VFD Automation

Technical Documentation

Leon Wu

Date: June 5, 2015

**Description**

A variable-frequency drive (VFD) (also termed adjustable-frequency drive, variable-speed drive, AC drive, micro drive or inverter drive) is a type of adjustable-speed drive used in electro-mechanical drive systems to control AC motor speed and torque by varying motor input frequency and voltage. Experienced building engineers at Rudin have used VFD on various buildings including 345 Park Avenue to achieve better control of the building’s thermal dynamics and better energy utilization. Control of the VFD has been done through manual operations whereas a building engineer has to monitor the temperature curves and occupancy changes continuously and adjust the VFD accordingly. The past operation history has indicated this manual process should be able to be automated with some defined criteria. VFD automation would lead to less workload on building operators, better accuracy due to reduced human error, and higher energy efficiency. Also it provides a scalable solution to other buildings with VFD to be installed and operated.

**Architecture**

Based on valuable inputs from building engineers and (building management system) BMS service companies, we designed this three-step occupancy based VFD automation solution, as illustrated in the Figure 1.

The first step starts with a computation module residing on the server in the datacenter that looks into the occupancy trend in the database. When the occupancy trend deviates from prior value for certain percentage (such as 2%), a trigger is generated and recorded into some database tables.

The second step is performed by two components running on the server in the building’s operation room. The first component is to retrieve data from the triggers tables and check whether any trigger (in the form of table row) has happened in the last few minutes (such as 5 minutes). If a trigger is detected, the second component translates that information into a custom-formatted BACnet message and broadcast that message through network that is visible to the BMS.

The third step is done by the procedure defined in the BMS. Whenever a BACnet message is received, first, the message or command is logged; then a pre-defined VFD rampdown procedure is started, either turning on or turning off. Also, this step allows building operators to override any automation actions so that when the automation fails or goes wrong, there is a manual way to correct it.



Figure 1: occupancy based VFD automation process

**Step 1**

**(to be added by Ashish)**

**Step 2**

1) Data Communication

Because the database server that hosts the occupancy data is at the data center and the network connectivity at the local server at the building does not have a direct connection to the remote server, an alternate data communication mechanism via XML file exchange is used. On the remote database, a process is added to generate an XML file using the data in the occupancy triggers table. On the local server at the building, a scheduled task, as follows, runs frequently (every five minutes) to retrieve the XML file and save it to a local XML file on the hard drive.

java -cp "D:\BACnet\Daemon\BACnetDaemon.jar" bacnetdaemon.VFDXML

2) Data Processing

A second scheduled task, as follows, runs after the XML retrieval task described above. This process reads and parses the data. It determines whether any new information has been added since last run, hence a new command is necessary.

