

Technical Document

Danfoss AK255 Driver Guide

July 20, 2012



Niagara^{AX-3.x} Danfoss AK255 Driver Guide

3951 Westerre Pkwy., Suite 350

Richmond

Virginia

23233

U.S.A.

Confidentiality Notice

The information contained in this document is confidential information of Tridium, Inc., a Delaware corporation ("Tridium"). Such information and the software described herein, is furnished under a license agreement and may be used only in accordance with that agreement.

The information contained in this document is provided solely for use by Tridium employees, licensees, and system owners; and, except as permitted under the below copyright notice, is not to be released to, or reproduced for, anyone else.

While every effort has been made to assure the accuracy of this document, Tridium is not responsible for damages of any kind, including without limitation 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 latest product specifications can be found by contacting our corporate headquarters, Richmond, Virginia.

Trademark Notice

BACnet and ASHRAE are registered trademarks of American Society of Heating, Refrigerating and Air-Conditioning Engineers. Microsoft, Excel, Internet Explorer, Windows, Windows Vista, Windows Server, and SQL Server are registered trademarks of Microsoft Corporation. Oracle and Java are registered trademarks of Oracle and/or its affiliates. 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, NiagaraAX Framework, and Sedona Framework are registered trademarks, and Workbench, WorkPlaceAX, and AXSupervisor, 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

Copyright and Patent Notice

This document may be copied by parties who are authorized to distribute Tridium products in connection with distribution of those products, subject to the contracts that authorize such distribution. It may not otherwise, 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.

Copyright © 2009-2012 Tridium, Inc. All rights reserved.

The product(s) described herein may be covered by one or more U.S. or foreign patents of Tridium.

CONTENTS

Document Change Log	v
Related documentation	ii–v

Introduction 1–1

Audience	1–1
Requirements	1–1
Module	1–1
Compatibility	1–1

Installation and setup 2–1

Using the driver	2–1
<i>Create a new station that includes the driver</i>	<i>2–1</i>
<i>Add the driver to an existing station</i>	<i>2–1</i>
<i>Add a device and set its URL</i>	<i>2–1</i>

Components and Plugins (Views)..... 3–1

AK255Network	3–1
AK255 Device Manager view	3–1
AK255Device	3–1
AK255AlarmDeviceExt	3–15
AK255 AlarmDetail	3–16
AK255 Alarm Manager View	3–16
AK255PointDeviceExt	3–17
AK255 Point Manager View	3–17
AK255DeviceFolder	3–17
AK255PointFolder	3–17
Ak255ProxyExt	3–17

PREFACE

Preface

- [Document Change Log](#)
- [Related documentation](#)

Document Change Log

Updates (changes/additions) to this *Danfoss AK255 Driver Guide* document are listed below.

- Initial publication July 20, 2012.

Related documentation

The following Danfoss document is related to the content in this document and may provide additional information on the topics it covers:

- *AK255 XML Interface* (document ID: AK255 XML Interface 2.101a)

CHAPTER 1

Introduction

This document describes how to use the AK255 driver (version AX-3.7 or later) to integrate a network of Danfoss AK255 refrigeration control devices into the NiagaraAXFramework.

Audience

Readers of this document are:

- Knowledgeable Danfoss AK255 users
- NiagaraAX certified

Requirements

- NiagaraAX-3.7 or later
- A license to use the **AK255** feature. Other device limits or proxy-point limits may apply to your license.

Module

The AK255 driver is contained in the **ak255.jar**.

Compatibility

- **Tested versions**
The AK255 driver was tested against the description contained in the Danfoss *AK255 XML Interface* document.
- **Platforms**
The Danfoss AK255 Driver runs on all NiagaraAX-3.7 or later platforms that support an Ethernet connection. The driver connects to the AK255 device network through the Ethernet connection using the HTTP protocol and TCP/IP.

CHAPTER 2

Installation and setup

From your PC, use the NiagaraAX Workbench 3.7 or later installed with the “installation tool” option (check box **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 various revision-named subfolders under the **sw** folder.

When installing Workbench on your PC, you should also select the **ak255** module.

Apart from installing the 3.7 version of the NiagaraAX distribution files in the JACE, make sure to install the **ak255** module also (if not already present, or upgrade if an older revision). For more details, see “About the Commissioning Wizard” in the *JACE NiagaraAX Install and Startup Guide*.

Following this, the JACE is now ready for AK255 software integration, as described in the rest of this document.

Using the driver

The following procedures describe how to setup the driver.

Create a new station that includes the driver

- Step 1 In Workbench, click **Tools > New Station**.
- Step 2 In the Nav tree, expand the new station and double-click the **Drivers** node.
This opens the **Driver Manager** view.
- Step 3 Open the **ak255** palette.
- Step 4 Drag the **AK255Network** from the **ak255** palette and drop it in the **Driver Manager** view pane.
- Step 5 In the Nav tree, right-click **config.bog** and click **Save**.
- Step 6 Download the new station into the JACE.
- Step 7 After the station starts, continue with “Add a device and set its URL” on page 1.

Add the driver to an existing station

- Step 1 Open the station in Workbench.
- Step 2 As a precaution, back up and save the station.
- Step 3 In the Nav tree, expand the station and double-click the **Drivers** node.
This displays the **Driver Manager** view.
- Step 4 Open the **ak255** palette.
- Step 5 Drag the **AK255Network** component from the palette and drop it on the **Driver Manager** view.
- Step 6 Save the station.

Add a device and set its URL

- Step 1 Open the running station in Workbench.
- Step 2 Open the **AK255 Device Manager** view of the **AK255Network** component.
- Step 3 Click the **New** button.
The **New** dialog appears.
- Step 4 Click the **OK** button.
Another **New** dialog appears.

- Step 5 Enter the **Url Address** for the device in the form of an IP address. You can ignore the other fields.
- Step 6 Click the **OK** button.
- Step 7 If needed, change the communication parameters to match the gateway device.
- Step 8 Configure the com port.
If there is a configuration problem, the AK255 device will be in fault. If the driver is unable to communicate to the device, the device will be down.
For more information about how to use the **Device Manager**, see “About the Device Manager” in the *Drivers Guide*.

