M211850EN-P

RESTRICTED

Administrator Guide

IRIS Focus Version 7.4

VAISALA

PUBLISHED BY

Vaisala Oyj
Vanha Nurmijärventie 21, FI-01670 Vantaa, Finland
P.O. Box 26, FI-00421 Helsinki, Finland
+358 9 8949 1
www.vaisala.com
docs.vaisala.com

Table of contents

1. About this document

1.1 Version information

This document provides information for installing, operating, and maintaining IRIS Focus software.

Table 1 Document versions (English)

Document code	Date	Description
M211850EN-P	April 2024	For IRIS Focus 7.4.
M211850EN-N	August 2023	For IRIS Focus 7.3.
M211850EN-M	January 2023	For IRIS Focus 7.2.

1.2 Related documents

Table 2 Related documents

Document code	Name
M211850EN	*IRIS Focus Administrator Guide*
M211849EN	*IRIS Focus User Guide*
M211904EN	*IRIS Focus Release Notes*
M212924EN	*IRIS and RDA Software Installation Guide (M212924EN)*

1.3 Trademarks

1.4 Documentation conventions

WARNING! Warning alerts you to a serious hazard. If you do not read and follow instructions carefully at this point, there is a risk of injury or even death.

CAUTION! Caution warns you of a potential hazard. If you do not read and follow instructions carefully at this point, the product could be damaged or important data could be lost.

Highlights important information on using the product.

Gives information for using the product more efficiently.

Lists tools needed to perform the task.

Indicates that you need to take some notes during the task.

2. IRIS Focus overview

IRIS Focus provides user-friendly, browser-based tools for viewing and analyzing weather data received from weather radars, WindCube Scan Lidars, and lightning sensors. Weather data is overlaid on a geographical map.

IRIS Focus forms the visualization and on-demand product generation front-end, while other software components handle device control, some product generation, and data distribution.

Weather radar data in the pictures of this chapter: courtesy of Meteorological Service of New Zealand Ltd. Lightning data: courtesy of Transpower New Zealand Ltd.

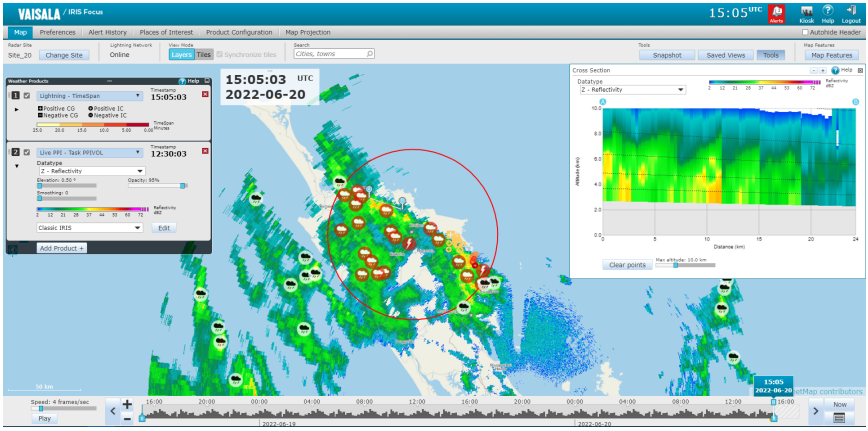


Figure 1 IRIS Focus main view showing the visualization of radar data

With the zoomable and draggable animation timeline, you can easily visualize recent, past, or nowcasted data.

Significant weather events such as thunderstorms, wind shear, or heavy rain are automatically detected and trigger alerts when they enter an area of interest.

The currently displayed weather product is automatically updated to the latest one available.

Nowcasting performs advection calculations on motion data from weather products to predict weather movement and severity up to 2 hours in the future.

Radar data

Radar data is gathered from a single weather radar or a network of radar sites through a composite. When viewing weather radar data, the map is centered on a selected radar site or composite site.

Lidar data

Windcube Scan Lidar data can be ingested to IRIS Focus in NetCDF format. IRIS Focus supports viewing lidar data from PPI and Fixed scans. Available on-demand products are currently PPI, RTI, and Turbulence. In addition, pre-generated products SHEAR, WARN, WIND are also available.

Lightning data

Lightning data is visualized through products such as **TimeSpan**, which provides information about recent lightning events on a customizable map.

With the zoomable animation timeline, you can easily visualize and animate recent data.

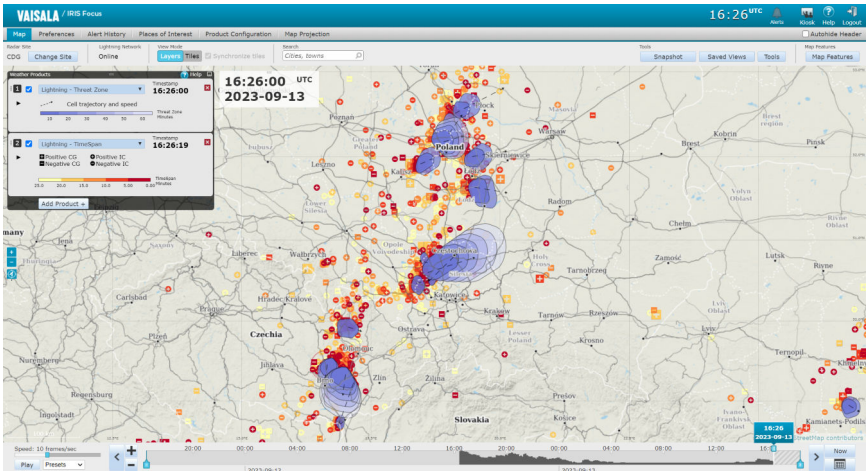


Figure 2 Lightning data displayed in a layered view

Weather products

The displayed data typically consists of radar, lidar, or lightning products.

Radar products are raw signal data from a radar receiver processed to provide information about current weather conditions. They provide information such as radar signal reflectivity or rain intensity for analysis by meteorologists.

Wind lidar products are either raw data measured by the sensor itself, like Doppler velocity, CNR (carrier-to-noise ratio), SNR (lidar reflectivity), pregenerated products from IRIS Analysis (WIND, SHEAR), or processed on-demand products in IRIS Focus (PPI, RTI, Turbulence). Lidar data enables precise measurements of wind fields, aerosol, and cloud layers in the atmosphere to give detailed observations about the lowest part of the atmosphere, that is, the boundary layer.

Lightning products visualize data from a lightning sensor network, produced by the Total Lightning Processor (TLP) software. Lightning products visualize, for example, the type and amplitude of lightning events.

More information

- On-demand radar products (page 29)
- Pregenerated weather products (page 30)

2.1 Data flow

IRIS Focus runs on a web server that users can connect to in an enterprise intranet or from an external location or the Internet.

The following figure shows a setup where IRIS Focus is used as part of a complete Vaisala weather device network consisting of 2 radar sites, 2 WindCube Scan Lidar sites, and an additional lidar or radar site.

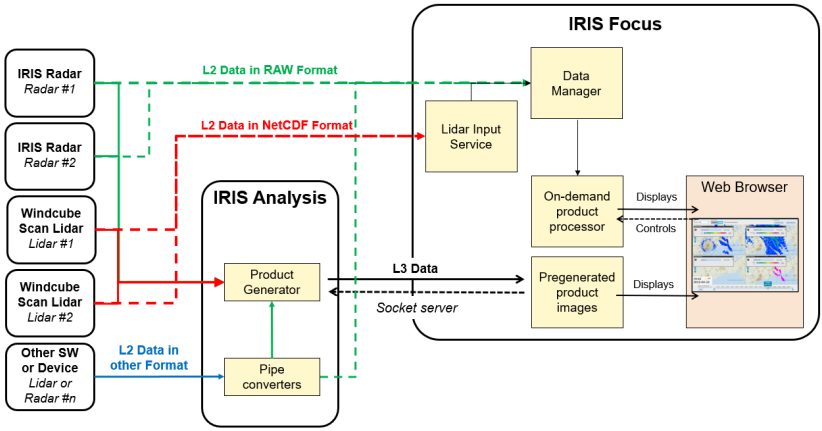


Figure 3 IRIS Focus data flow

In this case, IRIS Analysis, IRIS Radar, and wind lidar software can be considered back-end services for the IRIS Focus front-end interface. Network connections between IRIS Focus and IRIS Analysis processing back-end go through a socket server, a custom protocol over TCP/IP that delivers radar data from the IRIS back-end services to IRIS Focus. IRIS Focus polls the server for data and displays it on screen using the browser.

The components have the following functions:

- *IRIS Radar* - Operates the radar site and stores data gathered from the radar signals in RAW format.

- *Wind lidar software* - Operates the lidar site and stores data gathered from the lidar signals in NetCDF format.
- *IRIS Analysis* - Receives RAW data from a weather device through secure connection and processes it into displayable weather products.
- *IRIS Focus* - Polls pre-configured weather products from IRIS Analysis, displays them on the web interface, and generates on-demand weather products from RAW or NetCDF data.

The back-end collects data in different configurations, which are defined as *tasks* in IRIS Radar and wind lidar software. Tasks are sets of operating parameters for the device hardware and signal processing components, for example:

- Surveillance **PPI** scan at a single elevation angle
- Complete volume scan at multiple elevation angles
- Wind velocity scan

Each task type provides different source data. Users can select the task type when selecting an on-demand weather product to display in IRIS Focus.

2.2 IRIS Focus licensing

IRIS Focus requires a software license to run. To activate the license, you need a product key.

Vaisala delivers the product key when you purchase the software. If you have purchased the software and you have not received the product key, please contact Vaisala.

For server deliveries, Vaisala activates the product key in the factory, and a Vaisala representative sends you the key for future reference.

The license is mapped to the hardware of your IRIS Focus server or the ID of your virtual environment. If your hardware configuration changes and you need to re-install IRIS Focus, you must request a replacement license from your Vaisala representative.

An exception to this is the USB license key. If you have a USB license key, IRIS Focus runs when the USB license key is inserted in the server. If you install IRIS Focus on another server, you can move the USB license key to that server.

To view information about the license version, login to IRIS Focus as `admin`, and select **Admin > System > Licensing Management**.

License options

IRIS Focus has a basic license called *IRIS Focus Light*. This license enables users to view certain weather data on the map, but gives limited interaction with the tools. The full license is called *IRIS Focus*. This license provides access to the interactive features of IRIS Focus. The *IRIS Focus* license includes all the features of *IRIS Focus Light*.

There are separate licenses for weather radar/wind lidar data visualization and for lightning data visualization. One user can have access to both licenses. Access to licenses is defined by the IRIS Focus administrator in the user's profile.

IRIS Focus Light

IRIS Focus Light view has an unlimited number of seats. If there are no *IRIS Focus* license seats available, the user will be logged in with an *IRIS Focus Light* license. If the licence is missing, users cannot log in. This could happen, for example, if the USB license key has been removed or if this is a new installation, not from the factory, that requires an e-mail be sent to Vaisala to retrieve the license.

With an *IRIS Focus Light* license, the user sees the *IRIS Focus Light* map view. The following features are available:

- View one pregenerated weather product at a time (no on-demand products)
- See areas of interest with active alerts highlighted in the alert severity color when viewing current data
- View WMS map layers
- View the animation timeline
- View the cursor tool
- Create and edit personal color scales
- Change radar/lidar site
- Select map features
- Use the **Ruler Tool**
- Change user preferences

There are two variants of the *IRIS Focus Light* license:

- ***IRIS_Focus_Light_LGT***
  This license is for viewing lighting data.
- ***IRIS_Focus_Light_WR***
  This license is for viewing weather radar/ wind lidar data.

IRIS Focus

IRIS Focus licenses are based on a floating seat pool.

There are two variants of the *IRIS Focus* license:

- ***IRIS_Focus_Lightning***
  This license enables users to view the full-scale visualizations of lightning network sensor data, and to use all the related interactive tools.
- ***IRIS_Focus_Weather_Radar***
  This license enables users to view the full-scale visualizations of weather radar and wind lidar data, and to use all the related interactive tools.

The following features are available with the *IRIS Focus* license (in addition to all the *IRIS Focus Light* features):

- Create places of interest and set up alerting for them
- View alert icons on the map
- View alert history and the list of active alerts
- Advanced map features and tools

Advanced feature licenses

In addition to the *IRIS Focus Light* and *IRIS Focus* licenses, the following advanced feature licenses are available. These are system level licenses; one advanced feature license applies for all users.

Using the **NetworkHealth** product, **Turbulence** product, and Nowcasting also requires that the user has a Focus seat.

- ***IRIS_WMS***
  With the *IRIS_WMS* license, external WMS layers can be added to the system. Users can then access the layers through the weather product panel.
- ***IRIS_Nowcast***
  With the *IRIS_Nowcast* license, you get access to the nowcast algorithm for creating forecasts based on weather radar data up to 6 hours into the future. Using this feature also requires the *IRIS_Focus_Weather_Radar* license.
- ***IRIS_NetworkHealth_LGT***
  With the *IRIS_NetworkHealth_LGT* license, you can get the network performance information from the **Total Lightning Processor**, and display the information as **NetworkHealth** product in the product panel. Using this feature also requires the *IRIS_Focus_Lightning* license.
- ***IRIS_StormIntensity_LGT***
  With the *IRIS_StormIntensity_LGT* license, you can view the **Storm Intensity** product layer. Using this feature also requires the *IRIS_WMS* license.
- ***IRIS_ThreatZone_LGT***
  With the *IRIS_ThreatZone_LGT* license, you can view the **Lightning Threat Zone** product.
- ***IRIS_VHF_LGT***
  With the *IRIS_VHF_LGT* license, you can view VHF lightning data.
- ***IRIS_Turbulence***
  With the *IRIS_Turbulence* license, you can view the **Turbulence** product.

Seat-based license pool

IRIS Focus licenses are available in different configurations. To increase your seat count, you must replace the current license with a new one by contacting your Vaisala representative.

The seat count defines how many users can access IRIS Focus at the same time. For example, if there are 10 users with IRIS Focus privileges configured to the system, and there are only 5 IRIS Focus seats, then the first 5 users to access the system will be given *IRIS Focus* rights, whereas the remaining 5 users will enter the system with *IRIS Focus Light* credential.

Seat counts within a workstation are browser-based. For one license reservation, users may view IRIS Focus in as many instances or tabs of one browser, such as Firefox®, as they like. If a user opens IRIS Focus in a different browser, such as Google Chrome™, they reserve one license for each browser.

Licensing based on the number of weather radars

IRIS_Focus_Light_WR and *IRIS_Focus_Weather_Radar* licenses are valid for a defined number of weather radars. If you have more radars in the network than licenses, you need to define which radars the licenses are applied to. To do this, configure the `vsoweb-override.ini` file.

CAUTION! If you have more radars in the network than licenses, and you have not configured the list of radars to apply the licenses to, the system will not display any radar data.

For detailed instructions, see chapter *Configuring licensing based on the number of radars*.

Licenging based on the number of lidars

IRIS_Focus_Light_WR and *IRIS_Focus_Weather_Radar* licenses are valid for a defined number of lidars. If you have more lidars in the network than licenses, you need to define which lidars the licenses are applied to. To do this, configure the `vsoweb-override.ini` file.

CAUTION! If you have more lidars in the network than licenses, and you have not configured the list of lidars to apply the licenses to, the system will not display any lidar data.

For detailed instructions, see chapter *Configuring licensing based on the number of lidars*.

More information

- Configuring licensing based on the number of radars (page 57)
- User roles (page 165)

2.2.1 Focus Light user and Focus user differences

The following table summarizes differences between the IRIS Focus Light view (without the Focus role/Focus license) and full IRIS Focus view (with the Focus role and license).

Table 3 Focus Light user and Focus user

Feature	IRIS Focus Light view	IRIS Focus full view
View one pregenerated weather product at a time	✔	✔
View up to four weather products simultaneously (pregenerated and on-demand products)	-	✔
Create personal areas of interest and monitor these areas for weather events	-	✔

Feature	IRIS Focus Light view	IRIS Focus full view
View organization-level areas of interest	✓	✓
See areas of interest with active alerts highlighted in the alert severity color when viewing current data	✓	✓
View alert icons on the map	-	✓
View alert history and the list of active alerts	-	✓
Change user preferences	✓	✓
View WMS map layers	✓	✓
View the animated timeline	✓	✓
Use data analysis tools, like Tracking tool, Ruler tool, and Cursor tool	✓	✓
Select map features	✓	✓
Edit color scales	✓	✓
Advanced map features and tools	-	✓
Select radar/lidar site	✓	✓

3. Requirements

3.1 IRIS Focus hardware requirements

Table 4 Hardware requirements

Minimum	Recommended[1)]
• Modern 4-core CPU (Intel Xeon E5 series or similar) • 32 GB RAM • 1 TB HDD • 1400 x 1050 minimum screen resolution	• Modern 8-core CPU (Intel Xeon E5 series or similar) • 64 GB RAM • 2x 1 SAS TB HDD in RAID 1 configuration • 1920 x 1200 screen resolution

1) *The pre-installed IRIS Focus system delivery option uses the Dell PowerEdge R450 rack server unit, which meets the recommended hardware setup. See the Dell product data sheet for full specifications.*

When viewing IRIS Focus on minimum or low resolution, make sure that the browser zoom is set to 100% or lower.

The hardware capacity directly affects the performance of IRIS Focus. Multiple users can be logged in to IRIS Focus, and each user can have multiple weather and terrain layers rendered on screen at the same time. Each weather and terrain layer requires some resources from the system.

3.2 Software requirements

To enable IRIS Focus installation, and for IRIS Focus to function properly, the following software requirements must be met.

Root access

The root user rights need to be accessible during installation. When using the AlmaLinux installer provided by Vaisala, you can access the root user by logging in as **admin** and using the **sudo** command.

AlmaLinux 9.3

IRIS Focus 7.4 has been verified to work when installed on AlmaLinux 9.3. IRIS Focus will most likely work on other releases of AlmaLinux 9, but verification of this has not been performed by Vaisala.

Before installing IRIS Focus, you need to have either AlmaLinux 9.3 ISO image mounted on your server (offline installation) or a functional internet connection (online installation).

The installation script verifies the version of several core system packages during the installation and updates them from the mounted media or Internet.

IRIS Focus 7.4 has been verified to work when installed on AlmaLinux 9.3. IRIS Focus will most likely work on other releases of AlmaLinux 9, but verification of this has not been performed by Vaisala.

IRIS Analysis

The IRIS Analysis server provides radar products through a proprietary socket server connection. The socket server connection is enabled if at least one radar is connected to your IRIS Analysis server, at least one product is configured and generated in IRIS Analysis, and the IRIS Analysis server has IRIS software version 8.13.6 or newer installed. No further configuration is needed.

The map projection in the IRIS Focus web application depends on having a single radar or a group of radar sites to act as a center point to for map rendering.

In most IRIS Focus configurations, the radar product generator is an IRIS Analysis server that has been set up earlier on the radar site. For more information, please contact Vaisala.

If you have RAIN1 product created using a 3d CAPPI with R (rain intensity) as an input to the RAIN1, you need IRIS 9.1.0.

For information on configuring IRIS Analysis, see *IRIS and RDA Software Installation Guide (M212924EN)*.

Before beginning the IRIS Focus installation, make sure you know the hostname of your socket server.

Data Manager

Radar volume data is fetched from the Data Manager interface and processed to on-demand radar products in the IRIS Focus application.

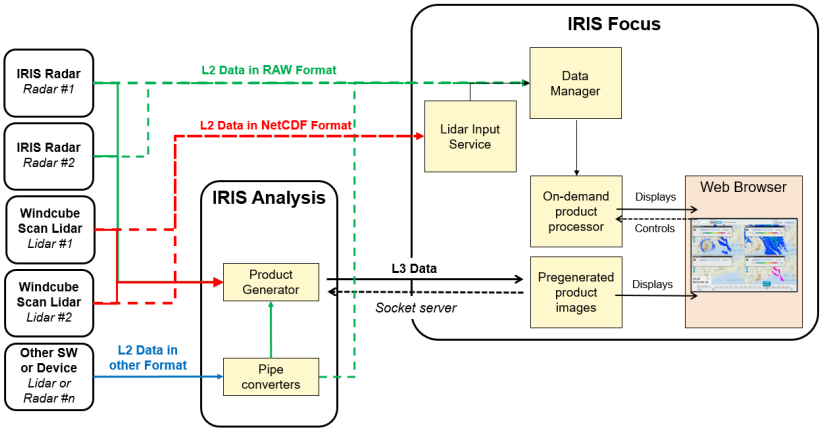


Figure 4 Generating on-demand IRIS products

Firewall

Configure the firewall to leave the following ports open:

- SSH access: port 22
- HTTP: port 80 (allows users to enter http: instead of https: in their browser; redirected to 443 immediately upon connection)
- HTTPS: port 443
- Kafka: port 9094 (only required for live lightning feed from external TLP)

Fixed Ethernet MAC Address

IRIS Focus system needs a fixed Ethernet MAC Address (Hexadecimal number assigned to each device / network board). Locking criteria used for the licenses are based on Ethernet MAC Address and Hostname. If you are using bonding Ethernet, the MAC address can change, which could lead to the impossibility to connect to the software.

DNS

Having DNS as part of the system is recommended. However, an offline installation without DNS is possible.

ipv4 and ipv6

Both ipv4 and ipv6 must be enabled.

Supported browsers

IRIS Focus supports current Microsoft Edge®, Mozilla Firefox®, and Google Chrome™ browsers.

On other browsers, the user interface features may work only partially, with poor performance, or not at all.

The recommended screen resolution is 1920 x 1200 (minimum 1400 x 1050).

More information

- Installation and configuration command options (page 48)
- Setting up Data Manager (page 60)
- Configuring DNS (page 145)

3.3 Network requirements

Security

IRIS Focus must be installed in an isolated, secure network. IRIS Focus as a web software is meant to be used within a network, not completely offline. For security reasons, direct access to IRIS Focus from the internet should be blocked.

Network bandwidth

Table 5 IRIS Network requirements

Item	Specification
Communication from IRIS Analysis and the TLP to IRIS Focus	
Network data transfer	>100 Mbit/s (1000 Mbit/s recommended)

3.4 Data Manager disk space requirements

The amount of radar data generated depends on a number of variables, including, for example:

- Size of the RAW files as determined by factors such as radar scan strategy, range, number of recorded data, and the amount of precipitation
- Number of radars in the network
- Amount of disk space reserved for the partition where data manager stores the data

The table below shows examples of how much disk space is required for Data Manager to store data collected over a certain time period. In addition, 400GB is needed for other purposes (*/srv* partition). Use the following formula to calculate the approximate disk space :

```
totalDiskSpace GB = 400+ (scanSize GB * numberOfRadars * (1440/
scanIntervalMinutes) * daysOfData)
```

Table 6 Approximate required disk space examples for an IRIS RAW file of 0.01 GB

Scan interval (minutes)	Number of radars	Days of Data				
		1 month	6 months	1 year	5 years	10 years
5	1	100 GB	500 GB	1 TB	5 TB	10 TB
10	1	50 GB	250 GB	500 GB	2.5 TB	5 TB
5	2	200 GB	1 TB	2 TB	10 TB	20 TB
10	2	100 GB	500 GB	1 TB	5 TB	10 TB
5	5	500 GB	2.5 TB	5 TB	25 TB	50 TB
10	5	200 GB	1.3 TB	2.6 TB	13 TB	26 TB
5	10	1 TB	5 TB	10 TB	50 TB	100 TB
10	10	500 GB	2.5 TB	5 TB	25 TB	50 TB
5	25	2 TB	13 TB	26 TB	130 TB	260 TB
10	25	1 TB	6 TB	13 TB	65 TB	130 TB

More information

- Data Manager (page 179)

4. IRIS Focus architecture

Architecture for radar products

IRIS Focus reads data in the formats produced by weather radar signal processors.

Usually this data is relayed to IRIS Focus through the signal processing and analysis component, IRIS Analysis, either as pregenerated radar products or as radar scan source data files which are processed and displayed as radar products by IRIS Focus.

IRIS Focus only accepts a single data source as its socket server. IRIS Analysis can be connected to an unlimited number of radar sites and relay their radar products to IRIS Focus.

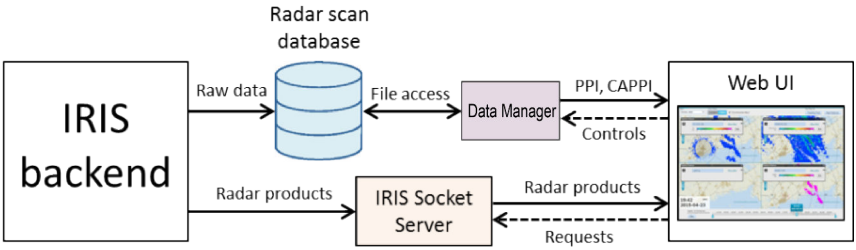


Figure 5 IRIS Focus architecture for radar products

Architecture for lidar products

Data from Vaisala WindCube Scan Lidars can be sent to IRIS Focus for visualization. Currently, PPI and FIXED scans are supported in IRIS Focus to be displayed or processed.

The Windforge software generates the data into a NetCDF file. The file is then sent to a specific directory in the Lidar Input Service, which in turn sends the file to Data Manager. IRIS Focus is compatible with the Windforge version 3.5.0.

IRIS Focus creates task names from ingested lidar data using the user-defined scan name in the lidar scan configuration. Lidar data previously ingested through IRIS Analysis may have a different scan naming scheme: the scan type and scan id (version of the configuration change in the lidar) separated by an underscore.

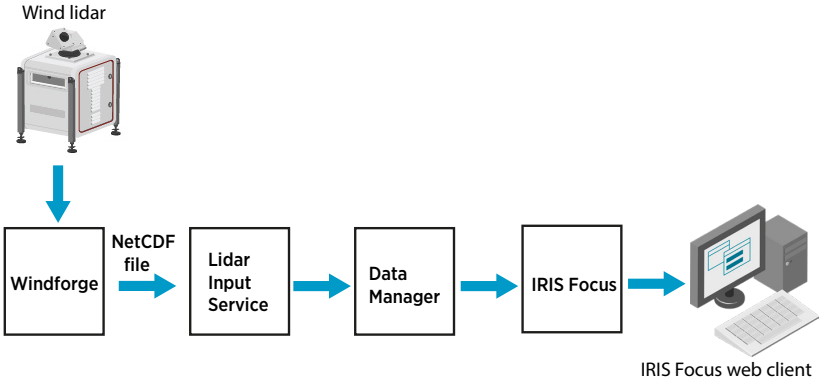


Figure 6 IRIS Focus lidar architecture

Windforge	Operates the Lidar site and stores data gathered from the radar signals in NetCDF format.
IRIS Analysis	Receives netCDF data from Windforge through secure connection and processes it into displayable lidar products.
IRIS Focus	Receives netCDF data from Windforge, polls pre-configured lidar products from IRIS Analysis, displays them on the web interface, and generates on-demand lidar products from netCDF data.

Architecture for lightning products

The data for lightning products in IRIS Focus originates from a Vaisala Lightning Detection System which uses multiple, remote sensors to detect signals emitted by lightning discharges, while filtering out the signals from non-lightning sources. Each sensor sends its data to the central processor (the **Total Lightning Processor**, TLP) where lightning locations are determined.

To ensure that the sensor data set applies to the same lightning event, the TLP compares the time at which the event was recorded by each sensor, and then calculates the precise location of the lightning event. the TLP also records several other descriptive characteristics of each lightning event. The data from the TLP is delivered to IRIS Focus. The data is ingested to the system in real time, after which it can be requested across specific time frames by lightning products.

A single TLP can consume and merge data sets from multiple other TLP systems to produce a superset of data. For example, if organizations from three neighboring countries share TLP data, they can have a superset of lightning solutions from all three countries on each of the TLP systems. From there, they can create subsets of data feeds by lightning characteristics or geographic regions. Each of these subsets can then be fed to a specific Kafka topic on a specific Kafka cluster. Each of these topics can feed several IRIS Focus systems.

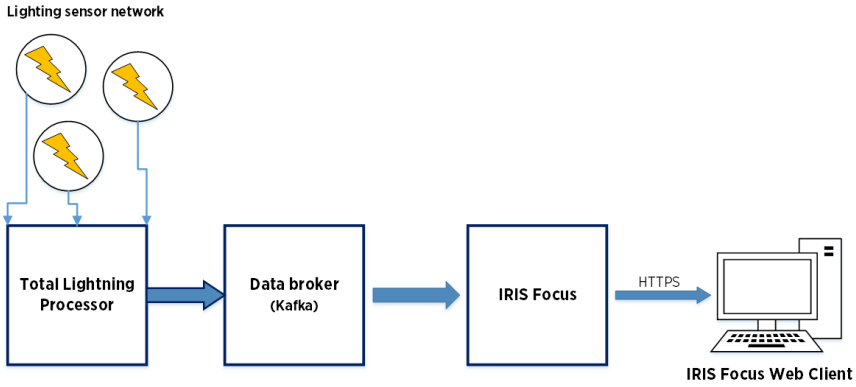


Figure 7 IRIS Focus lightning architecture

Visualization of products on the map

Each weather product is displayed on top of a map view, which is rendered by a GeoServer instance that is installed during the IRIS Focus installation. The map terrain and detail layers are always on the background, and the weather products are drawn on top of them. The user can change the order of weather product layers in real-time.

IRIS Focus can also display data received through WMS protocol, for example, satellite data. This data is also displayed as product layers over the map layer.

Most weather products have editable color scales. Color scales are stored on the IRIS Focus server and can be reused.

4.1 Map layers

The background map and the weather data visualizations are drawn as individual layers and then combined to form an overview of current weather conditions.

You can also view WMS layers from external sources, such as satellite image layers, as layers on the map.

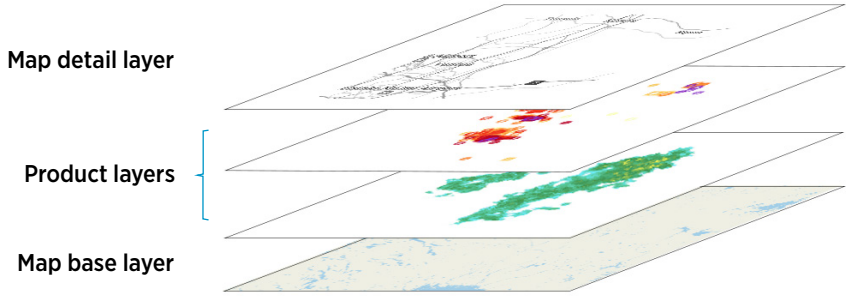


Figure 8 IRIS Focus map layers

Map layers

The background and foreground consist of non-interactive layers. At the bottom is the map base layer, which can be enhanced with the map detail layer containing roads, province boundaries, and other similar terrain features. The map detail layer will be projected on top of product layers.

Product layers

IRIS Focus users may have up to four product layers included in the map rendering, consisting of any combination of the IRIS Focus or external WMS products that the installation is licensed for.

4.2 GeoServer and maps

The map engine in IRIS Focus uses GeoServer architecture. When reading data from a single radar site, GeoServer renders the map using Azimuthal Equidistant projection, which means that all directions and distances are correct when measured from the point of origin, which is the radar site in this case. When reading data from a composite of multiple radar sites, Web Mercator projection is used.

The terrain data in IRIS Focus consists of a detailed vector map of Earth, separated into multiple layers. The base map content is licensed from the collaborative OpenStreetMap project, which provides all vector shapefiles for the base terrain.

Figure 9 Base map from GeoServer

To save system resources, the shapefiles are combined into different map detail levels that are rendered as a single layer where possible. For example, selecting the **Full detail** map level does not draw separate layers for terrain, roads, map labels, and other map features. Instead, all the content has been precompiled into a single layer in the IRIS Focus map package and then drawn on screen.

When a user opens the map view in IRIS Focus, GeoServer processes the vector data in the current view area into 256×256 PNG tiles that are displayed in the browser window. New tiles are calculated and generated every time the user pans or zooms on the map, so moving on the map may feel a bit sluggish in the beginning. To improve performance, GeoServer runs a caching component called GeoWebCache that stores the tiles for faster retrieval in the future.

GeoServer has a management web interface that runs at the following location:

`http://localhost:24180/geoserver`.

The default management account name is **`admin`**, and the password can be found in the following file:

`/etc/vaisala/radarsw/configuration/gis-override.ini`

The password is generated automatically during IRIS Focus installation.

The base map data is stored in a PostgreSQL database, which also stores all web application data.

More information

- Adding external map layers (page 177)

4.3 On-demand radar products

When displaying on-demand radar products, IRIS Focus fetches raw radar measurement data from the back-end and processes it in real-time. This provides hands-on control over the radar product parameters.

The full raw radar volume data is stored and can also be used later for on-demand product generation.

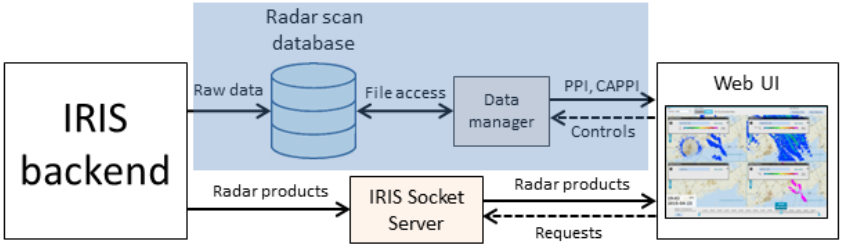


Figure 10 On-demand product components

Data for on-demand products comes from the **RAW** format files produced by the IRIS backend.

IRIS Focus reads the **RAW** data through the data manager.

When you select an on-demand radar product in IRIS Focus, the web application accesses the database and fetches the required data, not only for the current situation but for the whole recorded segment. The data is then processed and displayed in IRIS Focus.

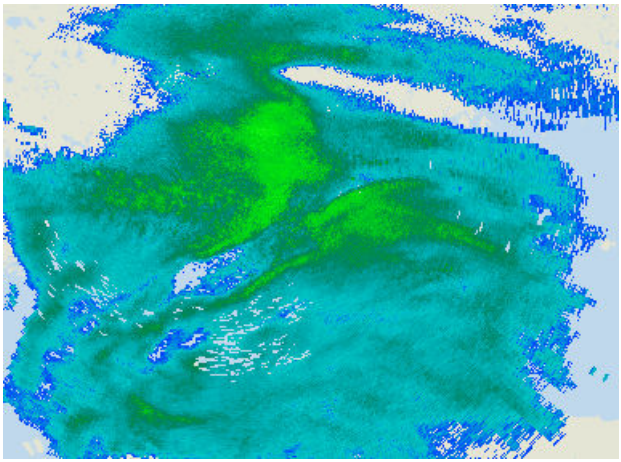


Figure 11 Visualized radar product

More information

- Data Manager (page 179)

4.4 Pregenerated weather products

Pregenerated products are generated by signal processing components in IRIS Analysis. IRIS Focus reads the list of products, and displays the products that the user requests on the IRIS Focus map view.

