such as MTConnect and OPC converters to FOCAS.

# Windows Issues

## Resetting File Permissions in Program Files to allow Saves

Beginning with Windows 7, it was recommended to not change any files in C:\Program Files as malware liked to write into this area and overwrite good stuff.

But the folder location selected by window does not work easily for MTConnect Agent.

So a superuser.bat script was written which will change all the file permission, you can find it here:

C:\Program Files (x86)\MTConnect\FanucLanMTConnectAgent\superuser.bat

**To run, right click superuser.bat and Run As Administrator, and it will change all the file permissions in the current folder to writeable.**

You can copy the file to some other directory and do the same thing.

# VC++ 2015 Compilationand Linking Issues

## Problem: redefinition of boost::noncopyable\_::noncopyable.

From: <https://sourceforge.net/p/dclib/discussion/442517/thread/cfd6538b/>

In file dlib/noncopyable.

#ifndef DLIB\_BOOST\_NONCOPYABLE\_HPP\_INCLUDED

#define DLIB\_BOOST\_NONCOPYABLE\_HPP\_INCLUDED

namespace dlib {

class noncopyable {

protected:

noncopyable() {}

~noncopyable() {}

private:

noncopyable(const noncopyable&);

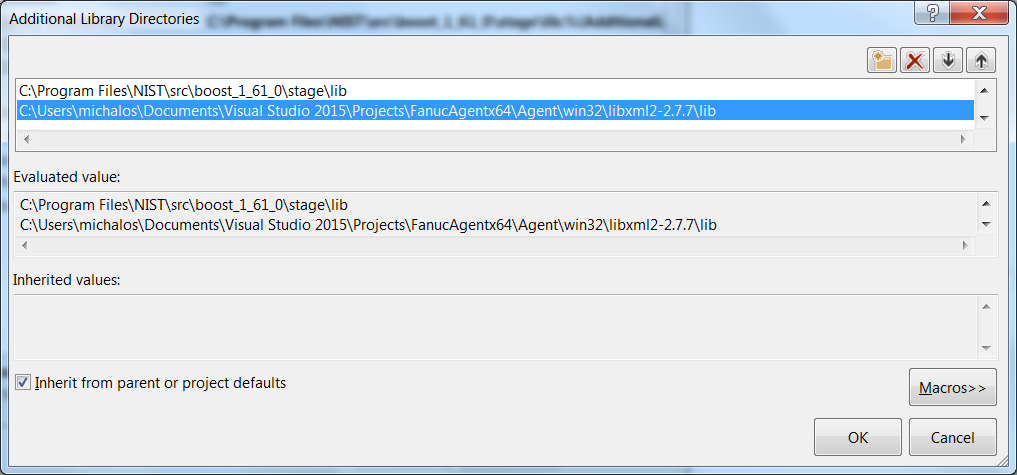
const noncopyable& operator=(const noncopyable&);

};

}

#endif // DLIB\_BOOST\_NONCOPYABLE\_HPP\_INCLUDED

## Missing lib Add to project properties-> Linker->General



## Create console project - no precompiled ATL, etc.

* Setup project as multibyte not Unicode string
* Make Project options: Compiler-> code generation-> Runtime static lib use of C lib

## Create MTCAgent folder

Copy in agent code – src/lib/win32.

## Preprocessor

Add these definitions:

* \_Windows
* WIN64 for 64 bit configurations for later #pragma lib includes described below
* \_NO\_CRT\_STDIO\_INLINE for problem (9) doesn't solve but should

## Include files

In project properties, Projects->Compiler->Command line Add: @IncludeDirs.txt  
You will have to adapt the full path to the location of your   
FIle IncludeDirs.txt:

-I"C:\Program Files\NIST\src\boost\_1\_61\_0"

-I.

-I"C:\Users\michalos\Documents\Visual Studio 2015\Projects\FanucAgentx64"

-I"C:\Users\michalos\Documents\Visual Studio 2015\Projects\FanucAgentx64\Agent\agent"

-I"C:\Users\michalos\Documents\Visual Studio 2015\Projects\FanucAgentx64\Agent\lib"

-I"C:\Users\michalos\Documents\Visual Studio 2015\Projects\FanucAgentx64\Agent\win32\libxml2-2.7.7\include"

## No gets() in service.cpp in VS 2015

*The most recent revision of the C standard (2011) has definitively removed this function from its specification. The function is deprecated in C++ (as of 2011 standard, which follows C99+TC3).*  
from <http://stackoverflow.com/questions/12893774/what-is-gets-equivalent-in-c11>

Subsituted:

while(gets\_s(line, sizeof(line)) != NULL) {

## timezone not declared

globals.cpp(241): error C2065: 'timezone': undeclared identifier

uint64\_t time = (mktime(&timeinfo) - timezone) \* 1000000;

Removed timezone to make it compile - but not really FIXED!

## l ibxml2d linking error

l ibxml2d\_a\_v120\_64.lib(error.obj) : error LNK2001: unresolved external symbol \_\_iob\_func

<http://stackoverflow.com/questions/30412951/unresolved-external-symbol-imp-fprintf-and-imp-iob-func-sdl2>

Added to project main file:

FILE \_iob[] = {\*stdin, \*stdout, \*stderr};  
extern "C" FILE \* \_\_cdecl \_\_iob\_func(void)  
{  
 return \_iob;  
}

## unresolved external symbol vfprintf

#pragma comment(lib, "legacy\_stdio\_definitions.lib")

## Add fwlib64.lib to linker path: used pragma with path. This will have to be customized. Lib is same for debug as well as release builds.

#define FANUCLIBPATH(X) "C:\\Users\\michalos\\Documents\\Visual Studio 2015\\Projects\\FanucAgentx64\\Fwlib64\\" ## X

#if defined( WIN64 ) && defined( \_DEBUG )

#pragma message( "DEBUG x64" )

#pragma comment(lib, FANUCLIBPATH("fwlib64.lib"))

# Replacement of Boost Functionality with C0x11

## Difference between C++11 std::bind and boost::bind

<http://stackoverflow.com/questions/10555566/difference-between-c11-stdbind-and-boostbind>

## C++ 11 Thread vs Boost Thread is there any difference?

<http://stackoverflow.com/questions/7241993/is-it-smart-to-replace-boostthread-and-boostmutex-with-c11-equivalents>

## 'INTMAX\_MAX': undeclared identifier

Sprinkled

#define INTMAX\_MAX 9223372036854775807i64

throughout code… ☹

Added

\_WIN64

to preprocessor

## No group\_thread

<http://stackoverflow.com/questions/9894263/boostthread-group-in-c11>

Replaced, cross fingers.