CHAPTER 3

Components and Plugins (Views)

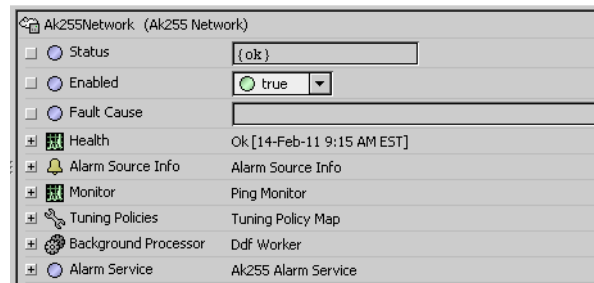
This section provides information about driver components and plugins (views).

Properties common to most NiagaraAX driver components are not described in this document. For related details, refer to the “Driver architecture” chapter in the *Drivers Guide*.

AK255Network

This component is a network-level component in the NiagaraAX architecture. It provides all the configuration properties necessary to allow the driver to communicate with a network of AK255 devices.

Figure 3-1 AK255Network properties



- **BackgroundProcessor** is a worker thread used for write functions. no configuration is required.
- **AlarmService** provides an alarm server socket to receive alarms from AK255 devices. Each device can be configured to notify this JACE when AK255 alarms occur.

AK255 Device Manager view

This table view is typical of many NiagaraAX device manager views. Each row of the table shows the status of an AK255 device connected to the network. For general information about this view, see “About the Device Manager” in the *Drivers Guide*.

Note: *This driver does not support device discovery.*

AK255Device

This component is a dynamic slot of the AK255Network and is used to model the existence of individual AK255 devices.

Figure 3-2 AK255 Device Properties Sheet

The screenshot shows the 'AK255Device (Ak255 Device)' configuration window. It contains several input fields and expandable sections. The fields include: Status (set to '{ok}'), Enabled (checked, set to 'true'), Fault Cause (empty), Health (Ok [14-Feb-11 12:02 PM EST]), Alarm Source Info (Alarm Source Info), Communicator (Ak255 Communicator), Device Id (Url Address 173.10.169.89), Ping Parameters (empty), Device Time (set to 'null'), Time Zone Offset (set to '-300 min'), Is Daylight Savings (unchecked, set to 'false'), Write Authorization Code (masked with dots), and Write Account Code (masked with dots). Below these are several expandable sections, each with a plus icon and a label: Store Hours (Ak255 Store Hours), Store Holidays (Ak255 Store Holidays), Schedules (Ak255 Schedules), Hvac Units (Ak255 Hvac Units), Meters (Ak255 Meters), Devices (Ak255 Devices), Alarms (Ak255 Alarm Device Ext), and Points (Ak255 Point Device Ext).

The following properties apply to AK255 devices:

- **Communicator** configures and displays communications status and functions. See “Communicator” on page 2 for detail.
- **Device Id**
- **Device Time** is the Url address of the AK255 device.
- **Time Zone Offset** is the number of minutes reported by the AK255 device to offset the time when taking into account the time zone. This value is not automatically updated. It can be forced to update by invoking the **readDateTime** action on the device.
- **Is Daylight Savings** is the daylight saving time indicator reported by the AK255 device. This value is not automatically updated. It can be forced to update by invoking the **readDateTime** action on the device.
- **Write Authorization Code** is used when messages are sent to the AK255 device that require an Authorization and Account Code. These messages modify values in the AK255 device.
- **Write Account Code** is used when messages are sent to the AK255 device that require an Authorization and Account Code. These messages modify values in the AK255 device.
- **Store Hours** models the store hours in the AK255. For details, see “Store Hours” on page 3.
- **Store holidays** models the store holidays in the AK255 device. For details, see “Store Holidays” on page 4
- **Hvac Units** models the HVAC units that may be in the AK255 device. For details, see “Hvac Units” on page 4
- **Meters** models the meters that may be in the AK255 device. For details, see “AK255 Meters” on page 7
- **Devices** models the refrigeration devices that may be in the AK255 device. For details, see “AK255 Devices” on page 8
- **Points** is a container object that holds proxy points corresponding to any AK255 command classes used for real-time monitoring and control that may be modeled as control points.

The following sections document the properties that expand when you click the plus (+) next to the property name.

Communicator

The **AK255Communicator** is a frozen slot of the **AK255Device** component and is used to configure and display communications status and functions.

- **Transmitter** provides information about HTTP messages transmitted.
 - **Transmission Attempts** is the number of HTTP message attempts.
 - **Transmission Counts** is the number of HTTP message attempts that completed successfully.
 - **Retransmission Counts** is the number of HTTP messages that were retried.
 - **Max Retry Count** specifies the maximum number of times an HTTP message will be retried.

- **Receiver** provides information about HTTP messages received.
 - **Response Timeout** is the amount of time the driver waits for a response to an HTTP message before timing out or retrying.
 - **num Frames Received** is the number of HTTP messages received.
- **TransactionManager** provides no end-user information.
- **Poll Scheduler** defines the polling rate: **Fast**, **Normal**, and **Slow**, as well as providing statistics on polling performance.
- **Unsolicited Mgr** provides no end-user information.
- **Credentials** are not used by the driver. Leave these fields blank.

Store Hours

AK255 Store Hours is a frozen slot of the **AK255Device** component that models the store hours in the AK255 device.

Figure 3-3 Store Hours properties

These properties define when daylight savings time goes into effect as well as opening and closing times for each day of the week. How to configure each property should be self-explanatory. The following actions are associated with these properties:

- **StoreHours.readStoreHours** is an action that, when invoked, sends a message to the AK255 device to read the daylight saving and daily store hours information.
- **Daylight Saving.modify** is an action used to modify the daylight saving setup in the AK255. When invoked, the user is prompted with the current values, which can be changed.

Figure 3-4 Modify dialog

Clicking **OK** sends the new values to the device.

