

Description: Guide to setting up BACnet and browsing for BACnet data

OS Requirement: Win 2000, XP Pro, Server 2003, Vista, Server 2008, Windows 7

General Requirement: Installation of GENESIS32 v9.2x

Introduction

BACnet is "a data communication protocol for Building Automation and Control networks." A data communication protocol is a set of rules governing the exchange of data over a computer network. The rules take the form of a written specification that spells out what is required to conform to the protocol. BACnet communication is special because the rules relate specifically to the needs of building automation and control equipment, i.e., they cover things like how to ask for the value of a temperature, define a fan operating schedule, or send a pump status alarm.

BACnet Client-Server Architecture

BACnet is based on a "client-server" model of the world; BACnet messages are called "service requests." A client machine sends a service request to a server machine, the server then performs the service and reports the result to the client. BACnet currently defines thirty-five message types that are divided into five groups or classes. For example, one class contains messages for accessing and manipulating the properties of the objects described above. A common one is the "ReadProperty" service request. This message causes the server machine to locate the requested property of the requested object and send its value back to the client. Other classes of services deal with alarms and events; file uploading and downloading; managing the operation of remote devices; and virtual terminal functions.

BACnet specification (135-2004) supports the following objects:

Binary Input	Calendar
Binary Output	Notification Class
Binary Value	Command
Analog Input	LifeSafetyPoint
Analog Output	File
Analog Value	Program
Averaging	Schedule
Life Safety Zone	Trend Log
Multi-state Input	Group
Multi-state Output	Event Enrollment
Multi-state value	Device
Loop	

NOTE: From ICONICS point of view, the above types are part of a known specification. But we can only perform read/write operations. That means, we cannot read whole buffer for Trend Log, we cannot

evaluate Events and etc. But, we can read how many events are there, how many samples for trend is there and etc.

Browsing BACnet Devices

Because BACnet application pool is generated in runtime and it can be browsed only remotely (it is expected that you have your BACnet device placed on the network), it is not possible to browse for local ICONICS BACnet Simulation data. Because of this, you need another PC (running ICONICS Simulator or real BACnet device) to provide the application pool for your browsing.

This application note describes how to browse for the data on the network and how to read the data by ICONICS GraphWorX32 client. You can follow the steps below to read the data from BACnet device:

- 1. Start GraphWorX32 from Start → Programs → ICONICS GENESIS32 → GraphWorX32 → GraphWorX32.
- 2. Create a new process point on your GraphWorX32 display and click on "Data Tags" button to browse for the BACnet data.
- 3. When Unified Data Browser is opened you need to click on BACnet tab (GenBroker Plugin) and select an appropriate input as shown on figure below.

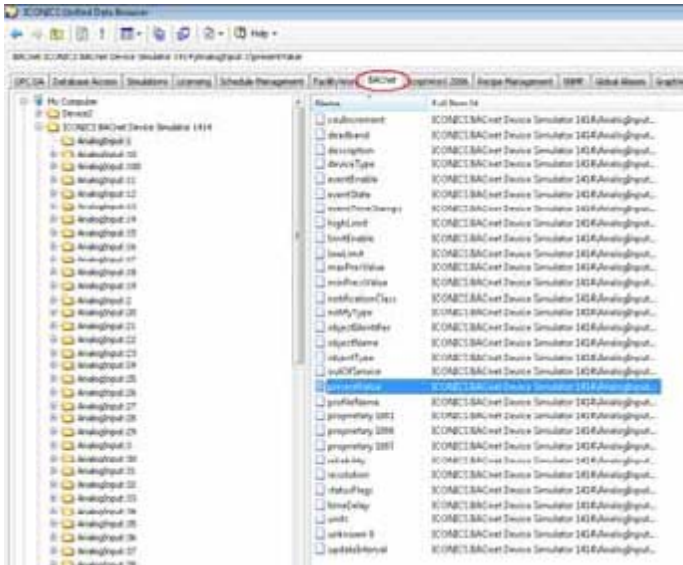


Figure 1 – Browsing BACnet device

NOTE: Usually, there is a "presentValue" which provides the current value from that device.

- 4. Click OK to confirm the selection.
- 5. Please notice PPT syntax for BACnet and click OK.

6. Switch to Runtime and you should see the similar output as shown on figure below.



Figure 2 – BACnet data in GraphWorX32 runtime

Understanding BACnet Application Pool

BACnet application pool contains the following objects below and it is very important to understand how it is designed. Here is an example on which it can be shown:

```
BACnet:ICONICS BACnet Device Simulator
498\AnalogInput:1\presentValue
```

where

BACnet: is the Genbroker plug-in prefix which indicates it is BACnet Process Point
 ICONICS BACnet Device Simulator 498 is the Device name with unique ID 498
 AnalogInput:1 translates to the <ObjectType>:<Instance ID>
 presentValue is the <Property name> (by default ID=85).