

Technical Document

Niagara^{AX-3.x} Dust Guide

Updated: February 19, 2008



Niagara^{AX} Dust Guide

Copyright © 2008 Tridium, Inc.

All rights reserved.

3951 Westerre Pkwy, Suite 350

Richmond

Virginia

23233

U.S.A.

Copyright Notice

The software described herein is furnished under a license agreement and may be used only in accordance with the terms of the agreement.

This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior written consent from Tridium, Inc.

The confidential information contained in this document is provided solely for use by Tridium employees, licensees, and system owners; and is not to be released to, or reproduced for, anyone else; neither is it to be used for reproduction of this Control System or any of its components.

All rights to revise designs described herein are reserved. While every effort has been made to assure the accuracy of this document, Tridium shall not be held responsible for damages, including consequential damages, arising from the application of the information contained herein. Information and specifications published here are current as of the date of this publication and are subject to change without notice.

The release and technology contained herein may be protected by one or more U.S. patents, foreign patents, or pending applications.

Trademark Notices

BACnet and ASHRAE are registered trademarks of American Society of Heating, Refrigerating and Air-Conditioning Engineers. Microsoft and Windows are registered trademarks, and Windows NT, Windows 2000, Windows XP Professional, and Internet Explorer are trademarks of Microsoft Corporation. Java and other Java-based names are trademarks of Sun Microsystems Inc. and refer to Sun's family of Java-branded technologies. Mozilla and Firefox are trademarks of the Mozilla Foundation. Echelon, LON, LonMark, LonTalk, and LonWorks are registered trademarks of Echelon Corporation. Tridium, JACE, Niagara Framework, Niagara^{AX} and Vykon are registered trademarks, and Workbench, WorkPlace^{AX}, and ^{AX}Supervisor, are trademarks of Tridium Inc. All other product names and services mentioned in this publication that is known to be trademarks, registered trademarks, or service marks are the property of their respective owners. The software described herein is furnished under a license agreement and may be used only in accordance with the terms of the agreement.

CONTENTS

Preface	iii
Document Change Log	iii
Dust Driver Installation	1-1
Dust Client Quick Start	2-1
Configure the DustNetwork	2-1
Add a DustNetwork	2-1
<i>To add a DustNetwork in the station</i>	<i>2-1</i>
Create a New SmartMeshClient	2-1
<i>To create a new SmartMeshClient</i>	<i>2-1</i>
Create Dust Client proxy points	2-2
Using online Discover to add Dust proxy points	2-2
<i>To discover Dust data items</i>	<i>2-2</i>
<i>To add discovered Mote channels as proxy points</i>	<i>2-2</i>
Manually adding Dust proxy points	2-3
Niagara Dust Client Concepts	3-1
About the SmartMesh network and NiagaraAX	3-1
NiagaraAX application	3-1
About SmartMesh	3-1
About dust palette components	3-2
Smart Mesh Client Manager notes	3-2
Dust Point Manager notes	3-3
Dust Point Manager "Discovered" notes	3-3
Dust Point Manager "Database" notes	3-4
Dust Plugin Guides	4-1
Dust Plugin Guides Summary	4-1
dust-DustPointManager	4-1
dust-SmartMeshClientManager	4-1
Dust Component Guides	5-1
Dust Component Guides Summary	5-1
dust-DustAlarmDeviceExt	5-1
dust-DustDeviceFolder	5-1
dust-DustNetwork	5-1
dust-DustPointDeviceExt	5-2
dust-DustPointFolder	5-2
dust-DustProxyExt	5-2
dust-NetworkConfig	5-2
dust-SmartMeshClient	5-2
dust-SystemConfig	5-3

CONTENTS

Preface

[Document Change Log](#)

Document Change Log

Updates (changes/additions) to this *NiagaraAX Dust Guide* document are listed below.

- Updated: February 19, 2008
Added component description for [DustAlarmDeviceExt](#) (Alarms device extension) for a Smart-MeshClient, which became available starting in build 3.0.100.
Converted document to “new look” print format. Changed document to reference the *NiagaraAX Drivers Guide*, a new document created from sections formerly in the *NiagaraAX User Guide*. Reversed the order of this change log to have latest entries first.
- Published: June 24, 2005
Added Copyright and Trademarks to preface. Changed document name to *NiagaraAX Dust User Guide*.
- Draft: June 15, 2005
The [SmartMeshClientManager](#) view was erroneously called the “Dust Device Manager” in several places—this was fixed. This was the only document change.
- Draft: May 26, 2005
(Initial change log). First document draft.