- **Modify day hours** changes the opening and closing times in the property sheet. When the values are saved, the AK255 device is updated.

Store Holidays

AK255StoreHolidays is a frozen slot of the **AK255Device** component that models the store holidays in the AK255 device.

Figure 3-5 Store Holidays properties

The screenshot shows a dialog box titled "Store Holidays (Ak255 Store Holidays)". It contains a list of holidays. The first holiday, "holiday1", is selected and its properties are displayed in a table below the list:

Property	Value
Descriptor	NewYearsDay
Id	1
Start Date	01-Jan-2012
Duration Days	1
Open Time	09:00 AM
Close Time	06:00 PM

Below the table, there are five more holidays listed: holiday2, holiday3, holiday4, holiday5, and holiday6, each with a plus icon to its left.

The following actions are associated with these properties:

- **StoreHolidays.readHolidays**, when invoked, forces the driver to send a message to read the current holiday configuration from the **AK255Device** component. If they do not exist, this adds **AK255StoreHoliday** components. If they do exist, this action updates them.
- **StoreHolidays.addHoliday**, when invoked, forces the driver to send a message to add a new holiday to the **AK255Device** and prompts the user for the holiday setup information.

Figure 3-6 Add Holiday dialog

The screenshot shows a dialog box titled "Add Holiday". It contains a list of properties for a new holiday. The properties are:

- Descriptor: (empty text box)
- Id: 0
- Start Date: 07-Mar-2011
- Duration Days: 1
- Open Time: 08:00 AM
- Close Time: 06:00 PM

At the bottom of the dialog are "OK" and "Cancel" buttons.

- **StoreHoliday.modify**, when invoked modifies an existing holiday defined in the **AK255Device** and prompts the user for the setup of the holiday to be modified.

Figure 3-7 Modify dialog

The screenshot shows a dialog box titled "Modify". It contains a list of properties for an existing holiday. The properties are:

- Descriptor: NewYearsDay
- Id: 2
- Start Date: 01-Jan-2012
- Duration Days: 1
- Open Time: 09:00 AM
- Close Time: 06:00 PM

At the bottom of the dialog are "OK" and "Cancel" buttons.

Schedules

AK255Schedules is a frozen slot of the **AK255Device** component and is used to configure and display Schedules in the AK255 device.

Hvac Units

AK255HvacUnits is a frozen slot of the **AK255Device** component and is used to configure and display information about HVAC applications in the AK255 device. One action is associated with this slot:

- **Read**, when invoked forces the driver to send a message to read the current HVAC units configuration from the AK255Device. If they do not exist, this adds **AK255HvacUnit** components. If they exist, this action updates them.

The sections document the dynamic slots that can be added to **AK255HvacUnits**.

- For **HvacUnit**, see “AK255HvacUnit” on page 5.
- For **HvacStage**, see “AK255HvacStage” on page 5.
- For **HVACDehumidStages**, see “AK255HvacDehumidStages” on page 6.
- For **HvacDehumidStage**, see “HvacDehumidStage” on page 7.

AK255HvacUnit

is a dynamic slot that is added to the **AK255HvacUnits** component and is used to configure and display information about the HVAC application in the AK255 device.

Figure 3-8 Hvac Unit properties

hvacUnit1 (Ak255 Hvac Unit)	
Status	{ok}
Fault Cause	
Poll Interval	+00000h 00m 30s
Last Update	08-Mar-2011 06:46:03 AM EST
Unit Index	1
Unit Name	MainSaleAC1
Current Status	Maint. Capacity-Heat
Current Value	20.00 °F (fault)
Cooling Stages	Status Component
Heating Stages	Status Component
Heat Reclaims	Status Component
humidity	Ak255 Dehumid Stages

The following action is associated with this slot:

- **Poll**, when invoked forces the driver to send a message to read the current HVAC unit configuration from the AK255Device.

The following explains the properties that are unique to the **AK255HvacUnit** component:

- **Poll Interval** is the period of time between automatic poll actions when the component is in a subscribed state.
- **Last Update** is a time stamp that indicates when this component was last polled and updated.
- **Unit Index** is the unit index of this HVAC unit.
- **Unit Name** is the name of the HVAC unit.
- **Current Status** is the current condition returned for the HVAC unit. This differs from the component Status, which is usually {ok}.
- **Current Value** is the current temperature returned for this HVAC unit.
- **Cooling Stages** is a container for the **AK255HvacStage** components used for cooling.
- **Heating Stages** is a container for the **AK255HvacStage** components used for heating.
- **Heat Reclaims** is a container for the **AK255HvacStage** components used for heat reclaim.
- **Humidity** is a container for the **AK255HvacStage** components used for humidity control.

AK255HvacStage

is a dynamic slot that is added to the **AK255HvacUnit** component's **Cooling Stages**, **Heating Stages**, and **Heat Reclaims** components. It is used to display information about an HVAC stage control in the AK255 device.

Figure 3-9 A255 Hvac Stage properties

coolingStage_1 (Ak255 Hvac Stage)	
Status	{ok}
Write Status	
Stage Name	Cooling 1
Index	1
Target Value	77.50 °F (fault)
Current Status	Off

The following action is associated with this slot:

- **Set Target**, when invoked forces the driver to send a message to modify the target value for the

stage.

The following explains the properties that are unique to the **AK255HvacStage** component:

- **Write Status** is a string that indicates the results of the last **Set Target** action.
- **Stage Name** is the name returned for this stage.
- **Index** is the index for this stage.
- **Target Value** is the intended control value for this stage.
- **Current Status** is the current condition of the device returned at this stage. This differs from the component Status, which is usually **{ok}**.

AK255HvacDehumidStages

is a dynamic slot that is added to the **AK255HvacUnits** component if the HVAC unit supports humidity control. It is used to configure and display information returned from the HVAC humidity control in the AK255 device. The humidity control may be a single-staged or multistage control.

- For single-stage control properties see,
- For multi-stage control properties see,

SingleStageDehumid properties

Figure 3-10 Single-stage humidity control properties

The following actions are associated with this slot (single-stage):

- **Modify Target**, when invoked, forces the driver to send a message to modify the intended value.
- **Modify Target Range**, when invoked, forces the driver to send a message to modify the intended range.

