

Testing and Validating Smart Energy Devices using the Daintree Networks Sensor Network Analyzer

Application Note AN029



Copyright © 2003–2008, Daintree Networks Inc
All rights reserved

Trademarks and acknowledgements

- ZigBee® is a registered trademark of the ZigBee Alliance.
- 802.15.4™ is a trademark of the Institute of Electrical and Electronics Engineers (IEEE).
- Pentium® is a registered trademark of Intel Corporation.
- Microsoft®, Windows®, and other Microsoft products mentioned herein are trademarks or registered trademarks of Microsoft Corporation.

These trademarks are registered by their respective owners in certain countries only. Other brands and their products are trademarks or registered trademarks of their respective holders and should be noted as such.

Disclaimer

This note and any examples it contains are provided as-is and are subject to change without notice. Except to the extent prohibited by law, Daintree Networks makes no express or implied warranty of any kind with regard to this guide, and specifically disclaims the implied warranties and conditions of merchantability and fitness for a particular purpose. Daintree Networks shall not be liable for any errors or incidental or consequential damage in connection with the furnishing, performance or use of this guide and the examples included.

The software described in this guide is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of those agreements.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or any means electronic or mechanical, including photocopying and recording, for any purpose other than the purchaser's personal use, without the written permission of Daintree Networks.

Sensor Network Analyzer Release 2.3 (2008-06-27)

About this application note

This application note describes procedures by which the Sensor Network Analyzer (SNA), together with the XML-based API provided, can be used to test and validate ZigBee Smart Energy devices.

About the Sensor Network Analyzer (SNA)

The SNA combines a powerful protocol analyzer with network visualization, measurements and diagnostics for IEEE 802.15.4 and ZigBee applications. It provides automatic display of network formation, topology changes, and router and coordinator state changes allowing rapid detection of incorrect network behavior and identification of device or network failures.

It also provides a powerful commissioning tool that helps to hide the complexity of the underlying technology, and provides straight-forward configuration, testing and troubleshooting capabilities. Its graphical representation makes it fast and easy for installers to monitor network formation and measure key parameters such as link quality and bindings.

About the SNA's API

The SNA provides an XML-based string over the TCP/IP socket API (Application Program Interface) to send and receive commands over-the-air using the 2400E Adapter. Detailed explanations of all API commands can be found in the *SNA API Reference Guide* (available from the SNA's Help menu).

About the 2400E Sensor Network Adapter

Daintree's 2400E Sensor Network Adapter can be used as an active device, which means that under the control of the SNA it is capable of active analysis.

Active devices are able to join an 802.15.4 or ZigBee network, interact with other devices on it, and actively poll devices to gain information not available through passive sniffing alone. They can also issue commands to network devices, such as configuration settings during commissioning, and the API commands described in this application note.

Visit www.daintree.net to find out more about Daintree products.



Before You Begin

This application note assumes use of the Daintree Networks Sensor Network Analyzer (SNA) Professional edition, and an active device (Daintree's 2400E, Embers EM250 or Integration dongle) which provides the following:

- Passive protocol analyzer that can capture and decode all observed packets
- Active functionality:
 - Starting/joining a network using Key Establishment security
 - An API that is used to inject ZDO requests or APS data into the network to simulate devices

Testing application profiles using the API

Using a combination of the SNA, an active device, and the API, you can quickly test and validate your device's Smart Energy implementation independently.

For more information about any of the API commands used, refer to the *SNA API Reference Guide* (available from the SNA's Help menu).

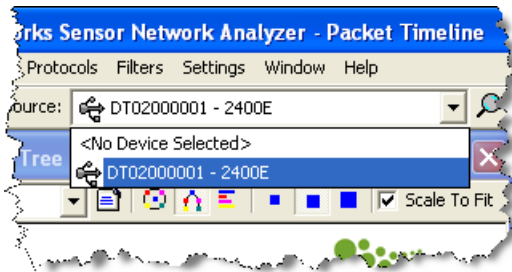
In summary, the steps required to test an application profile are as follows:

1. Start the SNA to monitor and control the test network.
2. Register the endpoints you are interested in.
3. Create the test network, with the SNA and the device under test.
4. Start the API.
5. Send the commands via the API and observe how the device under test responds.

