README

Friday, April 20, 2018

This document describes the data gathering for a Wiring machine that a log file. Multiple adapters to Wiring devices are possible, each contained within one MTConnect Agent.

This document presents a brief background on the mechanics of the MTConnect agent for the WIRING CMM. This document assumes the reader is familiar with MTConnect operation, and for deeper explanation of MTConnect, please refer to MTConnect URL: <http://www.mtconnect.org/> for more information. This document concerns itself with how an embedded Adapter operates. One or more embedded adapters read a configuration file, and then read a remote WIRING log files as logged from an WIRING device. This WIRING log file must be shared in Windows so the MTConnect embedded adapter can read the file, reinterpret the contents into MTConnect meaningful terminology, and then report this information to a canned MTConnect Agent (circa version 1.2), which handles management of devices(s) data, the web service, etc.

# Background

MTConnect is a new standard developed to facilitate the exchange of data on the manufacturing floor. The MTConnect open specification provides for cost effective data acquisition on the manufacturing floor for machine tools and related devices. MTConnect is based upon prevalent Web technology including XML and HTTP. Figure 1 shows the MT Connect architecture. An “MTConnect Device” is a piece of equipment – in this case a WIRING machine tool, which (optionally) includes an MTConnect Adapter so that we can get data from it. The “Agent” is a process that acts as a “bridge” between a device and a factory “Client Application”. To learn more about MTConnect visit: <http://www.mtconnect.org/>

Figure 1 shows a typical MTConnect WIRING Logging Agent system architecture (with one or more WIRING devices). Communication between two Windows PCs is assumed - one containing the WIRING log file and the other PC communicating over Ethernet to read the WIRING log file.



Figure 1 MTConnect WIRING Logging Agent system architecture

This document describes the data gathering for a WIRING device that updates a log file. Multiple adapters to WIRING software are possible, each contained within one MTConnect Agent.

The WIRING Agent contains back end adapters that read a log file generated from the periodically (typically when an event occurs within the CMM). The log file contains events and not samples, but all the events are time stamped and in absolute order of occurrence.

The file is specified as a Window cross-platform file, so it must contain the PC or computer name. UNC is short for Universal Naming Convention and specifies a Windows syntax to describe the location of a network resource, such as a shared file, directory, or printer. The UNC syntax for Windows systems has the generic form:

\\ComputerName\SharedResource

In our case the SharedResource is a shared folder that must be explicitly sharable. In order to use the UNC file, Microsoft File Operations: CreateFile, ReadFile and CloseFile are used as other generic C++ file operations did not work (but were originally tried.) UNC files on Windows seem to require Windows specific File operations. Note, the UNC file path must be accessible to other computers or it cannot be read.mInside the Agent are Adapters for each UNC file. Each Adapter runs as a thread, hence the distinction between 64 bit and 32 bit C++ solutions must be explicitly acknowledged in installing the binary exe. That is, 32-bit MTConnect agents do not on 64 bit platforms, although they may appear to.