CHAPTER 1

Dust Driver Installation

To use the NiagaraAX Dust driver, you must have a target JACE host that is licensed with the “dust” feature. In addition, other Dust “device limits” or proxy point limits may exist in your license.

From your PC, use the Niagara Workbench 3.*n.nn* installed with the “installation tool” option (checkbox “This instance of Workbench will be used as an installation tool”). This option installs the needed distribution files (.*dist* files) for commissioning various models of remote JACE platforms. The dist files are located under your Niagara install folder in a “sw\dist” subfolder.

Apart from installing the 3.*n.nn* version of the Niagara distribution file in the JACE, make sure to install the dust module too (if not already present, or upgrade if an older revision). For details, see [“About the Commissioning Wizard”](#) in the *JACE NiagaraAX Install and Startup Guide*.

Following this, the remote JACE is now ready for Dust client software configuration in its running station, as described in the rest of this document. See [“Niagara Dust Client Concepts”](#) on page 3-1.

Note: *Basic procedures for using the online learning features of the driver, including discovery of Mote channels (and proxy point representation in your Niagara station), is in the next section. See [“Dust Client Quick Start”](#) on page 2-1.*

CHAPTER 2

Dust Client Quick Start

This section provides a collection of procedures to use the Dust Client driver to add a SmartMeshClient and learn data items that originate in “Motes.” As with other NiagaraAX drivers, you can do most configuration from special “manager” views and property sheets using Workbench.

These are the main subsections:

- [Configure the DustNetwork](#)
- [Create Dust Client proxy points](#)

Configure the DustNetwork

To add and configure the DustNetwork, perform the following main tasks:

- [Add a DustNetwork](#)
- [Create a New SmartMeshClient](#)

Add a DustNetwork

To add a DustNetwork in the station

Use the following procedure to add a [DustNetwork](#) under the station’s Drivers container.


- Step 1 Double-click the station’s Drivers container, to bring up the **Driver Manager**.
- Step 2 Click the **New** button to bring up the New DeviceNetwork dialog. For more details, see “[Driver Manager New and Edit](#)” in the *Drivers Guide*.
- Step 3 Select “DustNetwork,” number to add: 1, and click **OK**.
This brings up a dialog to name the network.
- Step 4 Click **OK** to add the DustNetwork to the station.
You should have a DustNetwork named “DustNetwork” (or whatever you named it), under your Drivers folder.

Create a New SmartMeshClient

In the NiagaraAX Dust network architecture, a SmartMeshClient is the “device-level” component.

To create a new SmartMeshClient

You must know a number of things about the remote SmartMesh Client, including its hostname, and (if not using factory-defaults) a client login username, password, and control port.

- Step 1 Double-click the **DustNetwork**, or:
right-click the DustNetwork and select **Views > Smart Mesh Client Manager**.
This brings up the **Smart Mesh Client Manager**.
- Step 2 Click the **New** button  to create a Smart Mesh Client.
The **New** device dialog appears, with “Number to Add” as “1.” Click **OK**.
- Step 3 Another **New** dialog appears (see [Figure 3-2](#) on page 2).
In this dialog, note the following:
 - Hostname — enter a valid hostname or IP address for the remote SmartMesh Manager.
 - Username, Password, Control Port — enter values if necessary (“defaults” are supplied).
- Step 4 Click **OK** to add the SmartMeshClient to your station.

If the SmartMeshClient component is properly configured, it will show status of “ok” in the Smart Mesh Client Manager view of the network (Figure 3-3 on page 3).

See the next section: “Create Dust Client proxy points”.

Create Dust Client proxy points

As with device objects in other drivers, each SmartMeshClient has a Points extension that serves as the container for proxy points. The default view for any Points extension is the Point Manager (here, the DustPointManager). You use it to create Dust proxy points under any SmartMeshClient.

This section provides quick start procedures for both tasks, as follows:

- Using online Discover to add Dust proxy points
- Manually adding Dust proxy points

Note: For general information, see “About the Point Manager” in the Drivers Guide. For specific notes, see “Dust Point Manager notes” on page 3-3.