Each one of these steps will now be described in greater detail.

1. Start the SNA for the test network




1. Start the SNA application and connect the active device. If you need additional instructions, refer to the *SNA Quick Start Guide*, which is available from the SNA's Help menu.
2. Select the active device from the **Source** list.

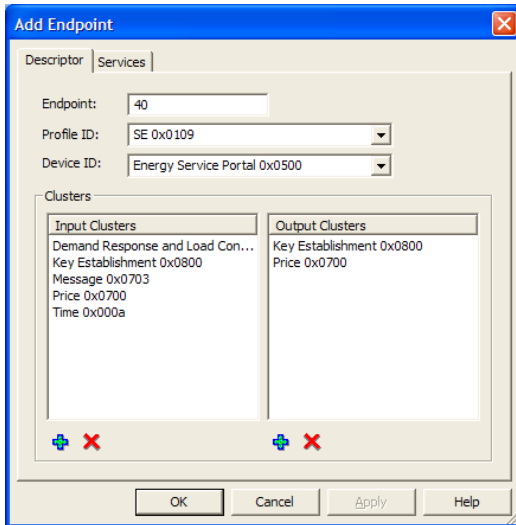


2. Register endpoints

Note that the endpoints you register must support your profile. The SNA provides a number of pre-defined endpoints. If none of the endpoints provided support the cluster/profile required, you need to create a new one or edit an existing one.

Create a new endpoint

1. Click  to open the Device Manager, and then select the **Commission** tab.
2. Click the active device to select it, and then click  to change its **Settings**.
3. Select the **Endpoints** tab, and then click  to add a new endpoint.
4. Specify the **Endpoint** number, **Profile ID**, **Device ID** and list of Input and Output **Clusters** to define your new endpoint. Then click **OK** to save the new endpoint definition.



Adding an endpoint creates a simple descriptor in the local ZigBee stack. Note that endpoints must be defined prior to joining or starting a ZigBee network and cannot be modified once the stack is running.

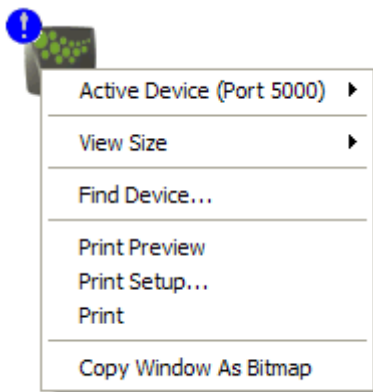
3. Create the test network with the SNA and the device under test

Instructions on how to do this using Key Establishment can be found in the Daintree Application Note AN027, *Implementing ZigBee Key Establishment Security using Daintree's SNA*.

4. Start the API

Once the SNA has connected to an active device, a TCP port becomes available to enable use of the API. This port can be used to send commands to the active device and to receive the corresponding responses and indications.

1. On the Visual Device Tree window, right-click the Active Device to show the TCP port to which it is bound.



2. Connect to the API via TCP:

```
$ tel net local host 5000
Trying 127.0.0.1...
Connected to local host.
Escape character is '^]'.
```

In some cases you may need to instruct your telnet program to perform echo locally. The mechanism for doing this differs between telnet programs. For the default Windows telnet, do the following:

1. Type "Ctrl-]"
2. Type "set localecho"
3. Press <Enter> twice to return control to telnet.

5. Start sending commands

Note: Refer to the *SNA API Reference Guide* (available from the SNA's Help menu) for detailed descriptions and examples of how to use all API commands.

The .xml commands will generate APS data frames, which will be transmitted from the active device to the device under test.

1. Open the file SE_commands.xml in a .xml reader, and decide which message to send.
2. Edit destination addresses and source/destination endpoints to match the network you created previously in steps 2 and 3.
3. Copy the updated command and paste it into the telnet session, and press <Enter> to send.
4. Observe the response of the device under test on both the SNA screen and your device's console.

The following commands (Load Control Event, Cancel Load Control Event, and Cancel All Load Control Events) are listed here as examples. More Smart Energy commands from the Price, DRLC, Simple Metering and Messaging cluster are shown in the SE_commands.xml file available from support@daintree.net.