The radar products and their settings are preconfigured, and only displayed in IRIS Focus. They cannot be edited in the IRIS Focus map view.

There is no upper limit to the number of preconfigured radar products that IRIS Focus can have.

The raw volume data is stored on a IRIS Analysis server. The data can be archived to tape or stored on a large disk array.

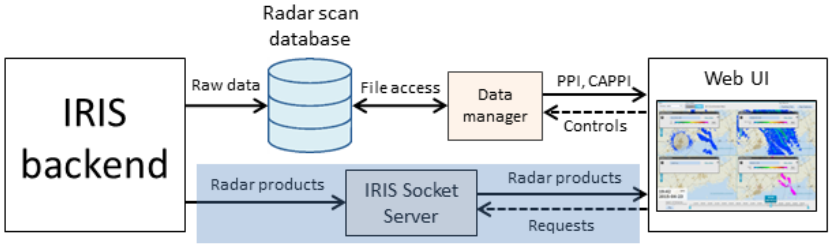


Figure 12 IRIS Analysis product data flow to IRIS Focus

The radar products are rasterized into 2D bitmap images, based on the back-end signal processing settings. The images are sent to the IRIS Focus web user interface through the IRIS Socket Server interface. The Socket Server uses TCP port 30735 to communicate with IRIS Focus.

When you select a pregenerated product in IRIS Focus, IRIS Focus polls the Socket Server and loads the image.

The resolution of pre-configured radar products is limited by the capacity of the processing module that produces them. For example, IRIS Analysis has the following limitations:

- Max number of **bins** in any **ray** at any time: 4200
- Max number of **rays** in a sweep: 1024
- Max number of **parameters** recorded in a **sweep**: 16
- Max number of **sweeps** per **scan**: 40

For information on setting up IRIS Analysis products, see *IRIS Product and Display Guide (M212928EN)*.

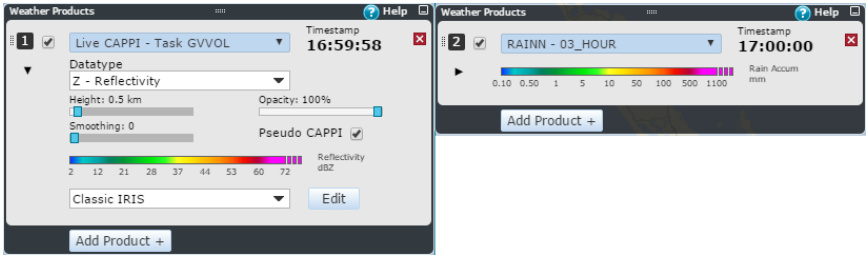


Figure 13 On-demand and pregenerated product settings

4.5 GLD360 lightning layer

Vaisala offers an optional subscription service for the Vaisala Global Lightning Dataset GLD360. GLD360 is a dedicated data stream that measures lightning strikes from the surface of the Earth, and its data is generated outside IRIS Focus.

GLD360 can be integrated with IRIS Focus and included as an additional WMS lightning layer in the web UI, where the user can view it just like radar product layers.

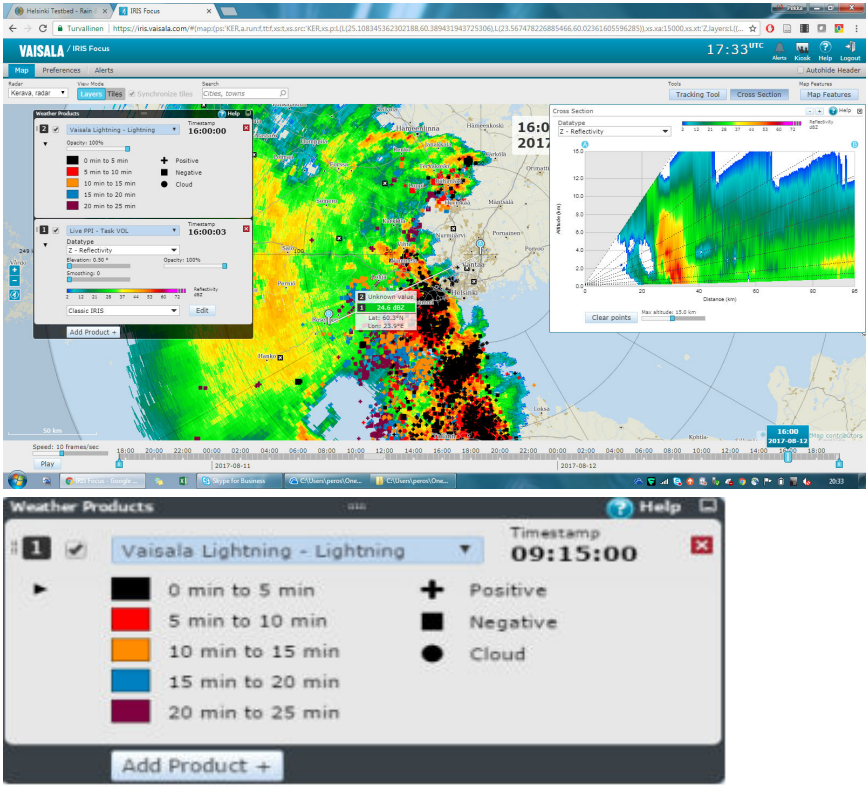


Figure 14 GLD360 lightning layer and controls

To take the GDL360 lightning layer into use, the IRIS Focus server must be online and your organization must have an active subscription to GLD360 data. For information on subscribing to GLD360 data, contact Vaisala Lightning Data Services.

More information

- Adding GLD360 lightning layer (page 173)

4.6 Web application

IRIS Focus supports current Microsoft Edge®, Mozilla Firefox®, and Google Chrome™ browsers.

IRIS Focus only accepts HTTPS connections. All requests to the standard HTTP port are redirected to the HTTPS port 443.

All application settings are stored in a PostgreSQL database on the IRIS Focus server.

More information

- Installing a CA certificate (page 184)
- Certificates (page 227)
- Encryption (page 227)

5. Installation for weather radar and/or wind lidar

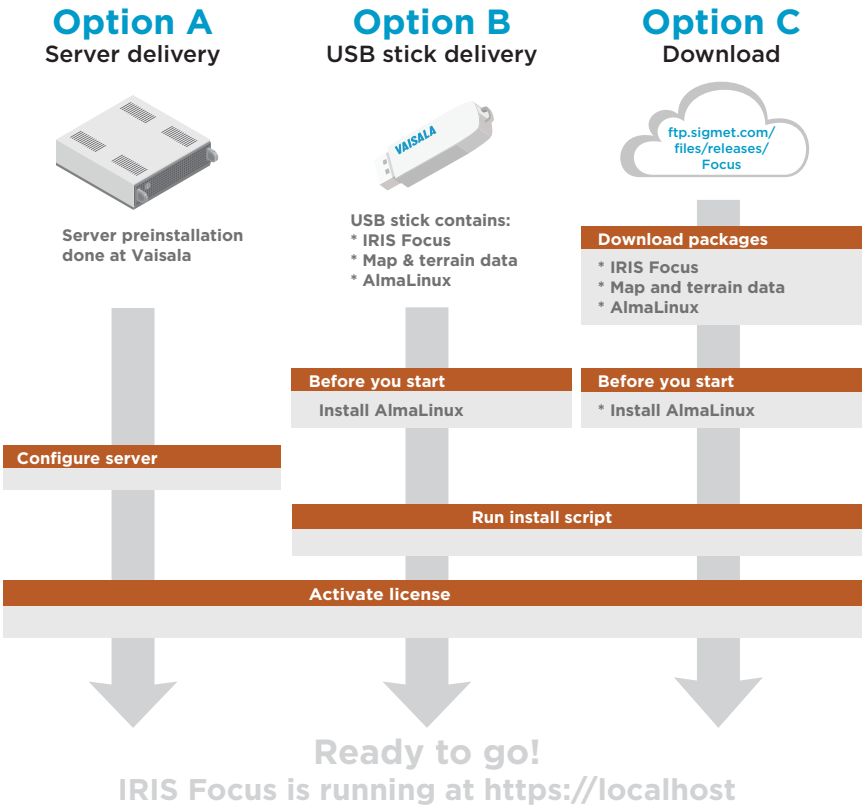


Figure 15 IRIS Focus delivery options

Option A Pre-installed system delivery from Vaisala. The "turnkey" option. Place an order and wait for delivery by Vaisala.

Option B Preconfigured USB stick containing the AlmaLinux operating system and all required files for installing IRIS Focus.

Option C Downloadable installation packages. Download the required packages to install IRIS Focus on your server.

More information

- Installation security notes (page 230)

5.1 Downloading installation packages

1. Connect to Vaisala Sigmet server (https://ftp.sigmet.vaisala.com) using a web browser or an FTP client.

   The host server allows read access for anonymous FTP connections.
   The files come in parts. Follow the steps in chapter *Verifying and joining files* to join the file parts together.
2. If using a web browser, navigate to */files/releases/Focus/<latest version>/Focus_install*, or if using an FTP client, navigate to */outgoing/releases/Focus/latest version>/Focus_install*.
3. Download the files inside the *installer* directory.

   The files are very large. Use a download tool such as CrossFTP that allows resuming downloads to fetch the files.

4. Navigate to */releases/Focus/vaisala-map-data*, and download the map files from */vaisala-iris-maps-v2* and terrain data files from */vaisala-iris-terrain-v2*.
5. If you require the AlmaLinux installation image, download it from:

   https://ftp.sigmet.vaisala.com/files/releases/AlmaLinux

   The AlmaLinux installation image is very large.

   You can skip the AlmaLinux installation image if you already have an appropriately configured AlmaLinux server installed.

5.1.1 Verifying and joining files

Each file has an associated *md5sum* file located in the same download directory.

In these instructions, `x_x` means the latest major and minor version.

After downloading the file(s), verify their integrity by checking each file's MD5 hash against the one provided at the installation site.

1. Verify the MD5 checksum values of the downloaded IRIS Focus installation files:
   - In AlmaLinux – Use the pre-installed **`md5sum`** command line tool: **`md5sum [filename]`**
   - In Microsoft Windows – Use the pre-installed **CertUtil** utility: **`certutil -hashfile [filename] MD5`**
2. Join the IRIS Focus installation file parts together to form a single tar file with the following command:

```
cat IRIS_Focus*_part_* >| IRIS_Focus_x_x_Installer.tar
```

3. Get the MD5 checksum value for the tar file that you created:

```
md5sum IRIS_Focus_x_x_Installer.tar
```

4. Verify that the MD5 checksum value matches the one shown in the ***`IRIS_Focus_x_x_Installer.tar.md5`*** file that you downloaded from https://ftp.sigmet.com
5. If you see any discrepancies in the hashes, download the mismatching file again.
6. Get the MD5 checksum value for the map files:

```
md5sum vaisala-iris-maps-v2-part* | tee mymd5sums
diff mymd5sums vaisala-iris-maps-v2.md5sum.txt && echo "Checksum verified
ok"
```

7. Get the MD5 checksum value for the map and terrain files:

```
md5sum vaisala-iris-terrain-v2-part* | tee mymd5sums
diff mymd5sums vaisala-iris-terrain-v2.md5sum.txt && echo "Checksum
verified ok"
```

8. Join the terrain data files together to form two zip files:

```
cat vaisala-iris-terrain-v2-part* >| terrain-v2.zip
unzip terrain-v2.zip
rm terrain-v2.zip
```

Leave the map files in parts.

5.2 Prerequisites for installation

Before installing IRIS Focus, make sure your environment meets the necessary hardware and software requirements.

More information

- IRIS Focus hardware requirements (page 19)
- Software requirements (page 19)

5.3 Installing AlmaLinux

A prerequisite for installing IRIS Focus is that AlmaLinux is installed on your intended IRIS Focus system.

IRIS Focus 7.4 has been verified to work when installed on AlmaLinux 9.3. IRIS Focus will most likely work on other releases of AlmaLinux 9, but verification of this has not been performed by Vaisala.

IRIS Focus has been tested with the security profile selection set to **None**.

If you do not have an AlmaLinux system running, select an installation image from Vaisala Sigmet server (https://ftp.sigmet.vaisala.com/files/releases/AlmaLinux/), and see instructions at Tecmint Linux Guides (https://www.tecmint.com/install-almalinux-9/) on how to perform the AlmaLinux installation.

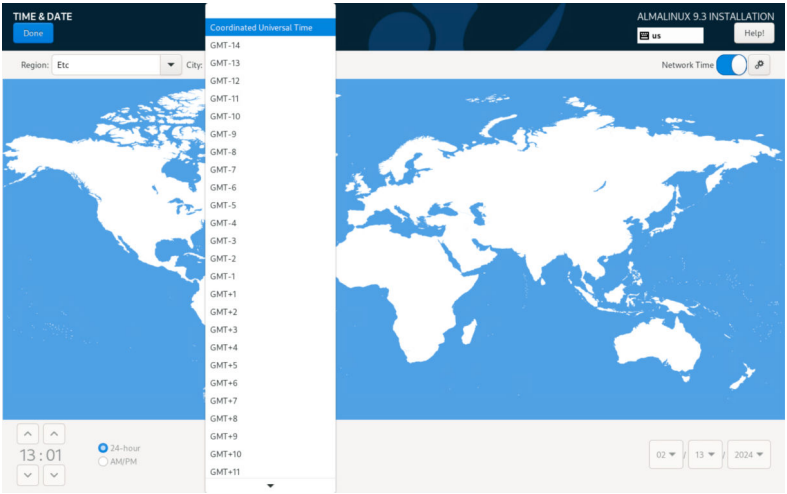
Install AlmaLinux according to the standard instructions, with the following changes.

Table 7 Recommended disk partitioning

Partition	File system type	Size
/home	XFS	50 GB
/boot	EXT4	500 MB
/boot/efi	EFI	600 MB
/var	XFS	100 GB
/	XFS	50 GB
swap	SWAP	size of RAM + 2 GB
/srv	XFS	All of the remaining disk space

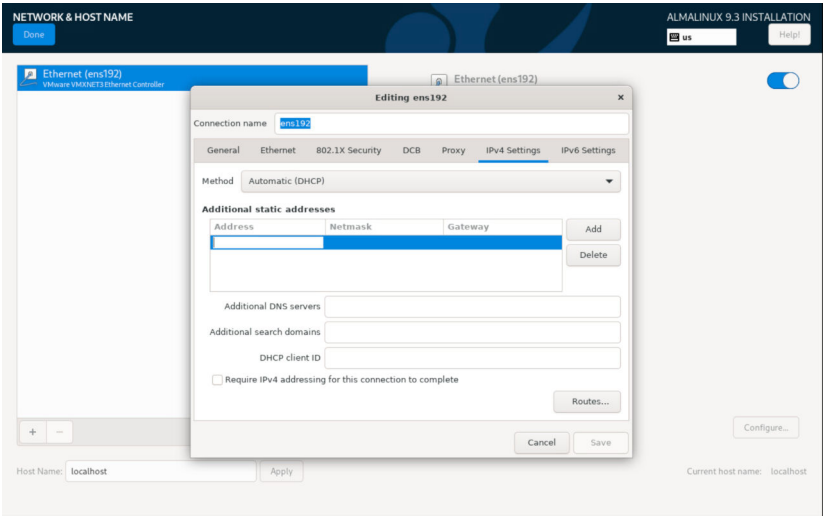
If there is only a little disk space, you can decrease the size of the */home* and */* partitions by 10-20 GB.

1. Select your installation language.
2. In **TIME & DATE**, set the system clock to Coordinated Universal Time (UTC) by choosing the following values:
   - Region: **Etc**
   - City: **Coordinated Universal Time**



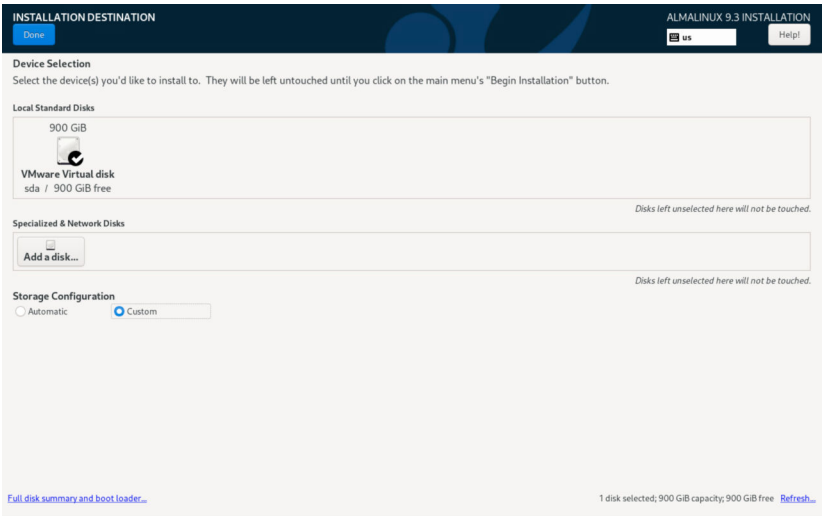
3. In **SOFTWARE SELECTION**, keep the default selection for **Base Environment Type** : **Server With GUI**.

4. In the AlmaLinux installation screen, select **Network & Host Name**.

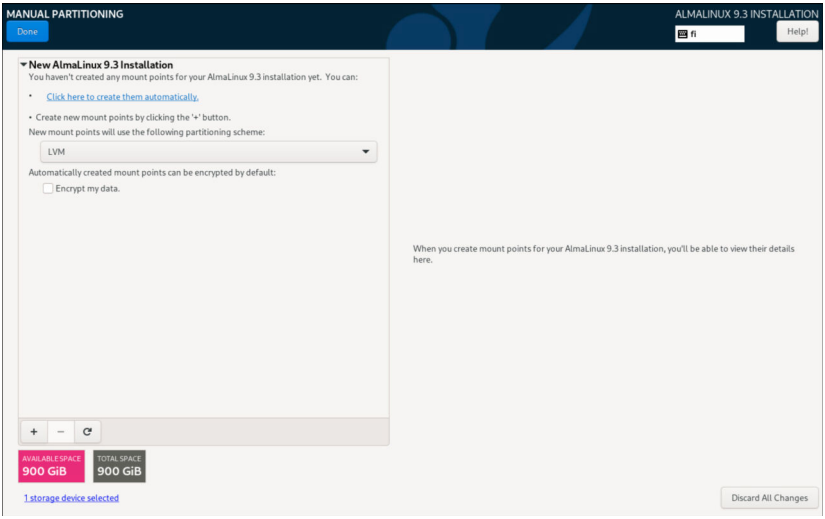


   a. Turn the network **ON**.
   b. Select **Configure**.
   c. In the **General** tab, select **Connect automatically with priority**.
   d. In the **IPv4 Settings** tab, select **Method > Manual**.
   e. In the **IPv4 Settings** tab, select **Add** to add your network IP address, Netmask, Gateway, and DNS servers.
   f. Select **Save**.
   g. In **Host Name**, type a name for this server.
   h. Select **Apply**.
   i. Select **Done**.

5. In **INSTALLATION DESTINATION**, start manual partitioning:
   a. Select the hard disk.
   b. Select **Storage Configuration**: **Custom**.
   c. Select **Done**.



6. In the **Manual partitioning** window, select **Click here to create them automatically**.

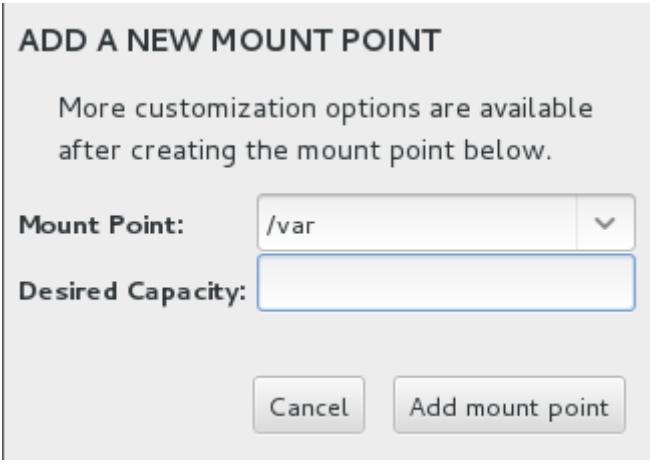


After creating the automatic partitions, you need to modify the partition manually in the next steps.

7. Modify the **/home** partition.
   a. Select the */home* partition.
   b. Under **Desired Capacity**, set the size of the home partition (*/home*) to **`50 GiB`**.
   c. Select **Update Settings**.

8. Create the */var* partition:
   a. Select the plus (**+**) icon.

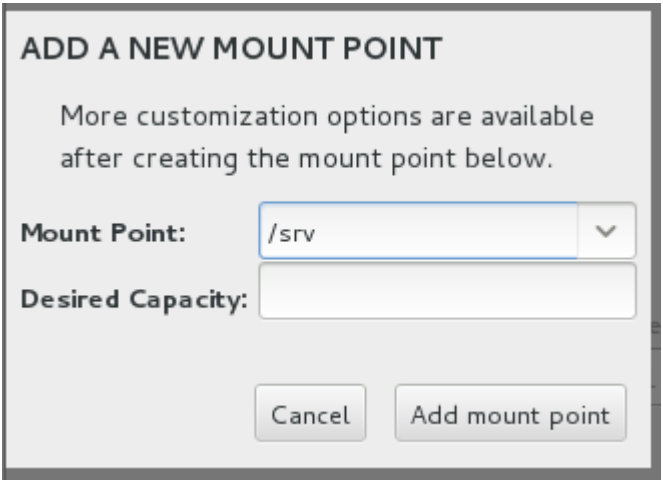
      The **ADD A NEW MOUNT POINT** dialog appears.



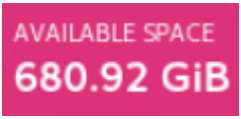
   b. In **Mount Point**, type */var*
   c. Under **Desired Capacity**, set the size of the */var* partition by typing **`100 GiB`**.
   d. Select **Add mount point**.
9. Select **/boot**.
   a. Under **Desired Capacity**, set the size of the */boot* partition by typing **`500 MiB`**.
   b. Select **Update Settings**.
10. Select **/**.
   a. Under **Desired Capacity**, set the size of the root partition (/) by typing **`100 GiB`**.
   b. Select **Update Settings**.
11. Select **swap**.
   a. Under **Desired Capacity**, set the size of the swap to the size that corresponds to RAM + 2 GB.
   b. Select **Update Settings**.

12. Create the */srv* partition:
    a. Select the plus (**+**) icon.

       The **ADD A NEW MOUNT POINT** dialog appears.

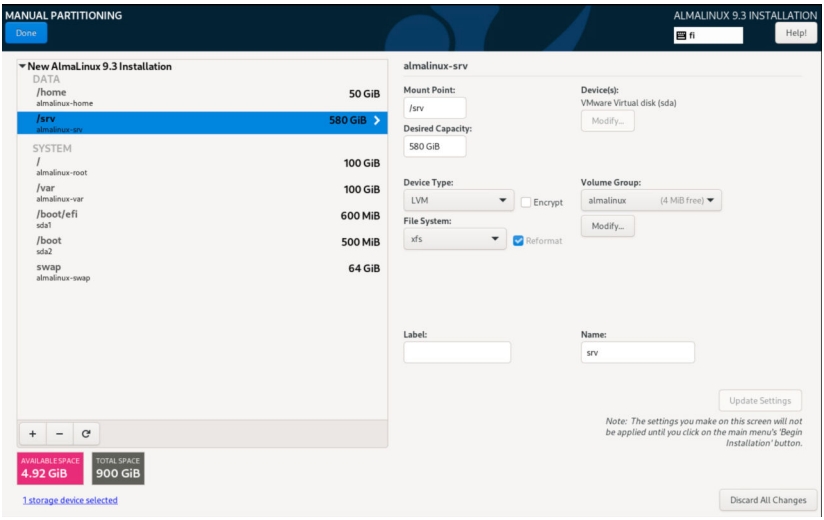


    b. In **Mount Point**, type */srv*
    c. Under **Desired Capacity**, use nearly all the available server space (indicated in the pink box) for the */srv* partition by typing, for example, **`680 GiB`**.



    d. Select **Add mount point**.
13. Select **Done**.

14. Check that the partitions are defined as follows (note that `/srv` has a different value):



15. Select **Done > Accept Changes**.

5.3.1 Setting the root password

If your system was pre-installed in Vaisala, the default password is `xxxxxxxx`.

1. Select **ROOT PASSWORD**.

   The **Root Password** window opens.

2. Enter your root password.

   Check the password strength meter. While Vaisala recommends a strong password, the software does not stop you from entering a weak one.

3. In the confirm text box, re-enter your root password.
4. In the upper left hand corner, select **Done** to return to the main configuration page.

   If your password is weak, you are prompted to select **Done** a second time.

5.3.2 Finalizing the installation

1. Select **USER CREATION**.

2. Create an account with the following properties:
   - User name: **radarop**
   - Password: **[choose password or use the default password xxxxxx]** Vaisala recommends using a non-default password.
3. In the upper left hand corner, select **Done** to return to the main configuration page.

   If your password is weak, you are prompted to select **Done** a second time.
4. Select **Begin Installation**.

   The installation will continue for a few minutes.
5. When prompted, selet **Reboot System**.
6. Select **LICENSE INFORMATION**.
7. Accept the license agreement.
8. Select **Done**.
9. Select **FINISH CONFIGURATION**.

The AlmaLinux installation is now complete. You are ready to install IRIS Focus.

5.4 Verify or override the FQDN of your server

You must determine or set the fully qualified domain name (FQDN) of the IRIS Focus server before installing the software. The FQDN must be the name that external clients will use when connecting to your IRIS Focus server. The installation assumes that this is correctly reported by the hostname command.

For example, if the final URL will be ***https://my-iris-focus.company.com/***, then the hostname command must report **iris-focus.company.com** as follows:

```
[root@my-iris-focus ~]# hostname --fqdn
my-iris-focus.company.com
[root@my-iris-focus ~]#
```

If your server does not report the correct host name, you can export an environment command to indicate what the correct host name should be. For example, if the output above had been "**my-iris**" and the correct value should have been "**my-iris-focus.company.com**", you should run the following command:

```
export HOST_FQDN=my-iris-focus.company.com
```

5.5 Installing IRIS Focus from a USB stick

The IRIS Focus installation USB contains the following file structure for the main version installation:

```
Focus_install
     vaisala-iris-maps-v2
     vaisala-iris-terrain-v2
     installer
     documentation
```

In the following instructions, **x.x** means the IRIS Focus major version and minor version number.

In the case of a patch release, the USB stick may also include an additional .tar file for the patch.

To install IRIS Focus from the USB stick, you must copy the files to the AlmaLinux server and prepare the files for installation.

5.5.1 Preparing the files on the USB stick

1. Reboot the system.
2. Log in to the server as **root**.
3. Insert the USB stick.

   If it is already plugged-in, remove and re-insert the stick.
4. In the pop-up dialog, select **Open With Files**.
5. Right-click a blank area and select **Open in Terminal**.
6. In the terminal, type **pwd** and press **ENTER**.

   The result is usually `/run/media/root/IRIS_FOCUS`.
7. Copy the *Focus_install* directory to the AlmaLinux server:

   ```
   mkdir /srv/Focus_install
   cp -r /run/media/root/IRIS_FOCUS/Focus_install/* /srv/Focus_install
   ```
8. Change to the *srv/Focus_install/installer* directory, and join the .tar file parts:

   ```
   cd /srv/Focus_install/installer
   cat IRIS_Focus_x_x_Installer_part_* >> IRIS_Focus_x_x_Installer.tar
   ```
9. To make sure that the file is now correct, run the following two commands and check that you get the same output:

   ```
   md5sum IRIS_Focus_x_x_Installer.tar
   cat IRIS_Focus_x_x_Installer.tar.md5
   ```

10. Extract the installation files into the default release directory:

```
tar -xvf IRIS_Focus_x_x_Installer.tar
```

11. Change to the */srv/Focus_install/vaisala-iris-terrain-v2* directory:

```
cd /srv/Focus_install/vaisala-iris-terrain-v2
```

    a. Join the terrain file parts:

    ```
    cat vaisala-iris-terrain-v2-part* > vaisala-iris-terrain-v2.zip
    ```

    Leave the map files in parts.

    b. Unzip the resulting terrain zip file:

    ```
    unzip vaisala-iris-terrain-v2.zip
    ```

    c. Remove the extra files:

    ```
    rm -rf vaisala-iris-terrain-v2-part*
    rm -rf vaisala-iris-terrain-v2.zip
    ```

5.5.2 Running the installation script

- If you do not want the system to reach any DNS server, use the offline installation method and the `--disable-dns` option.
- `<root application URL>` in the installation command example below corresponds to the hostname. If the hostname changes, you also need to change the `security.cors.origin.whitelist` parameter value in the *`vsoweb-override.ini`* file, and restart the application.
  The `cors-origin-whitelist (-cow)` switch determines the value of the `Access-Control-Allow-Origin` header. It must have the same value as the root application URL. The default value is the installation machine name.

1. Run the IRIS Focus installation script:

```
cd /srv/Focus_install/installer
./rsw-installer --online --gis-db-dump\
/srv/Focus_install/vaisala-iris-maps-v2 --terrain-dir\
/srv/Focus_install/vaisala-iris-terrain-v2 --radar -s <hostname or IP of
IRIS Analysis socket server> -cow <root application URL>
```

2. Reboot the system with the following command to cleanly bring up the services:

```
reboot
```

5.5.3 Installation and configuration command options

Table 8 Installation command options

Option	Description
--admin-password	Assign a non-default admin password.
--admin-user	Assign a non-default admin user.

--broken-dns	Only use this option if you network is unable to resolve the name of your IRIS Focus system using DNS and you can not use the **--fqdn FQDN** option to specify the correct name.
	`hostname --fqdn`
	`(default: False)`
	You can use either --offline or --online installation with this option.

-c OR --config-dir	Configuration directory

Option	Description
-cow	The **cors-origin-whitelist (-cow)** switch determines the value of the **Access-Control-Allow-Origin** header. It must have the same value as the root application URL. In the installation command, **<root application URL>** corresponds to the hostname. The default value is the installation machine name. If the hostname changes, you also need to change the **security.cors.origin.whitelist** parameter value in the *vsoweb-override.ini* file, and restart the application.
--deactivate-admin	Deactivate the admin account after running this script. Not needed for standard installations.
-d OR --debug	Get additional logging from the migration/ install.
--disable-dns	Turns off DNS in the */etc/nsswitch.conf* file. Only offline installation is possible with this option.
--docker-ip-addr 10.200.2.1	The IPv4 address use by containers running in Kubernetes to communicate with containers running in Docker. You only need to specify this option if the 10.200.2.1 default value is already in use by another system on the network.
--docker-subnet-bits 28	The number of bits in the subnet mask in the range of 8-28. You should not need to adjust this value unless there is a conflict between the default docker subnet (10.200.2.1/28) created by IRIS Focus and another subnet on your network.
--dry-run	List the steps that will be run (without running them).
--fqdn FQDN	During installation, the fully qualified domain name of the system is determined by using the **hostname --fqdn** command. Use this option if your network is set up so that **hostname --fqdn** returns the wrong name and you know the correct FQDN.
-g OR --geoserver-config-url	GeoServer configuration endpoint (default: http://localhost:24180/geoserver)
-gis-db-dump	Location of map files

Option	Description
-h or --help	Show help information.
--lightning	Allow configuration for lightning provider
--no-prompt	Fails (exits) on error without user confirmation
--offline	Disable online AlmaLinux base repository and require a local AlmaLinux base repository
--online	Allow online AlmaLinux base repository
--pg-data-dir	Use an alternative Postgres data directory location
--radar	Allow configuration for radar or lidar provider
-s	Socket server host
--skip-geoserver-installation	Do not install map server
--skip-geoserver-site-configuration	
--skip-os-version-check	Force the installation on a AlmaLinux version other than directly supported
--skip-terrain	Do not install terrain detail to the map server
--terrain-dir	Location of terrain files
--tlp TLP_ADDRESS	Address of the Total Lightning Processor
--wms -w	Basemap WMS address (default: /wms)

5.6 Installing IRIS Focus patch

If the delivery includes a separate patch file, first install the main version, and then the patch file.

The patch file is located in a separate folder on the USB stick.

In these instructions, `x.x` is the number of the version/patch.

1. Log in as `root`.
2. Copy the patch file `Vaisala_IRIS_installer-7.x.x.tar` and *`README.txt`* from the USB stick to a temporary directory.
3. Extract the tar file:

   ```
   tar -xvf Vaisala_IRIS_installer-7.x.x.tar
   ```

4. Follow the instructions in the *`README.txt`* to run the upgrade script.

5.7 Installing IRIS Focus components

The script automatically installs all necessary services, user accounts, and modules required to run IRIS Focus. The services start automatically.

For the list of IRIS Focus services and users, see IRIS Focus services and users (page 216).

1. Make sure you have an AlmaLinux server system set up, and that you have received the IRIS Focus installation files either as a USB delivery or as a download.
2. Make sure you have the IRIS Focus application installer, map data package, and terrain data package available.

   These are required because all IRIS Focus components are installed at the same time.
3. Mount the AlmaLinux ISO image. This was downloaded previously or provided on a USB stick.

   Although AlmaLinux is already set up, the IRIS Focus installer relies on some packages that are provided by the AlmaLinux repository.
4. Log in to the server as **`root`**.
5. Unpack the contents of the IRIS Focus installation file on the server, for example to the */srv/* directory.

   These files occupy approximately 40 Gb of space unpacked.
6. Navigate to the directory where you downloaded the files.
7. Launch the **`./rsw-installer`** script.

   The install script requires the following parameters:

   ```
   ./rsw-installer --offline --gis-db-dump [maps directory] --terrain-dir
   [terrain directory] -s [socket server hostname] --radar
   ```

   - **`--gis-db-dump`** – location for the map data
   - **`--terrain-dir`** – location for the terrain data
   - **`-s`** – hostname of the socket server that provides radar product data from IRIS Analysis
   - **`--radar`** – The --radar parameter is required when the IRIS Focus installation will be used to display radar or lidar data. This option should be omitted if the IRIS Focus installation will only be used to display lightning data.