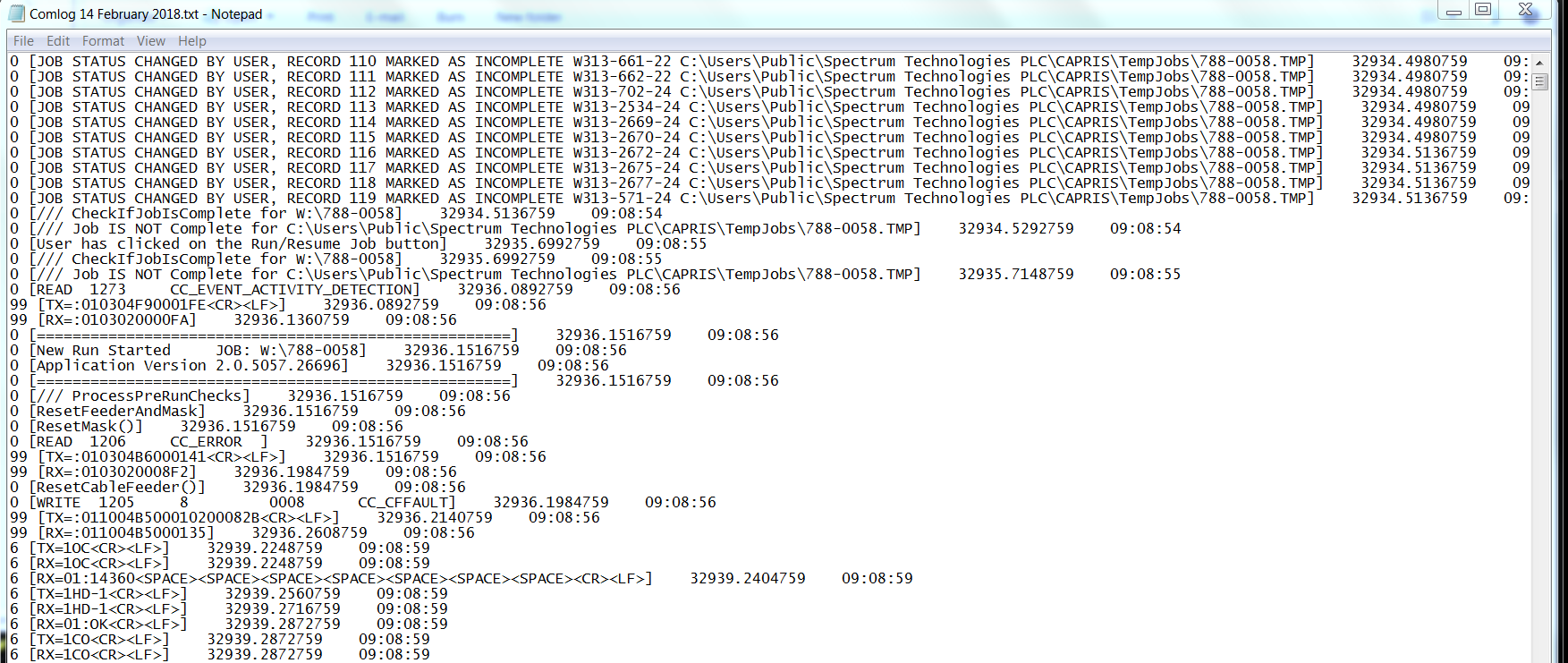
In the Wiring file, it saves all events with each line within the UNC shared file. Note, no error detection of runaway date or times is done.

The Wiring log file name convention is "Comlog xx Month Year.txt" which corresponds to todays date in naming; for example:

"Comlog 19 April 2018.txt"

Where the quotes are inserted to understand the name as blanks are used in the file name.

Below is a snippet from a sample log file generated by Wiring device:



Below is the translation of the Wiring status messages into MTConnect controller logic:

|  |  |
| --- | --- |
| Log file entry | MTConnect |
| New Run Started | power=ON  controllermode=MANUAL  execution=IDLE  error=""  program = name after JOB: |
| StartTheFeeder | power=ON  controllermode= AUTOMATIC  execution=EXECUTING |
| ASL START | power=ON  controllermode=MANUAL  execution= PAUSED  error="" |
| ASL STOP | power=ON  controllermode= MANUAL  execution=IDLE  error= |
| Feeder Has Stopped | power=ON  controllermode=AUTOMATIC  execution= IDLE  program="" |
| The Run Finished | power=ON  avail=AVAILABLE  controllermode=AUTOMATIC  execution= IDLE |
| Xyz and rpm | Faked based on AUTO, PROGRAM and EXECUTING |

Because of the deficiency of the MTConnect state logic, some side effects are generated to make the controller appear to be operating: positions for x,y,z and RPM change after every update if the controller is in automatic mode and executing.