Load Control Event API command:

```
<command name="APSDE_DATA_REQUEST" >
  <i denti fi cati on>
    <cns: descri pti on>Server to Client. Load control event. </cns: descri pti on>
  </i denti fi cati on>
  <fi el ds>
    <fi el d name="DestAddressMode" >
      <val ue>1</val ue>
    </fi el d>
    <fi el d name="DestAddress" >
      <val ue>0x8672</val ue>
    </fi el d>
    <fi el d name="DestEndpoi nt" >
      <val ue>32</val ue>
    </fi el d>
    <fi el d name="Profi l eID" >
      <val ue>0x109</val ue>
    </fi el d>
    <fi el d name="Cl usterID" >
      <val ue>0x701</val ue>
    </fi el d>
    <fi el d name="SrcEndpoi nt" >
      <val ue>204</val ue>
    </fi el d>
    <fi el d name="AsduLength" >
      <val ue>26</val ue>
    </fi el d>
    <fi el d name="Asdu" >
      <val ue>090b00000000001000000000000100010100000000000000</val ue>
    </fi el d>
    <fi el d name="TxOpti ons" >
      <val ue>0</val ue>
    </fi el d>
    <fi el d name="Radi usCounter" >
      <val ue>0</val ue>
    </fi el d>
  </fi el ds>
</command>
```

Resulting packet sent

```

+ Frame Control: 0x00
- Destination Endpoint: 0x20
- Cluster Identifier: Demand Response and Load Control (0x0701)
- Profile Identifier: SE (0x0109)
- Source Endpoint: 0xcc
- Counter: 0x10
ZigBee ZCL
+ Frame Control: 0x09
- Transaction Sequence Number: 0x0b
- Command Identifier: Load Control Event (0x00)
+ Load Control Event Command Payload
- Issuer Event ID: 0
+ Device Class
- Value: 0x0001
  .... 1 = HVAC Compressor or Furnace: Action required (0x01)
  .... 0 = Strip Heaters/Baseboard Heaters: No action required (0x00)
  .... 0 = Water Heater: No action required (0x00)
  .... 0 = Pool Pump / Spa / Jacuzzi: No action required (0x00)
  .... 0 = Smart Appliances: No action required (0x00)
  .... 0 = Irrigation Pump: No action required (0x00)
  .... 0 = Managed Commercial and Industrial Loads: No action required (0x00)
  .... 0 = Simple misc. (Residential On/Off) loads: No action required (0x00)
  .... 0 = Exterior Lighting: No action required (0x00)
  .... 0 = Interior Lighting: No action required (0x00)
  .... 0 = Electric Vehicle: No action required (0x00)
  .... 0 = Generation Systems: No action required (0x00)
  0000 = Reserved: 0x00
- Utility Enrolment Group: All Groups (0)
- Start Time: Now (0x00000000)
- Duration In Minutes: 1 minutes
- Criticality Level: Green (0x01)
- Cooling Temperature Offset: 1 x 0.1 degrees Celsius
- Heating Temperature Offset: 0 x 0.1 degrees Celsius
- Cooling Temperature Set Point: 0 x 0.01 degrees Celsius
- Heating Temperature Set Point: 0 x 0.01 degrees Celsius
- Average Load Adjustment Percentage: 0 %
- Duty Cycle: 0 % of the time
+ Event Control: 0x00
  .... 0 = Start Time: Randomized Start not Applied (0x00)
  .... 0 = End Time: Randomized End not Applied (0x00)
  0000 00.. = Reserved: 0x00

0000: 61 88 4e 94 24 70 49 00 00 48 02 70 49 00 00 1e  a.N.$PI..H.PI...
0010: 94 28 44 00 00 00 01 00 00 80 37 c2 50 00 00 00  .(D.....7BP...
0020: 20 01 07 09 01 cc 10 09 0b 00 00 00 00 00 01 00  ....L.....
0030: 00 00 00 00 00 01 00 01 01 00 00 00 00 00 00 00  .....
0040: 00 9d d4 70 57 .. .. .TpW..
    
```