Using online Discover to add Dust proxy points



This is the recommended way to add Dust proxy points under a SmartMeshClient. Use the following procedures:

- To discover Dust data items
- To add discovered Mote channels as proxy points

Note: If your DustNetwork has multiple SmartMesh Clients, repeat both procedures (discover and add) for each SmartMeshClient, until you have all needed points proxied in the station.

To discover Dust data items

Perform this task to discover Dust data items.

- Step 1 In the **Smart Mesh Device Manager**, double-click the **Points** icon  in the **Exts** column of the row representing the SmartMesh Client you wish to explore. This brings up its **Dust Point Manager**.
- Step 2 Click the **Discover** button  to learn the Motes and data items under the SmartMesh Client. When the discovery job completes, discovered Motes are listed in the *top pane* as *folders*, in the “Discovered” table. Initially, each Mote folder occupies one row (see Figure 3-4 on page 3).
- Note:** If you double-click (or select and click **Add**) on a discovered Mote folder, this automatically creates an empty **DustPointFolder** for that Mote in the station database. If desired, you can do this for all discovered Motes (the Dust Point Manager view is available on each DustPointFolder—in the lower Database pane, simply double-click the folder).
- Step 3 Beside any discovered Mote, click the plus (+) to see its data channels (Figure 3-5 on page 4).
- Note:** Channels that are listed in the Discovered table with “false” in the **Enable** column mean that they are disabled in the server. Any of these you add as proxy points will be unable to receive values.
- Step 4 To add Dust proxy points, see “To add discovered Mote channels as proxy points”.

To add discovered Mote channels as proxy points

Perform this task to add Dust proxy points.

- Step 1 Select the Mote channel(s) in the (top) Database pane of the DustPointManager.
- Step 2 You can map selected items in the station in different ways:
- Drag from the Discovered pane to Database pane (brings up an **Add** dialog).
 - Double-click an item in the Discovered pane (also brings up an **Add** dialog).
 - Click to select in Discovered, then press “a”. (“Quick Add”, meaning *no Add* dialog).
- This works the same as in other driver’s Point Manager views.
- Step 3 When the **Add** dialog appears, you often accept default values, except for Name and possibly Type. For complete details, see Dust Point Manager notes.
- The following brief summaries explain **Add** dialog fields:
- **Name** is the Dust proxy point name—you typically change this as needed.
 - **Type** is Niagara control point type to use for the proxy point.
Note: Unlike other entries in the Add dialog, you cannot edit the point’s Type later.
 - **Mote Id** is the Mote in which the point is located.

- **Channel Id** is the channel within the Mote (identified by Mote Id) for the point.
- **Mode** specifies whether the data item is read-only (RO), read-write (RW), or write-only (WO).
- **Enabled** determines whether Niagara communications occur (Enabled) or not (Disabled).
- **Facets** are the Dust proxy point's facets, for how the value should be displayed in Niagara.
- **Device Facets** reflect any facets learned from the Mote about this channel.

Step 4 When you have Mote channel(s) configured properly for your usage, click **OK**.

The proxy points are added to the station, and appear listed in the Database pane. See [Figure 3-6](#). For more details, see “[Dust Point Manager notes](#)” on page 3-3.

Manually adding Dust proxy points

You can manually add Dust proxy points, using the **New** button in the Dust Point Manager, or by dragging from the **dust** palette. However, this method is generally not recommended.

CHAPTER 3

Niagara Dust Client Concepts

This section provides key concepts about the NiagaraAX Dust driver.

The following main sections are included:

- [About the SmartMesh network and NiagaraAX](#)
- [About dust palette components](#)
- [Smart Mesh Client Manager notes](#)
- [Dust Point Manager notes](#)

About the SmartMesh network and NiagaraAX

The following sections provide a brief overview:

- [NiagaraAX application](#)
- [About SmartMesh](#)

NiagaraAX application

The NiagaraAX Dust client driver is a monitoring tool, not a native configuration tool. For configuration of a SmartMesh™ network, use the SmartMesh™ Console application available from Dust™ Networks, Inc.

About SmartMesh

The following terms and summaries provide a high-level view of a SmartMesh system:

- [SmartMesh network](#)
- [SmartMesh Manager](#)
- [Mote](#)
- [Control Port / Channel](#)
- [Notification Port / Channel](#)

SmartMesh network

A SmartMesh network is a wireless mesh network that can be used for a variety of monitoring applications, such as building automation, industrial monitoring, or remote site security.