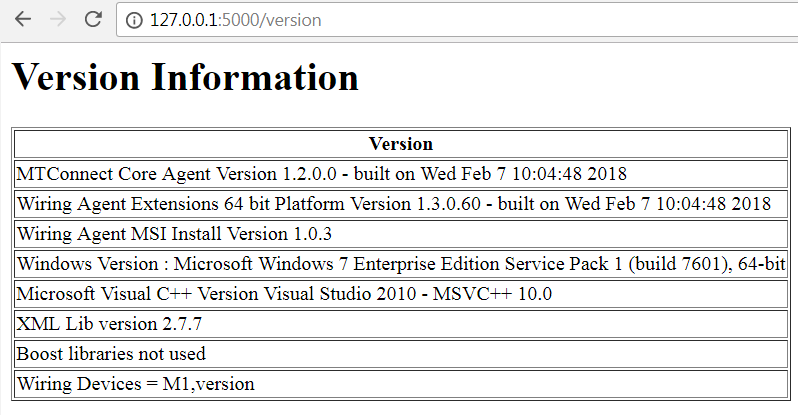
# Version Information from MTConnect Agent

The version information of the various MTConnect components is available through web browser access. It is recommended to use Google Chrome as it understand the XSLT formatting of the XML (Internet Explorer is baffled.)

The open source core MTConnect agent is downloaded from github and “frozen”. The version used is:

MTConnect Agent Version 1.2.0.0 - built on Sat Oct 12 13:30:24 2013

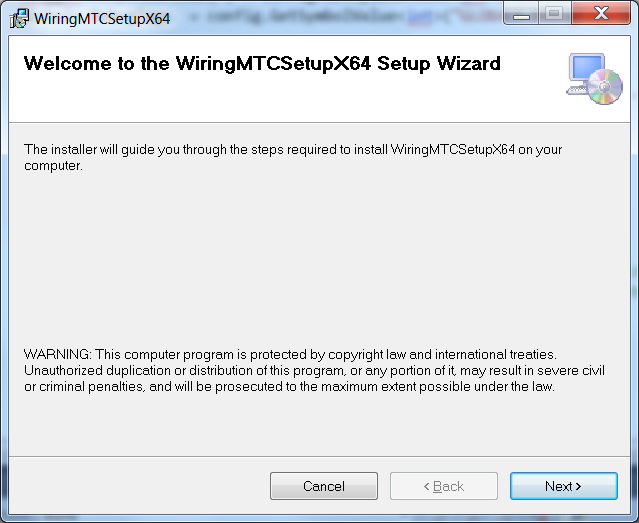
It is not perfect but the output shown below gives an indication of the software involved in the MTConnect Agent operation. It can be modified to include other version information, but requires a recompilation at this time.