Cancel Load Control Event API command

```

<command name="APSDE_DATA_REQUEST">
  <i denti fi cati on>
    <cns:descri pti on>Server to Client. Cancel Load control event.</cns:descri pti on>
  </i denti fi cati on>
  <fi el ds>
    <fi el d name="DestAddressMode">
      <val ue>1</val ue>
    </fi el d>
    <fi el d name="DestAddress">
      <val ue>0x1ddc</val ue>
    </fi el d>
    <fi el d name="DestEndpoi nt">
      <val ue>32</val ue>
    </fi el d>
    <fi el d name="Profi l eID">
      <val ue>0x109</val ue>
    </fi el d>
    <fi el d name="Cl usterID">
      <val ue>0x701</val ue>
    </fi el d>
    <fi el d name="SrcEndpoi nt">
      <val ue>204</val ue>
    </fi el d>
    <fi el d name="AsduLength">
      <val ue>15</val ue>
    </fi el d>
    <fi el d name="Asdu">
      <val ue>09 0b 01 00 00 00 00 01 00 00 00 00 00 00 00</val ue>
    </fi el d>
  </fi el ds>
</command>
    
```

Testing and Validating Smart Energy Devices using Daintree's SNA

```

</field name="TxOptions">
  <value>0</value>
</field>
<field name="RadiusCounter">
  <value>0</value>
</field>
</fields>
</command>

```

Resulting packet sent

The screenshot shows a packet capture window with two frames:

- ZigBee APS:** Frame Control: 0x00, Destination Endpoint: 0x20, Cluster Identifier: Demand Response and Load Control (0x0701), Profile Identifier: SE (0x0109), Source Endpoint: 0xcc, Counter: 0x13.
- ZigBee ZCL:** Frame Control: 0x09, Transaction Sequence Number: 0x0b, Command Identifier: Cancel Load Control Event (0x01). The payload is expanded to show:
 - Cancel Load Control Event Command Payload
 - Issuer Event ID: 0
 - Device Class: Value: 0x0001
 -1 = HVAC Compressor or Furnace: Action required (0x01)
 -0 = Strip Heaters/Baseboard Heaters: No action required (0x00)
 -0 = Water Heater: No action required (0x00)
 -0 = Pool Pump / Spa / Jacuzzi: No action required (0x00)
 -0 = Smart Appliances: No action required (0x00)
 -0 = Irrigation Pump: No action required (0x00)
 -0 = Managed Commercial and Industrial Loads: No action required (0x00)
 -0 = Simple misc. (Residential On/Off) loads: No action required (0x00)
 -0 = Exterior Lighting: No action required (0x00)
 -0 = Interior Lighting: No action required (0x00)
 -0 = Electric Vehicle: No action required (0x00)
 -0 = Generation Systems: No action required (0x00)
 -0 = Reserved: 0x00
 - Utility Enrolment Group: All Groups (0)
 - Cancel Control: 0x00
 -0 = Cancel Control: Terminate Event at the effective time (0x00)
 -0 = Reserved: 0x00
 - Effective Time: Now (0x00000000)

At the bottom, a hex dump shows the raw bytes of the packet:

```

0000: 61 88 75 94 24 70 49 00 00 48 02 70 49 00 00 1e  a.u.sPI..H.PI...
0010: b5 28 6a 02 00 00 01 00 00 80 37 c2 50 00 00 00  5(j).....7BP...
0020: 20 01 07 09 01 cc 13 09 0b 01 00 00 00 00 01 00  ....L.....
0030: 00 00 00 00 00 00 0e 94 0c 9f ..  ....^.....