SmartMesh Manager

A SmartMesh Manager is line-powered network node that controls and monitors a wireless mesh network of up to 50 “Motes.” The Manager coordinates routing, aggregates mote packets, and collects statistics. The Manager is also a server to a wired network (via an Ethernet port and TCP/IP), serving data in XML format.

Mote

A Mote is a small, low-power, IEEE 802.15.4 compliant transceiver that allows an OEM a way to add wireless communications to a device. For example, a sensor, actuator, or controller can be equipped with a mote. Motes are optimized to run SmartMesh software, and provide some number of analog inputs and digital inputs/outputs, collectively called “channels.”

Control Port / Channel

The [SmartMesh Manager](#) and client applications use the control channel to exchange commands and information about a SmartMesh Network. The first command from a client to the Manager is a login, and the Manager returns a token that the client uses in all subsequent commands.

Notification Port / Channel

The [SmartMesh Manager](#) uses the notification channel to stream data and network events to a client
To subscribe to the notification channel:

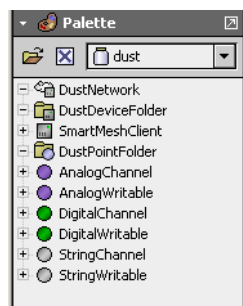
1. A client sends a subscribe method call over the control channel.
2. The Manager returns a token and port number at which the notification channel is available.
3. The client sends an authorization request containing the token to the notification port.
4. The Manager returns a stream of XML sensor data and network events.

The client application listens on the notification channel, processing the XML-wrapped data.

About dust palette components

In Workbench, open the **dust** palette in the Palette side bar to examine various Dust components, as shown in [Figure 3-1](#).

Figure 3-1 Components in dust palette

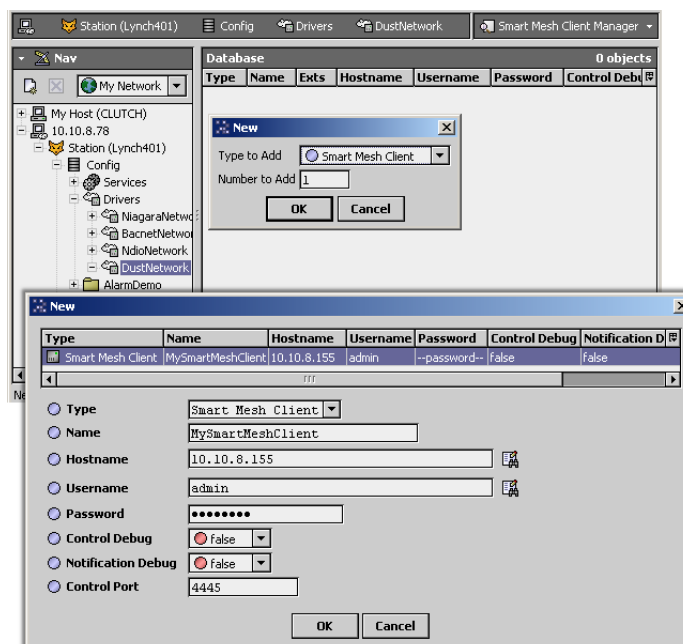


As shown, the Dust driver has relatively few components. Like most NiagaraAX drivers, you *rarely* work from the palette. Instead, various manager views simplify component creation, enforcing proper component hierarchy.

Smart Mesh Client Manager notes

The Smart Mesh Client Manager is the default view for the DustNetwork. Unlike many other driver's Device Manager, there is no "learn" of device-level ([SmartMeshClient](#)) components. To create SmartMeshClient components, use the **New** button. This produces two dialogs, see [Figure 3-2](#).

Figure 3-2 Adding SmartMeshClient in Smart Mesh Client Manager

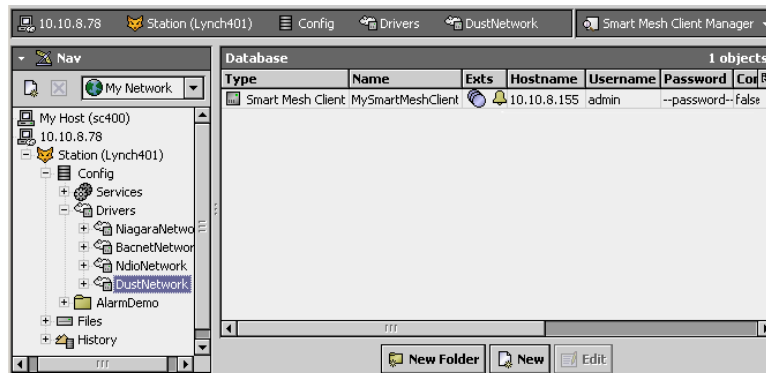


In the [Figure 3-2](#) example, all fields were left at defaults except for **Name** and **Hostname**. You must always enter the hostname of the [SmartMesh Manager](#) (without any port number).