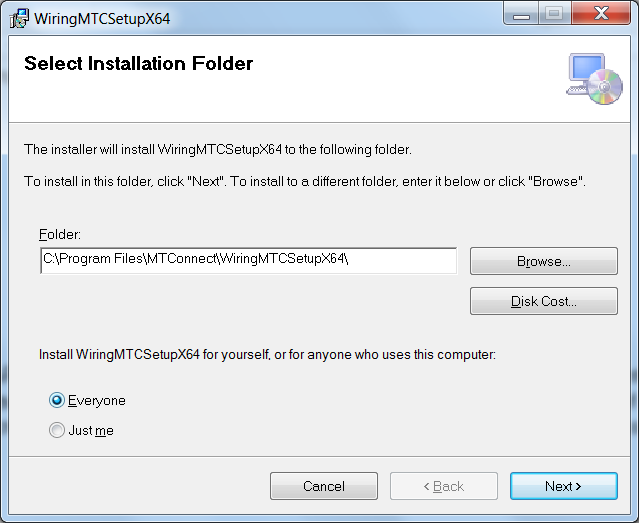


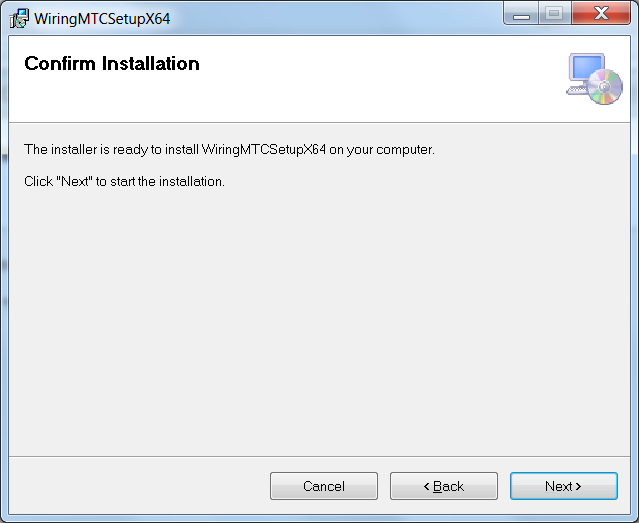
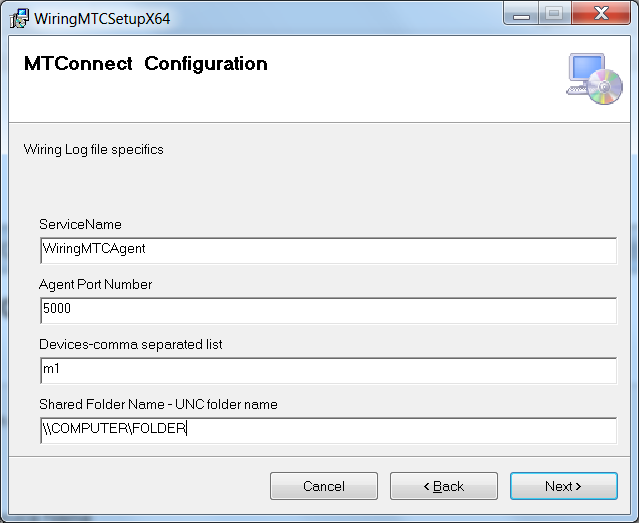
# Installation

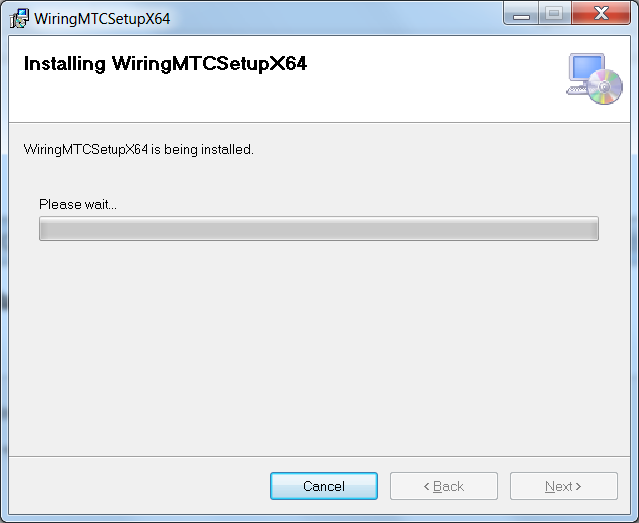
To install the Wiring Agent double click the (for 64 bit machines only).



