ALVASYS-DR-VKF // alvasysHailProtectionVKF

Software installation and configuration manual.

Ver: 1.0.1.2

Date: 21.10.2022

Author: MMA

1 History

Filename: ALVASYS-DR-VKF_Manual.odt			
Rev.	Date	Author	Description
1.0.1.2	21.10.2022	Marcello Meriano	First draft

2 Index

3 Confidentiality Notice

The information contained in this document is confidential information of alvasys automation ag (“ALVASYS”). Such information and the software described herein, is furnished under license agreement and may be used only in accordance with that agreement.
The information contained in this document is provided solely for use by ALVASYS employees, licenses and system owners. Contents of this document are not to be released to or reproduced for anyone else.
While every effort has been made to assure the accuracy of this document, ALVASYS 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 current as of the date to this publication and are subject to change without notice.
This document may be copied by parties who are authorised to distribute ALVASYS 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 ALVASYS.

4 . Introduction

Requirements

- Niagara >= 4.7
- A license to use the ALVASYS-DR-VKF driver. Other device limit or proxy-point limits may apply to your license. For license details and options, see the ALVASYS-DR-VKF price list.

Module

The ALVASYS-DR-VKF Driver is contained in one files:

`alvasysHailProtectionVKF-rt.jar`

Compatibility

Platforms

The ALVASYS-DR-VKF driver runs on Niagara >= 4.7 platforms.

Tested versions

Niagara 4.10.x

5 . License

The **has no limitation!**

6 . Installing the software

Installing the ALVASYS-DR-VKF driver is simple.

It requires a basic knowledge of the Tridium Niagara 4 and execute a few steps as described hereafter.
The driver, a Java "**.jar**" executable file, is usually shipped in a zip file.
Its name is generated according to the following structure:

alvasysHailProtectionVKF-rt.jar (for version 4.7+4.8)
alvasysHailProtectionVKF-rt.jar (for version >=4.9)

Installing the driver on your PC
The following procedures describe how to set-up the driver.

Step 1	First of all unzip the files which contains the driver and technical notes.
Step 2	Rename the files, changing theirs name into **alvasysHailProtectionVKF-rt.jar**!
Step 3	Copy the one **jar** files into the **modules** directory of your Niagara **Work Bench**.
Step 4	Restart your **Work Bench**.
Step 5	After restarting, the file should appears in the list of available software, which can be shown clicking on the **Software Manager** section of the **Platform** of your **Work Bench**.

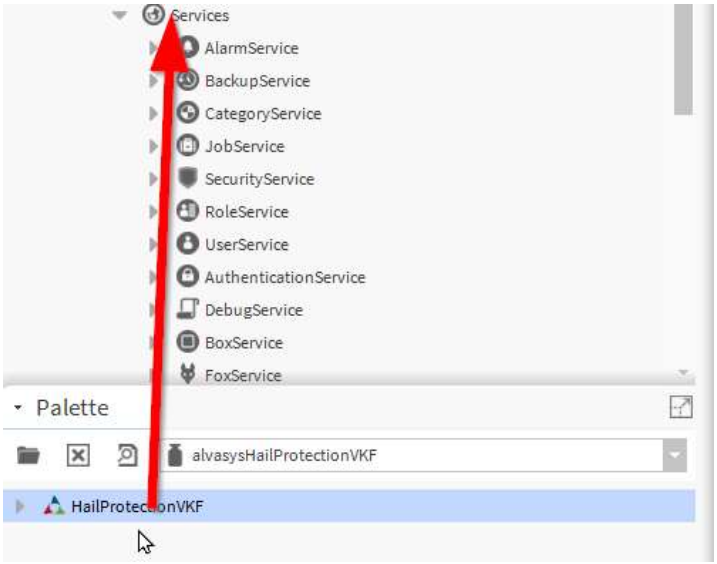
Installing the driver into the JACE/HAWK unit

Step 1	Through the **`Work Bench`** get connected to a JACE/HAWK/MAC/Supervisor running unit.
Step 2	Transfer the **`alvasysHailProtectionVKF-rt.jar`** module into the unit under the folder **`modules`**.
Step 3	This can be done by activating the standard Tridium procedure for software upgrading or simply copying the **`jar`** files by the **`File Transfer Client`** procedure, available under the list of the **`Platform`** options in your **`Work Bench`**. Destination directory inside the Jace8000 is: **`/opt/niagara/modules`** For further details on how to transfer files from **`Work Bench`** to JACE/HAWK units, refer to the official Tridium documentation.
Step 4	After copying the driver into the JACE/HAWK unit, force a reboot.