If the SmartMesh Manager is configured with “non-default” client-access credentials (hostname and password) or “non-default” control port, configure matching entries in this dialog.

After adding a SmartMeshClient, it immediately communicates to the SmartMesh Manager, and should list in the Device Manager table with an ok status, as shown in [Figure 3-3](#).

Figure 3-3 SmartMeshClient successfully added in Smart Mesh Client Manager



If the SmartMeshClient shows a *down* status (yellow), double-click it for an **Edit** dialog (similar to **New**, see [Figure 3-2](#)), and review the properties for hostname, credentials, and control port.

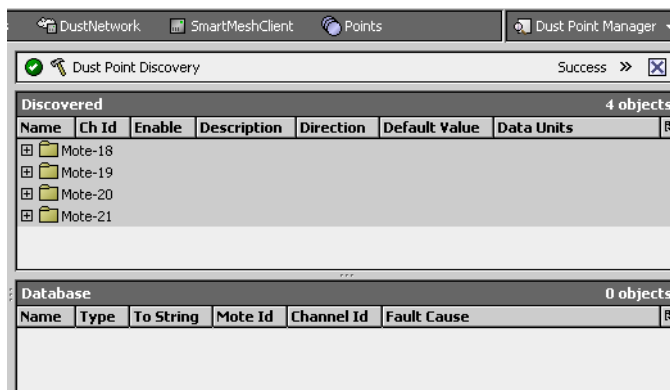
Dust Point Manager notes

The Dust Point Manager is the default view for the **Points** extension (or Points Folder) under a Smart-MeshClient. It works similar to other point managers that support online point discovery. See [“About the Point Manager”](#) in the *Drivers Guide* for general information.

Dust Point Manager “Discovered” notes

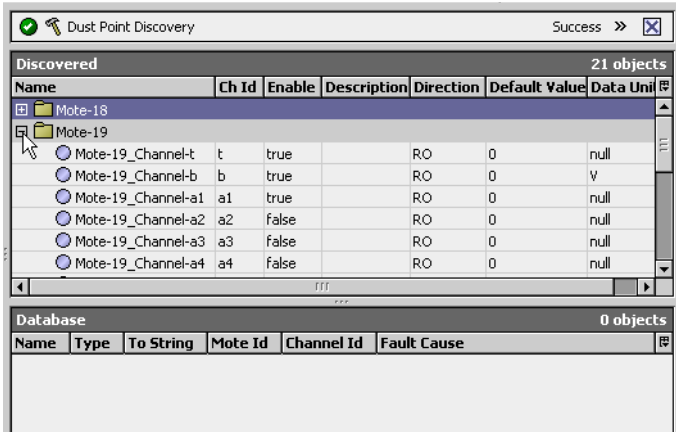
When a discover job completes, all found motes appear as expandable folders, each occupying one row in the top Discovered table ([Figure 3-4](#)).

Figure 3-4 Discovered motes appear as folders



As needed, click the plus icon “+” beside any mote folder to see all of its data channels—each one is a potential proxy point candidate ([Figure 3-5](#)).

Figure 3-5 Click plus (+) icon beside any discovered mote to see all channels



Note: Any channel showing “false” in the **Enable** column is a poor candidate for a proxy point. That channel is disabled in the server, and value updates cannot be received.

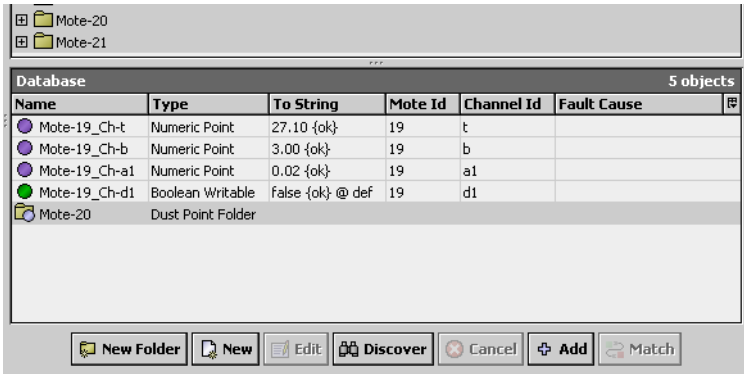
For organizational reasons, you may wish to make DustPointFolders for each discovered mote. The Dust Point Manager simplifies this—simply double-click on a discovered “mote folder” to create an identically-named DustPointFolder under the Points extension.