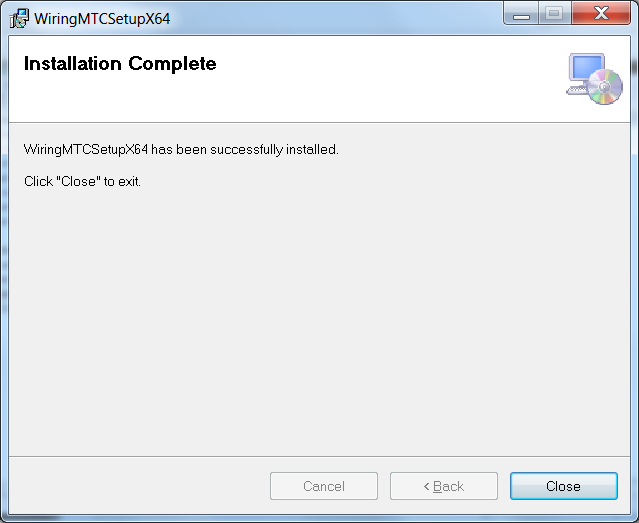


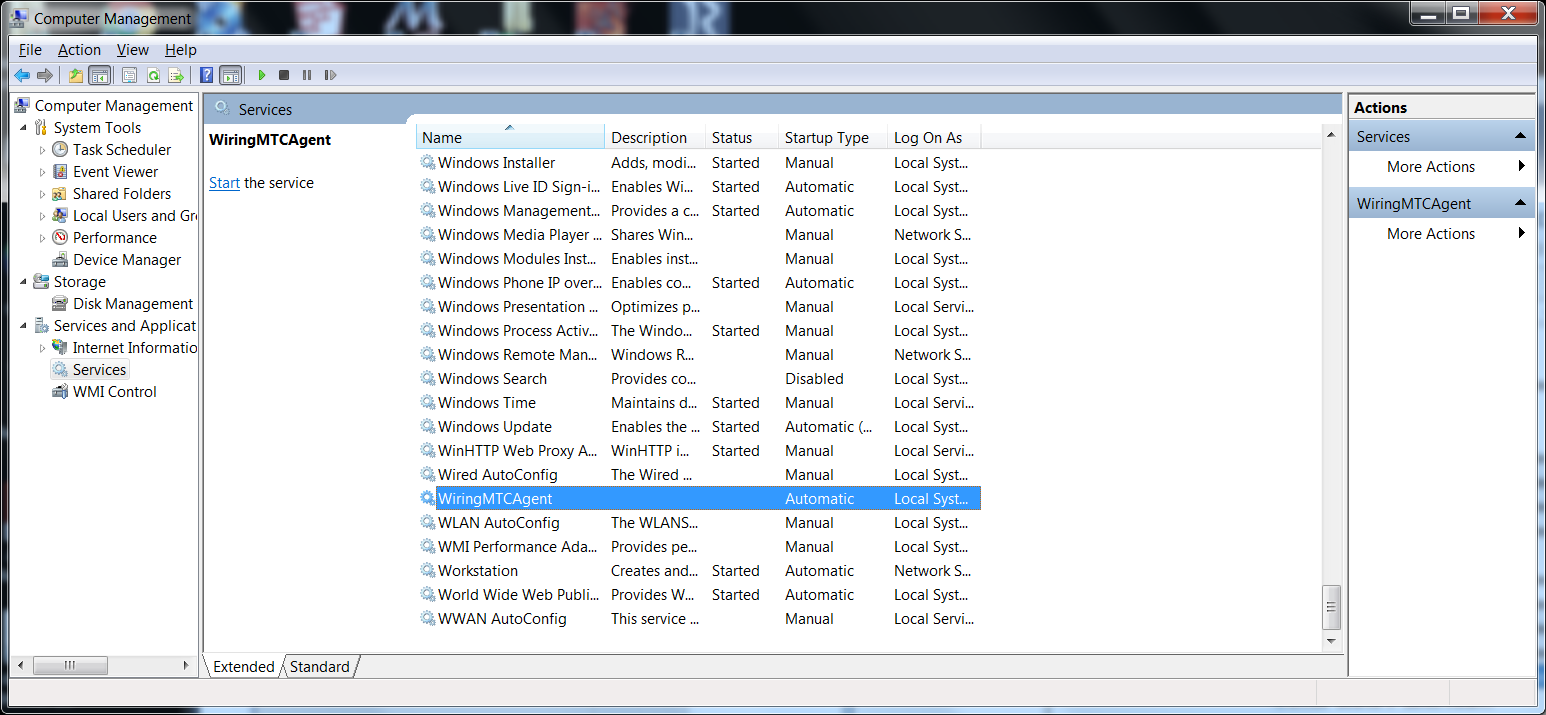






Acknowledge the installation permission challenge, and during the installation you should see some background black consoles pop up and disappear. then you should see the installation screen:





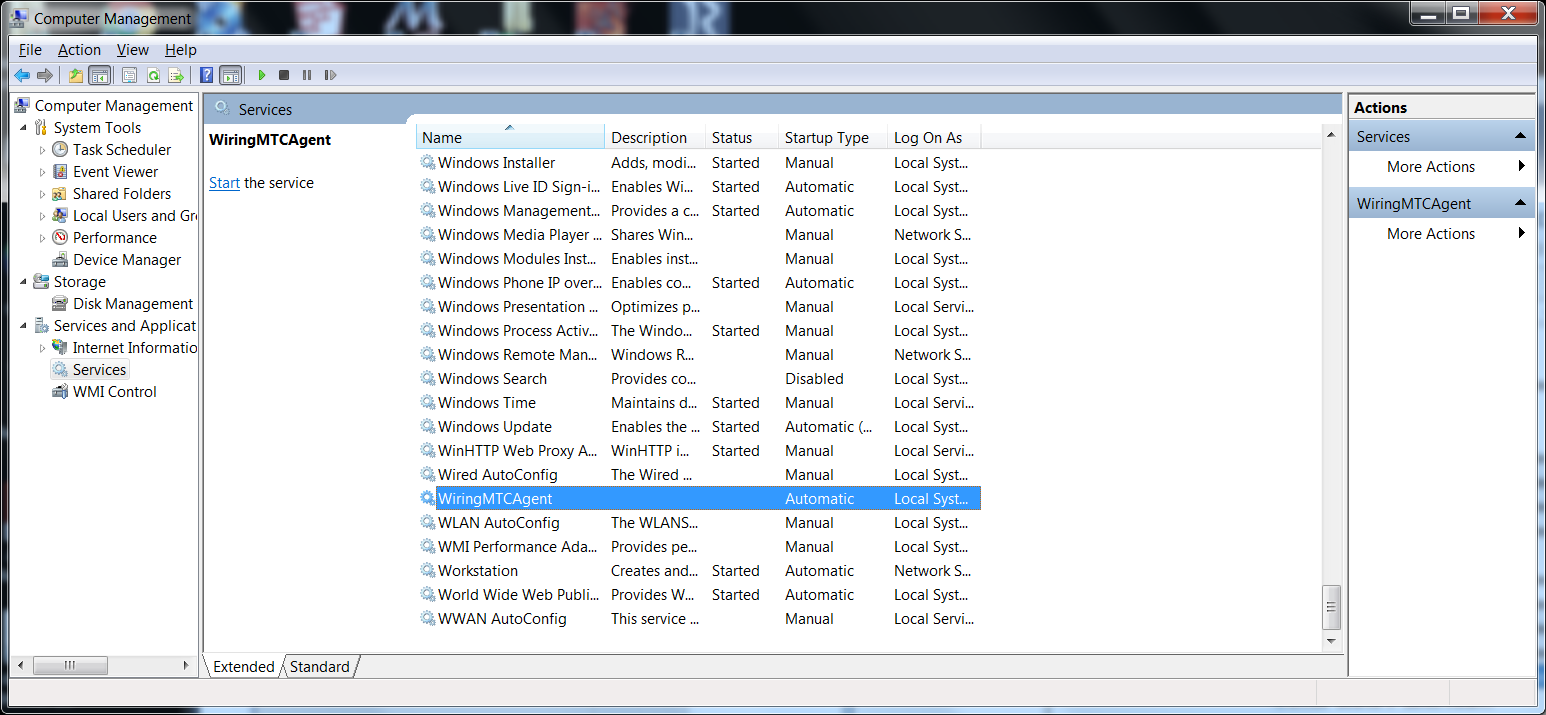
The agent will NOT start unless you tell it to START (unless you reboot).