The following explains the properties that are unique to the single-stage control for the **AK255HvacDehumidStages** component:

- **Write Status** is a string that indicates the results of the last **Set Target** action.
- **Dehumidify Type** is a value returned by the AK255 device.
- **Control** is a value returned by the device.
- **Target Value** is the intended value returned by the AK255 device
- **Target Range** is the intended range returned by the AK255 device.

MultiStageDehumid properties

Figure 3-11 Multistage humidity control properties

The following action is associated with this slot (multistage):

- **Modify Target Range**, when invoked, forces the driver to send a message to modify the intended range used by all stages.

The following explains the properties that are unique to the multistage control for the **AK255HvacDehumidStages** component:

- **Write Status** is a string that indicates the results of the last **Set Target** action.
- **Dehumidify Type** is a value returned by the AK255 device.
- **Control** is a value returned by the device.
- **Target Range** is the intended range returned by the AK255 device.

- **Stage n** is a dynamic **AK255DehumidStage** property that is added for each humidity control stage returned by the AK255 device.

HvacDehumidStage

AK255HvacDehumidStage is a dynamic slot that adds to the **AK255DehumidStages** component of an HVAC unit that supports humidity control. It is used to configure and display information about the HVAC humidity stage control in the AK255 device.

Figure 3-12 Dehumid Stage properties

The following action is associated with this slot:

- **Modify Target Range**, when invoked, forces the driver to send a message to modify the intended range used by this stage.

The following explains the properties that are unique to the **AK255HvacDehumidStage** component:

- **Write Status** is a string that indicates the results of the last **Set Target** action.
- **Index** is the index for this stage.
- **Target Value** is the intended control value returned by the device.
- **Pre-Delay** is a value returned by the device.
- **Post-Delay** is a value returned by the device.

AK255 Meters

AK255Meters is a frozen slot of the **AK255Device** component and is used to configure and display information about HVAC applications in the AK255 device.

The following action is associated with these properties:

- **Poll**, when invoked, forces the driver to send a message to read the current meter configuration from the AK255 device. This adds AK255Meter components if they do not exist, or updates them if they do exist.

For meter details, see “AK255 Meter” on page 7.

AK255 Meter

AK255Meter is a dynamic slot that is added to the **AK255Meters** component and is used to configure and display information about a meter application in the AK255 device.

Figure 3-13 Meter properties

- **Consumption Facets** is consumption unit information returned by the device for this meter.
- **Rate Facets** is rate unit information returned by the device for this meter.

- **Meter Name** is the name returned by the device.
- **Node Type** is the address type returned by the device.
- **Node** is the address returned by the device.
- **Consumption** is the current consumption value returned by the device.
- **Current Rate** is a value returned by the device.
- **Hourly Accumulation** is the current hourly accumulation returned by the device.
- **Daily Accumulation** is the current daily accumulation returned by the device.
- **Peak** represents the largest amount recorded by the meter.
- **Peak Time** returns the time the peak occurred.
- **Last Peak Reset** indicates the time the peak was reset.
- **Last Consumption Reset** indicates the most recent time that consumption was reset.

AK255 Devices

AK255Devices is a frozen slot of the AK255Device component and is used to configure and display information about the refrigeration devices in the AK255 device.

The following action is associated with the **AK255Devices** component:

- **Read**, when invoked, forces the driver to send a message to read the current devices configured in the **AK255Device** component. This adds **AK255SubDevice** and **AK255RackOnly** components to the device model if they do not exist. If they do exist, it updates them.

The following topics describe the dynamic device slots:

- **SubDevice**, see “AK255 SubDevice” on page 8.
- For **RackOnly**, see “RackOnly” on page 9.
- For **SuctionGroup**, see “SuctionGroup” on page 10.
- For **SuctionControl**, see “SuctionControl” on page 11.
- For **AlarmLimit**, see “AlarmLimit” on page 12.
- For **SuctionCircuit**, see “SuctionCircuit” on page 12.
- For **SuctionDefrost**, see “CircuitDefrost” on page 13.
- For **SensorValue**, see “SensorValue” on page 14.
- For **Condenser**, see “Condenser” on page 14.
- For **CondenserStages**, see “CondenserStages” on page 15.

AK255 SubDevice

is a dynamic slot that is added to the **AK255Devices** component and is used to configure and display information about a refrigeration device application in the AK255 device.

Note: *Some of the frozen properties of the device components are automatically hidden if the device does not return a value for the slot.*

Figure 3-14 Sub Device properties

<input type="radio"/> Status	{ok}
<input type="radio"/> Fault Cause	
<input type="radio"/> Poll Interval	+00000h 00m 30s
<input type="radio"/> Last Update	08-Mar-2011 08:24:09 AM EST
<input type="radio"/> Device Name	Suction Pres A
<input type="radio"/> Device Id	
<input type="radio"/> Device Type	Monitoring
<input type="radio"/> Node Type	2
<input type="radio"/> Node Type Enum	Sensor Input
<input type="radio"/> Device Addr	20-1.1
<input type="radio"/> Device Node	20
<input type="radio"/> Device Module	1
<input type="radio"/> Device Point	17
<input type="radio"/> File Name	
<input type="radio"/> Model	Conv 1
<input type="radio"/> Boolean Value	false {ok}
<input type="radio"/> Numeric Value	0.00 {ok}
<input type="radio"/> Device Status	OK
<input type="radio"/> Has Alarm	<input checked="" type="radio"/> true
<input type="radio"/> Has Main Switch	<input type="radio"/> false
<input type="radio"/> Has Lights	<input type="radio"/> false
<input type="radio"/> Has Night	<input type="radio"/> false
<input type="radio"/> Has Defrost	<input type="radio"/> false
<input type="radio"/> Has Cleaning	<input type="radio"/> false
<input type="radio"/> Has Shut Down	<input type="radio"/> false

The following action is associated with these properties:

- **Poll**, when invoked, forces the driver to send a message to read the current values for this device. The following defines the unique sub device properties:

- **Poll Interval** is the period of time between automatic polling actions when this component is in a subscribed state.
- **Last Update** is the time stamp at which this component was last polled and updated.
- **Device Name** is the name of the device as returned by the device.
- **Device Id** is the identification number returned by the device.
- **Device Type** is the type of device as returned by the device.
- **Node Type** is the type of node as returned by the device.
- **Node Type Enum** is the enumeration value returned by the device.
- **Device Addr** is the address returned by the device.
- **Device Node** is the address of the node returned by the device.
- **Device Module** is the address of the module returned by the device.
- **Device Point** is the address of the point returned by the device.
- **File Name** is the file name returned by the device.
- **Model** is the meter model name returned by the device.
- **Boolean Value** is the boolean value returned by the device.
- **Numeric Value** is the numeric value returned by the device.
- **Device Status** is the current status of the device as returned by the device.
- **Has Alarm** is a boolean value returned by the device.
- **Has Main Switch** is a boolean value returned by the device.
- **Has Lights** is a boolean value returned by the device.
- **Has Night** is a boolean value returned by the device.
- **Has Defrost** is a boolean value returned by the device.
- **Has Cleaning** is a boolean value returned by the device.
- **Has Shutdown** is a boolean value returned by the device.

RackOnly

is a dynamic slot that is added to the **AK255Devices** component and is used to configure and display information about a refrigeration device application in the AK255 device.

Note: Some of the frozen properties of the device components are automatically hidden if the device does not return a value for the slot.

Figure 3-15 Rack Only properties

Rack A (Ak255 Rack Only)	
Status	{ok}
Fault Cause	
Poll Interval	+00000h 00m 30s
Last Update	08-Mar-2011 12:25:45 PM EST
Rack Id	1
Num Suction Groups	2
Node Type	255
Node Type Enum	Empty Node
Device Node	0
Device Module	0
Device Point	0
Has Condenser	<input checked="" type="checkbox"/> true
suction1	Ak255 Suction Group
suction2	Ak255 Suction Group
condenser	Ak255 Condenser

The following properties are unique to the **AK255RackOnly** component.

- **Poll Interval** is the period of time between automatic polling actions when this component is in a subscribed state.
- **Last Update** is the time stamp at which this component was last polled and updated.
- **RackId** is the identification number for the rack returned by the device.
- **Num Suction Groups** is the number of suction groups returned by the device. A non-zero value adds **AK255SuctionGroup** components.
- **Node Type** is the type of node as returned by the device.
- **Node Type Enum** is the enumeration value returned by the device.
- **Device Node** is the address of the node returned by the device.
- **Device Module** is the address of the module returned by the device.
- **Device Point** is the address of the point returned by the device.
- **Has Condenser** is a boolean value returned by the device. A value of true adds an **AK255Condenser** component.

SuctionGroup

AK255SuctionGroup is a dynamic slot that is added to the **AK255RackOnly** component and is used to configure and display information about a suction group application in the AK255 device.

Note: Some of the frozen properties of the device components are automatically hidden if the device does not return a value for the slot.

Figure 3-16 Suction Group properties

suction1 (Ak255 Suction Group)	
Status	{ok}
Fault Cause	
Poll Interval	+00000h 00m 30s
Last Update	08-Mar-2011 12:38:12 PM EST
Write Status	
Rack Id	1
Suction Id	1
Group Name	-25 SUCTN A
Num Circuits	8
Suction Pressure	0.00 psi {ok}
Suction Target	17.10 psi {ok}
Suction Cutout	2.00 psi {ok}
Suction Control	Ak255 Suction Control
Alarm Hi Limit	Ak255 Alarm Limit
Alarm Lo Limit	Ak255 Alarm Limit
Defrost Auto Schedule	<input checked="" type="radio"/> false
curcuit1	Ak255 Suction Circuit
curcuit2	Ak255 Suction Circuit
curcuit3	Ak255 Suction Circuit
curcuit4	Ak255 Suction Circuit
curcuit5	Ak255 Suction Circuit
curcuit6	Ak255 Suction Circuit
curcuit7	Ak255 Suction Circuit
curcuit8	Ak255 Suction Circuit

The following actions are associated with this component:

- **Poll**, when invoked, forces the driver to send a message to read the current values for this device.
- **Modify Suction Target**, when invoked, prompts for, and then sends, a message to modify the suction target value for this suction group.
- **Modify Suction Count**, when invoked, prompts for, and then sends, a message to modify the suction cut-out value for this suction group.

The following properties are unique to this component:

- **Poll Interval** is the period of time between automatic polling actions when this component is in a subscribed state.
- **Last Update** is the time stamp at which this component was last polled and updated.
- **Write Status** is a string that indicates the results of the last **Set Target** action.
- **Rack Id** is an identification number for this rack.
- **Suction Id** is an identification number for this suction group.
- **Group Name** is the name returned for this suction group.
- **Num Circuits** is the number of circuits for this suction group. an **AK255SuctionCircuit** component is added for each circuit.
- **Suction Pressure** is the current suction pressure returned for this suction group.
- **Suction Target** is the current suction pressure target returned for this suction group.
- **Suction Cutout** is the current suction pressure cut-out returned for this suction group.
- **Suction Control** is the current suction control that the **AK255SuctionControl** component returned for this suction group.
- **Alarm Hi limit** is the current alarm high limit that the **Ak277AlarmLimit** component returned for this suction group.
- **Alarm Lo Limit** is the current alarm low limit that the **AK255AlarmLimit** component returned for this suction group.
- **Defrost Auto Schedule** is a boolean value returned for this suction group.
- **Circuit1 - n** is an **AK255SuctionCircuit** component for each circuit supported by this suction group.

SuctionControl

AK255SuctionControl is used to display information about the suction control of a suction group in the AK255 device.

Note: Some of the frozen properties of the device components are automatically hidden if the device does not return a value for the slot.