   If the computer is connected to the internet, you can run the installer with the **`--online`** flag.
   This fetches any additionally required AlmaLinux packages from the Internet.

The install process can take a significant amount of time, especially as the application database is first populated with map data.
Do not abort the installation if you do not see progress in a single step for up to 1 hour.

More information

- Security settings (page 227)
- Uninstalling IRIS Focus (page 240)

5.8 Activating license

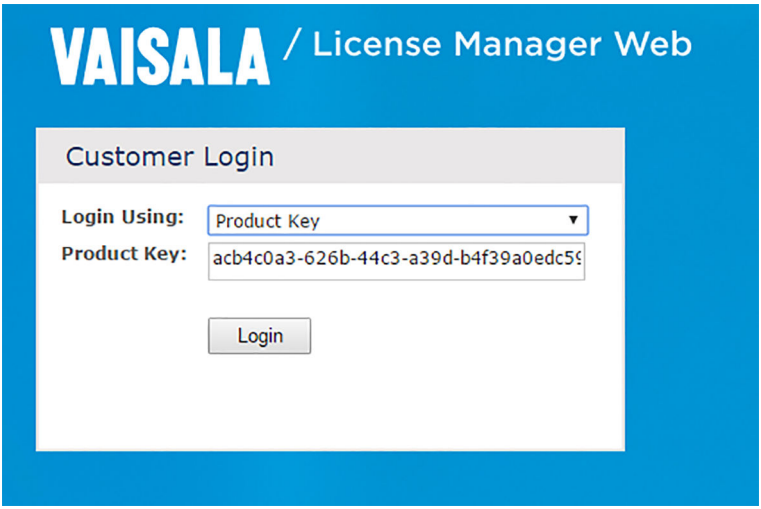
IRIS Focus provides several ways to activate the IRIS Focus software license on the server: with a USB license key, online, or offline without the USB license key.

5.8.1 Activating license – online

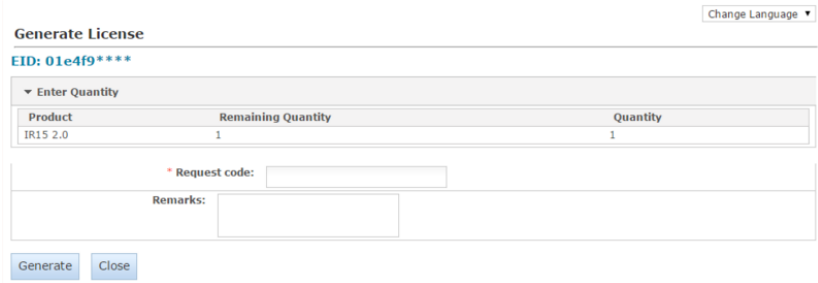
If you are using a USB license key, first insert the USB drive in the server for the license to work. See Using the USB license key (page 57).

1. Log in to the server as **`root`**.
2. Run the **`rsw-show-machine-code`** command on the IRIS Focus server to get the locking code specific to the server hardware.

3. Go to Vaisala License Manager Web at https://licensing.vaisala.com and select **Product Key** in the **Login Using** field.

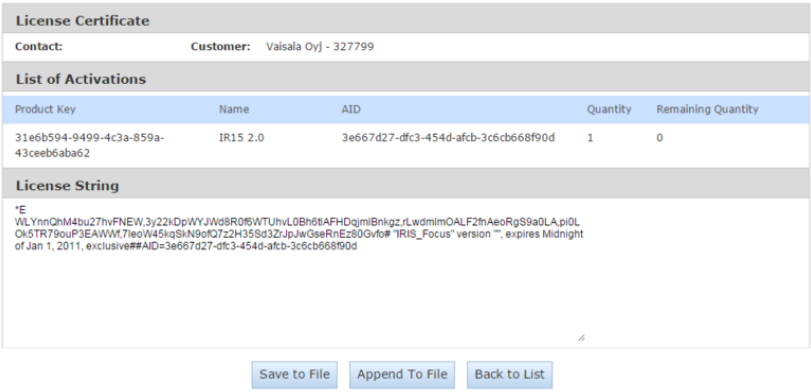


4. Enter your product key and select **Login**.
5. Enter the locking code in the **Request Code** field.



6. Select **Generate**.

   A popup window with the license string opens.



7. Select **Save to File** to save the license string to a file on disk.

   The file is saved by default with the name *lservrc*.

   Alternatively, use an SSH client to copy and paste the license string to a *.txt* file on the server.

8. Install the license with the **rsw-install-license <location-of-the-license-file>** command.
9. Restart the `vaisala-radarsw-webapp` service by typing:

```
systemctl restart vaisala-radarsw-webapp
```

10. Log in to IRIS Focus using an administrator account.
11. Select **Admin > System > Licensing Management** to view information about the license (seats, end date, and start date).

More information

- IRIS Focus licensing (page 14)

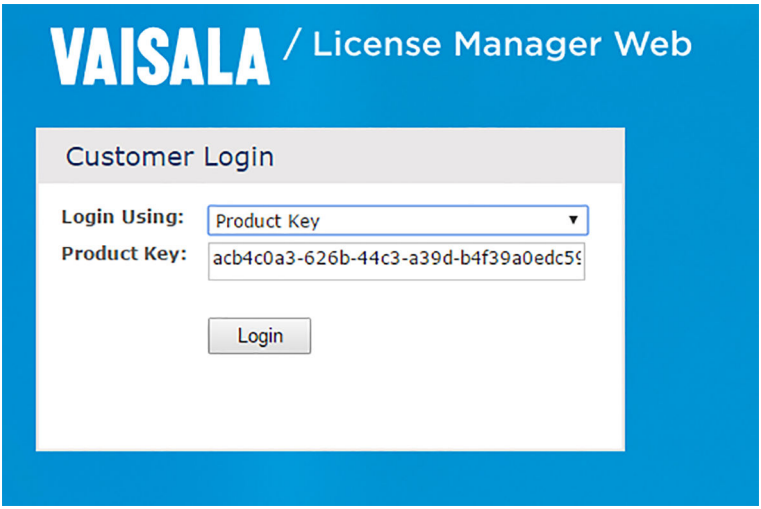
5.8.2 Activating license – offline

If the server running IRIS Focus is not connected to internet, you must activate the license by entering the IRIS Focus server locking code in **Vaisala License Manager Web** using an online computer. Then transfer the license file to the IRIS Focus server.

If you are using a USB license key, first insert the USB drive in the server for the license to work. See Using the USB license key (page 57).

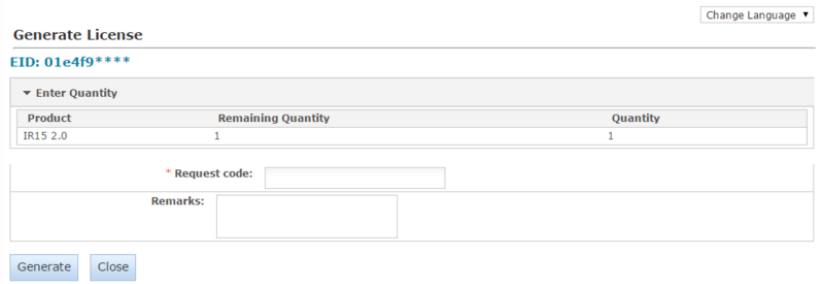
1. Run `rsw-show-machine-code > [filename]` command on the IRIS Focus server to get the product key specific to the server hardware.

   This stores the product key string in a file.
2. Copy the file to a removable media, such as a USB stick, and transfer it to the online computer.
3. Go to Vaisala License Manager Web at https://licensing.vaisala.com and select **Product Key** in the **Login Using** field.



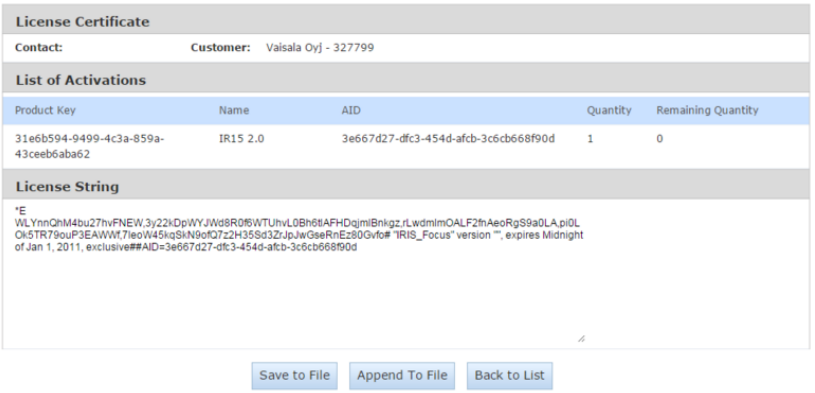
4. Enter your product key and select **Login**.

5. Enter the locking code in the **Request Code** field.



6. Select **Generate**.

   A popup window with the license string opens.



7. Select **Save to File** to save the license string to a file on disk.

   The file is saved by default with the name `lservrc`.

Alternatively, use an SSH client to copy and paste the license string to a `.txt` file on the server.

8. Copy the license file to a removable media and transport the file to the IRIS Focus server.

9. Install the license with the **`rsw-install-license <location-of-the-license-file>`** command.

More information

- IRIS Focus licensing (page 14)

5.9 Using the USB license key

The IRIS Focus license key can be provided on a USB drive. With the USB drive, you can transfer the license from one server to another.

After installing IRIS Focus, activate the license by linking the USB drive to the license file provided by Vaisala as described below.

For the license to remain active, the USB must remain in the server after completing this procedure.

If you transfer the license to another server, perform the activation procedure on the new server.

1. Insert the USB in the server machine.
2. Install the license with the following command:

   ```
   # rsw-install-license /srv/focus_license.txt
   ```

3. Restart the IRIS Focus web application:

   ```
   systemctl restart vaisala-radarsw-webapp
   ```

4. Log in to IRIS Focus using an administrator account.
5. Select **Admin > System > Licensing Management** to view information about the license (seats, end date, and start date).

5.10 Configuring licensing based on the number of radars

IRIS_Focus_Light_WR and *IRIS_Focus_Weather_Radar* licenses are valid for a defined number of weather radars. If you have more radars in the network than licenses, you need to define which radars the licenses are applied to. To do this, configure the ***`vsoweb-override.ini`*** file.

CAUTION! If you have more radars in the network than licenses, and you have not configured the list of radars to apply the licenses to, the system will not display any radar data.

1. Go to the file */etc/vaisala/radarsw/configuration/vsoweb-override.ini*.
2. Create a list of radars in numbered order.

   The format of the list entries is **radar.list.N**, where N is an integer.

Example: If you have two licenses and three radars called "MyRadarA", "MyRadarB", and "MyRadarC", and you want the license to apply to "MyRadarA" and "MyRadarC", list the radars as follows:

radar.list.1 = MyRadarA

radar.list.2 = MyRadarC

radar.list.3 = MyRadarB

5.11 Configuring licensing based on the number of lidars

IRIS_Focus_Light_WR and *IRIS_Focus_Weather_Radar* licenses are valid for a defined number of lidars. If you have more lidars in the network than licenses, you need to define which lidars the licenses are applied to. To do this, configure the *vsoweb-override.ini* file.

CAUTION! If you have more lidars in the network than licenses, and you have not configured the list of lidars to apply the licenses to, the system will not display any lidar data.

1. Go to the file */etc/vaisala/radarsw/configuration/vsoweb-override.ini*.
2. Create a list of lidars in numbered order.

   The format of the list entries is **radar.list.N**, where N is an integer.

Example: If you have two licenses and three lidars called "MyLidarA", "MyLidarB", and "MyLidarC", and you want the license to apply to "MyLidarA" and "MyLidarC", list the lidars as follows:

radar.list.1 = MyLidarA

radar.list.2 = MyLidarC

radar.list.3 = MyLidarB

5.12 Configuring IRIS for IRIS Focus

5.12.1 Configuring the firewall

IRIS Focus connects to IRIS Analysis using port 30735. By default, the firewall of the IRIS Analysis server blocks this port. When IRIS Analysis and IRIS Focus are installed on separate servers, you need to configure IRIS Analysis server to allow the connection to this port:

1. Log in to the IRIS Analysis server as **admin**.
2. Run the following commands:

```
sudo firewall-cmd --add-port=30735/tcp --permanent
sudo firewall-cmd -reload
```

5.12.2 Setting or changing the socket server

In order for IRIS Focus to configure the radar centers correctly, you need to have at least one PPI product in the socket server.

If needed, set or change the socket server:

1. Update the *vsoweb-override.ini* file with the following command:

```
/usr/vaisala/radarsw/configuration/bin/configure-vsoweb-ini -i
<socket_server_host_name>
```

2. Type the following command:

```
rsw-basemap-site-setup --socket-server <socket_server_host_name>
```

3. Restart the `vaisala-radarsw-webapp` service by typing:

```
systemctl restart vaisala-radarsw-webapp
```

5.12.3 Activating the socket server in IRIS Radar

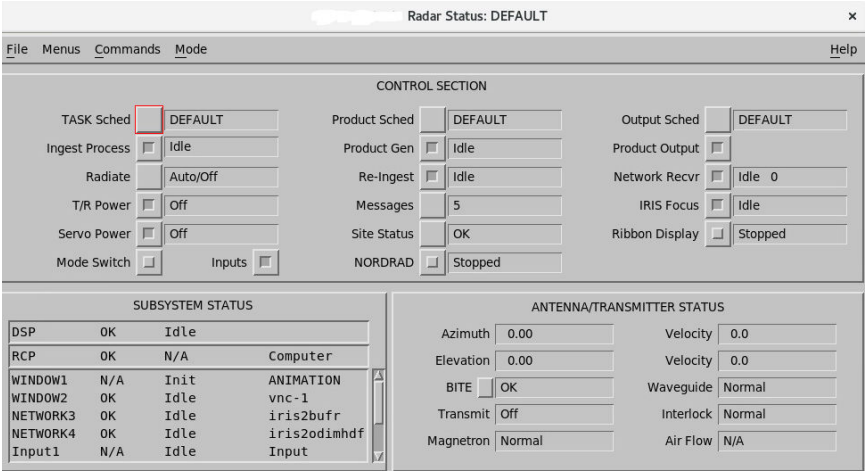


Figure 16 IRIS Radar status menu

If your system is running IRIS Focus server, you must enable the **IRIS Focus** option in IRIS Radar. For more information, see *IRIS Radar User Guide (M212926EN)*.

1. Make sure IRIS has started.
2. In IRIS Radar, select **Menus > Radar Status**.
3. Turn on the socket server by selecting the **IRIS Focus** checkbox.

   When this checkbox is selected, the field shows the status of the socket server process: **Idle**, **Running**, or **Stopped**.

5.12.4 Setting up Data Manager

The Data Manager service runs on the IRIS Focus server that receives radar scan volume data, stored in **RAW** file format, from the IRIS Analysis server and generates live radar products from the data in real-time.

During installation, IRIS Focus sets up all necessary services, databases, and user accounts for processing data. IRIS Focus features such as live products and dynamic composites require **RAW** files.

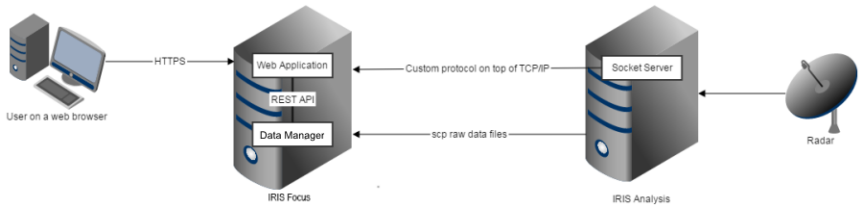


Figure 17 Radar data delivery paths

More information

- Data Manager (page 179)
- Data Manager does not work as expected (page 232)

5.12.4.1 Setting up Data Manager on IRIS Analysis server

To configure IRIS Analysis to send **RAW** files to IRIS Focus, you must set the target location on the IRIS Focus server as a network output device in IRIS Analysis.

The target location on IRIS Focus server is the following directory, which is owned by the **radardminput** user:

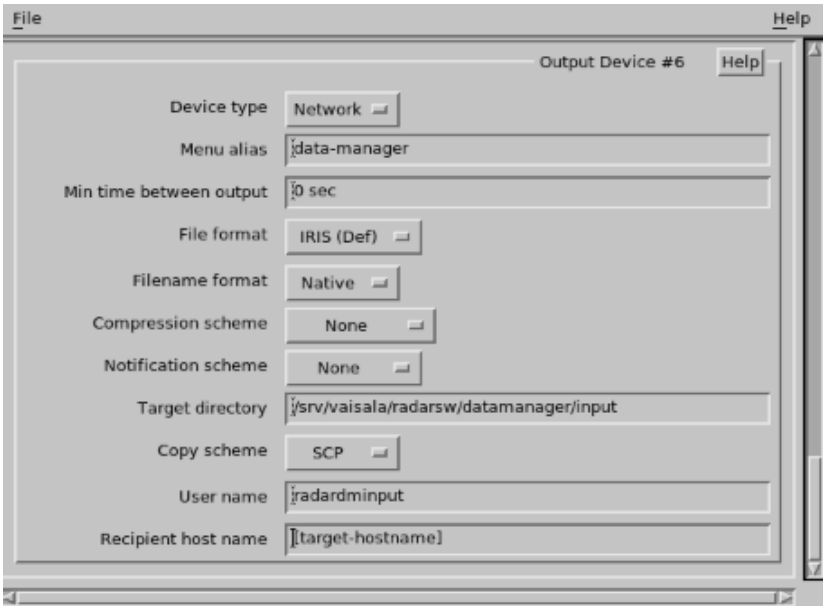
```
/srv/vaisala/radarsw/datamanager/input
```

1. Log in to the IRIS Analysis server as **radarop**.
2. In the terminal window, type: **setup&**

   The IRIS **Setup** utility opens
3. Select **Output**.
4. Create a new output device:
   a. In **Number of output devices**, increase the number of output devices by 1.
   b. Press **ENTER**.

      A new configurable output device is added to the end of the **Output Device** list.

5. In the configuration pane for the new output device, configure the new output device with the following settings:



   a. **Device type**: **Network**
   b. **Filename format**: **Native**
   c. **Target directory**: */srv/vaisala/radarsw/datamanager/input*
   d. **User name**: `radardminput`
   e. Host name: `[IRIS Focus server]`
   f. Select **File > Close**.
   g. Select **File > Save**.
   h. Select **File > Exit**.

6. Restart IRIS:

   a. Log in to the server as **root**.

   ```
   #su
       #<type password>
   ```

   b. Type:

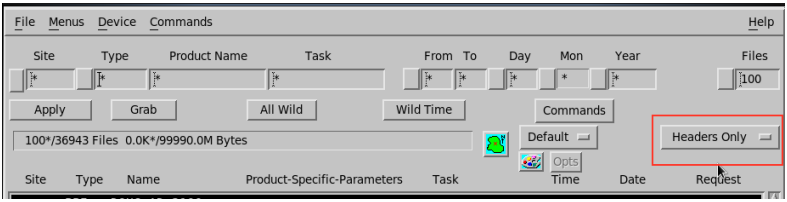
   ```
   systemctl stop iris.service
   systemctl start iris.service
   ```

7. In the terminal window, type: **`iris &`**

   a. Select **Menus > Product Output > Device**.

   b. Select the device you configured in the **Setup** utility.

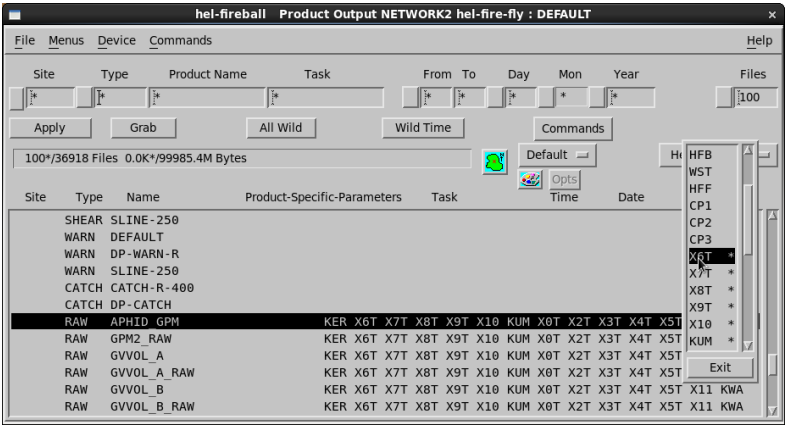
   c. In the drop down box on the far right of the window, select **Headers Only**.



   d. In the product list, select any **`RAW`** product.

   e. Right-click the far right of the product name and select a radar site.

      If needed, deselect any radar sites you do not want to include in the device configuration.



   f. Select **Apply**.

   g. Select **File > Save As**.

      Define a name for the new **Product Output** or use the **`DEFAULT`** option.

   h. Select **OK**.

   i. Select **Close**.

5.12.4.2 SSH connection for Data Manager

1. For an SSH connection between the IRIS Focus server and another server (for example, IRIS Analysis server), use the EdDSA (ed25519) scheme. If your server does not already have a file called */root/.ssh/id_ed25519.pub*, create it with the following command:

   ```
   ssh-keygen -t ed25519 -C "unique name to identify this key."
   ```

2. Copy the contents of **/root/.ssh/id_rsa.pub** to your clipboard.

   If this file does not exist in your system, generate the key in the */root/.ssh/* directory by typing **ssh-keygen -t rsa**, and pressing **ENTER** (no need to answer the questions).

3. Log in to the **root** account with the **su** command.

   When prompted, type the **root** password.

4. Launch a one-time SSH connection to the IRIS Focus server.

   ```
   ssh [IRIS Focus server IP address]
   ```

   This saves IRIS Focus server's hostname in the **known_hosts** file on the IRIS Analysis server.

5.12.4.3 Configuring IRIS Focus for transfer of WARN files

Set up SSH keys so that IRIS can send WARN files to Focus **warnreader** and allow alerts to be generated.

1. Log in to the IRIS Analysis server as **radardminput**.
2. Copy the contents of */root/<public_key_file>* to your clipboard.

   <public_key_file> may be, for example, *ssh/id_rsa.pub*.

3. Log in to the IRIS Focus server as **root**.
4. If it does not exist already, create the following *.ssh* file:

   ```
   mkdir /var/lib/warnreader/.ssh/
   vi /var/lib/warnreader/.ssh/authorized_keys
   ```

5. Paste the contents of your clipboard into: */var/lib/warnreader/.ssh/authorized_keys*

6. Type:

```
chmod 700 /var/lib/warnreader/.ssh
chmod 644 /var/lib/warnreader/.ssh/authorized_keys
chown warnreader:radarsw /var/lib/warnreader/.ssh/authorized_keys
chown warnreader:radarsw /var/lib/warnreader/.ssh/
chown warnreader:radarsw /var/lib/warnreader/
```

5.12.4.4 Setting up Data Manager on IRIS Focus server

RAW files on the IRIS Analysis server are handled by the local **root** user and **RAW** files on the IRIS Focus server by the local **radardminput** user.

You must add the IRIS Analysis **root** account's public SSH key to the IRIS Focus **radardminput** accepted keys list.

1. Log in to the IRIS Focus server as **root**.
2. If it does not exist already, create the following *.ssh* file:

```
# mkdir -m 700 /var/lib/radardminput/.ssh
# chown radardminput:radarsw /var/lib/radardminput/.ssh
```

3. Add the socket server key to the authorized SSH key store of the **radardminput** user:

   This enables file transfer from the IRIS Analysis root account to IRIS Focus **radardminput** user.

   a. Type:

```
# cd /var/lib/radardminput/.ssh
# ls
```

   b. If *authorized_keys* file already exists, type:

```
# vi authorized_keys
# rm socket-server-key
```

   Append the key you copied earlier to the file.

   c. If the *authorized_keys* file does not yet exist, add this file:

```
# vi authorized_keys
```

   Paste the key you copied earlier to your clipboard.

```
# chown radardminput:radarsw authorized_keys
# chmod 644 authorized_keys
```

4. Check that the expected on-demand product is visible in the IRIS Focus user interface.

   A data manager updater service records metadata of the files in a `PostgreSQL` database, which is accessed by the IRIS Focus web UI when it generates on-demand radar products from the data.

5.13 Verifying IRIS Focus installation

1. Check that the web user interface is running at the default HTTPS port, and the following default user accounts have been created in IRIS Focus during installation:

   - Username: `admin` / password: `admin123`
   - Username: `user` / password: `user123`

   Vaisala recommends changing the passwords after the installation.

2. Access the IRIS Focus web UI by opening a browser on the IRIS Focus server and navigating to *`https://localhost`*.

   You should see the login screen for IRIS Focus web application.

3. Log in with the default IRIS Focus user account.

   Make sure the application loads, and the map view is displayed.

4. Check that the expected on-demand product is visible in the IRIS Focus user interface.

   A data manager updater service records metadata of the files in a `PostgreSQL` database, which is accessed by the IRIS Focus web UI when it generates on-demand radar products from the data.

5. Verify that **Tracking Tool** and **Cross Section** buttons are visible in the application UI.

   This verifies that IRIS Focus features are enabled.

6. Enable the grid lines by selecting **Map Features Lat/lon grid**.

   Depending on where the map view is centered, you should see slightly distorted grid lines that are leading away from the equator. This verifies that the map projection is correct.

7. Confirm that Data Manager is running:

   a. Select **Weather Products > Add Product**.

   b. Add a new on-demand **PPI** or **CAPPI** product.

   c. Make sure you see weather data from the selected time on the screen.

6. Installation for lightning sensor network

6.1 Downloading installation packages

1. Connect to Vaisala Sigmet server (https://ftp.sigmet.vaisala.com) using a web browser or an FTP client.

   The host server allows read access for anonymous FTP connections.
   The files come in parts. Follow the steps in chapter *Verifying and joining files* to join the file parts together.
2. If using a web browser, navigate to */files/releases/Focus/<latest version>/Focus_install*, or if using an FTP client, navigate to */outgoing/releases/Focus/latest version>/Focus_install*.
3. Download the files inside the *installer* directory.

   The files are very large. Use a download tool such as CrossFTP that allows resuming downloads to fetch the files.

4. Navigate to */releases/Focus/vaisala-map-data*, and download the map files from */vaisala-iris-maps-v2* and terrain data files from */vaisala-iris-terrain-v2*.
5. If you require the AlmaLinux installation image, download it from:

   https://ftp.sigmet.vaisala.com/files/releases/AlmaLinux

   The AlmaLinux installation image is very large.

   You can skip the AlmaLinux installation image if you already have an appropriately configured AlmaLinux server installed.

6.1.1 Verifying and joining files

Each file has an associated *md5sum* file located in the same download directory.

In these instructions, `x_x` means the latest major and minor version.

After downloading the file(s), verify their integrity by checking each file's MD5 hash against the one provided at the installation site.

1. Verify the MD5 checksum values of the downloaded IRIS Focus installation files:
   - In AlmaLinux – Use the pre-installed **md5sum** command line tool:
     md5sum [filename]
   - In Microsoft Windows – Use the pre-installed **CertUtil** utility:
     certutil -hashfile [filename] MD5
2. Join the IRIS Focus installation file parts together to form a single tar file with the following command:

   ```
   cat IRIS_Focus*_part_* >| IRIS_Focus_x_x_Installer.tar
   ```

3. Get the MD5 checksum value for the tar file that you created:

   ```
   md5sum IRIS_Focus_x_x_Installer.tar
   ```

4. Verify that the MD5 checksum value matches the one shown in the *IRIS_Focus_x_x_Installer.tar.md5* file that you downloaded from https://ftp.sigmet.com
5. If you see any discrepancies in the hashes, download the mismatching file again.
6. Get the MD5 checksum value for the map files:

   ```
   md5sum vaisala-iris-maps-v2-part* | tee mymd5sums
   diff mymd5sums vaisala-iris-maps-v2.md5sum.txt && echo "Checksum verified
   ok"
   ```

7. Get the MD5 checksum value for the map and terrain files:

   ```
   md5sum vaisala-iris-terrain-v2-part* | tee mymd5sums
   diff mymd5sums vaisala-iris-terrain-v2.md5sum.txt && echo "Checksum
   verified ok"
   ```

8. Join the terrain data files together to form two zip files:

   ```
   cat vaisala-iris-terrain-v2-part* >| terrain-v2.zip
   unzip terrain-v2.zip
   rm terrain-v2.zip
   ```

   Leave the map files in parts.

6.2 Prerequisites for installation

Before installing IRIS Focus, make sure your environment meets the necessary hardware and software requirements.

More information

- IRIS Focus hardware requirements (page 19)
- Software requirements (page 19)

6.3 Installing AlmaLinux

A prerequisite for installing IRIS Focus is that AlmaLinux is installed on your intended IRIS Focus system.

IRIS Focus 7.4 has been verified to work when installed on AlmaLinux 9.3. IRIS Focus will most likely work on other releases of AlmaLinux 9, but verification of this has not been performed by Vaisala.

IRIS Focus has been tested with the security profile selection set to **None**.

If you do not have an AlmaLinux system running, select an installation image from Vaisala Sigmet server (https://ftp.sigmet.vaisala.com/files/releases/AlmaLinux/), and see instructions at Tecmint Linux Guides (https://www.tecmint.com/install-almalinux-9/) on how to perform the AlmaLinux installation.

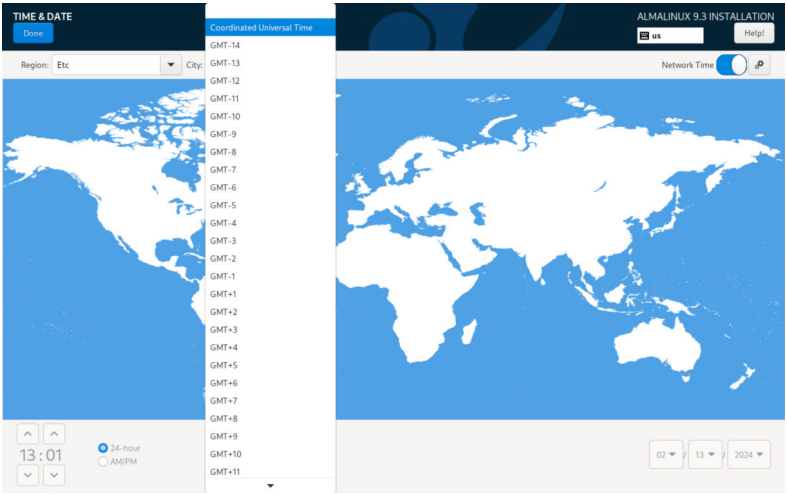
Install AlmaLinux according to the standard instructions, with the following changes.

Table 9 Recommended disk partitioning

Partition	File system type	Size
/home	XFS	50 GB
/boot	EXT4	500 MB
/boot/efi	EFI	600 MB
/var	XFS	100 GB
/	XFS	50 GB
swap	SWAP	size of RAM + 2 GB
/srv	XFS	All of the remaining disk space

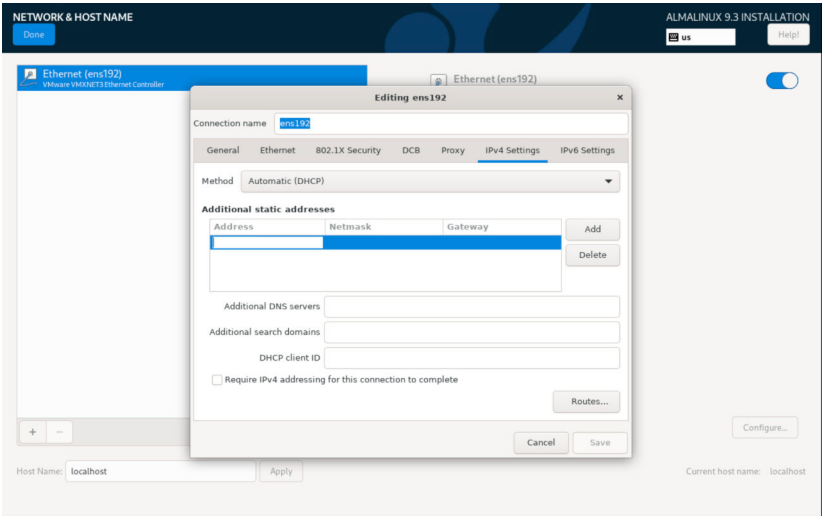
If there is only a little disk space, you can decrease the size of the */home* and */* partitions by 10-20 GB.

1. Select your installation language.
2. In **TIME & DATE**, set the system clock to Coordinated Universal Time (UTC) by choosing the following values:
   - Region: **Etc**
   - City: **Coordinated Universal Time**



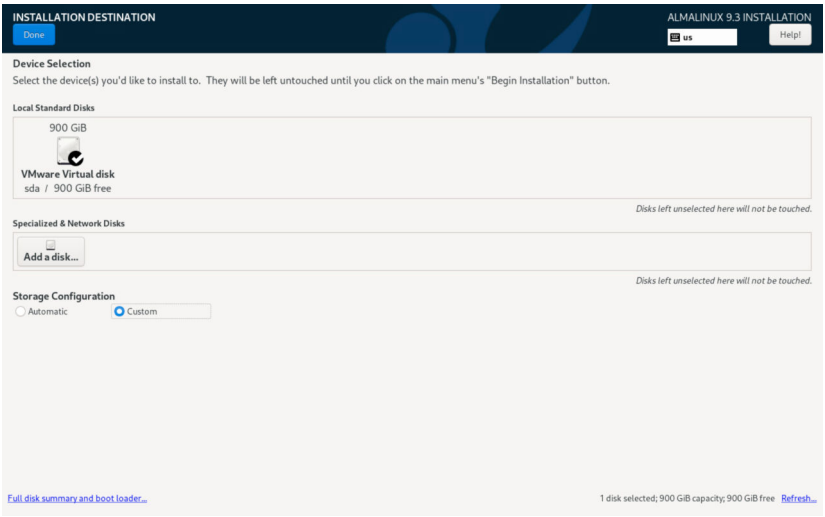
3. In **SOFTWARE SELECTION**, keep the default selection for **Base Environment Type** : **Server With GUI**.