# Uninstall

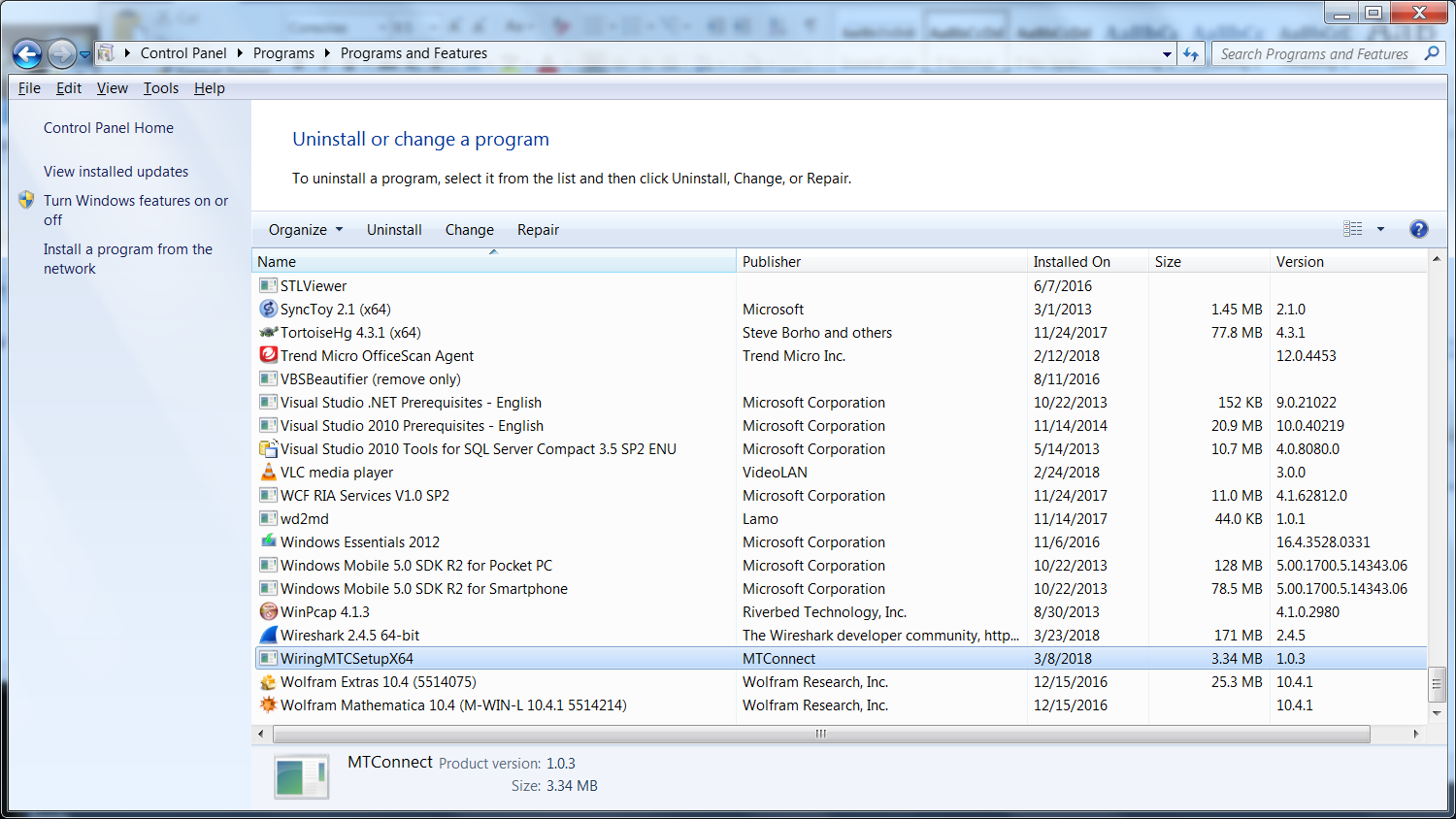
First make sure the Wiring Agent is stopped in the Service Control Manager:

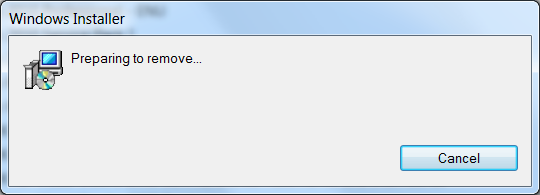
Right click on My Computer -> Select Manage -> Acknowledge UAC challenge

Select Services and Applications and then Services, scroll down to WiringAgent, and push Stop button.



After you have stopped the Wiring Agent service, go into the control panel and uninstall the program: MTConnectAgentWiringx64