```

Cancel All Load Control Events API Command

```

<command name="APSDE_DATA_REQUEST">
  <identi fication>
    <ns:descri ption>Server to Client. Cancel All Load control events.</ns:descri ption>
  </identi fication>
  <fields>
    <field name="DestAddressMode">
      <value>1</value>
    </field>
    <field name="DestAddress">
      <value>0x1ddc</value>
    </field>
    <field name="DestEndpoint">
      <value>32</value>
    </field>
    <field name="ProfileID">
      <value>0x109</value>
    </field>
    <field name="ClusterID">
      <value>0x701</value>
    </field>
    <field name="SrcEndpoint">
      <value>204</value>
    </field>
    <field name="AsduLength">
      <value>4</value>
    </field>
    <field name="Asdu">
      <value>09 0b 02 00</value>
    </field>
    <field name="TxOptions">
      <value>0</value>

```

Testing and Validating Smart Energy Devices using Daintree's SNA

```
</field>
<field name="RadiusCounter">
  <value>0</value>
</field>
</fields>
</command>
```

Resulting packet sent:

```
ZigBee APS
  Frame Control: 0x00
  Destination Endpoint: 0x20
  Cluster Identifier: Demand Response and Load Control (0x0701)
  Profile Identifier: SE (0x0109)
  Source Endpoint: 0xcc
  Counter: 0x14
ZigBee ZCL
  Frame Control: 0x09
  Transaction Sequence Number: 0x0b
  Command Identifier: Cancel All Load Control Events (0x02)
  Cancel All Load Control Events Command Payload
  Cancel Control: 0x00
    .... ..0 = Cancel Control: Terminate Event immediately (0x00)
    0000 000. = Reserved: 0x00

0000:  61 88 7d 94 24 70 49 00 00 48 02 70 49 00 00 1e  a.).$pI..H.pI...
0010:  bd 28 72 02 00 00 01 00 00 80 37 c2 50 00 00 00  =(r.....7BP...
0020:  20 01 07 09 01 cc 14 09 0b 02 00 3e ef c7 20 ..  ....L.....>0G .
0030:  ..
```

What next?

Now that you know how to send commands via the API, you can alter the commands sent to test your device's response.

```
</field>
<field name="SrcEndpoint">
  <value>204</value>
</field>
<field name="AsduLength">
  <value>4</value>
</field>
<field name="Asdu">
  <value>09 0b 02 00</value>
</field>
<field name="TxOptions">
  <value>0</value>
</field>
<field name="RadiusCounter">
  <value>0</value>
</field>
</fields>
</command>
```

```
ZigBee ZCL
  Frame Control: 0x09
  Transaction Sequence Number: 0x0b
  Command Identifier: Cancel All Load Control Events (0x02)
  Cancel All Load Control Events Command Payload
  Cancel Control: 0x00
    .... ..0 = Cancel Control: Terminate Event immediately (0x00)
    0000 000. = Reserved: 0x00

0000:  61 88 7d 94 24 70 49 00 00 48 02 70 49 00 00 1e  a.).$pI..H.pI...
0010:  bd 28 72 02 00 00 01 00 00 80 37 c2 50 00 00 00  =(r.....7BP...
0020:  20 01 07 09 01 cc 14 09 0b 02 00 3e ef c7 20 ..  ....L.....>0G .
0030:  ..
```

You can change any fields of the packet sent by altering the "Asdu" field (see A above). This field is decoded in green in the SNA packet decode window. If you highlight the field in the packet decode window, the associated byte is highlighted below. This makes it easy to identify which bytes to change if you want to change fields in the message sent. If you change the length of the packet, the "ASDULength" (see B above) field will also need to be changed.

If you are testing an in-premise display, you can

1. Ask it to display a message for 1 minute
2. Ask it to display a message for 5 minutes; then cancel the message after 2 minutes

If you are testing a Load Control Device you can

1. Send a Load Control Event, then check your device's console to see if it is listed
2. Cancel the Load Control Event, and make sure your device removes it from its list of events

You can also alter the security of the messages via the TxOptions field (See C on previous page). This means that you can try sending a message with the wrong security type, and check that your device reacts appropriately.

TxOptions field	Security used when in a KE network
0	NWK
1	NWK + APS
4	NWK (with APS Ack requested)
5	NWK + APS (with APS Ack requested)

Using the SNA in this way means you can check your device's reaction to messages easily without leaving the lab. This will help you find simple problems before you begin interacting with other devices.

For more information about any of the API commands used, refer to the *SNA API Reference Guide* (available from the SNA's Help menu).

Restrictions and limitations

The SNA's API does not support networks based upon the ZigBee 2004 Specification (now superseded).

If you use the Integration dongle as the active device, you'll need a second device and SNA instance to monitor the traffic.