4. In the AlmaLinux installation screen, select **Network & Host Name**.

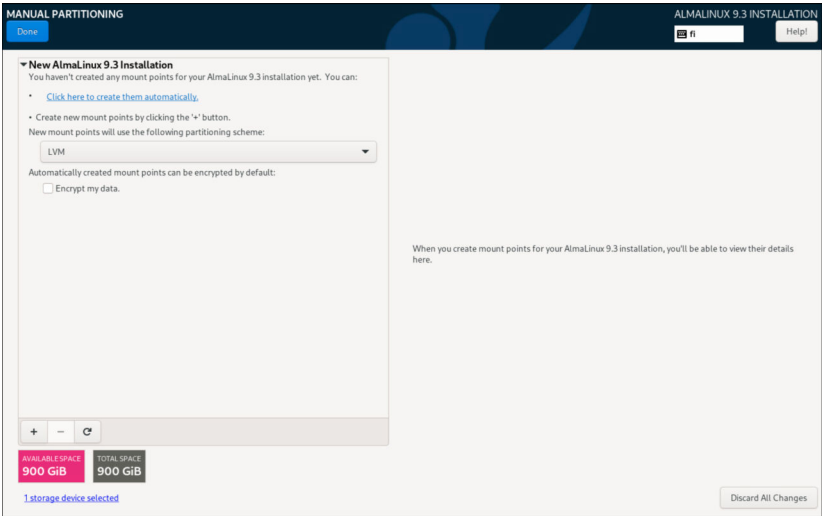


a. Turn the network **ON**.
b. Select **Configure**.
c. In the **General** tab, select **Connect automatically with priority**.
d. In the **IPv4 Settings** tab, select **Method > Manual**.
e. In the **IPv4 Settings** tab, select **Add** to add your network IP address, Netmask, Gateway, and DNS servers.
f. Select **Save**.
g. In **Host Name**, type a name for this server.
h. Select **Apply**.
i. Select **Done**.

5. In **INSTALLATION DESTINATION**, start manual partitioning:
   a. Select the hard disk.
   b. Select **Storage Configuration**: **Custom**.
   c. Select **Done**.



6. In the **Manual partitioning** window, select **Click here to create them automatically**.

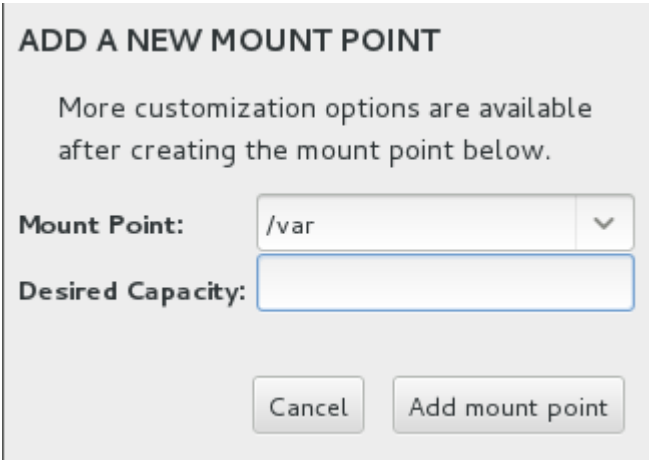


After creating the automatic partitions, you need to modify the partition manually in the next steps.

7. Modify the **/home** partition.
   a. Select the */home* partition.
   b. Under **Desired Capacity**, set the size of the home partition (*/home*) to **`50 GiB`**.
   c. Select **Update Settings**.

8. Create the */var* partition:
   a. Select the plus (**+**) icon.

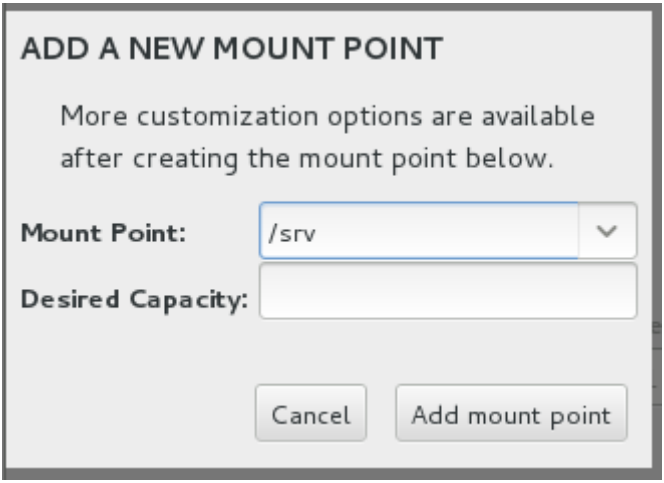
      The **ADD A NEW MOUNT POINT** dialog appears.



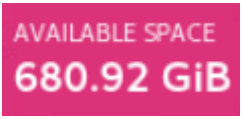
   b. In **Mount Point**, type */var*
   c. Under **Desired Capacity**, set the size of the */var* partition by typing **`100 GiB`**.
   d. Select **Add mount point**.
9. Select **/boot**.
   a. Under **Desired Capacity**, set the size of the */boot* partition by typing **`500 MiB`**.
   b. Select **Update Settings**.
10. Select **/**.
   a. Under **Desired Capacity**, set the size of the root partition (/) by typing **`100 GiB`**.
   b. Select **Update Settings**.
11. Select **swap**.
   a. Under **Desired Capacity**, set the size of the swap to the size that corresponds to RAM + 2 GB.
   b. Select **Update Settings**.

12. Create the */srv* partition:
    a. Select the plus (**+**) icon.

       The **ADD A NEW MOUNT POINT** dialog appears.

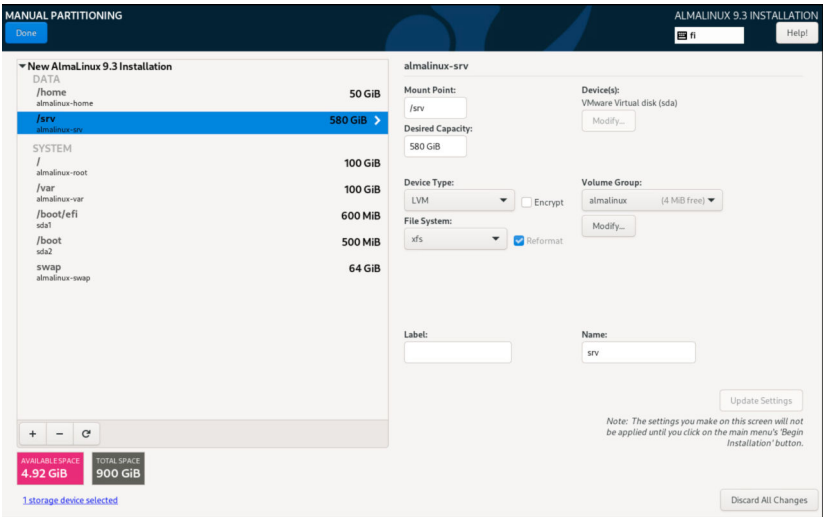


    b. In **Mount Point**, type */srv*
    c. Under **Desired Capacity**, use nearly all the available server space (indicated in the pink box) for the */srv* partition by typing, for example, **`680 GiB`**.



    d. Select **Add mount point**.
13. Select **Done**.

14. Check that the partitions are defined as follows (note that `/srv` has a different value):



15. Select **Done > Accept Changes**.

6.3.1 Setting the root password

If your system was pre-installed in Vaisala, the default password is `xxxxxxxx`.

1. Select **ROOT PASSWORD**.

   The **Root Password** window opens.

2. Enter your root password.

   Check the password strength meter. While Vaisala recommends a strong password, the software does not stop you from entering a weak one.

3. In the confirm text box, re-enter your root password.
4. In the upper left hand corner, select **Done** to return to the main configuration page.

   If your password is weak, you are prompted to select **Done** a second time.

6.3.2 Finalizing the installation

1. Select **USER CREATION**.

2. Create an account with the following properties:
   - User name: **radarop**
   - Password: **[choose password or use the default password xxxxxx]** Vaisala recommends using a non-default password.
3. In the upper left hand corner, select **Done** to return to the main configuration page.

   If your password is weak, you are prompted to select **Done** a second time.
4. Select **Begin Installation**.

   The installation will continue for a few minutes.
5. When prompted, selet **Reboot System**.
6. Select **LICENSE INFORMATION**.
7. Accept the license agreement.
8. Select **Done**.
9. Select **FINISH CONFIGURATION**.

The AlmaLinux installation is now complete. You are ready to install IRIS Focus.

6.4 Verify or override the FQDN of your server

You must determine or set the fully qualified domain name (FQDN) of the IRIS Focus server before installing the software. The FQDN must be the name that external clients will use when connecting to your IRIS Focus server. The installation assumes that this is correctly reported by the hostname command.

For example, if the final URL will be ***https://my-iris-focus.company.com/***, then the hostname command must report **iris-focus.company.com** as follows:

```
[root@my-iris-focus ~]# hostname --fqdn
my-iris-focus.company.com
[root@my-iris-focus ~]#
```

If your server does not report the correct host name, you can export an environment command to indicate what the correct host name should be. For example, if the output above had been "**my-iris**" and the correct value should have been "**my-iris-focus.company.com**", you should run the following command:

```
export HOST_FQDN=my-iris-focus.company.com
```

6.5 Installing IRIS Focus from a USB stick

The IRIS Focus installation USB contains the following file structure for the main version installation:

```
Focus_install
     vaisala-iris-maps-v2
     vaisala-iris-terrain-v2
     installer
     documentation
```

In the following instructions, **x.x** means the IRIS Focus major version and minor version number.

In the case of a patch release, the USB stick may also include an additional .tar file for the patch.

To install IRIS Focus from the USB stick, you must copy the files to the AlmaLinux server and prepare the files for installation.

6.5.1 Preparing the files on the USB stick

1. Reboot the system.
2. Log in to the server as **root**.
3. Insert the USB stick.

   If it is already plugged-in, remove and re-insert the stick.
4. In the pop-up dialog, select **Open With Files**.
5. Right-click a blank area and select **Open in Terminal**.
6. In the terminal, type **pwd** and press **ENTER**.

   The result is usually `/run/media/root/IRIS_FOCUS`.
7. Copy the *Focus_install* directory to the AlmaLinux server:

   ```
   mkdir /srv/Focus_install
   cp -r /run/media/root/IRIS_FOCUS/Focus_install/* /srv/Focus_install
   ```
8. Change to the *srv/Focus_install/installer* directory, and join the .tar file parts:

   ```
   cd /srv/Focus_install/installer
   cat IRIS_Focus_x_x_Installer_part_* >> IRIS_Focus_x_x_Installer.tar
   ```
9. To make sure that the file is now correct, run the following two commands and check that you get the same output:

   ```
   md5sum IRIS_Focus_x_x_Installer.tar
   cat IRIS_Focus_x_x_Installer.tar.md5
   ```

10. Extract the installation files into the default release directory:

```
tar -xvf IRIS_Focus_x_x_Installer.tar
```

11. Change to the */srv/Focus_install/vaisala-iris-terrain-v2* directory:

```
cd /srv/Focus_install/vaisala-iris-terrain-v2
```

a. Join the terrain file parts:

```
cat vaisala-iris-terrain-v2-part* > vaisala-iris-terrain-v2.zip
```

Leave the map files in parts.

b. Unzip the resulting terrain zip file:

```
unzip vaisala-iris-terrain-v2.zip
```

c. Remove the extra files:

```
rm -rf vaisala-iris-terrain-v2-part*
rm -rf vaisala-iris-terrain-v2.zip
```

6.5.2 Running the installation script

- If you do not want the system to reach any DNS server, use the offline installation method and the **--disable-dns** option.
- **<root application URL>** in the installation command example below corresponds to the hostname. If the hostname changes, you also need to change the **security.cors.origin.whitelist** parameter value in the *vsoweb-override.ini* file, and restart the application.
  The **cors-origin-whitelist (-cow)** switch determines the value of the **Access-Control-Allow-Origin** header. It must have the same value as the root application URL. The default value is the installation machine name.

1. Run the IRIS Focus installation script:

```
cd /srv/Focus_install/installer
./rsw-installer --online --gis-db-dump\
/srv/Focus_install/vaisala-iris-maps-v2 --terrain-dir\
/srv/Focus_install/vaisala-iris-terrain-v2\
--lightning -cow <root application URL>
```

   To limit access to port 9094 (kafka) to the TLP, if you know the IP address of your **Total Lightning Processor** (TLP), you can include `--tlp IP_ADDRESS` in the command line. If this option is omitted, port 9094 will be opened to all systems on the network.

2. Reboot the system with the following command to cleanly bring up the services:

```
reboot
```

6.5.3 Installation and configuration command options

Table 10 Installation command options

Option	Description
--admin-password	Assign a non-default admin password.
--admin-user	Assign a non-default admin user.

--broken-dns	Only use this option if you network is unable to resolve the name of your IRIS Focus system using DNS and you can not use the **--fqdn FQDN** option to specify the correct name.
	`hostname --fqdn`
	`(default: False)`
	You can use either --offline or --online installation with this option.

-c OR --config-dir	Configuration directory

Option	Description
-cow	The **cors-origin-whitelist (-cow)** switch determines the value of the **Access-Control-Allow-Origin** header. It must have the same value as the root application URL. In the installation command, **<root application URL>** corresponds to the hostname. The default value is the installation machine name. If the hostname changes, you also need to change the **security.cors.origin.whitelist** parameter value in the *vsoweb-override.ini* file, and restart the application.
--deactivate-admin	Deactivate the admin account after running this script. Not needed for standard installations.
-d OR --debug	Get additional logging from the migration/ install.
--disable-dns	Turns off DNS in the */etc/nsswitch.conf* file. Only offline installation is possible with this option.
--docker-ip-addr 10.200.2.1	The IPv4 address use by containers running in Kubernetes to communicate with containers running in Docker. You only need to specify this option if the 10.200.2.1 default value is already in use by another system on the network.
--docker-subnet-bits 28	The number of bits in the subnet mask in the range of 8-28. You should not need to adjust this value unless there is a conflict between the default docker subnet (10.200.2.1/28) created by IRIS Focus and another subnet on your network.
--dry-run	List the steps that will be run (without running them).
--fqdn FQDN	During installation, the fully qualified domain name of the system is determined by using the **hostname --fqdn** command. Use this option if your network is set up so that **hostname --fqdn** returns the wrong name and you know the correct FQDN.
-g OR --geoserver-config-url	GeoServer configuration endpoint (default: http://localhost:24180/geoserver)
-gis-db-dump	Location of map files

Option	Description
-h or --help	Show help information.
--lightning	Allow configuration for lightning provider
--no-prompt	Fails (exits) on error without user confirmation
--offline	Disable online AlmaLinux base repository and require a local AlmaLinux base repository
--online	Allow online AlmaLinux base repository
--pg-data-dir	Use an alternative Postgres data directory location
--radar	Allow configuration for radar or lidar provider
-s	Socket server host
--skip-geoserver-installation	Do not install map server
--skip-geoserver-site-configuration	
--skip-os-version-check	Force the installation on a AlmaLinux version other than directly supported
--skip-terrain	Do not install terrain detail to the map server
--terrain-dir	Location of terrain files
--tlp TLP_ADDRESS	Address of the Total Lightning Processor
--wms -w	Basemap WMS address (default: /wms)

6.6 Installing IRIS Focus patch

If the delivery includes a separate patch file, first install the main version, and then the patch file.

The patch file is located in a separate folder on the USB stick.

In these instructions, `x.x` is the number of the version/patch.

1. Log in as `root`.
2. Copy the patch file `Vaisala_IRIS_installer-7.x.x.tar` and *`README.txt`* from the USB stick to a temporary directory.
3. Extract the tar file:

   ```
   tar -xvf Vaisala_IRIS_installer-7.x.x.tar
   ```

4. Follow the instructions in the *`README.txt`* to run the upgrade script.

6.7 Installing IRIS Focus components

The script automatically installs all necessary services, user accounts, and modules required to run IRIS Focus. The services start automatically.

For the list of IRIS Focus services and users, see IRIS Focus services and users (page 216).

1. Make sure you have an AlmaLinux server system set up, and that you have received the IRIS Focus installation files either as a USB delivery or as a download.
2. Make sure you have the IRIS Focus application installer, map data package, and terrain data package available.

   These are required because all IRIS Focus components are installed at the same time.
3. Mount the AlmaLinux ISO image. This was downloaded previously or provided on a USB stick.

   Although AlmaLinux is already set up, the IRIS Focus installer relies on some packages that are provided by the AlmaLinux repository.
4. Log in to the server as **`root`**.
5. Unpack the contents of the IRIS Focus installation file on the server, for example to the */srv/* directory.

   These files occupy approximately 40 Gb of space unpacked.
6. Navigate to the directory where you downloaded the files.
7. Launch the **`./rsw-installer`** script.

   The install script requires the following parameters:

   ```
   ./rsw-installer --offline --gis-db-dump [maps directory] --terrain-dir
   [terrain directory] -s [socket server hostname] --radar
   ```

   - **`--gis-db-dump`** – location for the map data
   - **`--terrain-dir`** – location for the terrain data
   - **`-s`** – hostname of the socket server that provides radar product data from IRIS Analysis
   - **`--radar`** – The --radar parameter is required when the IRIS Focus installation will be used to display radar or lidar data. This option should be omitted if the IRIS Focus installation will only be used to display lightning data.

   If the computer is connected to the internet, you can run the installer with the **`--online`** flag.
   This fetches any additionally required AlmaLinux packages from the Internet.

The install process can take a significant amount of time, especially as the application database is first populated with map data.
Do not abort the installation if you do not see progress in a single step for up to 1 hour.

More information

- Security settings (page 227)
- Uninstalling IRIS Focus (page 240)

6.8 Installing Storm Intensity layer

To add the **Lightning Storm Intensity** WMS layer to IRIS Focus, run the following command immediately after the initial installation of IRIS Focus:

```
/usr/vaisala/radarsw/configuration/bin/configure-map -u /wms --add-ltz-wms
admin <admin password>
```

The `configure-map` script resets all the map layers, so that if you have installed any third-party WMS layers, they are deleted. Therefore, it is easiest to install the **Lightning Storm Intensity** layer right after installation with this script. However, if you choose to add this layer after having already added third party WMS layers and you want to keep them, use the following command instead of the `configure-map` script:

```
rsw-layer-add --layername "Lightning Storm Intensity" \
  --layerurl /ltzwms --layer \

"futurelightning:storm_intensity,futurelightning:storm_centroid_path_10min
_all"\
  -o 120 -rr 600 -c -m "storm,density" \
  -s "http://localhost:9973/geoserver/www/strike-intensity-tracking.sld" \
  --uiheight 70 -d -r admin -p <admin password>
```

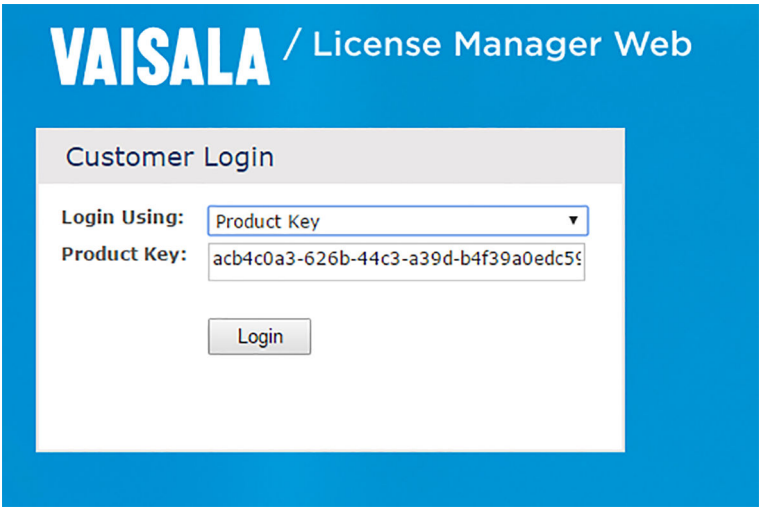
6.9 Activating license

IRIS Focus provides several ways to activate the IRIS Focus software license on the server: with a USB license key, online, or offline without the USB license key.

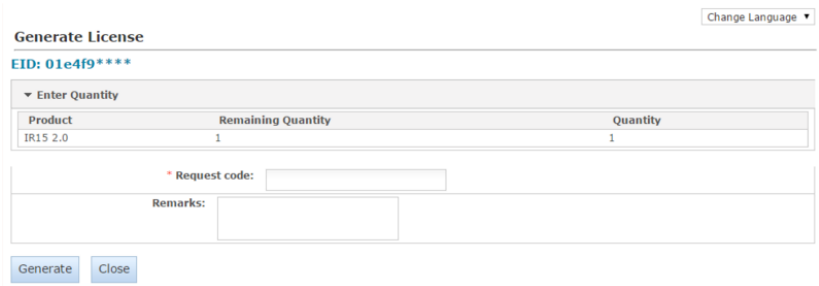
6.9.1 Activating license – online

If you are using a USB license key, first insert the USB drive in the server for the license to work. See Using the USB license key (page 57).

1. Log in to the server as **`root`**.
2. Run the **`rsw-show-machine-code`** command on the IRIS Focus server to get the locking code specific to the server hardware.
3. Go to Vaisala License Manager Web at https://licensing.vaisala.com and select **Product Key** in the **Login Using** field.

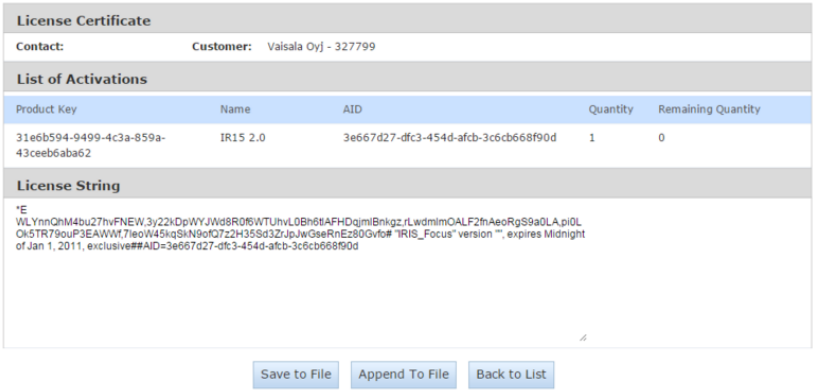


4. Enter your product key and select **Login**.
5. Enter the locking code in the **Request Code** field.



6. Select **Generate**.

   A popup window with the license string opens.



7. Select **Save to File** to save the license string to a file on disk.

   The file is saved by default with the name *`lservrc`*.

   Alternatively, use an SSH client to copy and paste the license string to a *`.txt`* file on the server.

8. Install the license with the **`rsw-install-license <location-of-the-license-file>`** command.
9. Restart the `vaisala-radarsw-webapp` service by typing:

```
systemctl restart vaisala-radarsw-webapp
```

10. Log in to IRIS Focus using an administrator account.
11. Select **Admin > System > Licensing Management** to view information about the license (seats, end date, and start date).

More information

- IRIS Focus licensing (page 14)

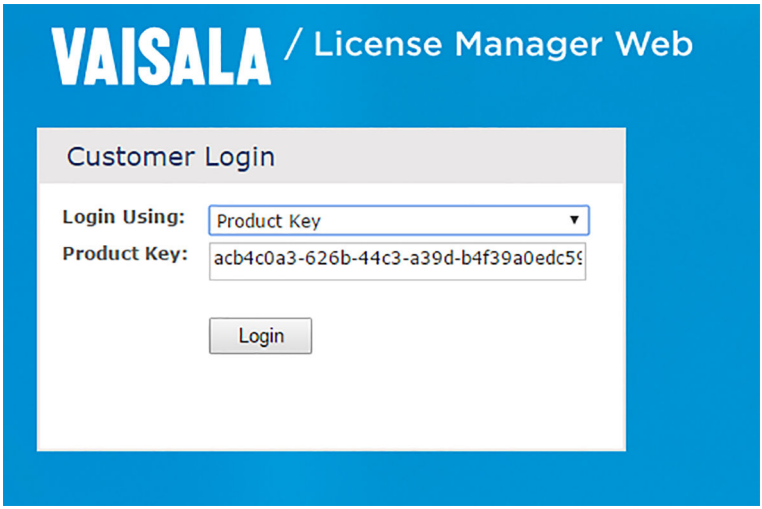
6.9.2 Activating license – offline

If the server running IRIS Focus is not connected to internet, you must activate the license by entering the IRIS Focus server locking code in **Vaisala License Manager Web** using an online computer. Then transfer the license file to the IRIS Focus server.

If you are using a USB license key, first insert the USB drive in the server for the license to work. See Using the USB license key (page 57).

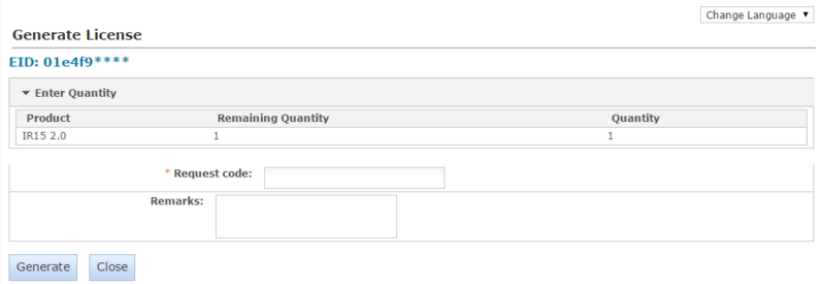
1. Run `rsw-show-machine-code > [filename]` command on the IRIS Focus server to get the product key specific to the server hardware.

   This stores the product key string in a file.
2. Copy the file to a removable media, such as a USB stick, and transfer it to the online computer.
3. Go to Vaisala License Manager Web at https://licensing.vaisala.com and select **Product Key** in the **Login Using** field.



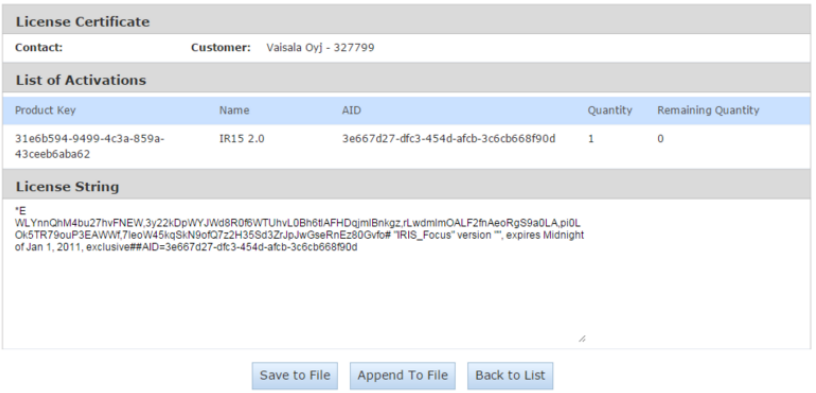
4. Enter your product key and select **Login**.

5. Enter the locking code in the **Request Code** field.



6. Select **Generate**.

   A popup window with the license string opens.



7. Select **Save to File** to save the license string to a file on disk.

   The file is saved by default with the name `lservrc`.

Alternatively, use an SSH client to copy and paste the license string to a `.txt` file on the server.

8. Copy the license file to a removable media and transport the file to the IRIS Focus server.

9. Install the license with the **`rsw-install-license <location-of-the-license-file>`** command.

More information

- IRIS Focus licensing (page 14)

6.10 Using the USB license key

The IRIS Focus license key can be provided on a USB drive. With the USB drive, you can transfer the license from one server to another.

After installing IRIS Focus, activate the license by linking the USB drive to the license file provided by Vaisala as described below.

For the license to remain active, the USB must remain in the server after completing this procedure.

If you transfer the license to another server, perform the activation procedure on the new server.

1. Insert the USB in the server machine.
2. Install the license with the following command:

```
# rsw-install-license /srv/focus_license.txt
```

3. Restart the IRIS Focus web application:

```
systemctl restart vaisala-radarsw-webapp
```

4. Log in to IRIS Focus using an administrator account.
5. Select **Admin > System > Licensing Management** to view information about the license (seats, end date, and start date).

6.11 Connecting the TLP system

Follow this procedure to add the **Total Lightning Processor** system to the IRIS Focus system to retrieve lightning data.

These steps are typically done automatically by the **`./rsw-installer`** script when you include the **`--lightning`** option. You only need to perform these steps if you did not include the **`--lightning`** option when running **`./rsw-installer`**. Otherwise, you can skip to section Configuring the TLP for IRIS Focus (page 91).

1. To enable lightning in the Web application, edit the *vsoweb-override.ini* configuration file in the */etc/vaisala/radarsw/configuration* directory. Change (or create, if not present) the **[PROVIDERS]** section to the following:

   ```
   [PROVIDERS]
   radar.enabled = true
   lightning.enabled = true
   ```

2. Restart the Web application by typing:

   ```
   systemctl restart vaisala-radarsw-webapp
   ```

3. Configure the firewall.

   The **Total Lightning Processor** connects to the Kafka data broker on port **9094** on the IRIS Focus system. If you are running the **firewalld** service, configure the firewall to allow this connection.
   Example: If the TLP system IP address is **10.55.11.2**, run the following firewall commands on the IRIS Focus system to allow **10.55.11.2** access to port **9094**:

   ```
   firewall-cmd --permanent --zone=public --add-rich-rule='rule family="ipv4"
   source address="10.55.11.2/32" port protocol="tcp" port="9094" accept'

   firewall-cmd --reload
   ```

4. Configure the **Total Lightning Processor**.

   At this point, the IRIS Focus system should be set up and ready for lightning data provided by the Total Lightning Processor. Follow the instructions in Configuring the TLP for IRIS Focus (page 91) to start the flow of lightning data from the TLP to IRIS Focus.

6.12 VHF or high data rate adjustments

If your TLP system will be providing lightning data at very high data rates, the lightning cache size of the lightning-websocket service should be increased. If you expect that your lightning data may exceed more than 100 000 events a day, you should increase the lightning cache size as indicated in section Increasing buffer capacity for lightning data (page 150).

6.13 Configuring the TLP for IRIS Focus

If you have the **Total Lightning Processor** (TLP) system that will be providing lightning data to IRIS Focus, you need to add a new service to the TLP system to push the lightning data into the kafka data broker service running on the IRIS Focus system. Your TLP must be running version 1.2.7 or later.

In the following procedure, you need the directory */opt/vai/tlp/etc*. If it does not exist, install it:

1. Log in to your TLP system as **root** user, or use the **su** or **sudo** command to gain root access.
2. Run the command:

```
dnf install -y vaisala-tlp-to-kafka
```

6.13.1 Verifying the installation of vaisala-tlp-to-kafka package

Before configuring a TLP system to send information to the Kafka data broker running on IRIS Focus, verify that the necessary software packages have been installed.

1. Log into your TLP system using the **root** user account.
2. Run the following command to make sure the necessary software packages are installed:

```
rpm -q vaisala-tlp-to-kafka || dnf install -y vaisala-tlp-to-kafka
```

6.13.2 Changing `regstatd2` report frequency

The **regstatd2** service generates a regional network health report periodically that is used to provide the **Network Health** product layer on IRIS Focus. In a default installation, the **regstatd2** service updates this report once an hour. It is recommended that you configure **regstatd2** on the TLP to produce this report at a more frequent 10-minute interval.

1. Log into your TLP system using the **vops** user account.
2. Go to the *regstatd2.cfg* file in the */opt/vai/tlp/etc* directory.
3. Edit the file to set the **updateIntervalMinutes** parameter to 10 minutes by typing:

```
updateIntervalMinutes   10
```

4. Stop the **regstatd2** service by typing:

```
lpstart stop regstatd2
```

5. Start the **regstatd2** service again by typing:

```
lpstart start regstatd2
```

6.13.3 Adding the `tlp-to-kafka` service

This instruction applies to TLP 1.2.7. If you have a later release of TLP, see the **`tlp-to-kafka man page`**.

In IRIS Focus 7, the access to the Kafka cluster is on a different port than it was in Focus 6. Access now requires an authentication token. The details are described in step 5.

The steps below require that the **`vaisala-tlp-to-kafka`** package is installed on your TLP system. If this package is missing, you can install it by logging in as the **`root`** user and running:

```
dnf install -y vaisala-tlp-to-kafka
```

1. Log into your TLP system using the **`vops`** user account.
2. Go to the *`startup.cfg`* file in the */opt/vai/tlp/etc directory*.
3. Add the following line to the file:

   ```
   core n java tlp-to-kafka -jar /opt/vai/tlp/lib/tlp-to-kafka.jar
   ```

4. Edit the *tlp-to-kafka.cfg* file in the */opt/vai/tlp/etc* directory according to how you want the lightning events to be sent to IRIS Focus:

   - If you want the lightning events sent to IRIS Focus to be composite flash events produced by the TLP, set the **lp.tokafka.smqLightning** parameter to **"smq://fdata"**.
   - If you want the lightning events sent to IRIS Focus to include the individual lightning strokes produced by the TLP, set the **lp.tokafka.smqLightning** parameter to **"smq://RLFxStrokeData"**.
   - You can use any shared memory queue of lightning data as the source for IRIS Focus. For example, if your TLP system is producing solutions from both VHF and LF based lightning sensors you can use the standard VHF event queue **"smq://sdata3d"**, the standard VHF flash queue **"smq://fdata3d"**, a merged data set **"smq://tldata or smq://wmdata"**, or some customer filtered queue. If you choose a data set that includes VHF data, you will need the *IRIS_VHF_LGT* feature enabled in your IRIS Focus license. Depending on your use cases for IRIS Focus, there may be limited use of the forwarding all of the raw VHF data points available in the **"smq://sdata3d"** shared memory queue as there can be many VHF event points for each lightning discharge.
   - If you have the **Lightning Threat Zone** feature licensed, make sure that the lightning data source you select includes LF data. The **Lightning Threat Zone** engine ignores all VHF lightning events in the data stream and only uses the LF events that is sees in the data stream.