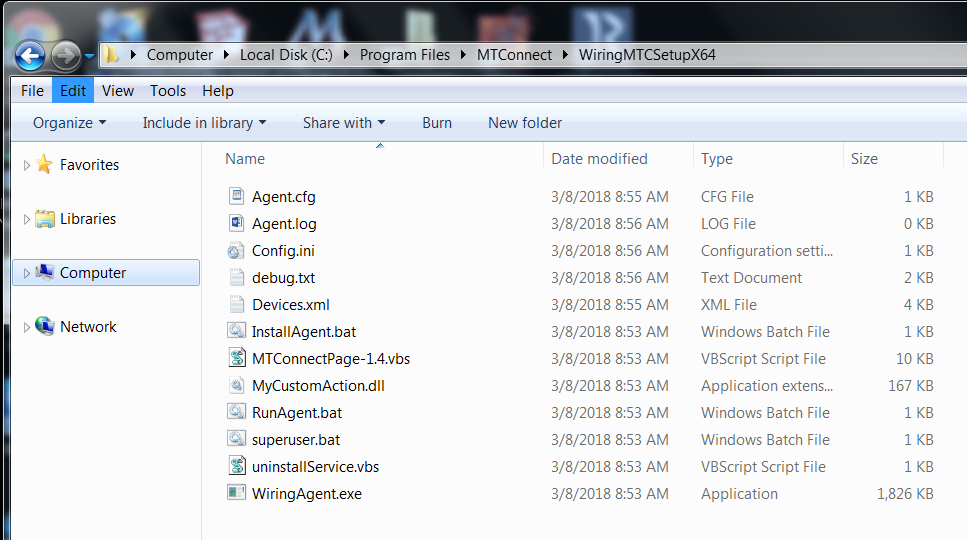


Please wait .. acknowledge UAC permission to uninstall challenge (you must be administrator or have administrator privileges). It will uninstall and you may see a black console screen popup in the background momentarily.

And then in the Service Manager click Action->Refresh, and the Wiring Agent service should be removed. The agent code in C:\Program Files\MTConnect\MTConnectAgentWiringx64 should also be removed.

# Configuration

The installation wizard installs the log file Agent into the folder: C:\Program Files\MTConnect\WiringMTCSetupX64 where x64 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 in the can be used to verify that the log file Agent is working. MTConnectPage-1.4.vbs reads the data from the agent via <http://127.0.0.1:5000> and then formats the data. (assuming you have configure the agent port to 5000).

# Configuration

The Wiring MTConnect Agent uses the open-source MTconnect agent version 1.2 as the backbone to read http requests, generate XML responses, and update the underlying data. There is a NIST log file adapter addition to remotely read local or remote windows UNC files and then parse. The adapter can handle multiple device log file and update the core MTConnect agent.

The configuration is done at installation time during the wizard screens. The user must supply pairs of device names and UNC log folder names, e.g., 1,\\servername\C$\Program Files\WiringLog From Wikipedia: The Microsoft Windows UNC, short for Universal Naming Convention or Uniform Naming Convention, specifies a common syntax to describe the location of a network resource, such as a shared file, directory, or printer. The UNC syntax for Windows systems has the generic form:

\\ComputerName\SharedFolder\Resource

Microsoft often refers to this as a "network path". You can use local paths, but if you are doing remote access of the log file, you must use the UNC convention.

The user can modify the config.ini file in the C:/Program Files/MTConnect/MTConnectAgentWiringx64 folder. However, the devices.xml and the agent.cfg files necessary for the core MTConnect agent are generated at installation time, and thus a reinstallation would be required to modify these files.

In the config.ini file, you can change the ServiceName, Agent port and query times of the log files. These changes will take if you stop/restart the Agent service or reboot the machine.

[GLOBALS]

ServiceName=WiringAgent

AgentPort=5000

QueryServer=10000

ServerRate=5000

Debug=0

ResetAtMidnite=false

MTConnectDevice=Wiring1

logging\_level=FATAL

[Wiring1]

NetworkShare=\\servername\c

Thus, the agent it reads the config.ini file for list of “MTConnectDevice”s under the Globals section. The configuration file also sets the Global flags: QueryServer, ServerRate, Debug, AgentPort(default 5000), and ResetAtMidnite.

# DIRECTIONS TO CONFIGURE WIRING LOG FILE AGENT.

Modify Config.ini in C:\Program Files\MTConnect\MTConnectAgentWiringx64directory

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

[GLOBALS]

Config=NEW

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

MTConnectDevice=M1, M2, M3

1. Make sure there is an ini file “section” for each device (in this case M1, M2, M3) and ProductionLog tag that points to the UNC (Windows Universal Naming Convention) path to the log file as in:

[M1]

NetworkShare=\\grandflorio\c$\logfolder

[M2]

NetworkShare =\\rufous\c$\logfolder

[M3]

NetworkShare =\\synchro\c$\logfolder

1. Start Wiring 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 config.ini tag should say :”Config=UPDATED” if a problem tag will say: “Config=ERROR”