java -cp "D:\BACnet\Daemon\BACnetDaemon.jar" bacnetdaemon.VFDRampdown

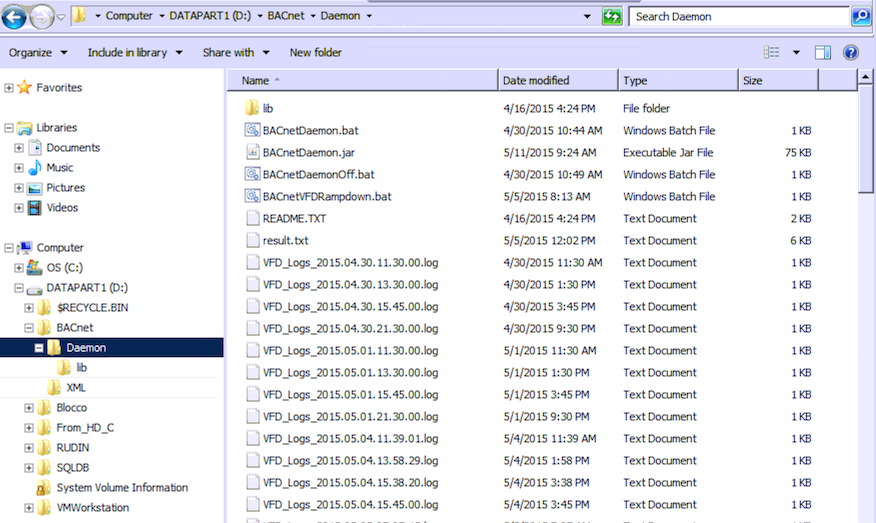
3) BACnet Command

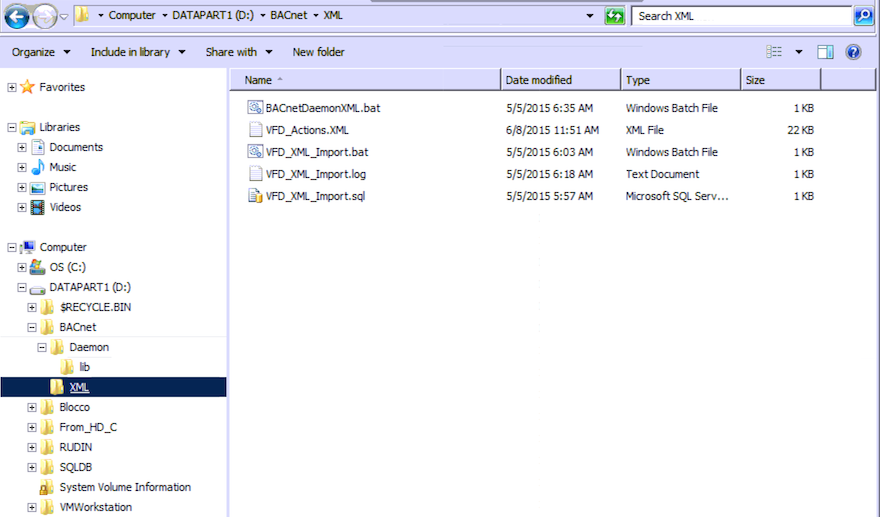
Once a new command is detected and constructed, this process format a custom BACnet message targeting a specific device and one of its specific changeable value (both were provided by the BMS). And then the process broadcast the message via the network out utilizing a BACnet API library (BACnet4J). At the same time, a log of the newly generated message is created and saved to the hard drive as a text file. The log is for checking of the system status.

4) Software

All the executable files, XML files, batch files, and log files locate inside one master directory D:\BACnet on the local server at the building.

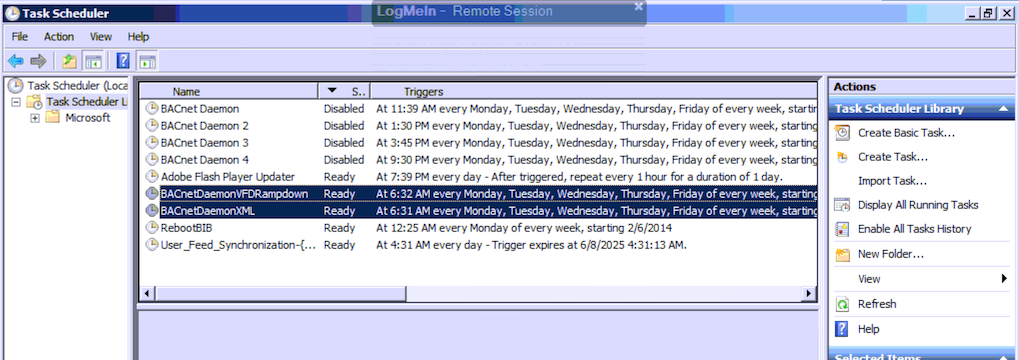
* D:\BACnet\Daemon contains BACnetDaemon.jar, the main program, \*.bat batch files, and log files, along with third party libraries.
* D:\BACnet\XML contacts \*.bat batch files for XML processing, and \*.XML file that contains replicated trigger data.





5) Scheduled Tasks

As explained above, two scheduled tasks with every five minutes frequency have to be configured on the local server at the building.



**Step 3**

**(to be added by Cory)**