   To set the value, type:

   ```
   lp.tokafka.smqLightning <parameter-value>
   ```

   For example:

   ```
   lp.tokafka.smqLightning "smq://RLFxStrokeData"
   ```

5. Access to the Kafka cluster requires an authentication token. The authentication token is randomly generated during the IRIS Focus 7 installation, and it is used in the password field.

   a. To find the value of this token, run the following command as **root** on the IRIS Focus system (in the example output below, the token is **L5KpD55KqxI7kGUuM0mQrmCh9QqONKI4**)

   ```
   [root@iris-focus ~]# grep kafka.*.ScramLoginModule /etc/vaisala/
   focus/k8s/vaisala-focus.yaml | head -1
               config:
   org.apache.kafka.common.security.scram.ScramLoginModule required
   username="focus-kafka" password="L5KpD55KqxI7kGUuM0mQrmCh9QqONKI4";
   [root@iris-focus ~]#
   ```

   b. When you have identified the fully qualified domain name and the authentication token for the IRIS Focus connection, go to the */opt/vai/tlp/etc* directory on the TLP system, locate the *kafka-producers.properties* file there, and change the lines as follows:

   ```
   bootstrap.servers=helsinki.rd.vaisala.com:9094
   security.protocol=SASL_PLAINTEXT
   sasl.mechanism=SCRAM-SHA-512
   sasl.jaas.config=org.apache.kafka.common.security.scram.ScramLoginModule
    required \
      username="focus-kafka" \
      password="L5KpD55KqxI7kGUuM0mQrmCh9QqONKI4";

   # How many acknowledgements are required before considering the request
   complete



   acks=all
   ```

   This example assumes that the fully qualified domain name of the IRIS Focus server is *helsinki.rd.vaisala.com* and that the randomly generated authentication token generated on the IRIS Focus server is **L5KpD55KqxI7kGUuM0mQrmCh9QqONKI4**. Make the appropriate substitutions for your installation.

6. Start the **tlp-to-kafka** service by typing:

   ```
   lpstart start tlp-to-kafka
   ```

The **tlp-to-kafka man** page provides more information on configuring and running the **tlp-to-kafka** service on a TLP system.

6.14 Verifying IRIS Focus installation

1. Check that the web user interface is running at the default HTTPS port, and the following default user accounts have been created in IRIS Focus during installation:

   - Username: `admin` / password: `admin123`
   - Username: `user` / password: `user123`

   Vaisala recommends changing the passwords after the installation.

2. Access the IRIS Focus web UI by opening a browser on the IRIS Focus server and navigating to *`https://localhost`*.

   You should see the login screen for IRIS Focus web application.

3. Log in with the default IRIS Focus user account.

   Make sure the application loads, and the map view is displayed.

4. Check that the expected on-demand product is visible in the IRIS Focus user interface.

   A data manager updater service records metadata of the files in a `PostgreSQL` database, which is accessed by the IRIS Focus web UI when it generates on-demand radar products from the data.

5. Verify that **Tracking Tool** and **Cross Section** buttons are visible in the application UI.

   This verifies that IRIS Focus features are enabled.

6. Enable the grid lines by selecting **Map Features Lat/lon grid**.

   Depending on where the map view is centered, you should see slightly distorted grid lines that are leading away from the equator. This verifies that the map projection is correct.

7. Confirm that Data Manager is running:

   a. Select **Weather Products > Add Product**.

   b. Add a new on-demand **PPI** or **CAPPI** product.

   c. Make sure you see weather data from the selected time on the screen.

7. Installation for lightning sensor network and weather radar

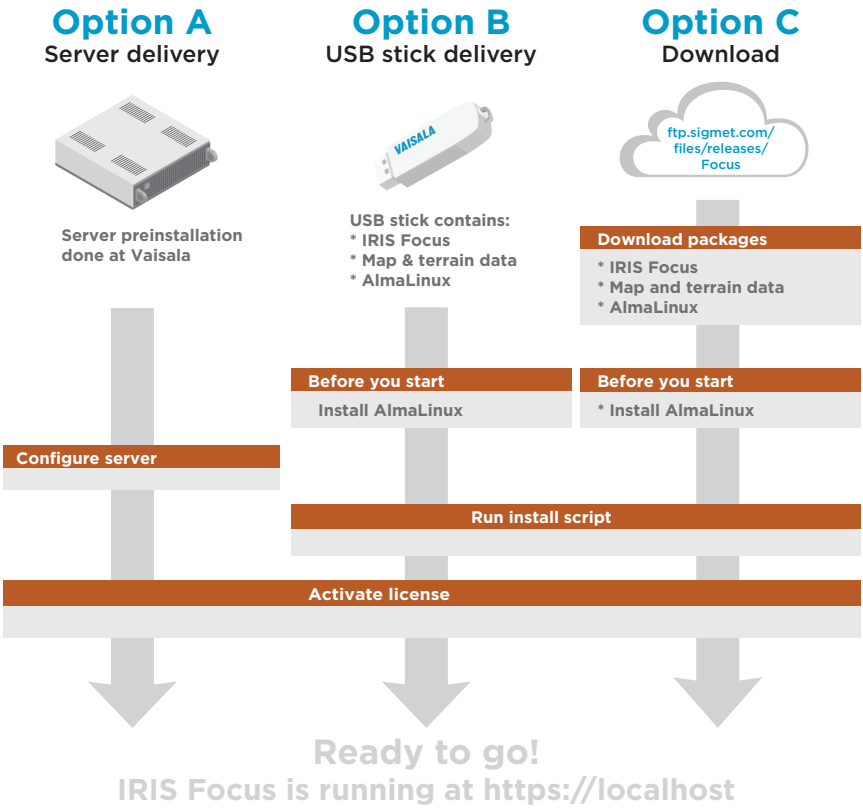


Figure 18 IRIS Focus delivery options

Option A Pre-installed system delivery from Vaisala. The "turnkey" option. Place an order and wait for delivery by Vaisala.

Option B Preconfigured USB stick containing the AlmaLinux operating system and all required files for installing IRIS Focus.

Option C Downloadable installation packages. Download the required packages to install IRIS Focus on your server.

More information

- Installation security notes (page 230)

7.1 Downloading installation packages

1. Connect to Vaisala Sigmet server (https://ftp.sigmet.vaisala.com) using a web browser or an FTP client.

   The host server allows read access for anonymous FTP connections.
   The files come in parts. Follow the steps in chapter *Verifying and joining files* to join the file parts together.
2. If using a web browser, navigate to */files/releases/Focus/<latest version>/Focus_install*, or if using an FTP client, navigate to */outgoing/releases/Focus/latest version>/Focus_install*.
3. Download the files inside the *installer* directory.

   The files are very large. Use a download tool such as CrossFTP that allows resuming downloads to fetch the files.

4. Navigate to */releases/Focus/vaisala-map-data*, and download the map files from */vaisala-iris-maps-v2* and terrain data files from */vaisala-iris-terrain-v2*.
5. If you require the AlmaLinux installation image, download it from:

   https://ftp.sigmet.vaisala.com/files/releases/AlmaLinux

   The AlmaLinux installation image is very large.

   You can skip the AlmaLinux installation image if you already have an appropriately configured AlmaLinux server installed.

7.1.1 Verifying and joining files

Each file has an associated *md5sum* file located in the same download directory.

In these instructions, `x_x` means the latest major and minor version.

After downloading the file(s), verify their integrity by checking each file's MD5 hash against the one provided at the installation site.

1. Verify the MD5 checksum values of the downloaded IRIS Focus installation files:
   - In AlmaLinux – Use the pre-installed **`md5sum`** command line tool: **`md5sum [filename]`**
   - In Microsoft Windows – Use the pre-installed **CertUtil** utility: **`certutil -hashfile [filename] MD5`**
2. Join the IRIS Focus installation file parts together to form a single tar file with the following command:

   ```
   cat IRIS_Focus*_part_* >| IRIS_Focus_x_x_Installer.tar
   ```

3. Get the MD5 checksum value for the tar file that you created:

   ```
   md5sum IRIS_Focus_x_x_Installer.tar
   ```

4. Verify that the MD5 checksum value matches the one shown in the *`IRIS_Focus_x_x_Installer.tar.md5`* file that you downloaded from https://ftp.sigmet.com
5. If you see any discrepancies in the hashes, download the mismatching file again.
6. Get the MD5 checksum value for the map files:

   ```
   md5sum vaisala-iris-maps-v2-part* | tee mymd5sums
   diff mymd5sums vaisala-iris-maps-v2.md5sum.txt && echo "Checksum verified
   ok"
   ```

7. Get the MD5 checksum value for the map and terrain files:

   ```
   md5sum vaisala-iris-terrain-v2-part* | tee mymd5sums
   diff mymd5sums vaisala-iris-terrain-v2.md5sum.txt && echo "Checksum
   verified ok"
   ```

8. Join the terrain data files together to form two zip files:

   ```
   cat vaisala-iris-terrain-v2-part* >| terrain-v2.zip
   unzip terrain-v2.zip
   rm terrain-v2.zip
   ```

   Leave the map files in parts.

7.2 Prerequisites for installation

Before installing IRIS Focus, make sure your environment meets the necessary hardware and software requirements.

More information

- IRIS Focus hardware requirements (page 19)
- Software requirements (page 19)

7.3 Installing AlmaLinux

A prerequisite for installing IRIS Focus is that AlmaLinux is installed on your intended IRIS Focus system.

IRIS Focus 7.4 has been verified to work when installed on AlmaLinux 9.3. IRIS Focus will most likely work on other releases of AlmaLinux 9, but verification of this has not been performed by Vaisala.

IRIS Focus has been tested with the security profile selection set to **None**.

If you do not have an AlmaLinux system running, select an installation image from Vaisala Sigmet server (https://ftp.sigmet.vaisala.com/files/releases/AlmaLinux/), and see instructions at Tecmint Linux Guides (https://www.tecmint.com/install-almalinux-9/) on how to perform the AlmaLinux installation.

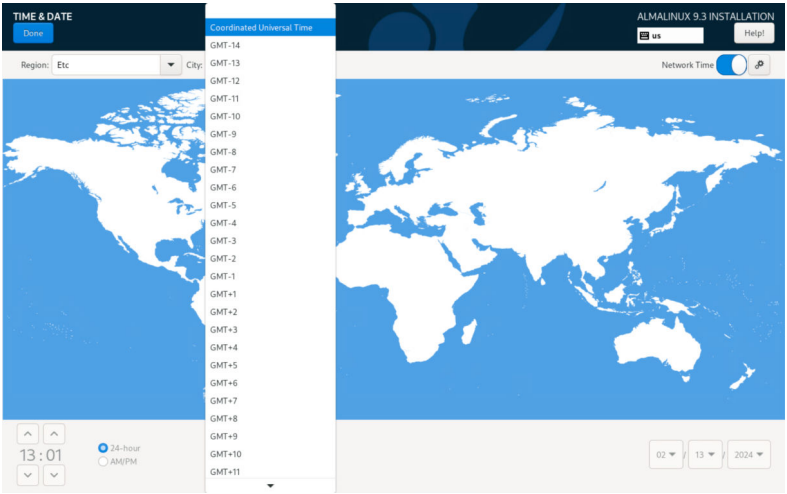
Install AlmaLinux according to the standard instructions, with the following changes.

Table 11 Recommended disk partitioning

Partition	File system type	Size
/home	XFS	50 GB
/boot	EXT4	500 MB
/boot/efi	EFI	600 MB
/var	XFS	100 GB
/	XFS	50 GB
swap	SWAP	size of RAM + 2 GB
/srv	XFS	All of the remaining disk space

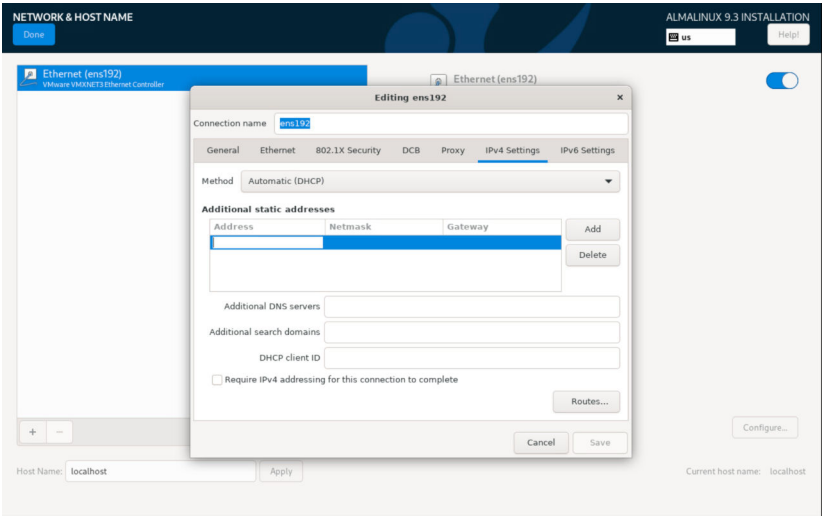
If there is only a little disk space, you can decrease the size of the */home* and */* partitions by 10-20 GB.

1. Select your installation language.
2. In **TIME & DATE**, set the system clock to Coordinated Universal Time (UTC) by choosing the following values:
   - Region: **Etc**
   - City: **Coordinated Universal Time**



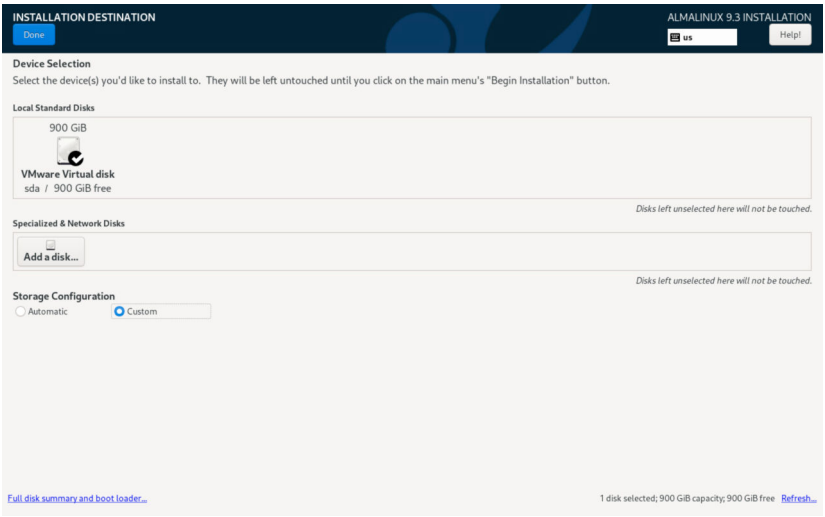
3. In **SOFTWARE SELECTION**, keep the default selection for **Base Environment Type** : **Server With GUI**.

4. In the AlmaLinux installation screen, select **Network & Host Name**.

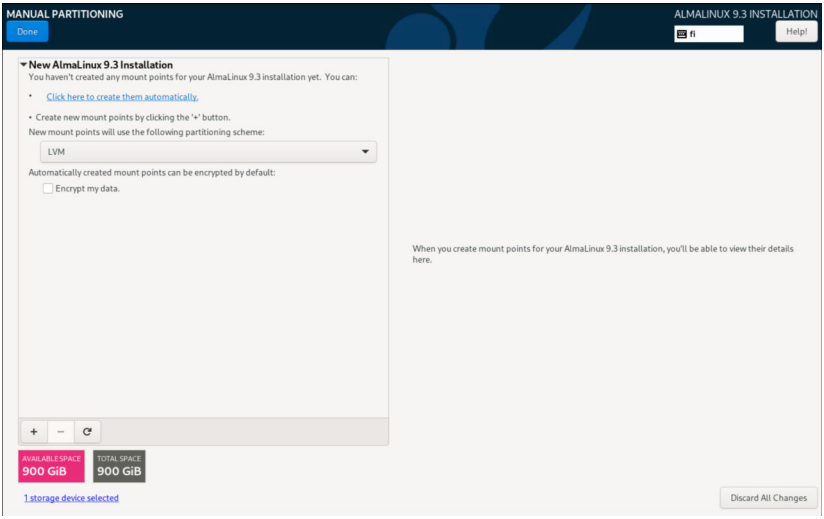


a. Turn the network **ON**.
b. Select **Configure**.
c. In the **General** tab, select **Connect automatically with priority**.
d. In the **IPv4 Settings** tab, select **Method > Manual**.
e. In the **IPv4 Settings** tab, select **Add** to add your network IP address, Netmask, Gateway, and DNS servers.
f. Select **Save**.
g. In **Host Name**, type a name for this server.
h. Select **Apply**.
i. Select **Done**.

5. In **INSTALLATION DESTINATION**, start manual partitioning:
   a. Select the hard disk.
   b. Select **Storage Configuration**: **Custom**.
   c. Select **Done**.



6. In the **Manual partitioning** window, select **Click here to create them automatically**.

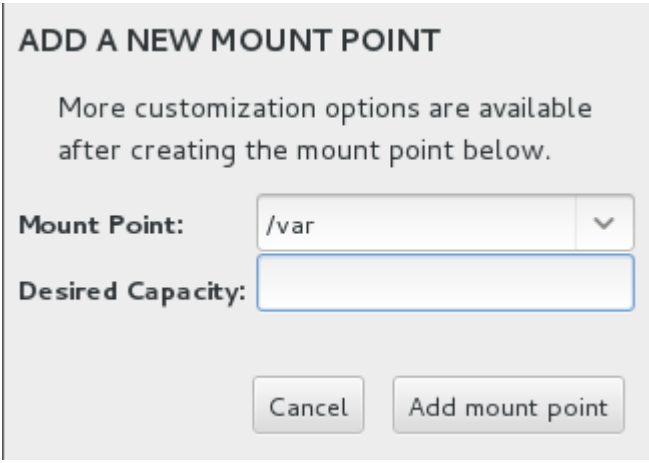


After creating the automatic partitions, you need to modify the partition manually in the next steps.

7. Modify the **/home** partition.
   a. Select the */home* partition.
   b. Under **Desired Capacity**, set the size of the home partition (*/home*) to **`50 GiB`**.
   c. Select **Update Settings**.

8. Create the */var* partition:
   a. Select the plus (**+**) icon.

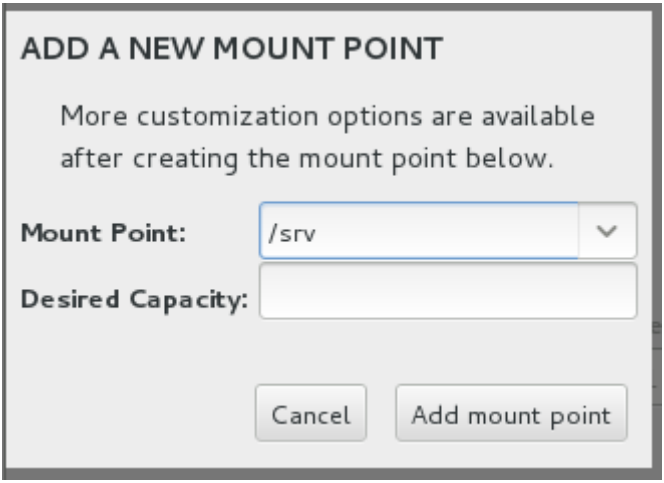
      The **ADD A NEW MOUNT POINT** dialog appears.



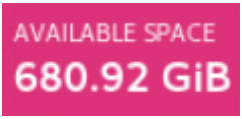
   b. In **Mount Point**, type */var*
   c. Under **Desired Capacity**, set the size of the */var* partition by typing **`100 GiB`**.
   d. Select **Add mount point**.
9. Select **/boot**.
   a. Under **Desired Capacity**, set the size of the */boot* partition by typing **`500 MiB`**.
   b. Select **Update Settings**.
10. Select **/**.
   a. Under **Desired Capacity**, set the size of the root partition (/) by typing **`100 GiB`**.
   b. Select **Update Settings**.
11. Select **swap**.
   a. Under **Desired Capacity**, set the size of the swap to the size that corresponds to RAM + 2 GB.
   b. Select **Update Settings**.

12. Create the */srv* partition:
    a. Select the plus (**+**) icon.

       The **ADD A NEW MOUNT POINT** dialog appears.

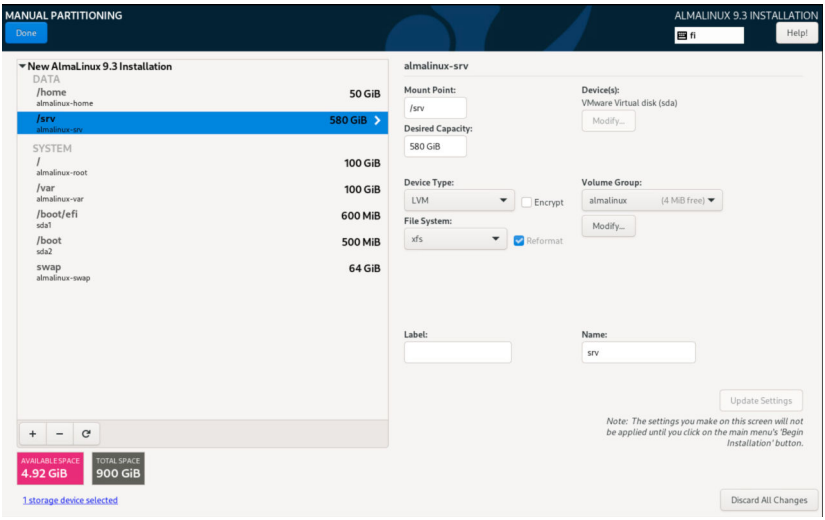


    b. In **Mount Point**, type */srv*
    c. Under **Desired Capacity**, use nearly all the available server space (indicated in the pink box) for the */srv* partition by typing, for example, **`680 GiB`**.



    d. Select **Add mount point**.
13. Select **Done**.

14. Check that the partitions are defined as follows (note that `/srv` has a different value):



15. Select **Done > Accept Changes**.

7.3.1 Setting the root password

If your system was pre-installed in Vaisala, the default password is `xxxxxxxx`.

1. Select **ROOT PASSWORD**.

   The **Root Password** window opens.

2. Enter your root password.

   Check the password strength meter. While Vaisala recommends a strong password, the software does not stop you from entering a weak one.

3. In the confirm text box, re-enter your root password.
4. In the upper left hand corner, select **Done** to return to the main configuration page.

   If your password is weak, you are prompted to select **Done** a second time.

7.3.2 Finalizing the installation

1. Select **USER CREATION**.

2. Create an account with the following properties:
   - User name: **radarop**
   - Password: **[choose password or use the default password xxxxxx]** Vaisala recommends using a non-default password.
3. In the upper left hand corner, select **Done** to return to the main configuration page.

   If your password is weak, you are prompted to select **Done** a second time.
4. Select **Begin Installation**.

   The installation will continue for a few minutes.
5. When prompted, selet **Reboot System**.
6. Select **LICENSE INFORMATION**.
7. Accept the license agreement.
8. Select **Done**.
9. Select **FINISH CONFIGURATION**.

The AlmaLinux installation is now complete. You are ready to install IRIS Focus.

7.4 Verify or override the FQDN of your server

You must determine or set the fully qualified domain name (FQDN) of the IRIS Focus server before installing the software. The FQDN must be the name that external clients will use when connecting to your IRIS Focus server. The installation assumes that this is correctly reported by the hostname command.

For example, if the final URL will be ***https://my-iris-focus.company.com/***, then the hostname command must report **iris-focus.company.com** as follows:

```
[root@my-iris-focus ~]# hostname --fqdn
my-iris-focus.company.com
[root@my-iris-focus ~]#
```

If your server does not report the correct host name, you can export an environment command to indicate what the correct host name should be. For example, if the output above had been "**my-iris**" and the correct value should have been "**my-iris-focus.company.com**", you should run the following command:

```
export HOST_FQDN=my-iris-focus.company.com
```

7.5 Installing IRIS Focus from a USB stick

In these instructions, **x.x** is the number of the version/patch.

The IRIS Focus installation USB contains the following file structure for the main version installation:

```
Focus_install
     vaisala-iris-maps-v2
     vaisala-iris-terrain-v2
     installer
     documentation
```

In the case of a patch release, the USB stick may also include an additional .tar file for the patch.

To install IRIS Focus from the USB stick, you must copy the files to the AlmaLinux server and prepare the files for installation.

7.5.1 Preparing the files on the USB stick

1. Reboot the system.
2. Log in to the server as **`root`**.
3. Insert the USB stick.

   If it is already plugged-in, remove and re-insert the stick.
4. In the pop-up dialog, select **Open With Files**.
5. Right-click a blank area and select **Open in Terminal**.
6. In the terminal, type **pwd** and press **ENTER**.

   The result is usually `/run/media/root/IRIS_FOCUS`.
7. Copy the *`Focus_install`* directory to the AlmaLinux server:

   ```
   mkdir /srv/Focus_install
   cp –r /run/media/root/IRIS_FOCUS/Focus_install/* /srv/Focus_install
   ```

8. Change to the *`srv/Focus_install/installer`* directory, and join the .tar file parts:

   ```
   cd /srv/Focus_install/installer
   cat IRIS_Focus_x_x_Installer_part_* >> IRIS_Focus_x_x_Installer.tar
   ```

9. To make sure that the file is now correct, run the following two commands and check that you get the same output:

   ```
   md5sum IRIS_Focus_x_x_Installer.tar
   cat IRIS_Focus_x_x_Installer.tar.md5
   ```

10. Extract the installation files into the default release directory:

    ```
    tar –xvf IRIS_Focus_x_x_Installer.tar
    ```

11. Change to the */srv/Focus_install/vaisala-iris-terrain-v2* directory:

```
cd /srv/Focus_install/vaisala-iris-terrain-v2
```

a. Join the terrain file parts:

```
cat vaisala-iris-terrain-v2-part* > vaisala-iris-terrain-v2.zip
```

Leave the map files in parts.

b. Unzip the resulting terrain zip file:

```
unzip vaisala-iris-terrain-v2.zip
```

c. Remove the extra files:

```
rm -rf vaisala-iris-terrain-v2-part*
rm -rf vaisala-iris-terrain-v2.zip
```

7.5.2 Running the installation script radar and lightning

- If you do not want the system to reach any DNS server, use the offline installation method and the **--disable-dns** option.
- **<root application URL>** in the installation command example below corresponds to the hostname. If the hostname changes, you also need to change the **security.cors.origin.whitelist** parameter value in the *vsoweb-override.ini* file, and restart the application.
  The **cors-origin-whitelist (-cow)** switch determines the value of the **Access-Control-Allow-Origin** header. It must have the same value as the root application URL. The default value is the installation machine name.

1. Run the IRIS Focus installation scripts:

```
cd /srv/Focus_install/installer
./rsw-installer --online --gis-db-dump\
/srv/Focus_install/vaisala-iris-maps-v2 --terrain-dir\
/srv/Focus_install/vaisala-iris-terrain-v2\
--radar -s <hostname or IP of IRIS Analysis socket server>\
--lightning -cow <root application URL>
```

2. Reboot the system with the following command to cleanly bring up the services:

```
reboot
```

7.5.3 Installation and configuration command options

Table 12 Installation command options

Option	Description
--admin-password	Assign a non-default admin password.
--admin-user	Assign a non-default admin user.

--broken-dns	Only use this option if you network is unable to resolve the name of your IRIS Focus system using DNS and you can not use the **--fqdn FQDN** option to specify the correct name.
	`hostname --fqdn`
	`(default: False)`
	You can use either --offline or --online installation with this option.

-c OR --config-dir	Configuration directory
-cow	The `cors-origin-whitelist (-cow)` switch determines the value of the `Access-Control-Allow-Origin` header. It must have the same value as the root application URL. In the installation command, `<root application URL>` corresponds to the hostname. The default value is the installation machine name. If the hostname changes, you also need to change the `security.cors.origin.whitelist` parameter value in the *`vsoweb-override.ini`* file, and restart the application.
--deactivate-admin	Deactivate the admin account after running this script. Not needed for standard installations.
-d OR --debug	Get additional logging from the migration/ install.

Option	Description
--disable-dns	Turns off DNS in the */etc/nsswitch.conf* file. Only offline installation is possible with this option.
--docker-ip-addr 10.200.2.1	The IPv4 address use by containers running in Kubernetes to communicate with containers running in Docker. You only need to specify this option if the 10.200.2.1 default value is already in use by another system on the network.
--docker-subnet-bits 28	The number of bits in the subnet mask in the range of 8-28. You should not need to adjust this value unless there is a conflict between the default docker subnet (10.200.2.1/28) created by IRIS Focus and another subnet on your network.
--dry-run	List the steps that will be run (without running them).
--fqdn FQDN	During installation, the fully qualified domain name of the system is determined by using the **hostname --fqdn** command. Use this option if your network is set up so that **hostname --fqdn** returns the wrong name and you know the correct FQDN.
-g OR --geoserver-config-url	GeoServer configuration endpoint (default: http://localhost:24180/geoserver)
-gis-db-dump	Location of map files
-h or --help	Show help information.
--lightning	Allow configuration for lightning provider
--no-prompt	Fails (exits) on error without user confirmation
--offline	Disable online AlmaLinux base repository and require a local AlmaLinux base repository
--online	Allow online AlmaLinux base repository
--pg-data-dir	Use an alternative Postgres data directory location
--radar	Allow configuration for radar or lidar provider
-s	Socket server host
--skip-geoserver-installation	Do not install map server
--skip-geoserver-site-configuration	
--skip-os-version-check	Force the installation on a AlmaLinux version other than directly supported

Option	Description
--skip-terrain	Do not install terrain detail to the map server
--terrain-dir	Location of terrain files
--tlp TLP_ADDRESS	Address of the Total Lightning Processor
--wms -w	Basemap WMS address (default: /wms)

7.6 Installing IRIS Focus patch

If the delivery includes a separate patch file, first install the main version, and then the patch file.

The patch file is located in a separate folder on the USB stick.

In these instructions, **x.x** is the number of the version/patch.

1. Log in as **root**.
2. Copy the patch file **Vaisala_IRIS_installer-7.x.x.tar** and ***README.txt*** from the USB stick to a temporary directory.
3. Extract the tar file:

   ```
   tar -xvf Vaisala_IRIS_installer-7.x.x.tar
   ```

4. Follow the instructions in the ***README.txt*** to run the upgrade script.

7.7 Installing IRIS Focus components

The script automatically installs all necessary services, user accounts, and modules required to run IRIS Focus. The services start automatically.

For the list of IRIS Focus services and users, see IRIS Focus services and users (page 216).

1. Make sure you have an AlmaLinux server system set up, and that you have received the IRIS Focus installation files either as a USB delivery or as a download.
2. Make sure you have the IRIS Focus application installer, map data package, and terrain data package available.

   These are required because all IRIS Focus components are installed at the same time.
3. Mount the AlmaLinux ISO image. This was downloaded previously or provided on a USB stick.

   Although AlmaLinux is already set up, the IRIS Focus installer relies on some packages that are provided by the AlmaLinux repository.
4. Log in to the server as **root**.

5. Unpack the contents of the IRIS Focus installation file on the server, for example to the */root/IRIS* directory.

   These files occupy approximately 40 Gb of space unpacked.

6. Navigate to the directory where you downloaded the files.
7. Launch the **`./rsw-installer`** script.

   The install script requires the following parameters when it will be connected to an IRIS Analysis system and running a local map server to provide map tiles:

   ```
   ./rsw-installer --offline --gis-db-dump [maps directory] --terrain-dir
   [terrain directory] -s [socket server hostname] --radar --lightning
   ```

   - **`--gis-db-dump`** – location for the map data
   - **`--terrain-dir`** – location for the terrain data
   - **`--radar`** – use this parameter if you are connecting weather radars to IRIS Focus
   - **`-s`** – hostname of the socket server that provides radar product data from IRIS Analysis
   - **`--lightning`** – use this parameter if you are connecting a Total Lightning Processor system to IRIS Focus.

   If the computer is connected to the internet, you can run the installer with the **`--online`** flag.
   This fetches any additionally required AlmaLinux packages from the internet.

The install process can take a significant amount of time, especially as the application database is first populated with map data.
Do not abort the installation if you do not see progress in a single step for up to 1 hour.

7.8 Installing Storm Intensity layer

To add the **Lightning Storm Intensity** WMS layer to IRIS Focus, run the following command immediately after the initial installation of IRIS Focus:

```
/usr/vaisala/radarsw/configuration/bin/configure-map -u /wms --add-ltz-wms
admin <admin password>
```

The **`configure-map`** script resets all the map layers, so that if you have installed any third-party WMS layers, they are deleted. Therefore, it is easiest to install the **Lightning Storm Intensity** layer right after installation with this script. However, if you choose to add this layer after having already added third party WMS layers and you want to keep them, use the following command instead of the **`configure-map`** script:

```
rsw-layer-add --layername "Lightning Storm Intensity" \
  --layerurl /ltzwms --layer \

"futurelightning:storm_intensity,futurelightning:storm_centroid_path_10min
_all"\
  -o 120 -rr 600 -c -m "storm,density" \
  -s "http://localhost:9973/geoserver/www/strike-intensity-tracking.sld" \
  --uiheight 70 -d -r admin -p <admin password>
```

7.9 Activating license

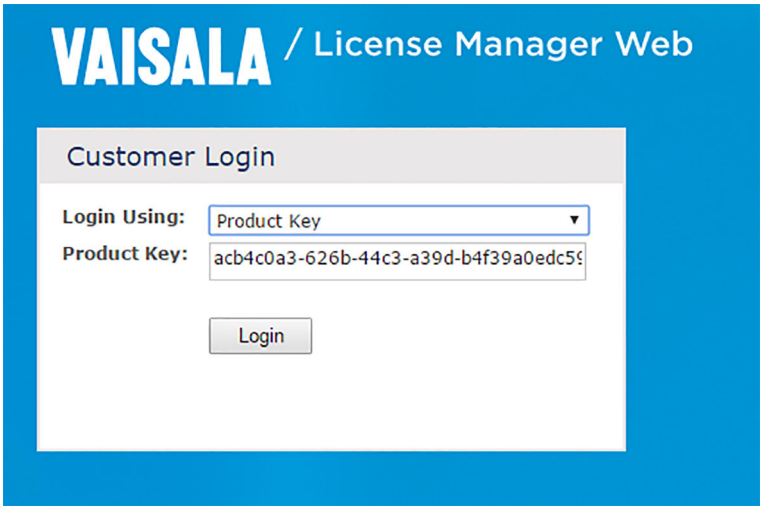
IRIS Focus provides several ways to activate the IRIS Focus software license on the server: with a USB license key, online, or offline without the USB license key.

7.9.1 Activating license – online

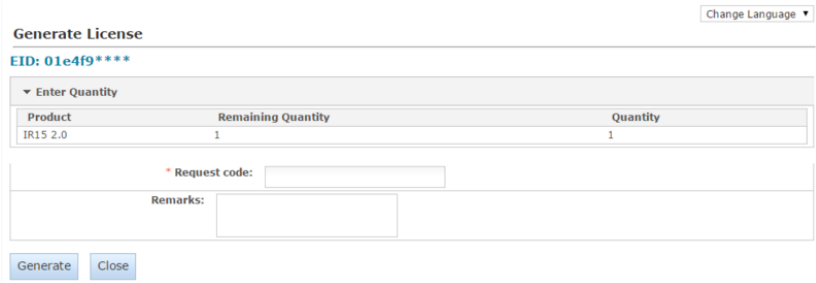
If you are using a USB license key, first insert the USB drive in the server for the license to work. See Using the USB license key (page 57).