Figure 3-17 Suction Control properties

Suction Control (Ak255 Suction Control)	
<input type="radio"/> Control Type	Ak2 Sc255
<input type="radio"/> Temp Target Value	0.00 °F
<input type="radio"/> Temp Range Value	2.00 Δ°F
<input type="radio"/> Max Press Value	4.00 psi
<input type="radio"/> Post Defrost Delay Value	20.00 min

The following properties are associated with this component:

- **Control Type** is the control type returned by the device.
- **Temp Target Value** is the target temperature returned by the device.
- **Temp Range Value** is the target range (high and low) for the temperature as returned from the device.
- **Max Press Value** is the maximum pressure value returned by the device.
- **Post DefrostTemp Delay Value** is a value returned by the device.

AlarmLimit

AK255AlarmLimit is used to configure and display information about the alarm limit in the AK255 device.

Note: Some of the frozen properties of the device components are automatically hidden if the device does not return a value for the slot.

Figure 3-18 Alarm Limit properties

Alarm Hi Limit (Ak255 Alarm Limit)	
<input type="radio"/> Value	55.00 psi
<input type="radio"/> Duration	5 min

The following actions are associated with this component:

- **Modify Limit**, when invoked, prompts for, and then sends, a message to modify the value for this alarm limit.
- **Modify Duration**, when invoked, prompts for, and then sends, a message to modify the alarm limit duration value for this alarm limit.

The following properties are associated with this component:

- **Value** is the current alarm limit value returned by the device
- **Duration** is the current alarm limit duration returned by the device.

SuctionCircuit

AK255SuctionCircuit is used to display information about the suction circuit in the AK255 device.

Note: Some of the frozen properties of the device components are automatically hidden if the device does not return a value for the slot.

Figure 3-19 Suction Circuit properties

circuit1 (Ak255 Suction Circuit)	
<input type="radio"/> Status	{ok}
<input type="radio"/> Fault Cause	
<input type="radio"/> Poll Interval	+00000h 00m 30s
<input type="radio"/> Last Update	08-Mar-2011 01:23:37 PM EST
<input type="radio"/> Write Status	
<input type="radio"/> Rack Id	2
<input type="radio"/> Suction Id	1
<input type="radio"/> Circuit Id	1
<input type="radio"/> Circuit Name	MeatPrep 7 B 1
<input type="radio"/> Circuit Status	Thermostat Cutin
<input type="radio"/> Temperature Target	55.00 °F {ok}
<input type="radio"/> Temperature Range	1.00 Δ°F {ok}
<input checked="" type="radio"/> Defrost	Ak255 Circuit Defrost
<input type="radio"/> Sensor Count	1
<input checked="" type="radio"/> Meat Prep B 7	Ak255 Sensor Value

The following actions are associated with this component:

- **Poll**, when invoked, forces the driver to send a message to read the current values for this device.
- **Modify Temperature Target**, when invoked, prompts for, and then sends, a message to modify the target temperature value for this suction circuit.
- **Modify Temperature Range**, when invoked, prompts for, and then sends, a message to modify the temperature range for this suction circuit.
- **Manual Defrost**, when invoked, prompts for, and then sends, a message to initiate a manual defrost cycle.

The following properties are unique to this component:

- **Poll Interval** is the period of time between automatic polling actions when this component is in a subscribed state.
- **Last Update** is the timestamp at which this component was last polled and updated.
- **Write Status** is a string property that indicates the results of the last Set Target action.
- **Rack Id** is the identification number for the suction circuit rack.
- **Suction Id** is an identification number for the suction circuit.
- **Circuit Id** is an identification number for the suction circuit.
- **Circuit Name** is the name returned for this suction circuit.
- **Circuit Status** is the current status returned for this suction circuit.
- **Temperature Target** is the current target temperature for this suction circuit.
- **Temperature Range** is the current target temperature range for this suction circuit.
- **Defrost** is the current **AK255CircuitDefrost** component for this suction circuit.
- **Sensor Count** is the number of sensors used for this suction circuit.
- **Sensor 1 - n** is the **AK255SensorValue** component to be added for each sensor supported.

CircuitDefrost

AK255CircuitDefrost is used to display information about the circuit defrost in the AK255 device.

Note: Some of the frozen properties of the device components are automatically hidden if the device does not return a value for the slot.

Figure 3-20 Circuit Defrost properties

The following actions are associated with this component:

- **Modify Duration**, when invoked, prompts for, and then sends, a message to modify the defrost duration value for this suction circuit.
- **Modify Duration**, when invoked, prompts for, and then sends, a message to modify the defrost duration value for this suction circuit. The parent **Suction Group's Defrost Auto Schedule** property must be set to **false** for this action to be visible.

The following properties are associated with this component:

- **Defrost Type** is a designator returned by the device that identifies the defrost cycle.
- **Termination Type** is an indicator returned by the device.
- **Termination Value** is a value returned by the device.
- **Drip Delay** is a value returned by the device.
- **Min Defrost** is the minimum amount of time for the defrost as returned by the device.
- **Defrost Duration** is the amount of time to defrost the refrigerator as returned by the device.
- **Num Defrost Schedules** is the number of defrost schedules configured.
- **Schedule1 - n** are the schedules of defrost times.

SensorValue

AK255SensorValue is used to display information about the sensor value of the suction circuit in the AK255 device.

Note: Some of the frozen properties of the device components are automatically hidden if the device does not return a value for the slot.

Figure 3-21 SensorValue properties

The screenshot shows a properties window titled "MDfZ Meat A 1 (Ak255 Sensor Value)". It contains several properties, each with a status icon (a circle with a dot) and a value field:

- Status: {ok}
- Sensor Name: MDfZ Meat A 1
- Sensor Number: 1
- Value: -14.40 °F {ok}
- Alarm Hi Limit: Ak255 Alarm Limit
- Alarm Lo Limit: Ak255 Alarm Limit

The following are the unique properties associated with this component:

- **Sensor Name** is the name returned by the device.
- **Sensor Number** is the number returned by the device.
- **Value** is the current value returned by the device/
- **Alarm Hi Limit** is the current upper limit for the **AK255AlarmLimit** component as returned by the device for this sensor.
- **Alarm Lo Limit** is the current low limit for the **AK255AlarmLimit** component as returned by the device for this sensor.