If you do this, note that the “Database” pane in the Point Manager for each DustPointFolder shows only Dust proxy points that reside under that folder. However, a “Discover” from any Dust Point Manager always shows *all* discovered motes.

Dust Point Manager “Database” notes

The “Database” table contains existing Dust proxy points and (if added) DustPointFolders, where each appears as a row in the table (Figure 3-4).

Figure 3-6 Database shows Dust proxy points in the station



If you created DustPointFolders under the Points container, you can see all proxy points in the Smart-MeshClient from the main (root) Points Manager using “All Descendants” tool. For details, see the section “Points New Folder and New” under “About the Point Manager” in the Drivers Guide.

CHAPTER 4

Dust Plugin Guides

Plugins provide *views* of components, and can be accessed many ways—for example, double-click a component in the tree for its *default* view. In addition, you can right-click a component, and select from its **Views** menu. For summary documentation on any view, select **Help > On View** (F1) from the Workbench menu, or press F1 while the view is open.


Summary information is provided here about the different [Dust views](#).

Dust Plugin Guides Summary

Summary information is provided on views specific to components in the `dust` module, with views listed in alphabetical order as follows:

- [DustPointManager](#)
- [SmartMeshClientManager](#)


dust-DustPointManager

 Use the DustPointManager to discover, add, edit, and delete Dust proxy points under a [SmartMeshClient](#). The DustPointManager is the default view for the [DustPointDeviceExt](#) (`Points` container) under an [SmartMeshClient](#). The DustPointManager is also the default view for any [DustPointFolder](#) under the `Points` container of an [SmartMeshClient](#).

To view, double-click a [DustPointDeviceExt](#) or [DustPointFolder](#), or click it and select **Views > Dust Point Manager**.

For general information, see “[About the Point Manager](#)” in the *Drivers Guide*. For Dust-specific details, see “[Dust Point Manager notes](#)” on page 3-3. For a related procedure, see “[Using online Discover to add Dust proxy points](#)” on page 2-2.

dust-SmartMeshClientManager

 Use the SmartMeshClientManager to create, edit, and view [SmartMeshClients](#) under a [DustNetwork](#). The DustDeviceManager is the default view on the [DustNetwork](#). To view, double-click the [DustNetwork](#), or right-click it and select **Views > Smart Mesh Client Manager**.

For general information, see “[About the Device Manager](#)” in the *Drivers Guide*. For Dust-specific details, see “[Smart Mesh Client Manager notes](#)” on page 3-2. For a related procedure, see “[Create a New Smart-MeshClient](#)” on page 2-1.

Note: Online discovery of “device-level” [SmartMeshClient](#) components is not available.

CHAPTER 5

Dust Component Guides


These component guides provides summary help on [Dust components](#).

Dust Component Guides Summary

Summary information is provided on components specific to the `dust` module, listed in alphabetical order as follows:

- [DustAlarmDeviceExt](#)
- [DustDeviceFolder](#)
- [DustNetwork](#)
- [DustPointDeviceExt](#)
- [DustPointFolder](#)
- [DustProxyExt](#)
- [NetworkConfig](#)
- [SmartMeshClient](#)
- [SystemConfig](#)

dust-DustAlarmDeviceExt


 `DustAlarmDeviceExt` (Alarms) is a device extension for a [SmartMeshClient](#), available starting in build 3.0.100, and in AX-3.1 and later. The Alarms extension has no special views, and its property sheet contains two properties:

- **Alarm Class**
To specify the local alarm class in the station to which to route received Dust events.
- **Last Time Received**
Status timestamp of last received Dust event.