1. Log in to the server as **root**.
2. Run the **rsw-show-machine-code** command on the IRIS Focus server to get the locking code specific to the server hardware.

3. Go to Vaisala License Manager Web at https://licensing.vaisala.com and select **Product Key** in the **Login Using** field.

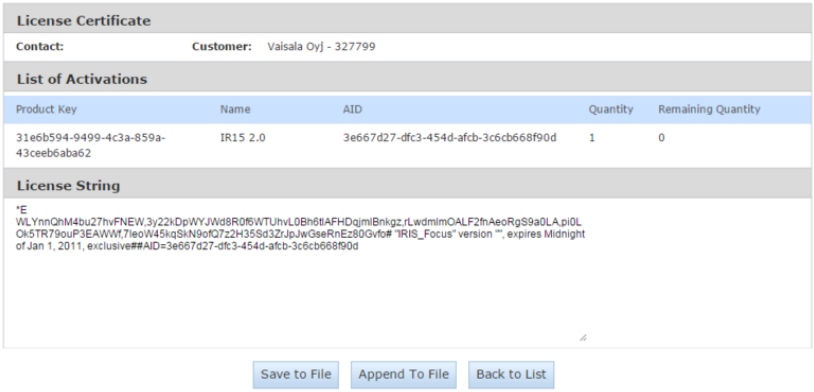


4. Enter your product key and select **Login**.
5. Enter the locking code in the **Request Code** field.



6. Select **Generate**.

   A popup window with the license string opens.



7. Select **Save to File** to save the license string to a file on disk.

   The file is saved by default with the name *`lservrc`*.

   Alternatively, use an SSH client to copy and paste the license string to a *`.txt`* file on the server.

8. Install the license with the **`rsw-install-license <location-of-the-license-file>`** command.
9. Restart the `vaisala-radarsw-webapp` service by typing:

```
systemctl restart vaisala-radarsw-webapp
```

10. Log in to IRIS Focus using an administrator account.
11. Select **Admin > System > Licensing Management** to view information about the license (seats, end date, and start date).

More information

- IRIS Focus licensing (page 14)

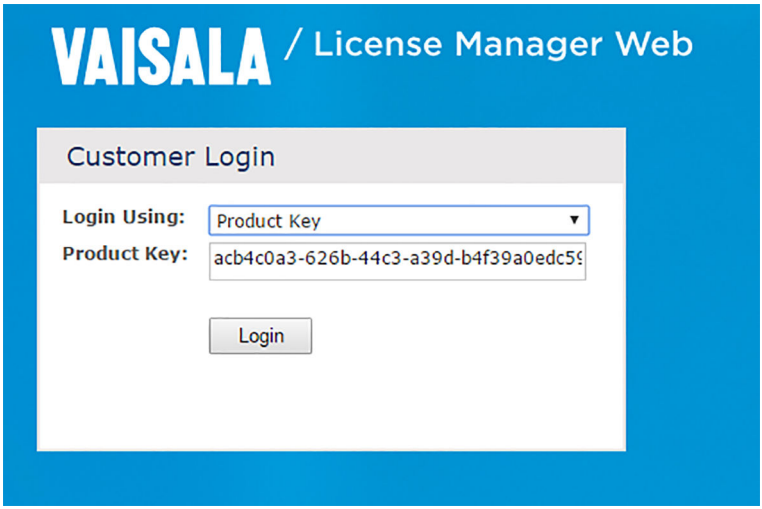
7.9.2 Activating license – offline

If the server running IRIS Focus is not connected to internet, you must activate the license by entering the IRIS Focus server locking code in **Vaisala License Manager Web** using an online computer. Then transfer the license file to the IRIS Focus server.

If you are using a USB license key, first insert the USB drive in the server for the license to work. See Using the USB license key (page 57).

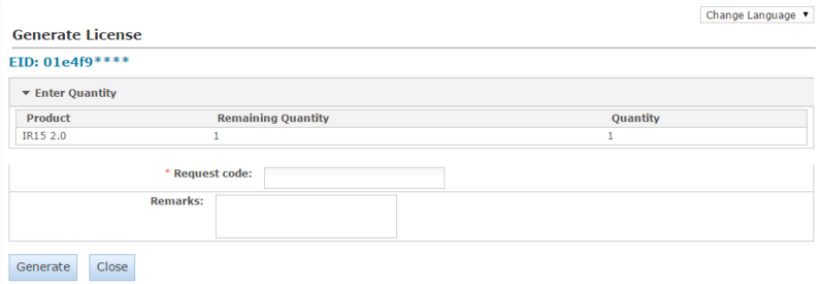
1. Run `rsw-show-machine-code > [filename]` command on the IRIS Focus server to get the product key specific to the server hardware.

   This stores the product key string in a file.
2. Copy the file to a removable media, such as a USB stick, and transfer it to the online computer.
3. Go to Vaisala License Manager Web at https://licensing.vaisala.com and select **Product Key** in the **Login Using** field.



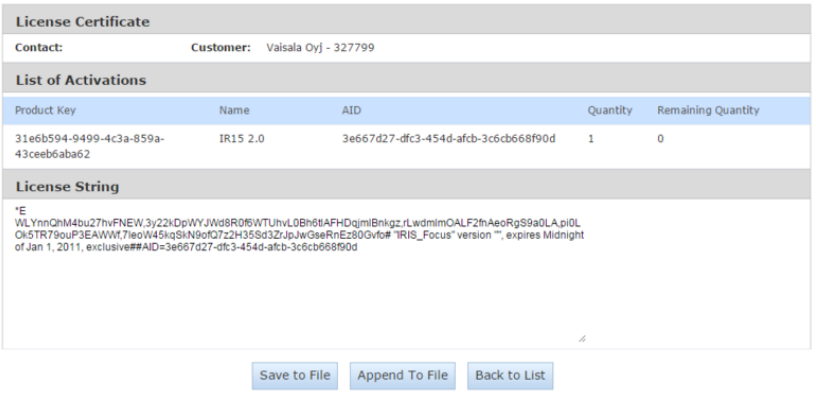
4. Enter your product key and select **Login**.

5. Enter the locking code in the **Request Code** field.



6. Select **Generate**.

   A popup window with the license string opens.



7. Select **Save to File** to save the license string to a file on disk.

   The file is saved by default with the name `lservrc`.

Alternatively, use an SSH client to copy and paste the license string to a `.txt` file on the server.

8. Copy the license file to a removable media and transport the file to the IRIS Focus server.

9. Install the license with the **`rsw-install-license <location-of-the-license-file>`** command.

More information

- IRIS Focus licensing (page 14)

7.10 Using the USB license key

The IRIS Focus license key can be provided on a USB drive. With the USB drive, you can transfer the license from one server to another.

After installing IRIS Focus, activate the license by linking the USB drive to the license file provided by Vaisala as described below.

For the license to remain active, the USB must remain in the server after completing this procedure.

If you transfer the license to another server, perform the activation procedure on the new server.

1. Insert the USB in the server machine.
2. Install the license with the following command:

   ```
   # rsw-install-license /srv/focus_license.txt
   ```

3. Restart the IRIS Focus web application:

   ```
   systemctl restart vaisala-radarsw-webapp
   ```

4. Log in to IRIS Focus using an administrator account.
5. Select **Admin > System > Licensing Management** to view information about the license (seats, end date, and start date).

7.11 Configuring licensing based on the number of radars

IRIS_Focus_Light_WR and *IRIS_Focus_Weather_Radar* licenses are valid for a defined number of weather radars. If you have more radars in the network than licenses, you need to define which radars the licenses are applied to. To do this, configure the ***`vsoweb-override.ini`*** file.

CAUTION! If you have more radars in the network than licenses, and you have not configured the list of radars to apply the licenses to, the system will not display any radar data.

1. Go to the file */etc/vaisala/radarsw/configuration/vsoweb-override.ini*.
2. Create a list of radars in numbered order.

   The format of the list entries is **radar.list.N**, where N is an integer.

Example: If you have two licenses and three radars called "MyRadarA", "MyRadarB", and "MyRadarC", and you want the license to apply to "MyRadarA" and "MyRadarC", list the radars as follows:

radar.list.1 = MyRadarA

radar.list.2 = MyRadarC

radar.list.3 = MyRadarB

7.12 Configuring IRIS for IRIS Focus

7.12.1 Configuring the firewall

IRIS Focus connects to IRIS Analysis using port 30735. By default, the firewall of the IRIS Analysis server blocks this port. When IRIS Analysis and IRIS Focus are installed on separate servers, you need to configure IRIS Analysis server to allow the connection to this port:

1. Log in to the IRIS Analysis server as **admin**.
2. Run the following commands:

```
sudo firewall-cmd --add-port=30735/tcp --permanent
sudo firewall-cmd -reload
```

7.12.2 Setting or changing the socket server

In order for IRIS Focus to configure the radar centers correctly, you need to have at least one PPI product in the socket server.

If needed, set or change the socket server:

1. Update the *vsoweb-override.ini* file with the following command:

```
/usr/vaisala/radarsw/configuration/bin/configure-vsoweb-ini -i
<socket_server_host_name>
```

2. Type the following command:

```
rsw-basemap-site-setup --socket-server <socket_server_host_name>
```

3. Restart the **vaisala-radarsw-webapp** service by typing:

```
systemctl restart vaisala-radarsw-webapp
```

7.12.3 Activating the socket server in IRIS Radar

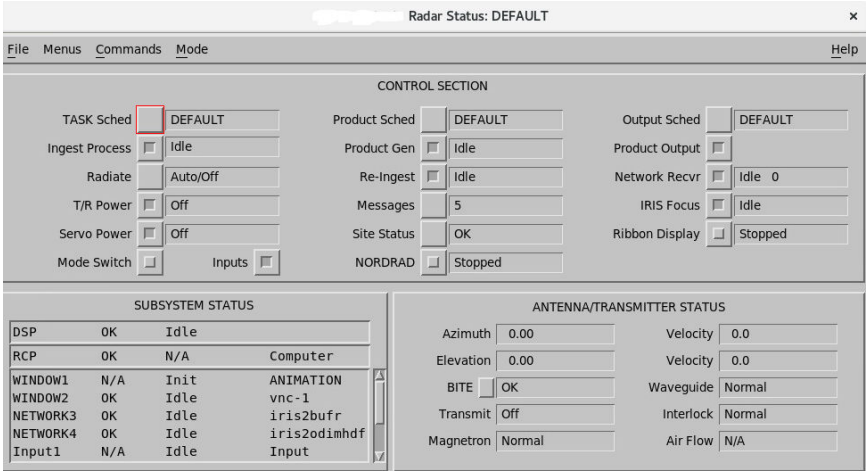


Figure 19 IRIS Radar status menu

If your system is running IRIS Focus server, you must enable the **IRIS Focus** option in IRIS Radar. For more information, see *IRIS Radar User Guide (M212926EN)*.

1. Make sure IRIS has started.
2. In IRIS Radar, select **Menus > Radar Status**.
3. Turn on the socket server by selecting the **IRIS Focus** checkbox.

   When this checkbox is selected, the field shows the status of the socket server process: **Idle**, **Running**, or **Stopped**.

7.12.4 Setting up Data Manager

The Data Manager service runs on the IRIS Focus server that receives radar scan volume data, stored in **RAW** file format, from the IRIS Analysis server and generates live radar products from the data in real-time.

During installation, IRIS Focus sets up all necessary services, databases, and user accounts for processing data. IRIS Focus features such as live products and dynamic composites require **RAW** files.

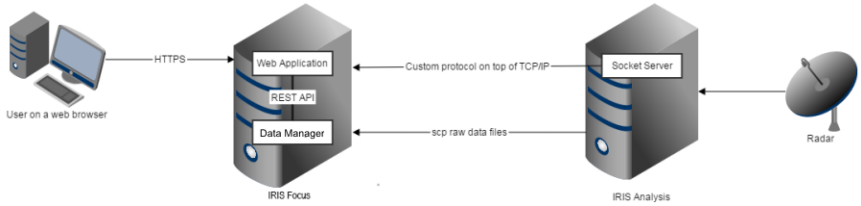


Figure 20 Radar data delivery paths

More information

- Data Manager (page 179)
- Data Manager does not work as expected (page 232)

7.12.4.1 Setting up Data Manager on IRIS Analysis server

To configure IRIS Analysis to send **RAW** files to IRIS Focus, you must set the target location on the IRIS Focus server as a network output device in IRIS Analysis.

The target location on IRIS Focus server is the following directory, which is owned by the **radardminput** user:

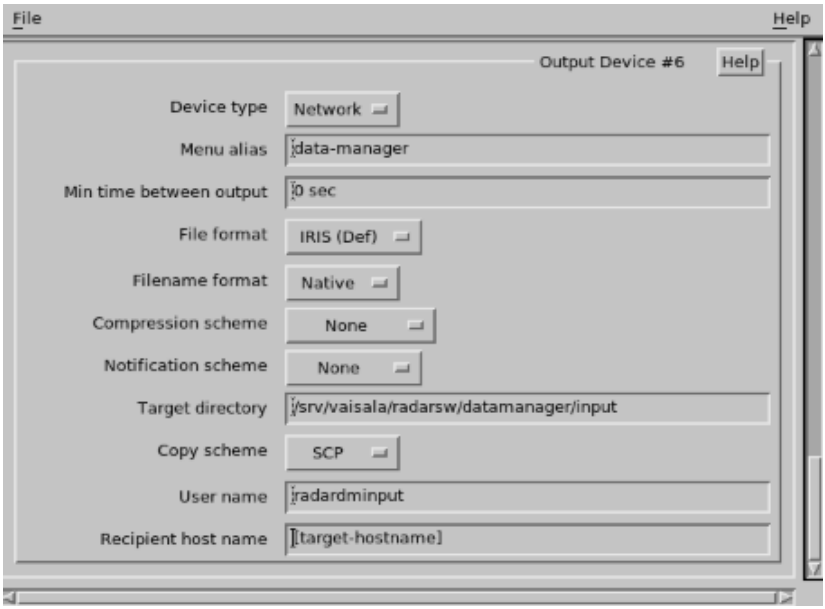
```
/srv/vaisala/radarsw/datamanager/input
```

1. Log in to the IRIS Analysis server as **radarop**.
2. In the terminal window, type: **setup&**

   The IRIS **Setup** utility opens
3. Select **Output**.
4. Create a new output device:
   a. In **Number of output devices**, increase the number of output devices by 1.
   b. Press **ENTER**.

      A new configurable output device is added to the end of the **Output Device** list.

5. In the configuration pane for the new output device, configure the new output device with the following settings:



a. **Device type**: **Network**
b. **Filename format**: **Native**
c. **Target directory**: */srv/vaisala/radarsw/datamanager/input*
d. **User name**: `radardminput`
e. Host name: `[IRIS Focus server]`
f. Select **File > Close**.
g. Select **File > Save**.
h. Select **File > Exit**.

6. Restart IRIS:
   a. Log in to the server as **root**.

```
#su
    #<type password>
```

   b. Type:

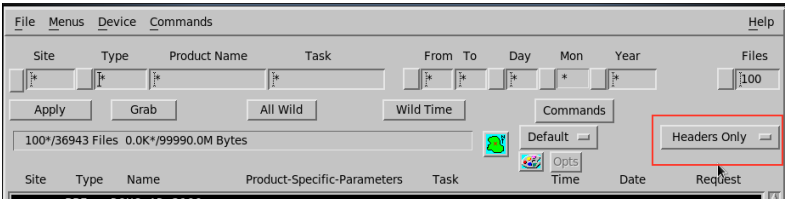
```
systemctl stop iris.service
systemctl start iris.service
```

7. In the terminal window, type: **`iris &`**

   a. Select **Menus > Product Output > Device**.

   b. Select the device you configured in the **Setup** utility.

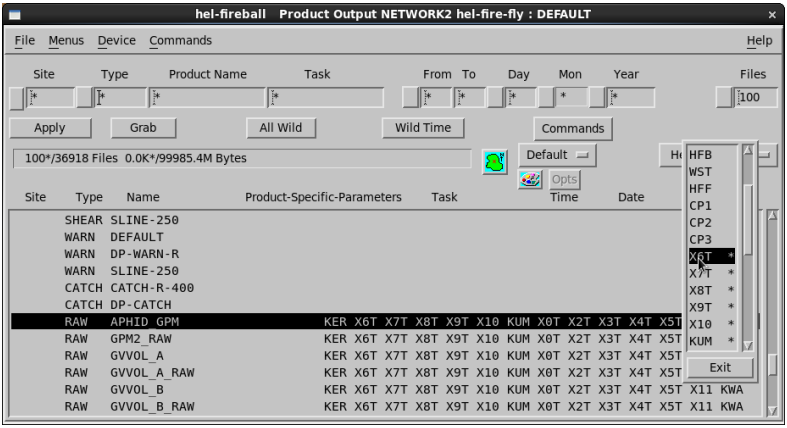
   c. In the drop down box on the far right of the window, select **Headers Only**.



   d. In the product list, select any **`RAW`** product.

   e. Right-click the far right of the product name and select a radar site.

      If needed, deselect any radar sites you do not want to include in the device configuration.



   f. Select **Apply**.

   g. Select **File > Save As**.

      Define a name for the new **Product Output** or use the **`DEFAULT`** option.

   h. Select **OK**.

   i. Select **Close**.

7.12.4.2 Setting up Data Manager on IRIS Focus server

RAW files on the IRIS Analysis server are handled by the local **root** user and **RAW** files on the IRIS Focus server by the local **radardminput** user.

You must add the IRIS Analysis **root** account's public SSH key to the IRIS Focus **radardminput** accepted keys list.

1. Log in to the IRIS Focus server as **root**.
2. If it does not exist already, create the following *.ssh* file:

```
# mkdir -m 700 /var/lib/radardminput/.ssh
# chown radardminput:radarsw /var/lib/radardminput/.ssh
```

3. Add the socket server key to the authorized SSH key store of the **radardminput** user:

   This enables file transfer from the IRIS Analysis root account to IRIS Focus **radardminput** user.

   a. Type:

```
# cd /var/lib/radardminput/.ssh
# ls
```

   b. If *authorized_keys* file already exists, type:

```
# vi authorized_keys
# rm socket-server-key
```

   Append the key you copied earlier to the file.

   c. If the *authorized_keys* file does not yet exist, add this file:

```
# vi authorized_keys
```

   Paste the key you copied earlier to your clipboard.

```
# chown radardminput:radarsw authorized_keys
# chmod 644 authorized_keys
```

4. Check that the expected on-demand product is visible in the IRIS Focus user interface.

   A data manager updater service records metadata of the files in a **PostgreSQL** database, which is accessed by the IRIS Focus web UI when it generates on-demand radar products from the data.

7.13 Connecting the TLP system

Follow this procedure to add the **Total Lightning Processor** system to the IRIS Focus system to retrieve lightning data.

These steps are typically done automatically by the `./rsw-installer` script when you include the `--lightning` option. You only need to perform these steps if you did not include the `--lightning` option when running `./rsw-installer`. Otherwise, you can skip to section Configuring the TLP for IRIS Focus (page 91).

1. To enable lightning in the Web application, edit the *`vsoweb-override.ini`* configuration file in the */etc/vaisala/radarsw/configuration* directory. Change (or create, if not present) the `[PROVIDERS]` section to the following:

```
[PROVIDERS]
radar.enabled = true
lightning.enabled = true
```

2. Restart the Web application by typing:

```
systemctl restart vaisala-radarsw-webapp
```

3. Configure the firewall.

   The **Total Lightning Processor** connects to the Kafka data broker on port **9094** on the IRIS Focus system. If you are running the `firewalld` service, configure the firewall to allow this connection.
   Example: If the TLP system IP address is `10.55.11.2`, run the following firewall commands on the IRIS Focus system to allow `10.55.11.2` access to port **9094**:

```
firewall-cmd --permanent --zone=public --add-rich-rule='rule family="ipv4"
source address="10.55.11.2/32" port protocol="tcp" port="9094" accept'

firewall-cmd --reload
```

4. Configure the **Total Lightning Processor**.

   At this point, the IRIS Focus system should be set up and ready for lightning data provided by the Total Lightning Processor. Follow the instructions in Configuring the TLP for IRIS Focus (page 91) to start the flow of lightning data from the TLP to IRIS Focus.

7.14 VHF or high data rate adjustments

If your TLP system will be providing lightning data at very high data rates, the lightning cache size of the lightning-websocket service should be increased. If you expect that your lightning data may exceed more than 100 000 events a day, you should increase the lightning cache size as indicated in section Increasing buffer capacity for lightning data (page 150).

7.15 Configuring the TLP for IRIS Focus

If you have the **Total Lightning Processor** (TLP) system that will be providing lightning data to IRIS Focus, you need to add a new service to the TLP system to push the lightning data into the kafka data broker service running on the IRIS Focus system. Your TLP must be running version 1.2.7 or later.

In the following procedure, you need the directory */opt/vai/tlp/etc*. If it does not exist, install it:

1. Log in to your TLP system as **root** user, or use the **su** or **sudo** command to gain root access.
2. Run the command:

```
dnf install -y vaisala-tlp-to-kafka
```

7.15.1 Verifying the installation of vaisala-tlp-to-kafka package

Before configuring a TLP system to send information to the Kafka data broker running on IRIS Focus, verify that the necessary software packages have been installed.

1. Log into your TLP system using the **root** user account.
2. Run the following command to make sure the necessary software packages are installed:

```
rpm -q vaisala-tlp-to-kafka || dnf install -y vaisala-tlp-to-kafka
```

7.15.2 Changing regstatd2 report frequency

The **regstatd2** service generates a regional network health report periodically that is used to provide the **Network Health** product layer on IRIS Focus. In a default installation, the **regstatd2** service updates this report once an hour. It is recommended that you configure **regstatd2** on the TLP to produce this report at a more frequent 10-minute interval.

1. Log into your TLP system using the **vops** user account.
2. Go to the ***regstatd2.cfg*** file in the */opt/vai/tlp/etc* directory.

3. Edit the file to set the **updateIntervalMinutes** parameter to 10 minutes by typing:

```
updateIntervalMinutes   10
```

4. Stop the **regstatd2** service by typing:

```
lpstart stop regstatd2
```

5. Start the **regstatd2** service again by typing:

```
lpstart start regstatd2
```

7.15.3 Adding the tlp-to-kafka service

This instruction applies to TLP 1.2.7. If you have a later release of TLP, see the **tlp-to-kafka man page**.

In IRIS Focus 7, the access to the Kafka cluster is on a different port than it was in Focus 6. Access now requires an authentication token. The details are described in step 5.

The steps below require that the **vaisala-tlp-to-kafka** package is installed on your TLP system. If this package is missing, you can install it by logging in as the **root** user and running:

```
dnf install -y vaisala-tlp-to-kafka
```

1. Log into your TLP system using the **vops** user account.
2. Go to the *startup.cfg* file in the */opt/vai/tlp/etc directory*.
3. Add the following line to the file:

```
core n java tlp-to-kafka -jar /opt/vai/tlp/lib/tlp-to-kafka.jar
```

4. Edit the *tlp-to-kafka.cfg* file in the */opt/vai/tlp/etc* directory according to how you want the lightning events to be sent to IRIS Focus:

   - If you want the lightning events sent to IRIS Focus to be composite flash events produced by the TLP, set the **lp.tokafka.smqLightning** parameter to **"smq://fdata"**.
   - If you want the lightning events sent to IRIS Focus to include the individual lightning strokes produced by the TLP, set the **lp.tokafka.smqLightning** parameter to **"smq://RLFxStrokeData"**.
   - You can use any shared memory queue of lightning data as the source for IRIS Focus. For example, if your TLP system is producing solutions from both VHF and LF based lightning sensors you can use the standard VHF event queue **"smq://sdata3d"**, the standard VHF flash queue **"smq://fdata3d"**, a merged data set **"smq://tldata or smq://wmdata"**, or some customer filtered queue. If you choose a data set that includes VHF data, you will need the *IRIS_VHF_LGT* feature enabled in your IRIS Focus license. Depending on your use cases for IRIS Focus, there may be limited use of the forwarding all of the raw VHF data points available in the **"smq://sdata3d"** shared memory queue as there can be many VHF event points for each lightning discharge.
   - If you have the **Lightning Threat Zone** feature licensed, make sure that the lightning data source you select includes LF data. The **Lightning Threat Zone** engine ignores all VHF lightning events in the data stream and only uses the LF events that is sees in the data stream.

   To set the value, type:

   ```
   lp.tokafka.smqLightning <parameter-value>
   ```

   For example:

   ```
   lp.tokafka.smqLightning "smq://RLFxStrokeData"
   ```

5. Access to the Kafka cluster requires an authentication token. The authentication token is randomly generated during the IRIS Focus 7 installation, and it is used in the password field.

   a. To find the value of this token, run the following command as **root** on the IRIS Focus system (in the example output below, the token is **L5KpD55KqxI7kGUuM0mQrmCh9QqONKI4**)

   ```
   [root@iris-focus ~]# grep kafka.*.ScramLoginModule /etc/vaisala/
   focus/k8s/vaisala-focus.yaml | head -1
               config:
   org.apache.kafka.common.security.scram.ScramLoginModule required
   username="focus-kafka" password="L5KpD55KqxI7kGUuM0mQrmCh9QqONKI4";
   [root@iris-focus ~]#
   ```

   b. When you have identified the fully qualified domain name and the authentication token for the IRIS Focus connection, go to the */opt/vai/tlp/etc* directory on the TLP system, locate the *kafka-producers.properties* file there, and change the lines as follows:

   ```
   bootstrap.servers=helsinki.rd.vaisala.com:9094
   security.protocol=SASL_PLAINTEXT
   sasl.mechanism=SCRAM-SHA-512
   sasl.jaas.config=org.apache.kafka.common.security.scram.ScramLoginModule
    required \
      username="focus-kafka" \
      password="L5KpD55KqxI7kGUuM0mQrmCh9QqONKI4";

   # How many acknowledgements are required before considering the request
   complete



   acks=all
   ```

   This example assumes that the fully qualified domain name of the IRIS Focus server is *helsinki.rd.vaisala.com* and that the randomly generated authentication token generated on the IRIS Focus server is **L5KpD55KqxI7kGUuM0mQrmCh9QqONKI4**. Make the appropriate substitutions for your installation.

6. Start the **tlp-to-kafka** service by typing:

   ```
   lpstart start tlp-to-kafka
   ```

The **tlp-to-kafka man** page provides more information on configuring and running the **tlp-to-kafka** service on a TLP system.

7.16 Verifying IRIS Focus installation

1. Check that the web user interface is running at the default HTTPS port, and the following default user accounts have been created in IRIS Focus during installation:
   - Username: `admin` / password: `admin123`
   - Username: `user` / password: `user123`

   Vaisala recommends changing the passwords after the installation.

2. Access the IRIS Focus web UI by opening a browser on the IRIS Focus server and navigating to *`https://localhost`*.

   You should see the login screen for IRIS Focus web application.

3. Log in with the default IRIS Focus user account.

   Make sure the application loads, and the map view is displayed.

4. Check that the expected on-demand product is visible in the IRIS Focus user interface.

   A data manager updater service records metadata of the files in a `PostgreSQL` database, which is accessed by the IRIS Focus web UI when it generates on-demand radar products from the data.

5. Verify that **Tracking Tool** and **Cross Section** buttons are visible in the application UI.

   This verifies that IRIS Focus features are enabled.

6. Enable the grid lines by selecting **Map Features Lat/lon grid**.

   Depending on where the map view is centered, you should see slightly distorted grid lines that are leading away from the equator. This verifies that the map projection is correct.

7. Confirm that data manager is running:
   a. Select **Weather Products > Add Product**.
   b. Add a new on-demand **PPI** or **CAPPI** product.
   c. Make sure you see weather data from the selected time on the screen.

8. Verify that you can add the **TimeSpan** and **Network Health** products on the map. If there is lightning occurring, check that you can see lightning data appearing on the map, as well as the regional health of your lightning network.

   If you have just completed the installation, it may take a while until the first network health report arrives.

7.17 Running nowcasting on a different server

Making use of the nowcasting, the load on your nowcast service may cause performance issues: IRIS Focus may become slower in returning results to users.

You can compensate for this by moving nowcasting to a separate server.

On the new (blank AlmaLinux, non-Focus) machine that will have the nowcast server on it, do the following steps:

1. Setup firewall rules first.
2. Set `ALLOW_IP` to IP address of machine that needs to access nowcast, or set to nothing to allow all machines access:

```
declare ALLOW_IP=""
declare -i ALLOW_PORT=31004
if systemctl status firewalld &> /dev/null && (( ALLOW_PORT > 0 )); then
if [ "${ALLOW_IP}" != "" ]; then
```

3. Limit access to just the machine specified:

```
firewall-cmd --permanent --zone=public --add-rich-rule="rule family=
\"ipv4\" source address=\"${ALLOW_IP}/32\" port protocol=\"tcp\" port=\"$
{ALLOW_PORT}\" accept" else
```

4. Allow everyone access instead:

```
firewall-cmd --permanent --add-port="${ALLOW_PORT}/tcp"
fi
fi
firewall-cmd --reload
```

5. scp the *`cloud-nowcast-service.tar`* from the *`<Focus installation files dir>/k8s/images`* to the nowcast server machine:

```
scp root@<Focus_machine>:/Focus_installation_files/k8s/images/cloud-
nowcast-service.tar .
```

6. Load and create nowcast container:

```
podman image load < cloud-nowcast-service.tar
podman run --name nowcast -d -p 31004:31004 com.vaisala.fire/cloud-nowcast-
service:7.x.x /app/bin/nowcast-server 31004
```

   where `x.x` is the number of the version/patch.

7. Check that you can reach nowcast on local server:

```
curl --request POST http://localhost:31004/focus-nowcast/api/v2/health;
echo
```

   You should see the following output:

```
{"status":"UP"}
```

8. To manage with **`systemd`**, use these commands:

```
podman generate systemd --new --name vaisala-radarsw-nowcast >| /etc/
systemd/system/vaisala-radarsw-nowcast.service
chmod 644 /etc/systemd/system/vaisala-radarsw-nowcast.service
systemctl enable --now vaisala-radarsw-nowcast
systemctl status vaisala-radarsw-nowcast
```

9. Any time firewall rules are changed, you need to restart the nowcast service with the following command:

```
systemctl restart vaisala-radarsw-nowcast
```

   a. Example on restarting without system control:

```
podman stop nowcast
podman start nowcast
```

10. To view the log, use the following command:

```
podman logs nowcast
```

11. On the IRIS-Focus machine, check that you can reach nowcast from remote server:

```
curl --request POST http://<nowcast_machine>:31004/focus-nowcast/api/v2/
health; echo
```

   You should see the following output:

```
{"status":"UP"}
```

12. Change the line in *`vsoweb-override.ini`* (use the hostname where the nowcast is):

   nowcast.http.server.url = http://*<Focus_machine>*:31004/focus-nowcast/api/v2/mvf/

13. Restart the webapp with this command:

```
systemctl restart vaisala-radarsw-webapp
```

8. One-server installation of IRIS Focus and IRIS Analysis

Follow this procedure when you install IRIS Analysis and IRIS Focus on the same server.

A prerequisite for installing IRIS Analysis and IRIS Focus is that AlmaLinux is installed on the server.

1. Install AlmaLinux and IRIS/RDA software with Kickstart according to instructions in *IRIS and RDA Software Installation Guide (M212924EN)*.

   The Kickstart installation process will automatically perform all the necessary installation steps, such as creating the correct partitions.
2. Verify or override the FQDN of the server. See Verify or override the FQDN of your server (page 45).
3. Install IRIS Focus:
   a. If needed, download the installation packages. See Downloading installation packages (page 35) .
   b. Install IRIS Focus. See Installing IRIS Focus from a USB stick (page 108).
   c. Install the IRIS Focus components. See Installing IRIS Focus components (page 113).
4. Configure IRIS Analysis for IRIS Focus. See Configuring IRIS for IRIS Focus in one-server installation (page 137).
5. Enable the graphical desktop environment. See Enabling a graphical desktop environment (page 142).
6. Verify IRIS Focus installation. See Verifying IRIS Focus installation (page 67).
7. Activate the IRIS Focus license. See Activating license – online (page 52), Activating license – offline (page 55), or Using the USB license key (page 57).

8.1 Configuring IRIS for IRIS Focus in one-server installation

The Data Manager service enables IRIS Focus to receive radar scan volume data from IRIS Analysis.

During installation, IRIS Focus sets up all necessary services, databases, and user accounts for processing data. IRIS Focus features such as live products and dynamic composites require RAW files.

8.1.1 Setting up data manager on IRIS Analysis server

To configure IRIS Analysis to send **RAW** files to IRIS Focus, you must set the target location on the IRIS Focus server as a network output device in IRIS Analysis.

The target location on IRIS Focus server is the following directory, which is owned by the **radardminput** user:

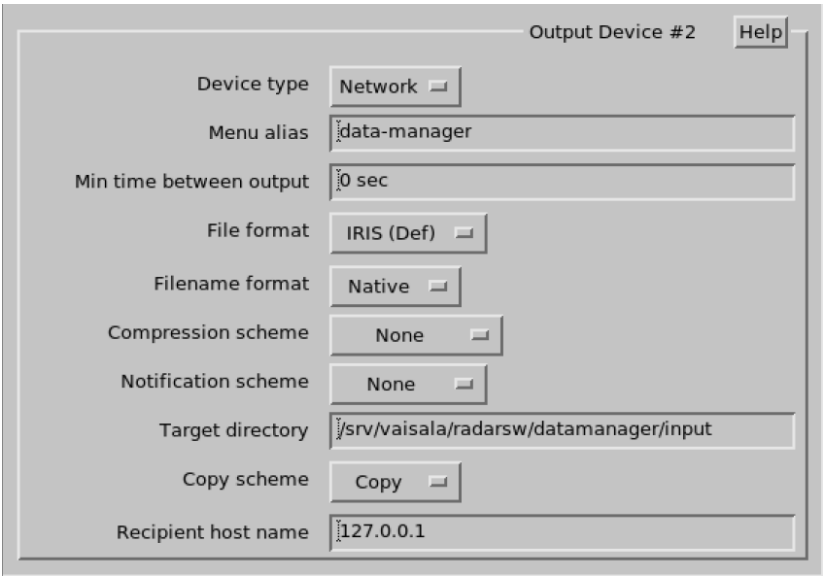
```
/srv/vaisala/radarsw/datamanager/input
```

1. Log in to the IRIS Analysis server as **radarop**.
2. In the terminal window, type: **setup&**

   The IRIS **Setup** utility opens.
3. Select **Output**.
4. Create a new output device:
   a. In **Number of output devices**, increase the number of output devices by 1.
   b. Press **ENTER**.