Condenser

AK255Condenser is used to configure and display information about the condenser in the AK255 device.

Note: Some of the frozen properties of the device components are automatically hidden if the device does not return a value for the slot.

Figure 3-22 Condenser properties

The screenshot shows a properties window titled "condenser (Ak255 Condenser)". It contains many properties, each with a status icon and a value field:

- Status: {ok}
- Fault Cause: (empty field)
- Poll Interval: +00000h 00m 30s
- Last Update: 08-Mar-2011 02:16:38 PM EST
- Write Status: (empty field)
- Rack Id: 2
- Condenser Name: Condenser B
- Value: 163.00 psi {ok}
- Condenser Type: Air Cooled
- Fan Type: Single Fan
- Num Fans: 1
- Control Sensor: Disch Press
- Control Type: Rate Of Change
- Control Method: Cut In Out
- Alarm Hi Limit: Ak255 Alarm Limit
- Alarm Lo Limit: Ak255 Alarm Limit
- CutinStages: Ak255 Condenser Stages
- CutoutStages: Ak255 Condenser Stages

The following action is associated with this slot:

- **Poll**, when invoked forces the driver to send a message to read the current values for this device.
- The following properties are associated with this component:

- **Poll Interval** is the period of time between automatic polling actions when this component is in a subscribed state.
- **Last Update** is the time stamp at which this component was last polled and updated.
- **Write Status** is a string that indicates the results of the last **Set Target** action.
- **Rack Id** is an identification number for this rack.
- **Condenser Name** is the name of the condenser as returned by the device.

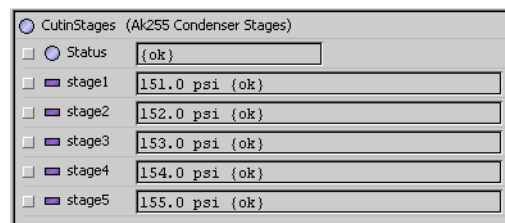
- **Value** is the current value returned by the device.
- **Condenser Type** is the type of condenser returned by the device.
- **Fan Type** is the type of fan returned by the device.
- **Num Fans** is the number of fans returned by the device.
- **Control Sensor** is the control sensor returned by the device.
- **Control Type** is the control method returned by the device.
- **Alarm Hi Limit** is the upper limit of the **AK255AlarmLimit** component as returned for this condenser.
- **Alarm Lo Limit** is the low limit of the **AK255AlarmLimit** component as returned for this condenser.
- **CutinStages** adds an **AK255CondenserStages** component to the model if the **Control Method** property is set to **CutInOut**.
- **CutoutStages** adds an **AK255CondenserStages** component to the model if the **Control Method** property is set to **CutInOut**.

CondenserStages

AK255CondenserStages is used to configure display information about the condenser cut-in and cut-out stage controls in the AK255 device.

Note: Some of the frozen properties of the device components are automatically hidden if the device does not return a value for the slot.

Figure 3-23 Condenser Stages properties



The following action is associated with this component:

- **Modify Stages**, when invoked, prompts for, and then sends, a message to modify the condenser stage control values for this condenser.

The following property is unique for this component:

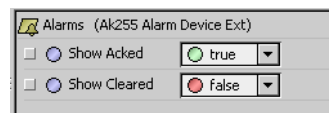
- **Stage1** - n provides multiple stage control values for this condenser.

AK255AlarmDeviceExt

AK255AlarmDeviceExt (Alarms) is a frozen slot of the **AK255Device** component and is used to display and acknowledge alarms stored in the AK255 device.

The default view is the Ak255 Alarm Manager. See “AK255 Alarm Manager View” on page 16.

Figure 3-24 Alarm Device Extension properties



This action is associated with alarms:

- **Read Alarms**, when invoked, forces the driver to send a series of messages to read the alarms from the device. This adds an **AK255Detail** component to the **AK255Alarms** component.

These properties are associated with alarms:

- **Show Acked** is a boolean value that controls the showing of acknowledged alarms in the **Alarm Manager View** of the **AK255Alarms** component.
- **Show Cleared** is a boolean value that controls the showing of cleared alarms in the **Alarm Manager View** of the **AK255Alarms** component.

For alarm detail information, see “AK255 AlarmDetail” on page 16.

AK255 AlarmDetail

The **AK255Detail** component represents an alarm in the AK255 device.

Figure 3-25 Alarm detail properties

The screenshot shows a form titled "alm_1138 (AK255 Alarm Detail)". It contains the following fields:

- Ref**: 1138
- Status**: active
- Occur Time**: 07-Mar-2011 07:17 AM EST
- Ack Time**: 08-Mar-2011 10:18 AM EST
- Normal Time**: null
- Alarm Name**: I/O Comm error
- Trip Value**: Trip
- Current Value**: OffLn
- Store Name**:
- Unit Number**: 0
- Unit Name**: Facility Contrl
- Device Name**: AK2-CM Module
- Device Model**:
- Node Type**: Not Def
- Addr**: 29-2..
- Alarm Class**:
- Action**: Normal
- Clearable**: false
- Setting**:
- Acknowledgement**: Yes (1-1) 03/08/11 11:18

- **Ref** is an integer reference number for this alarm.
- **Status** is the condition of this alarm: **active** or **cleared**.
- **Occur Time** is the time stamp when the alarmed was tripped.
- **Ack Time** is the time stamp when the alarm was acknowledged.
- **Normal Time** is the time that the To Normal event occurred.
- **Alarm Name** is the name of the alarm returned by the device.
- **Trip Value** is information returned from the device for this alarm.
- **Current Value** is the current value returned by the device for this alarm.
- **Store Name** is a name returned by the device for this alarm.
- **Unit Number** is a number returned by the device for this alarm.
- **Unit Name** is a name returned by the device for this alarm.
- **Device Name** is the name of the device as returned by the device for this alarm.
- **Device Model** is the model of the device as returned by the device for this alarm.
- **Node Type** is the type of node for this device as returned by the device for this alarm.
- **Addr** is the address returned by the device for this alarm
- **Alarm Class**. is the alarm group returned for this alarm.
- **Action** is the action returned for this alarm.
- **Clearable** is a flag returned for this alarm.
- **Setting** is a value returned for this alarm.
- **Acknowledgment** is a value returned for this alarm.