The following types of Dust events are supported:

- `notifications/event/alarmOpen`
- `notifications/event/alarmClose`
- `notifications/event/netMoteJoin`
- `notifications/event/netMoteLive`
- `notifications/event/netMoteUnknown`

dust-DustDeviceFolder

 `DustDeviceFolder` is the Dust implementation of a folder under a [DustNetwork](#). Typically, you add such folders using the **New Folder** button in the [SmartMeshClientManager](#) view of the `DustNetwork`. Each `DustDeviceFolder` has its own [SmartMeshClientManager](#) view.

dust-DustNetwork

 `DustNetwork` represents a tree of SmartMesh Manager clients and ancillary objects. This network-level component is a NiagaraAX Framework convention, and has no physical equivalent in any Dust Networks system.


The `DustNetwork` has the standard network components and status properties such as status and enabled—see “[Common network components](#)” in the *Drivers Guide* for more details.

In addition, the following `DustNetwork` properties are unique or have special importance:

- **Status**
Will be fault if the Dust driver is not properly licensed. The license feature entry is “dust”.


- **Thread Pool**
This controls the number of threads used to execute all actions of all Dust components in the network. This includes most communications with remote devices, which can be multi-threaded, so if there are performance issues, you can increase the number of threads. Data updates and events on the notification port cannot be optimized with the number of threads.
Max Threads: Default value is 4.

dust-DustPointDeviceExt

 DustPointDeviceExt (Points) is a device extension for a [SmartMeshClient](#). It functions as the container for Dust proxy points, each representing a channel in a Mote. You can also add one or more [DustPointFolder](#)s under the Points container.


The default view of the DustPointDeviceExt is the DustPointManager.

dust-DustPointFolder

 DustPointFolder is the Dust implementation of a folder under a [SmartMeshClient](#)'s Points extension. Typically, you add this folder by double-clicking a discovered “Mote folder” in the DustPointManager. This creates a DustPointFolder in the station named the same as the discovered mote.

Or, you add such folders using the **New Folder** button in the DustPointManager. Each DustPointFolder has its own DustPointManager view.

dust-DustProxyExt

 DustProxyExt is the proxy extension used by any Dust proxy point. Each Dust proxy point represents a specific channel in a Mote. Typically, a channel represents some value in a sensor or actuator.

The DustProxyExt has common ProxyExt properties, such as status and enabled. See “[ProxyExt properties](#)” in the *Drivers Guide* for more details.


In addition, the following DustProxyExt properties are unique or have special importance:

- **Mote Id**
The ID of the mote in which the channel originates.
- **Channel Id**
Channel within the mote (identified by Mote Id) of the point.
- **Sample Time**
Server time of the current value.
- **Async Notification**
Whether or not the point is currently subscribed to receive values from the notification channel.
Note: This does not mean a subscription with the backend system—it is internal to the client driver only.

dust-NetworkConfig

 NetworkConfig is a child component of a [SmartMeshClient](#), and contains read-only diagnostic properties updated when the client first connects to the server.

dust-SmartMeshClient

 SmartMeshClient represents a remote [SmartMesh Manager](#), a TCP/IP node that controls and monitors a [SmartMesh network](#), a wireless mesh network of up to 50 [Motes](#). A Mote is a wireless transceiver that has connections for analog, digital, and serial sensors and actuators.

The SmartMeshClient is the “device-level” component in the NiagaraAX network architecture. It has two device extensions: Points and Alarms. It also has the standard device component properties such as status and enabled. See “[Common device components](#)” in the *Drivers Guide* for more details.

In addition, the following SmartMeshClient properties are unique or have special importance:

- **Hostname**
The TCP/IP host name or address of the SmartMesh Manager. Must not include a port number.
- **Username / Password**
Credentials understood by the SmartMesh Manager for access by a client.
- **Control Debug**
Diagnostic tool for printing control messages to the station’s standard output.
- **Control Port**
TCP/IP port number of the Dust control interface.

- **Control Token**
Diagnostic property. Server-assigned identifier for the client.
- **Notification Debug**
Diagnostic tool for printing notification messages to the station's standard output.
- **Notification Port**
Diagnostic property. Server-assigned TCP/IP port of the notification interface.
- **Notification Token**
Diagnostic property. Server-assigned identifier for the client.
- **NetworkConfig**
Container for read-only diagnostic information, updated when the client first connects to the server.
- **SystemConfig**
Container for read-only diagnostic information, updated during ping.
- **State**
The read-only current client state, either:
 - Attached
 - Attaching
 - Detached
 - Detaching

dust-SystemConfig

-  SystemConfig is a child component of a [SmartMeshClient](#), and contains read-only diagnostic properties updated during a ping.