      A new configurable output device is added to the end of the **Output Device** list.

5. In the configuration pane for the new output device, configure the new output device with the following settings:



   a. **Device type**: **Network**
   b. **Filename format**: **Native**
   c. **Target directory**: */srv/vaisala/radarsw/datamanager/input*
   d. **User name**: `radardminput`
   e. Host name: `127.0.0.1`
   f. Select **File > Close**.
   g. Select **File > Save**.
   h. Select **File > Exit**.

6. Restart IRIS:
   a. Log in to the server as **root**.

```
#su
    #<type password>
```

   b. Type:

```
systemctl stop iris.service
systemctl start iris.service
```

7. Allow access to the Data Manager input directory:
   a. Log in to the server as **root**.
   b. Type:

```
chmod 777 /srv/vaisala/radarsw/datamanager/input/
```

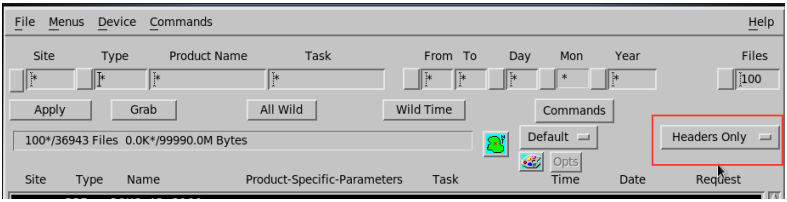
   This setting allows members of the **radarsw** group to copy RAW files into this directory.

8. In the terminal window, type: **`iris &`**

   a. Select **Menus > Product Output > Device**.

   b. Select the device you configured in the **Setup** utility.

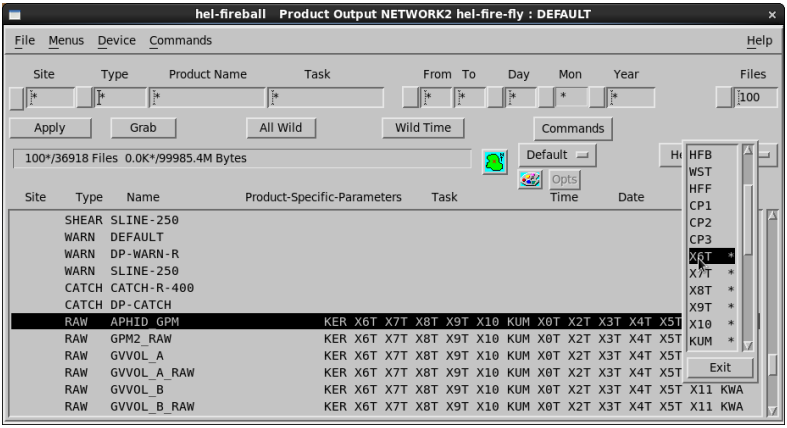
   c. In the drop down box on the far right of the window, select **Headers Only**.



   d. In the product list, select any **`RAW`** product.

   e. Right-click the far right of the product name and select a radar site.

      If needed, deselect any radar sites you do not want to include in the device configuration.



   f. Select **Apply**.

   g. Select **File > Save As**.

      Define a name for the new **Product Output** or use the `DEFAULT` option.

   h. Select **OK**.

   i. Select **Close**.

8.2 Enabling a graphical desktop environment

IRIS Focus does not include any graphical applications. For security and performance reasons it is preferred to run IRIS Focus in the text based multi-user mode. This reduces the number of running services.

IRIS Analysis, on the other hand, does include graphical applications that require a graphical desktop environment when run locally. If you intend to run graphical applications directly on the system where IRIS Analysis is installed and the system is currently operating in a text based multi-user mode, you will need to switch to graphical mode.

1. To determine whether the graphical environment is active or inactive, run the following command:

```
[root@fire-test-iris ~]# systemctl is-active graphical.target
inactive
[root@fire-test-iris ~]#
```

2. To activate the graphical desktop environment, use the following command:

```
[root@fire-test-iris ~]# systemctl isolate graphical
[root@fire-test-iris ~]#
```

3. To make the graphical desktop environment the default when the system is started, use the following command:

```
[root@fire-test-iris ~]# systemctl set-default graphical
[root@fire-test-iris ~]#
```

4. If you need to disable and stop the graphical desktop environment, use the following commands to switch back to the text based multi-user mode:

```
[root@fire-test-iris ~]# systemctl set-default multi-user
[root@fire-test-iris ~]# systemctl isolate multi-user
[root@fire-test-iris ~]#
```

9. Upgrading IRIS Focus

9.1 Migrating to IRIS Focus 7.4

If you have IRIS Focus 6.x or 7.x in use, you can take release 7.4 into use through a migration. In a migration, the following configurations will remain intact:

- User details
- Identity configuration
- Password configuration
- Organizations
- Application subscription
- Map layers
- Map view context
- Welcome and notification text
- Alert notification settings
- Weather alert message
- Technical alert messages
- Color scales
- DataFlow alert configuration
- GLD360 WMS layer
- Places of interest and events
- Nowcast configuration
- Saved views
- Predefined composites
- Projection setting

Use the version of the **`rsw-migrate-install`** tool that matches with the version of the target system where the new install will be done. For example, if you are migrating from 6.1 to 7.4, you need to use the **`rsw-migrate-install`** tool from the IRIS Focus 7.4 release.

IRIS Focus 7.4 requires a server computer running on AlmaLinux 9.

1. If you do not already have the ***`focus-migrate-tool-7.4.0.tar.gz`*** file, generate it by running the following command in the Focus 7.4 release directory:

   ```
   ./rsw-migrate-install --create-tool-tar
   ```

2. Copy the ***`focus-migrate-tool-7.4.0.tar.gz`*** file to the server you want to migrate from.
3. Untar the file:

   ```
   tar -xvf focus-migrate-tool-7.4.0.tar.gz
   ```

4. Run the migration tool to get a backup file:

```
./rsw-migrate-install --backup
```

5. The migration tool tells you where the resulting migration backup tar file is located. Copy the file to a safe place.
6. Install Focus 7.4 on a server running AlmaLinux 9.
7. Copy the migration backup tar file to the Focus 7.4 server.
8. On the Focus 7.4 server, run the migration tool from the Focus 7.4 release directory to restore the backup tar (`-s` is optional and will run the `rsw-basemap-site-setup` during the migration):

```
./rsw-migrate-install -s <IRIS Analysis socket server> --restore-archive <path to
backup tar>
```

If you want to copy the settings from one 7.4 server to another, you can do a migration between the servers. Note that the migration tar file stores a number of additional configuration files from the source server that are not automatically migrated. So, if you are missing some settings from the 7.4 server after the migration, check for them in the tar file.

10. Configuration

10.1 Configuring server after changing IP address

If the IP address of the IRIS Focus server is changed, and after the change you experience problems using kubectl, microk8s, or k9s commands from the command line, do the following:

1. Log in to the server as **root**.
2. Run the following command:

```
microk8s config | tee ~/.kube/config
chmod 600 ~/.kube/config
```

10.2 Configuring DNS

Use the `rsw-manage-network` script if you need to make changes to the DNS configuration after installation.

1. View list of options for the script:

```
rsw-manage-network --help
```

2. Determine if DNS is enabled on the system in the */etc/nsswitch.conf* file:

```
rsw-manage-network dns-is-enabled
```

3. If DNS is currently enabled, you can use the following command to disable DNS on the system and apply the necessary changes to the IRIS Focus related services, such the `coredns` service running in Kubernetes:

```
rsw-manage-network dns-disable
```

4. If DNS is currently disabled, and the */etc/resolv.conf* file contains the IP address of DNS name servers, you can use the following command to enable DNS and apply the necessary changes to the IRIS Focus related services, such as the `coredns` service running in Kubernetes:

```
rsw-manage-network dns-enable
```

5. If you have manually modified the DNS configuration, use the following command to apply your changes to the **coredns** service running in Kubernetes. If you have enabled DNS, this will update the **coredns** configmap so that it uses the name servers listed in */etc/resolv.conf* when it needs to forward a request. If you have disabled the DNS, this command will remove any forwarding entries from the **coredns** configuration:

```
rsw-manage-network dns-apply
```

More information

- IRIS Focus fails to resolve host names (page 231)

10.3 Configuring vsoweb-override.ini file

Use this procedure to change the following settings:

radar.enabled = true/false

lightning.enabled = true/false

iris.socket.server.host

security.cors.origin.whitelist

1. Go to the */etc/vaisala/radarsw/configuration* directory.
2. To update any entry in the *vsoweb-override.ini* file, use the command:

```
configure-vsoweb-ini
```

Example:

```
$/usr/vaisala/radarsw/configuration/bin/configure-vsoweb-ini --radar false --
lightning true --cors-origin-whitelist localhost --iris_host
iris_server.mydomain.com
```

10.4 Adding/removing radars

When new radar sites are added or removed as data sources on the IRIS Analysis server, radar settings on the IRIS Focus server must be re-syncronized. Settings requiring updates include updating the radar site location in GeoServer and calculating new map projections.

1. Run radar site setup script:

```
rsw-basemap-site-setup --socket-server [socket_server_host_name]
```

2. Restart the **vaisala-radarsw-webapp** service by typing:

```
systemctl restart vaisala-radarsw-webapp
```

10.5 Configuring nowcast

You must have a nowcasting license to use nowcasting in IRIS Focus.
See IRIS Focus licensing (page 14).

If you have a license for the nowcast service, you can enable the IRIS Focus web application to make nowcast projections available in the web interface.

To do this, you may need to make changes to the ***vsoweb-override.ini*** file found in the /etc/vaisala/radarsw/configuration directory.

1. Log in to the server as **root**.
2. Edit */etc/vaisala/radarsw/configuration/vsoweb-override.ini*.
3. In the **[NOWCAST]** section of the **vsoweb-override.ini** file, check that the nowcast server is enabled:

```
nowcast.mvf.run = true
```

4. Check the nowcast server URL:

```
nowcast.http.server.url = http://localhost:31000/focus-nowcast/api/v2/mvf/
```

5. If you made any changes to the **vsoweb-orverride.ini** configuration file, you must restart the web application.

```
systemctl restart vaisala-radarsw-webapp
```

10.6 Running nowcasting on a different server

Making use of the nowcasting, the load on your nowcast service may cause performance issues: IRIS Focus may become slower in returning results to users.

You can compensate for this by moving nowcasting to a separate server.

On the new (blank AlmaLinux, non-Focus) machine that will have the nowcast server on it, do the following steps:

1. Setup firewall rules first.
2. Set **ALLOW_IP** to IP address of machine that needs to access nowcast, or set to nothing to allow all machines access:

```
declare ALLOW_IP=""
declare -i ALLOW_PORT=31004
if systemctl status firewalld &> /dev/null && (( ALLOW_PORT > 0 )); then
if [ "${ALLOW_IP}" != "" ]; then
```

3. Limit access to just the machine specified:

```
firewall-cmd --permanent --zone=public --add-rich-rule="rule family=
\"ipv4\" source address=\"${ALLOW_IP}/32\" port protocol=\"tcp\" port=\"$
{ALLOW_PORT}\" accept" else
```

4. Allow everyone access instead:

```
firewall-cmd --permanent --add-port="${ALLOW_PORT}/tcp"
fi
fi
firewall-cmd --reload
```

5. scp the *cloud-nowcast-service.tar* from the *<Focus installation files dir>/k8s/images* to the nowcast server machine:

```
scp root@<Focus_machine>:/Focus_installation_files/k8s/images/cloud-
nowcast-service.tar .
```

6. Load and create nowcast container:

```
podman image load < cloud-nowcast-service.tar
podman run --name nowcast -d -p 31004:31004 com.vaisala.fire/cloud-nowcast-
service:7.x.x /app/bin/nowcast-server 31004
```

where **x.x** is the number of the version/patch.

7. Check that you can reach nowcast on local server:

```
curl --request POST http://localhost:31004/focus-nowcast/api/v2/health;
echo
```

   You should see the following output:

```
{"status":"UP"}
```

8. To manage with **`systemd`**, use these commands:

```
podman generate systemd --new --name vaisala-radarsw-nowcast >| /etc/
systemd/system/vaisala-radarsw-nowcast.service
chmod 644 /etc/systemd/system/vaisala-radarsw-nowcast.service
systemctl enable --now vaisala-radarsw-nowcast
systemctl status vaisala-radarsw-nowcast
```

9. Any time firewall rules are changed, you need to restart the nowcast service with the following command:

```
systemctl restart vaisala-radarsw-nowcast
```

   a. Example on restarting without system control:

```
podman stop nowcast
podman start nowcast
```

10. To view the log, use the following command:

```
podman logs nowcast
```

11. On the IRIS-Focus machine, check that you can reach nowcast from remote server:

```
curl --request POST http://<nowcast_machine>:31004/focus-nowcast/api/v2/
health; echo
```

   You should see the following output:

```
{"status":"UP"}
```

12. Change the line in *`vsoweb-override.ini`* (use the hostname where the nowcast is):

   nowcast.http.server.url = http://*<Focus_machine>*:31004/focus-nowcast/api/v2/mvf/

13. Restart the webapp with this command:

```
systemctl restart vaisala-radarsw-webapp
```

10.7 Increasing buffer capacity for lightning data

The **lightning-websocket** service provides lightning events to the web browser. For performance reasons, the lightning events are kept in a cache so that data may quickly be provided to end users. The factory default configuration sets the size of this cache so that it can hold up to 700,000 events. This is typically a large enough number to provide up to a week's worth of historical data for high precision lightning networks that use LF signal processing to detect the electrical discharge of each lightning event.

VHF lightning detection networks detect events related to the channel that the electrical discharge of a lightning event flows through, rather than the the single discharge that flows through the channel. VHF lightning detection networks typically provide several events for each discharge and produce a lot of lightning data. If you connect IRIS Focus to a lightning data feed containing events produced by a VHF lightning detection network, the default cache size of 700,000 events will probably be too small. In that case, increase the size of the cache.

Increasing the cache size causes an increase of the memory requirements on your server, and causes a longer initialization time of the **lightning-websocket** service, as it loads its cache from the Kafka cluster at the startup. You may need to add or allocate more RAM to the system if you increase the cache size to a very large size.

1. Go to the *vaisala-focus-lightning.yaml* file in the */etc/vaisala/focus/k8s* directory.

   The size of the cache is controlled by two parameters (the example shows the default values):

```
# Internal lightning cache configuration, total capacity is count * size
    lightning.cache.buf.count = 701
    lightning.cache.buf.size = 1000
```

2. To increase the cache size from 700,000 to 10,000,000, change the **lightning.cache.buf.count** parameter to 10001 using a text editor:

```
# Internal lightning cache configuration, total capacity is count * size
   lightning.cache.buf.count = 10001
   lightning.cache.buf.size = 1000
```

   Alternatively, you can change the size from the command line:

```
sed -e 's,^\(    lightning.cache.buf.count\).*,\1 = 10001,' -i /etc/
vaisala/focus/k8s/vaisala-focus-lightning.yaml
```

3. To stop the **lightning-websocket** service and apply the changes, run the following commands:

```
kubectl delete --namespace vaisala-focus-lightning deployment/lightning-
websocket
kubectl apply -f /etc/vaisala/focus/k8s/vaisala-focus-lightning.yaml
```

   Kubernetes will start the **lightning-websocket** service when the *vaisala-focus-lightning.yaml* file is applied.

10.8 Configuring alert notifications

IRIS Focus can send notifications to users when weather alerts are triggered. In addition, IRIS Focus can send notifications about technical alerts to users with **administrator** role.

Configure the email and SMS settings for the system so that it can send notifications.

For SMS gateway, IRIS Focus supports MessageBird (https://www.messagebird.com.) IRIS Focus also supports email-to-text services.

1. Log in to the IRIS Focus web application as **administrator**.
2. Select **Admin > System > Notification settings**.
3. Fill in the required parameters for email and SMS notification message service.
4. To test the email and SMS service, enter the address or phone number in the **Email verification** or **SMS verification** field, and select **Send**.

   You must save your settings before sending the test message.

10.8.1 Editing default messages for weather alerts

Write the default content for the notification messages that users receive when weather alerts are triggered. When users set notifications for their own areas of interest, they can either use the default content or replace it with their own message text.

Select whether users by default receive a notification when the alert is cleared. Users can change this in their personal settings.

If some recipients' phones do not support HTML formatting, use the plain-text email message fields.

Depending on the service provider, SMS messages that exceed the limit of 160 characters may get broken up into multiple messages.

1. Log in to the IRIS Focus web application as **`administrator`**.
2. Select **Admin > System > Weather alert default messages**.
3. Fill in the email and SMS fields.

   You can select macros that will fill in the message with predefined values when the message is sent. The content can be, for example, the name of the area of interest and severity of the alert.

Table 13 Email message field

Field	Description
Email to	Default: the address set for the user account of the user who created the area of interest. If the user only has the **focus** user role, then only the user can receive the notification. If the user has the **poweruser** role, the user can add other recipients.
Email subject	You can use macros to fill in information, such as the severity of the alert and the name of the area of interest.
Email text (HTML)	The content of the email. You can use macros to fill in information.
Email text (plain text)	The content of the email. You can use macros to fill in information. Use this field if the recipients' devices do not support HTML. If you are using an email-to-SMS service, and some recipients' phones do not support HTML formatting, use the SMS message fields instead of the email message fields.
Email subject when cleared	The subject of the email that is sent when the alert is cleared. You can use macros to fill in information.
Email text when cleared (HTML)	The content of the email that is sent when the alert is cleared. You can use macros to fill in information.

Field	Description
Email text when cleared (plain text)	The content of the email that is sent when the alert is cleared. You can use macros to fill in information. Use this field if the recipients' devices do not support HTML. If you are using an email-to-SMS service, and some recipients' phones do not support HTML formatting, use the SMS message fields instead of the email message fields.

Table 14 SMS message fields

Field	Description
Send to	Default: the number set for the user account of the user who created the area of interest. If the user only has the **focus** user role, then only the user can receive the notification. If the user has the **poweruser** role, the user can add other recipients.
SMS text	You can use macros to fill in information, such as the severity of the alert, and the name of the area of interest. Character limit: 160 Messages that exceed the character limit (160 characters) will be broken up into multiple messages.
SMS text when cleared	The content of the SMS that is sent when the alert is cleared. You can use macros to fill in information.

10.8.2 Editing messages for technical alerts

You can configure IRIS Focus to send notifications about technical alerts to users with `administrator` role. Technical alerts include, for example, alerts about dataflow problems.

You can view information about the technical alerts in the **Alert history** view, if you have a `focus` user role.

Set the content of the notification messages:

1. Log in to the IRIS Focus web application as **`administrator`**.
2. Select **Admin > System > Technical alert default messages**.
3. Fill in the required parameters for email and SMS notification messages.

If you want to receive notifications, they must be enabled in you personal **Preferences**

10.9 Setting up housekeeping for events and alerts database

You can set IRIS Focus to clean the alerts database when it is getting full, and to give an alert when the database load is approaching the database size limit. By default, this feature is enabled. The database size limit is set automatically depending on the partition/disk size reported by the operating system during the installation, but you can change this limit. The default is 10% of the hard disk partition. By default, the database is installed in the */srv* partition.

You can select the limit that triggers the alert. The default is 90% of the size limit. You can also set the clean-up target. The clean-up target tells how many of the latest alerts will be kept in the database.

If you want to save the old alerts, do one of the following when you get the alert about the approaching clean-up:

- Take a manual backup of the database.
- Add disk space to the partition. Restart the webapp after this.
- Increase the configured database size limit (%). Restart the webapp after this.

The alerts that are erased from the database during the clean-up are also deleted from the **Alert history** table. This means that if an alert has persisted for a long period time, and the housekeeping has erased alerts from that period, you will only see the latest timestamps for the alert.

1. Log in to the server as **`root`**.
2. Go to the *`vsoweb-override.ini`* file in the */etc/vaisala/radarsw/configuration* directory.
3. Set the maximum percentage of disk partition to use (database size limit) by setting the value:

   ```
   events.alerts.housekeeping.trigger.partition.percentage
   ```

4. Set the limit that triggers the alert (percentage of the maximum number of alerts) by setting the value:

   ```
   events.alerts.housekeeping.alert.percent.full
   ```

5. Set the clean-up target by setting the value:

```
events.alerts.housekeeping.target.limit
```

6. If you want to disable the database housekeeping, set the following key to **`false`**:

```
events.alerts.housekeeping.do.housekeeping = false
```

7. If you want to disable the alerts for housekeeping, set the following key to **`false`**:

```
events.alerts.housekeeping.alert.before = false
```

8. Restart the web application.

10.10 Configuring alert engine

Processing data (particularly radar data) for alert detection can be system intensive. In the case of a complex alert configuration, the system may sometimes be unable to keep up. When this occurs, the delay from the time when data arrives until the time when you are notified will start to increase.

When the alert engine detects that there is too much delay in alerts from some combination of area of interest and event, the alert engine clears all of the pending data for that area/event combination. As a result, some data gets dropped, but the system is able to catch up to real-time. The timeout period before data is dropped is adjustable. The default is 2 minutes.

To configure parameters related to alert engine:

1. Go to the */etc/vaisala/radarsw/webapp* directory.
2. Open the *`application.yaml`* file.
3. Adjust these parameters:
   - **`threads`**: Defines the number of threads to make available for alert processing.
   - **`flushTimeOut`**: Defines the timeout period (the period after which data is dropped), in seconds.

10.11 Configuring visualization of hybrid tasks

When you use hybrid tasks, you can select whether partially finished hybrid scans are displayed on IRIS Focus or not. By default, partial hybrid scans are displayed.

If you want to display only completed volume scans, follow these steps:

1. Log in to the server as **`root`**.

2. Go to the *vsoweb-override.ini* file in the */etc/vaisala/radarsw/configuration* directory.
3. Set the **HYBRID_PRODUCT_TIMES** parameter to **false**:

```
use.partial.hybrid.times = false
```

4. Restart the web application.

If you want to reset IRIS Focus to display partial hybrid scans, reset the **HYBRID_PRODUCT_TIMES** parameter to **true**, and restart the web application.

10.12 Scheduling image exports from IRIS Focus

If you want to share interesting weather events on, for example, your website, use a **REST POST** method to schedule image exports from IRIS Focus saved views.

CAUTION! Depending on setup of the target website, the image export can be a bit slow. Take this into account when planning your export volumes and schedules.

10.12.1 Exporting images as .png files

Use this procedure to export images as .png files.

1. In the IRIS Focus **Map** view, set-up the view you want to save.

   For example, you can save the settings for:
   - **Weather Products**
   - Map tools such as the cross-section and tracking tools
   - Zoom level
2. Select **Saved Views > Save**.
3. Name the view and select **Save**.

   The new view is added to the **Saved Views** list for your future use.
4. Configure your web server to access the IRIS Focus image export service:

```
@Request: POST <your IRIS Focus URL>/focus-webapp/api/v2/image-export/get-image
@Produces: "image/png"
```

5. Configure the following parameters:

Parameter	Description
`username`	For security reasons, Vaisala recommends that you configure a specific user for exporting images.
`password`	IRIS Focus password for the user.
`time`	Time, in ISO-8601 format: `2021-06-18T17:55:23.000Z`
`widthPx`	Width of the exported image, in pixels.
`heightPx`	Height of the exported image, in pixels.
`savedViewName`	The name of the saved view you created in step 3.
`savedViewUser`	Optional value. Used if you configure a specific user for exporting images (recommended).

6. Instead of step 4 and step 5, you can run the export from the command line by creating a script and setting-up a **cron** job. For example:

   a. Create a Python script for the image export such as the following:

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
from requests_futures.sessions import FuturesSession
import datetime
APP_URL = "your_url_here"
IMAGE_EXPORT_LOC = "/focus-webapp/api/v2/image-export/get-image"
FILE_PATH = "yourpath_and_nameofoutputimagesinpng_here"
USERNAME = "username_here"
PASSWORD = "password_here"
TIME = datetime.datetime.utcnow().isoformat()
WIDTH = "1000"
HEIGHT = "700"
VIEW = "view_name_here"
def main():
    session = FuturesSession()
    req_params = {"username": USERNAME, "password": PASSWORD, "time":
TIME, "savedViewName": VIEW, "widthPx": WIDTH, "heightPx": HEIGHT}
    future_one = session.post(APP_URL + IMAGE_EXPORT_LOC,
params=req_params, verify=False)    # wait for the request to complete,
if it hasn't already
    res = future_one.result()
    print('{0} response status: {1}'.format(TIME, res.status_code))
    if res.status_code == 200:
        with open(FILE_PATH, 'wb') as f:
            f.write(res.content)

if __name__ == '__main__':
    main()
```

   Although the example **image-export.py** script saves only one snapshot, you can edit it to loop a set number of times and get multiple snapshots at a time.

   b. Type **crontab -e** in the terminal and add, for example, the following line to the *crontab* file (add your own paths and arguments).

```
*/15 * * * * /usr/bin/python
        /path/to/script/image-export.py >> /path/to/log/export.log 2>&1
```

   This executes the **image-export.py** script every 15 minutes and saves a single snapshot as a PNG file to the server.

10.12.2 Exporting images as .geotiff files

You can also export images as geoTIFF files.

The procedure is otherwise similar to Exporting images as .shp files (page 160), but to configure your web server to access the IRIS Focus image export service, use the following command:

```
@Request: POST <server-name>/focus-webapp/api/v2/image-export/geotiff
@Produces: "image/tiff"
```

The image is exported as a ***.tiff*** file.

Note that you can use the sample Python script shown in Exporting images as .shp files (page 160) to grab geotiff files by setting the TYPE to "geotiff".

10.12.3 Exporting images as .shp files

Use this procedure to export images as shape files (.shp). The output is a zip file containing all the files for the shape file.

1. In the IRIS Focus **Map** view, set-up the view you want to save.

   For example, you can save the settings for:
   - **Weather Products**
   - Map tools such as the cross-section and tracking tools
   - Zoom level
2. Select **Saved Views > Save**.
3. Name the view and select **Save**.

   The new view is added to the **Saved Views** list for your future use.
4. Configure your web server to access the IRIS Focus image export service:

   ```
   @Request: POST <server-name>/focus-webapp/api/v2/image-export/shp
   @Produces: "application/octet-stream"
   ```

   The image is exported as a zip file.

5. Configure the following parameters:

Parameter	Description
`username`	A valid IRIS Focus username. For security reasons and for smooth user experience, Vaisala recommends that you configure a specific user for exporting images. If you are using the username of an active user, and that user is logged when a scheduled export takes place, the user will get logged out, because a user cannot be logged in from two machines at the same time.
`password`	IRIS Focus password for the user.
`time`	Time, in ISO-8601 format: `2021-06-18T17:55:23.000Z`
`savedViewName`	The name of the saved view you created.
`savedViewUser`	Optional value. Used if you configure a specific user for exporting images (recommended).

6. Instead of steps 4 and 5, you can run the export from the command line by creating a script and setting-up a `cron` job. For example:

   a. Create a Python script for the image export such as the following:

```
#!/usr/bin/python3
from requests.sessions import Session
from datetime import datetime, timedelta

# Change to host name of IRIS Focus if run externally
APP_URL = "https://localhost"

# User account to login with to render image
USERNAME = "image-export"
PASSWORD = "USER_PASSWORD"

# Name of saved view and user account that created the saved view
VIEW = "SAVED_VIEW_NAME"
VIEW_USER = "USER_THAT_SAVED_VIEW"

# You can change these values
OUTPUT_DIR = '.'   # Directory to write output file to
FILE_BASE_NAME = "image-export" # Name of file sans extension
SSL_VERIFY = False # Set to True if you have a valid certificate
TYPE = "shp"       # Can be "shp" or "geotiff"

# Example of backing up 5 minutes from "now" (no data at time causes
404)
TIME = datetime.utcnow() - timedelta(days=0, hours=0, minutes=5)

def main():
    ext = ".tiff"
    if TYPE == "shp":
        ext = ".zip"
    file_path = OUTPUT_DIR + "/" + FILE_BASE_NAME + ext
```

```
    session = Session()
    time_str = TIME.isoformat()
    url  = APP_URL + "/focus-webapp/api/v2/image-export/" + TYPE
    req_params = {"username": USERNAME, "password": PASSWORD,
                  "time": time_str,
                  "savedViewName": VIEW, "savedViewUser": VIEW_USER}
    res = session.post(url, params=req_params, verify=SSL_VERIFY)
    print('{0} response status: {1}'.format(time_str, res.status_code))
    if res.status_code == 200:
        with open(file_path, 'wb') as f:
            f.write(res.content)
        print('Created file: {0}'.format(file_path))

if __name__ == '__main__':
    main()
```

Although the example **image-export.py** script saves only one snapshot, you can edit it to loop a set number of times and get multiple snapshots at a time.

b. Type **crontab -e** in the terminal and add, for example, the following line to the *crontab* file (add your own paths and arguments).