AK255 Alarm Manager View

This table view shows the alarms read from AK255 devices.

Figure 3-26 Alarm Manager View

Ref	Status	Occur Time	Ack Time	Normal Time	Alarm Name	Trip Value	Current Value	Store Name	Unit Number	Unit Name
1150	active	07-Mar-11 8:12 AM EST	08-Mar-11 10:18 AM EST	null	Low zone temp	20.0 -F	20.0 -F		0	Facility Contrl
1149	active	07-Mar-11 7:43 AM EST	08-Mar-11 10:18 AM EST	null	Low zone temp	-50.0 -F	-50.0 -F		0	Facility Contrl
1148	active	07-Mar-11 7:17 AM EST	08-Mar-11 10:18 AM EST	null	I/O Comm error	Trip	OffLn		0	Facility Contrl
1147	active	07-Mar-11 7:17 AM EST	08-Mar-11 10:18 AM EST	null	I/O Comm error	Trip	OffLn		0	Facility Contrl
1146	active	07-Mar-11 7:17 AM EST	08-Mar-11 10:18 AM EST	null	I/O Comm error	Trip	OffLn		0	Facility Contrl
1145	active	07-Mar-11 7:17 AM EST	08-Mar-11 10:18 AM EST	null	I/O Comm error	Trip	OffLn		0	Facility Contrl
1144	active	07-Mar-11 7:17 AM EST	08-Mar-11 10:18 AM EST	null	I/O Comm error	Trip	OffLn		0	Facility Contrl
1143	active	07-Mar-11 7:17 AM EST	08-Mar-11 10:18 AM EST	null	I/O Comm error	Trip	OffLn		0	Facility Contrl

The following columns appear on the Alarm Manager View:

- **Ref** is the alarm number and icon. For a description of the icons, see “alarm-AlarmConsole” in the *User Guide*.
- **Status** is the current status read from the AK255 device.
- **Occur Time** is the time stamp when the alarm occurred.
- **Ack Time** is the time stamp when the alarm was acknowledged.
- **Normal Time** is the time that the To Normal event occurred.
- **Alarm Name** is the name of the alarm.
- **Trip Value** is
- **Current Value** is the current value of the device.
- **Store Name** is the name of the store.
- **Unit Number** is the device number.
- **Unit Name** is the name of the refrigerator.

For more details, see “AK255 AlarmDetail” on page 16. For how to acknowledge and clear these alarms, see “Managing alarms” in the *User Guide*.

AK255PointDeviceExt

AK255PointDeviceExt (Points) is a frozen slot of the **AK255Device** component that serves as a container object to hold the AK255 proxy points that correspond to any AK255 command classes that are used for real-time monitoring and control, and that may be modeled as control points. You can control points for inputs, relays, sensors and analog variable outputs in the AK255 device.

The default view is the Ak255 Point Manager view. See “AK255 Point Manager View” on page 17.

AK255 Point Manager View

This view is typical of all driver point manager views. The driver supports point discovery and provides the ability to add **ControlPoints** to expose inputs, relays, sensors and analog variable outputs discovered in the AK255 device.

Figure 3-27 Point Manager View

The screenshot shows the AK255 Discovery window with two main tables: 'Discovered' and 'Database'.

Discovered Table (137 objects):

To Type	To Name	To Addr	To Node Addr	To Module Addr	To Point Addr	To Units Enum	To Node Addr	To Module Addr	To Point Addr
input	Compressor A 1	31-4-1	31	4	1	No Units	31	4	1
input	Compressor A 3	31-4-3	31	4	3	No Units	31	4	3
input	Compressor A 2	31-4-2	31	4	2	No Units	31	4	2

Database Table (7 objects):

Name	Type	Out	To Type	To Name	To Addr	To Node Addr	To Module Addr	To Point Addr	To Units Enum
Misc Relay 1	Boolean Writable	false (ok) @ def	relay	Misc Relay 1	29-1.1	29	1	9	No Units
Misc Relay 2	Boolean Writable	true (ok) @ def	relay	Misc Relay 2	29-2.1	29	2	9	No Units
Misc Relay 3	Boolean Writable	true (ok) @ def	relay	Misc Relay 3	29-3.1	29	3	9	No Units
Misc Relay 4	Boolean Writable	true (ok) @ def	relay	Misc Relay 4	30-1.1	30	1	9	No Units
Misc Relay 5	Boolean Writable	true (ok) @ def	relay	Misc Relay 5	30-2.1	30	2	9	No Units
Compr Room Temp	Numeric Point	-50.00 °F (ok)	sensor	Compr Room Temp	24-1.6	24	1	22	DegF
Phase Loss A	Boolean Point	false (ok)	input	Phase Loss A	31-5.2	31	5	2	No Units

For more general information about this view, see “About the Point Manager” in the *Drivers Guide*.

AK255DeviceFolder

This is an available folder for organizing Ak255Device components. The default view is the **Ak255 Device Manager**. See “AK255 Device Manager view” on page 1.

AK255PointFolder

Use this folder to organize AK255 proxy points. The default view is the Ak255 Point Manager. See “AK255 Point Manager View” on page 17.

Ak255ProxyExt

Ak255proxyExt is the proxy extension used for any type of point made from the Ak255 Point Manager. included are the typical common proxy extension properties, as well as two container slots that are unique to the AK255 driver:

- Read Parameters
- Write Parameters

Both hold properties used in the protocol to fetch data points, and are automatically populated when you add a discovered point from the learn pane.

For general information on common proxy extension properties, refer to “ProxyExt properties” in the *Drivers Guide*.