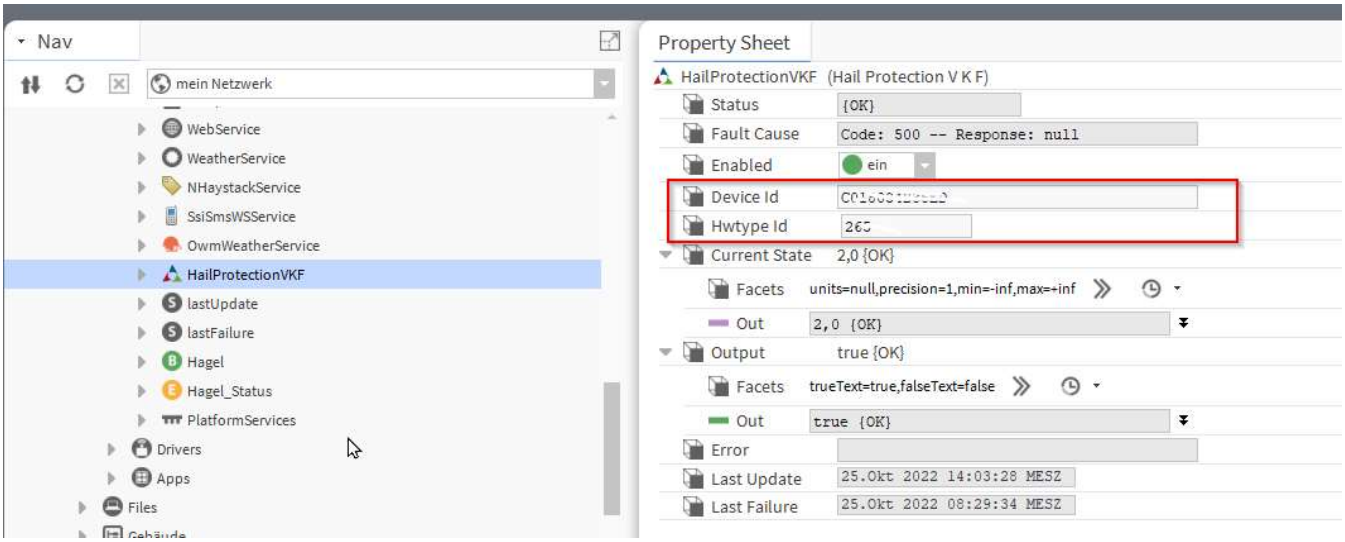
7 . Driver configuration

Installing the alvasysHailProtectionVKF-rt.jar

The first step of the driver configuration is the installation of the **`HailProtectionVKF`** under the station running in the JACE/HAWK/Supervisor unit. Copy the HailProtectionVKF in the Service!



Configuring the Network Parameters



- **Status:** status of Service.
  - It will be “fatal fault” if the Driver is not licensed.
  - It will be “down” if last API call return with errors (Timeout, Code: 404, 400, …).
- **Fault Cause:**
  - It will show “[Driver not licensed]” if the Driver is not licensed.
  - It will show error codes, in case of API error.
- **DeviceId:** dynamic field, to set Device ID in the API call. You received this from *
- **HwtypeId:** dynamic field, to set Hardware Type ID in the API call. You received this from *

*https://www.hagelschutz-einfach-automatisch.ch/eigentuemer-verwaltungen/das-system-erklaert/schnittstelle.html

Treiber Model: Niagara 4

- **Current State:** the value (**currentState**) returned from netitservices API is mapped to Output
  - currentState = 0 → no Hail
  - currentState = 1 → Hail
  - currentState = 2 → Test Hail
-
- **Output:** the value (**currentState**) returned from netitservices API is mapped to Output BooleanPoint in this way (only if API returns with response code 200):
  - currentState = 0 → FALSE
  - currentState = 1 → TRUE
  - currentState = 2 → TRUE
- **Error:** This property is populated in case of an error (Timeout, Code: 404, 400, …), using the **message** field in the API response body.
- **Last Update:** Date and time of the last call made to the API.

- **Last Failure:** Date and time of the last call made to the API with errors.

Response OK (Code: 200):


```
{
  "currentState": 0,
  "newProgVer": 0,
  "hailState": 0
}
```


Response KO (Code: 404, 400):


```
{
  "timestamp": "2022-10-23T08:58:15.800+00:00",
  "exception": "HardwareTypeException",
  "message": "The wrong hardware type for the device was specified"
}
```


Behaviour:

On API ok response:

- Last Update is updated to “now”.
- Output value is updated to currentState value in API response body and its status is set to OK.
- Last Failure, Error and Fault Cause are not cleared, continuing to refer to the last failed API call.

On API error:

- Last Update is updated to “now”.
- Last Failure is updated to “now”.
- Output status is set to DOWN. Its value is not modified.
- Error value is updated to message value in API response body.
- Fault Cause is set to API response code.

Update Interval are fix:

- 120s

Action:

- Update the Hail manual (but is only needed when you do the initialization)