```
*/15 * * * * /usr/bin/python3
       /path/to/script/image-export.py >> /path/to/log/export.log 2>&1
```

This executes the **image-export.py** script every 15 minutes and creates a single ZIP file containing the shape file components.

10.13 Exporting NetCDF files from lidar systems to IRIS Focus

The following instructions show how to export NetCDF files from lidar systems to IRIS Focus.

The NetCDF files are created in the lidar system and delivered to IRIS Focus using the SFTP file transfer protocol.

Lidar volumes that contain multiple sweeps must be sent as a single NetCDF file.

10.13.1 Preparing IRIS Focus for transferring NetCDF files

The **dminput** user account was created during installation with the necessary settings for transferring NetCDF files. The account is disabled by default.

To enable the **dminput** user account, set up a password. Log in as the root user and use the following command:

```
su -
passwd dminput
```

10.13.2 Configuring the lidar system

For full instructions, see chapter *Configuring the FTP* in *WindCube Scan software suite User Manual (M212324EN)*.

1. Set the IP address of your IRIS Focus system as the host name.
2. Set the user to **dminput**.
3. Set the password to match the **dminput** account password.
4. Set the directory to */srv/pv/lidar-input-service*.

11. System administration

11.1 User roles

A user's access to IRIS Focus features depends on the roles assigned to the user. For example, the administration features are available to user accounts with the `administrator` role. A user may have several user roles, and when they log in, they have the features of all their roles available.

User roles can be divided into two categories:

- `Focus` roles are needed for full-scale remote sensing data visualization. Logging in with a `Focus` role reserves a seat from the license seat pool.
- System roles are needed for system purposes. They do not reserve seats from the pool, and they do not offer the full-scale features. For full-scale features, the user also needs a `Focus` role.

Focus roles

`Focus` roles reserve a `Focus` seat from the license seat pool when logging in.

Table 15 Focus roles

`Focus Weather Radar` In the **Add user** screen, this role is called `focus-radar`.	Can access the full IRIS Focus feature set for visualizing weather radar or wind lidar data, such as: • Configuring product generation • Using data analysis tools, like Tracking tool • Creating personal areas of interest and monitoring these areas for weather events created by `poweruser`
`Focus Lightning` In the **Add user** screen, this role is called `focus-lightning`.	Can access the full IRIS Focus feature set for visualizing lighting data, such as: • Configuring product generation • Using data analysis tools, like Tracking tool • Creating personal areas of interest and monitoring these areas for weather events created by `poweruser`

IRIS Focus Light

A user without a `focus` role enters the *IRIS Focus Light* view when logging in.

IRIS Focus Light view consists of a predefined map view with limited features. The following features are available:

- View one pregenerated weather product at a time (no on-demand products)
- See areas of interest with active alerts highlighted in the alert severity color when viewing current data

- View WMS map layers
- View the animation timeline
- View the cursor tool
- Create and edit personal color scales
- Change radar/lidar site
- Select map features
- Use the **Ruler Tool**
- Change user preferences

IRIS Focus Light view has an unlimited number of seats. If there are no *IRIS Focus* license seats available, the user will be logged in with an *IRIS Focus Light* license. If the licence is missing, users cannot log in. This could happen, for example, if the USB license key has been removed or if this is a new installation, not from the factory, that requires an e-mail be sent to Vaisala to retrieve the license.

Seat allocation and restrictions

A user with a `Focus Lightning` role reserves one of the *IRIS_Focus_Lightning* seats associated with the license.

A user with a `Focus Weather Radar` role reserves one of the *IRIS_Focus_Weather_Radar* seats associated with the license.

When the user logs out, the seat is released.

If a user with one of the `Focus` roles (`Focus Lightning` or `Focus Weather Radar`) logs in and there are no seats available, the user is directed to the IRIS Focus Light view. When an *IRIS Focus* license is available, the user is provided with an opportunity to switch to the full-scale IRIS Focus view.

The user is also directed to the *IRIS Focus Light* view in a situation where the user has both `Focus Lightning` and `Focus Weather Radar` roles, and the system has run out of free *IRIS_Focus_Weather_Lightning* or *IRIS_Focus_Weather_Radar* seats. In other words, both seats have to be available for this user to see the full-scale IRIS Focus.

System roles

System roles are needed for various system management tasks and functionalities. System roles do not reserve a Focus seat from the seat pool.

When logging in, a user that has one or more of these roles, but no `Focus` role, enters the *IRIS Focus Light* view.

Table 16 System roles

Role	Description
`administrator`	Can access all administration features, such as: • User and licensing management • Map management and configuration • Alert notification settings (email and SMS) • Dataflow monitoring • Creating global color scales (requires also a **focus** role)
`poweruser`	Can access `poweruser` features: • Can create new weather events. • Can create places of interest that are visible to all users in an organization, and adding weather events to monitor on these areas. (Only applies to the `root` organization.) • Can set up and manage pre-defined composites. • Can configure MVFs to be used in nowcasting. • Can select an organization-level map projection. (Only applies to the `root` organization.) All `poweruser` tasks are described in chapter *Poweruser tasks* in *IRIS Focus User Guide*.
`user`	Can access various features of the base software. This role must be assigned as an additional role to every user account with `focus`, `poweruser`, or `kiosk` role.
`kiosk`	Identical to the `User` role with the exception that an account with the `Kiosk` role will not be automatically logged out after a period of inacivity.

Considerations for assigning user roles

- `user` role should be assigned to every user account, even if they also have other roles.
- To create users that always enter the *IRIS Focus Light* view (so-called *"Light users"*), only assign system roles to these users. Do not assign Focus roles to them.
- Users with the `poweruser` role also need a `focus` role to access the full set of IRIS Focus features.
- To avoid reserving a `focus` license when performing administration tasks, the default `administrator` account does not have the `focus` role.
- To see both weather radar and lightning data, a user must have both `Focus Lightning` and `Focus Weather Radar` roles.

The following table shows the recommended user types (Admin, Poweruser, Focus user, Light user, and Kiosk user) and the roles recommended for them so that they can perform their tasks efficiently but without unnecessarily reserving Focus license seats.

Table 17 Recommended user types

	Focus Weather Radar / Focus Lightning	user	kiosk	poweruser	administrator
Admin		✔			✔
Poweruser	✔	✔		✔	
Focus user	✔	✔			
Light user		✔			
Kiosk user			✔		

More information

- IRIS Focus licensing (page 14)
- IRIS Focus licensing (page 14)

11.1.1 Managing user accounts

The system automatically creates a user account called image-export. Do not delete this account! This account is used by the system for performing automatic image exports. Deleting it disables the image exports.

1. Log in to the IRIS Focus web application as **administrator**.
2. Select **Admin** in the upper right corner.
3. Select **Users** to add, edit, or delete users.
4. When changing a user's role, the change will not take effect while the user is logged in. To log out the user, select **Logged In Users > Actions > Log out user**.

Administrator can also set and change the state of a user account:

- **Active**: the user account is active, and the user can log in to IRIS Focus.
- **Locked**: the user account is not active, and the user cannot log in. This is one way to deactivate user accounts. An alternative way is to delete the user account.
- **Expired**: the user has to change the password during the next login.

11.1.2 Creating user accounts after first install

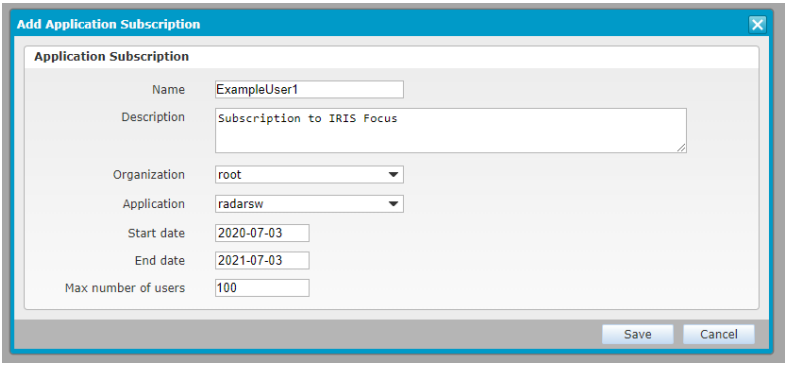
After a fresh installation, create the user accounts.

1. Log in to the IRIS Focus web application as **administrator**.
2. Select **Admin > Organizations**.

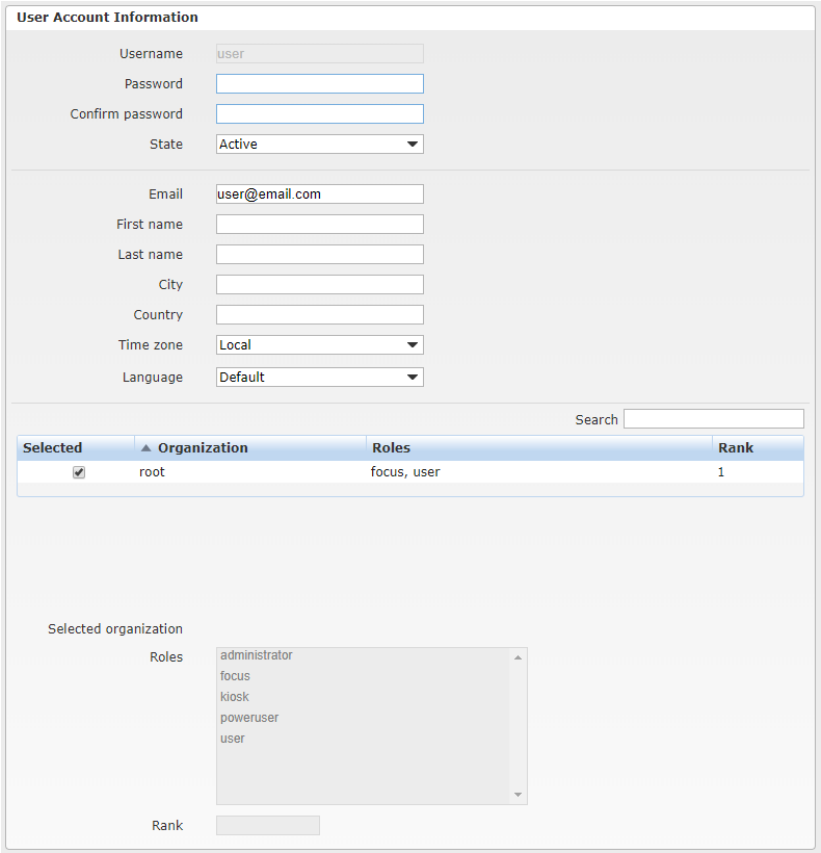
3. Choose which organization you want to create your users in.

> Users with **poweruser** or **administrator** role must belong to the **root** organization, and **root** organization must be assigned rank #1.

4. In the **Application Subscriptions** tab:
   a. Select the **radarsw** application.
   b. Enter the validity period.
   c. Enter the maximum number of users. This is the maximum number of users in this organization who can be logged in to IRIS Focus at the same time, including Focus users and Light view users.



5. To add users to the organization, select **Admin > Users > Add New User**.



a. Add user details.

b. Select an organization for user.

   If a user account belongs to multiple organizations, the user roles are applied according to the organization that has the highest **Rank**.

6. Assign roles to the user.

To avoid reserving an IRIS Focus license when performing administration tasks, the default administrator account does not have the **focus** role.

   a. In the organization list pane, make sure the organization is highlighted.
   b. In the **Roles** pane, select the role.

      To assign multiple roles to a user account, press **SHIFT**+**CTRL** and select roles from the list.
   c. To enable IRIS Focus features for a user account, select both the **user** and **focus** roles.
   d. To enable advanced IRIS Focus features, such as configuring events, select the **poweruser** role, which must be associated with the root organization, in addition to a Focus role.

11.1.3 Removing user accounts

1. Log in to the IRIS Focus web application as **administrator**.
2. Select **Admin > User > Users**.
3. Select a user and then **Delete**.

   The user is no longer listed as a user in IRIS Focus. However, the user name of the deleted account remains in the system database. This keeps log files intact, as references to deleted users remain in the audit logs.
   IRIS Focus does not allow you to create a new user with the same username as an existing one. This applies even when the account has been removed earlier, because the account name remains in the database.

11.1.4 Unlocking administrator account

If an **administrator** account is accidentally locked, unlock it as follows:

1. Log in to the server as **root**.
2. Run the following command:

```
rsw-db-tool reset-admin-password
```

11.2 Managing organizations

Each user account belongs to one or more organizations. You can use organizations to manage:

- How many users of each organization can be logged in at the same time.
- The visibility of organization-level places of interest: these are only visible to the members of the same organization as the poweruser who created them.

Users with `poweruser` or `administrator` role must belong to the `root` organization, and `root` organization must be assigned rank #1.

11.3 Map management

The standard installation of IRIS Focus includes a complete world map that is suitable for most scenarios.

The map consists of separate layers that are further separated into base layers and non-base layers. One base layer and one non-base layer are always rendered on the screen. Typically, base maps contain the underlying terrain and the non-base layers contain additional details that can be displayed on top of the base map.

Map data is served to the IRIS Focus web interface by GeoServer map server using Web Map Service (WMS) protocol. To improve performance, instead of calling for new map data each time the map view changes, the maps are cached in pre-rendered PNG tiles using GeoWebCache.

Administrators can add custom map layers or edit existing layers.

IRIS Focus users can select which map layers they see in the **Map** view, and edit the view by selecting **Map Features**.

11.3.1 Adding and editing map layers

1. Log in to the IRIS Focus web application as `administrator`.
2. Select **Admin > Map > Map Layers**.

   The **Map Layers** view lists the available map data layers. Each layer has the following properties:
   - **Base layer** - Enable to set this layer as a base layer
   - **Title** - Layer name
   - **Type** - WMS layers
   - **URL** - Address for the WMS server
   - **Layer** - Title of the layer on the server

3. To add a new layer, select **Add New Layer**.
   a. Type the layer information, including **Title**, **URL**, and **Layer**.
   b. Define map layer properties such as:
      - **Transparent** - Enable to use PNG or GIF alpha channel for transparency
      - **MIME type** - Select image type
   c. If you want to use a dark version of the layer with the dark mode map, create a separate dark layer with the same name, and append "`_dark`" at the end of the name. This name will be automatically requested when the user selects the dark map mode in the **Map Features** panel.

   When you are adding a WMS layer from an external source, note the following:
   - Get the URL from the layer provider.
   - You can set any values for **Realtime offset** and **Refresh rate**, but if the exact value is not available from the layer provider, the system will give you a time closest what you defined.
   - In order for the system to query for the cursor tool data, check the **Usable in map cursor tool** checkbox.
   - **Layer style** defines the availability of the color legend in the map view. IRIS Focus supports both `.sld` files and WMS methods of providing the legend.
   - If you do not want the layer to be visible to users, after adding a layer, go to the **Map View Contexts** screen, and uncheck the **Visibility** checkbox.
   - The user can see the added external WMS layer in the **Add Product** drop-down list of the **Weather Products** pane.
4. To edit a layer, select **Edit** for that layer and make your changes.

   The **Map Layer Information** window for that layer opens.
5. Select **Save**.

More information

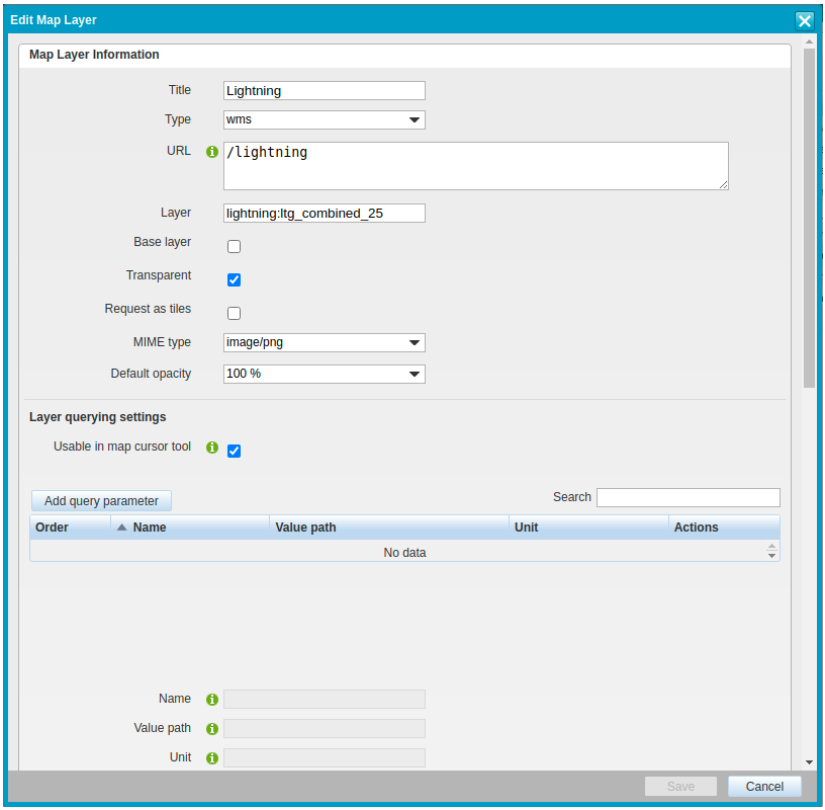
- Map layer configuration options (page 244)

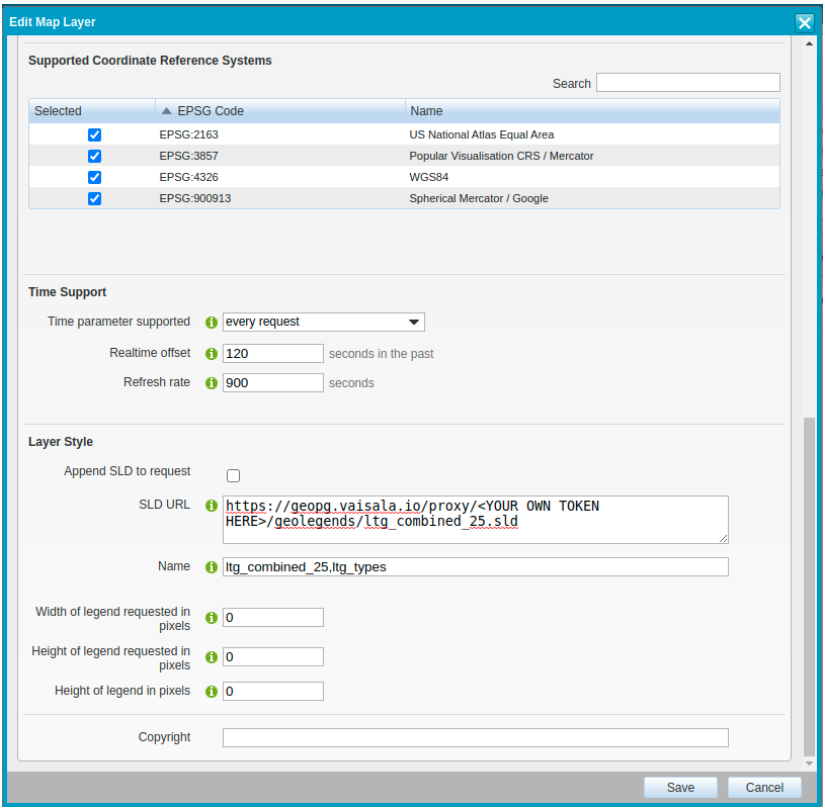
11.3.2 Adding GLD360 lightning layer

To take the GDL360 lightning layer into use, the IRIS Focus server must be online and your organization must have an active subscription to GLD360 data. For information on subscribing to GLD360 data, contact Vaisala Lightning Data Services.

1. Log in to the IRIS Focus web application as `administrator`.
2. Select **Admin > Map > Map Layers**.
3. Select **Add New Layer**.

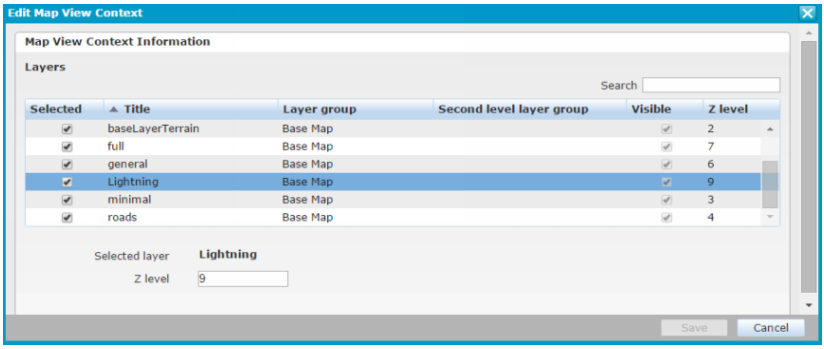
4. In **Map Layer Information**, enter the following values on the layer properties:
   a. **URL**: `/lightning`
   b. **Layer**: `lightning:ltg_combined_25`
   c. **Transparent**: Checkbox selected
   d. **Usable in map cursor tool**: Checkbox selected
   e. **SLD URL**: https://storm.vaisala.com/geolegends/ltg_combined_25.sld
   f. **Name**: `ltg_combined_25.ltg_types`





5. Select **Save**.
6. Select **Map > Map View Contexts**
7. Edit the default map context **TheMap**.

8. Select the newly created lightning layer and set its **Z level** higher than all base map layers in the map context.



In the web application, the new layer is listed in the product selection list.

More information

- GLD360 lightning layer (page 32)

11.3.3 Map View Context

The **Map View Contexts** view lists all defined maps.

Only the default TheMap context is available. Perform all map layer customization in the default TheMap context. Do not create new map contexts for custom map layers.

To edit TheMap, select **Edit**.

- To make a map layer available for users in the map view, select the **Selected** checkbox in the **Edit Map View Contexts**.
- To set the order in which multiple map layers are rendered on screen, change the **Z level** of map layers.
  The lowest number is rendered first, and higher numbers rendered on top of that.

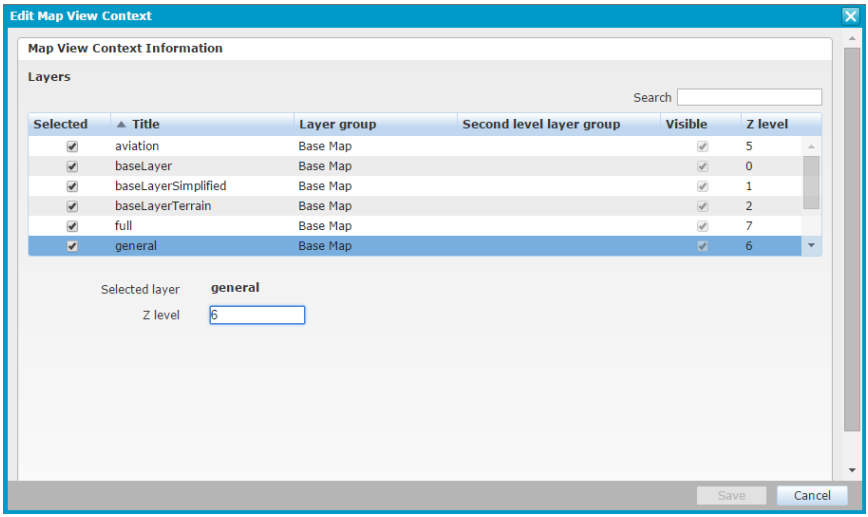


Figure 21 Editing the Map Context

11.3.4 Adding external map layers

You can import an external map layer, such as a shapefile, into Geoserver for IRIS Focus to display on the map.

For information on adding WMS layers from external sources, see *IRIS Focus Administrator Guide (M211850EN)*.

1. Make sure you have a shapefile (`.shp`) available.

   For an example resource with shapefiles available for download, see the WGS84 projection examples at:
   https://osmdata.openstreetmap.de/data/coastlines.html
2. Use an `scp` client or similar application to copy the shapefile to a directory on the IRIS Focus server such as `/srv/container/mnt/geoserver/inspire`.
3. Login to the server as `root`.
4. Open the file: `/etc/vaisala/radarsw/configuration/gis-override.ini`
5. Copy the `geoserver.admin.password` .

   This password is autogenerated during installation.

6. Using a browser, login to IRIS Focus Geoserver at:

   http://<IRIS_Focus_server_name>:24180/geoserver/web/
   Login using the username **`admin`** and the password you copied earlier.

   > Depending on your own network configuration you may need to do this at the server, over a remote console, or by using your local browser.

7. Add a new **Store**:
   a. Select **Stores > Add New Store**.
   b. Choose the data source:

      Shapefile - ESRI(tm) Shapefiles (*.shp)

   c. Select the following (the following list shows example values).

      - **Workspace**: `vaisala`
      - **Data Source Name**: `coastlines`
      - **Description**: leave blank
      - **Shapefile location**: browse to the shapefile
        For example: `\files\lines.shp`

   d. Leave the other fields as default.
   e. Select **Save**.
8. Publish the layer:
   a. Check that the **New Layer** menu opens.
   b. If the **New Layer** menu does not open automatically, select **Layers > Add New Layer**.
   c. In the **Add layer from** list, find the new layer.
   d. Select **Publish**.

   The **Edit Layer** menu shows the new layer name. For example, `vaisala:coastlines`.
9. In the **Edit Layer** menu:
   a. Leave all inputs as they are except:

      - **Name**: `coastlines`
      - **Title**: `coastlines`
      - **Coordinate Reference Systems > Declared SRS**
      - Select **Find** and search for `4326 (WSG 84)`.

   b. To fill the bounding boxes, select **Compute from data** and **Compute from native bounds**.
   c. Select **Save**.

10. Select **Layer Groups**.
    a. Select an existing layer group (for example, `vai_full_en`) and then select **Add Layer**.
    b. Find the new layer and add it.

       The layer is now listed in the **Layers** table.
    c. Select **Save**.
11. Login to IRIS Focus as `user`.
12. To confirm that the new layer is visible, select **Map Features > Map Detail > Full Detail**.
13. Open IRIS Focus UI and login as `administrator`.
14. Go to **Admin > Maps > Map layers > Add new layer**:
    a. Select the following:
       - **Title**: `coastlines`
       - **URL**: `/wms`
       - **Layer**: `vaisala:[layer_name]`
       - Select **Find** and search for `4326 (WSG 84)`.
       - **Save**
       - **Request as tiles**: `yes`
15. Go to **Admin > Maps > Map layers > Map view contexts** and edit **TheMap**.
16. Enable the layer by selecting it.
    a. Set **Z level** to something bigger than existing layers so that it would appear on top of the other map layers.
17. Go back to the application an reload the page.

More information

- GeoServer and maps (page 27)

11.4 Data Manager

Data manager is the HTTP/REST interface that provides raw data for on-demand (Live) radar products.

More information

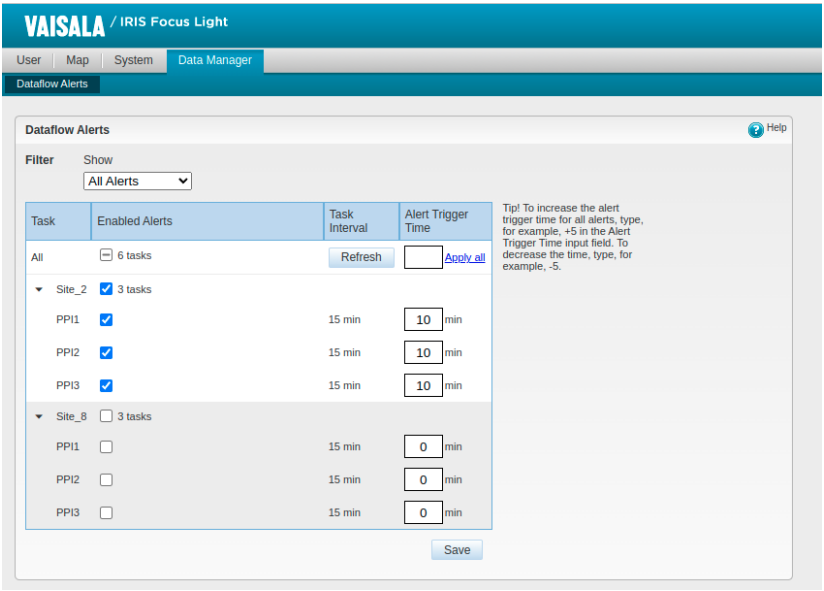
- Data Manager disk space requirements (page 22)
- Setting up Data Manager (page 60)
- Managing dataflow alerts (page 180)
- Viewing dataflow alerts (page 181)
- On-demand radar products (page 29)

11.4.1 Managing dataflow alerts

Enable and set-up data flow alerts to monitor the flow of radar data to IRIS Focus through Data Manager.

1. Log in on an account that has **`administrator`** rights.
2. Run the radar system for some time to allow the Data Manager database to populate.
3. Select **Admin > Data Manager > Dataflow Alerts**.

   The **Dataflow Alerts** page opens, and you can see Enabled Alerts.



Task
Radar task associated with the dataflow.

Alerts
If selected, IRIS Focus generates an alert if the dataflow for that task is interrupted.

Task Interval
Shows the interval between task run times (minutes).
Data Manager re-calculates the frequency automatically each time you open the **Dataflow Alerts** page. To refresh the times manually, select **Refresh**.
The timestamp shows the last detected date for received data.

Alert trigger time
The time (minutes) after which IRIS Focus generates an alert if the dataflow is interrupted.

4. To receive alerts about interruptions to the flow of task data:
   a. In the **Alerts** column, select the check box.
   b. In the **Alert trigger time** column, set a time that is higher than the expected data flow interval.
   c. To manage all enabled alerts in the same way, fill in the **Global alert trigger time**, and then select **Apply**:
      - To set the same trigger time for all alerts, type a number in the input field.
      - To increase the alert trigger time for all alerts, type, for example, +5 in the input field. To decrease the time, type, for example, -5.
      - To set a trigger time that is the same as detected interval between task run times for all alerts, leave the input field blank.
5. Select **Save**.

More information

- Data Manager (page 179)

11.4.2 Viewing dataflow alerts

If there is a break in the flow of radar product data, IRIS Focus sends a dataflow alert.

1. On the right side of the main menu, select **Alerts > Technical**.
2. In the **Alerts** pane, acknowledge the alert.

   The acknowledgement records who has seen the alert and when. Acknowledging alerts has no effect on the alert status.
3. Dataflow alerts are available to view in the **Alert history** area.

More information

- Data Manager (page 179)

11.4.3 Setting up Data Manager housekeeping service

When Data Manager exceeds its allocated disk space, the background housekeeping service begins to delete volume scans, starting from the oldest.

Data Manager disk space is allocated during installation, but can be modified later on.

1. Log in to the server as `root`.

2. Open the file */etc/vaisala/focus/k8s/vaisala-focus.yaml* in your favourite editor, for example vi or emacs, and edit the necessary parameters under *datamanager*:

```
volumedir:maxSizeMB: 66850
fixedDelay:          ms: 60000
fixedRate:          ms: 3600000
```

3. The disk space allocation for Data Manager is configured during installation. If you wish to change the allocation later on, use the **datamanager.volumeDir.maxSizeMB** parameter. For example: **datamanager.volumeDir.maxSizeMB = 1000**

```
datamanager.volumeDir.maxSizeMB = 1000
```

4. Define how often housekeeping checks for disk overusage (milliseconds).

```
datamanager.housekeeping.fixedRate.ms = 60000
```

   Vaisala recommends running this check once a day.
   While this checks runs, other operations on Data Manager slow down.

5. Define the delay for when housekeeping first runs after Data Manager has been started or restarted (milliseconds).

```
datamanager.housekeeping.fixedDelay.ms = 60000
```

6. After making any changes, run:

```
kubectl apply -f /etc/vaisala/focus/k8s/vaisala-focus.yaml
kubectl get all -n vaisala-focus | grep pod/data-manager (this will show
you the correct name for the next step)
kubectl delete pod <pod name> -n vaisala-focus        (example pod name:
data-manager-service-5c9cd95ccb-b8str)
```

11.4.4 Running Data Manager clear data script

Use the **rsw-data-manager-clear-data** script if the Data Manager data storage becomes corrupt or if there is a need to remove all data from Data Manager.

CAUTION! Running the script deletes all radar data from IRIS Focus, including Nowcasting configurations, pre-defined composite configurations, and RAW radar data.

1. Run the script:

```
DM_RESET=yes rsw-data-manager-clear-data
```

If there is a lot of RAW radar data in Data Manager, it may take some time to run the script.

CAUTION! Do not interrupt the script execution.

When the script is complete, Data Manager restarts automatically and you can continue using IRIS Focus.

11.5 Creating alert message log files

You can configure the system to create and send log files that contain information about each triggered alert. You can use these files, for example, in your message distribution systems to send alerts through channels not covered by the alert notification system.

The log files contain single-line JSON messages for each alert. The logs are created hourly. Messages are logged to an open log file as they appear. A delayed message may appear in a later log file.

You can customize the service: for example, how often new log files are created or whether empty log files are created.

By default, the log files are stored in the */srv/pv/log/alerts* directory.

There is no automated clean up of log files.

The service attempts to backfill using Kafka's group ID. You can take the service down for several minutes, and when you bring it back, it recovers any log messages that appeared during the outage and appends them to the active log file.

1. To enable the service, run the following command:

```
install -D -d /srv/pv/log/alerts
kubectl create -f /etc/vaisala/focus/k8s/vaisala-focus-alert-logger.yaml
```

2. To customize the service, modify the configuration file:

```
vi /etc/vaisala/focus/k8s/vaisala-focus-alert-logger.yaml
```

3. To disable the service, run the following command:

```
kubectl delete -f /etc/vaisala/focus/k8s/vaisala-focus-alert-logger.yaml
rm -fr /srv/pv/log/alerts # This is optional if you want to clear all
alert files
```

11.6 Installing a CA certificate

The web application comes with a temporary, self-signed SSL certificate that secures the connection between the IRIS Focus server and the user's web browser.

Consider acquiring and using a trusted certificate from a certificate authority (CA), especially if you plan to offer access to IRIS Focus outside your organization.

1. Acquire a certificate that has been signed by a trusted authority.

   This is usually done by an IT department or an external organization, who purchase the certificate from an external certificate authority (CA). You can use any trusted certificate authority.

   a. Create a certificate signing request (CSR).

      - The CN (Common Name) attribute is currently neither required and nor sufficient, so the certificate signing request must include the SAN attribute, with the DNS name of the service.
      - For details, contact the certificate authority that you are going to use.

   b. Send the CSR to the certificate authority to be signed.

   c. The certificate authority provides the certificate.

2. Back-up your current configuration by running:

```
run /usr/vaisala/radarsw/backup/bin/do-backups
```

   This backs up all the configuration files as a *.tar* file to */srv/vaisala/radarsw/backup/configuration*.

3. Install a copy of your certificate *pem* file under the */etc/vaisala/radarsw/webapp-proxy/certificates* directory.

   Give the file a name to match the host name that your users will use. Example: if your users connect to https://focus.acme.com/, use *focus.acme.com.pem* as the name for the pem file. IMPORTANT:

   CAUTION! Do NOT replace or remove the *localhost.pem* file in the directory, as this is required for inter-service connections.

   Use the following command:

   ```
   install -m 400 -o haproxy -g root focus.acme.com.pem /etc/vaisala/radarsw/
   webapp-proxy/certificates/focus.acme.com.pem
   ```

4. *Optional*: If you have other files related to the *pem* file that you want to keep organized, you can install them in the same directory. This is optional, as haproxy should ignore them. For example, if you have a *crt* and *key* file that correspond to your *pem* file, you can install copies of them:

   ```
   install -m 400 -o haproxy -g root focus.acme.com.crt /etc/vaisala/radarsw/
   webapp-proxy/certificates/focus.acme.com.crt
   install -m 400 -o haproxy -g root focus.acme.com.key /etc/vaisala/radarsw/
   webapp-proxy/certificates/focus.acme.com.key
   ```

5. Edit the */etc/haproxy/haproxy.cfg* configuration file so that the proxy server knows to offer the "*focus.acme.com.pem*" certificate to users that connect to "https://focus.acme.com/" and the *localhost.pem* certificate to local services that connect to "https://localhost/". To do this:

   a. Comment out the bind configuration line that binds all hosts to the same certificate file. To do this, insert a "#" symbol at the start of the line.

      In other words, change this line:

      ```
      bind *:443 ssl crt /etc/vaisala/radarsw/webapp-proxy/certificates/
      localhost.pem no-sslv3 ciphers EDH+aRSA+AESGCM:EDH+aRSA+AES:DHE-RSA-
      AES256-SHA:EECDH+aRSA+AESGCM:EECDH+aRSA+AES:ECDHE-RSA-AES256-SHA:ECDHE-
      RSA-AES128-SHA:RSA+AESGCM:RSA+AES+SHA:DES-CBC3-SHA:-DHE-RSA-AES128-SHA:!
      aNULL:!eNULL:!LOW:!MD5:!EXP:!PSK:!DSS:!RC4:!SEED:!ECDSA:!ADH:!IDEA
      ```

      to the following:

      ```
      #  bind *:443 ssl crt /etc/vaisala/radarsw/webapp-proxy/certificates/
      localhost.pem no-sslv3 ciphers EDH+aRSA+AESGCM:EDH+aRSA+AES:DHE-RSA-
      AES256-SHA:EECDH+aRSA+AESGCM:EECDH+aRSA+AES:ECDHE-RSA-AES256-SHA:ECDHE-
      RSA-AES128-SHA:RSA+AESGCM:RSA+AES+SHA:DES-CBC3-SHA:-DHE-RSA-AES128-SHA:!
      aNULL:!eNULL:!LOW:!MD5:!EXP:!PSK:!DSS:!RC4:!SEED:!ECDSA:!ADH:!IDEA
      ```

   b. Enable the two bind configuration lines that configure haproxy to use two separate certificates. To do this, uncomment two lines in the file, and change MY_DOMAIN to your fully qualified host name that users connect to ("focus.acme.com" in this example).

      Change:

      ```
      #  bind MY_DOMAIN:443 ssl crt /etc/vaisala/radarsw/webapp-proxy/
      certificates/MY_DOMAIN.pem no-sslv3 ciphers EDH+aRSA+AESGCM:EDH+aRSA
      +AES:DHE-RSA-AES256-SHA:EECDH+aRSA+AESGCM:EECDH+aRSA+AES:ECDHE-RSA-
      AES256-SHA:ECDHE-RSA-AES128-SHA:RSA+AESGCM:RSA+AES+SHA:DES-CBC3-SHA:-
      DHE-RSA-AES128-SHA:!aNULL:!eNULL:!LOW:!MD5:!EXP:!PSK:!DSS:!RC4:!SEED:!
      ECDSA:!ADH:!IDEA
      #  bind localhost:443 ssl crt /etc/vaisala/radarsw/webapp-proxy/
      certificates/localhost.pem no-sslv3 ciphers EDH+aRSA+AESGCM:EDH+aRSA
      +AES:DHE-RSA-AES256-SHA:EECDH+aRSA+AESGCM:EECDH+aRSA+AES:ECDHE-RSA-
      AES256-SHA:ECDHE-RSA-AES128-SHA:RSA+AESGCM:RSA+AES+SHA:DES-CBC3-SHA:-
      DHE-RSA-AES128-SHA:!aNULL:!eNULL:!LOW:!MD5:!EXP:!PSK:!DSS:!RC4:!SEED:!
      ECDSA:!ADH:!IDEA
      ```

      to the following (remove the leading comment character, and change MY_DOMAIN to your fully qualified host name):

```
bind focus.acme.com:443 ssl crt /etc/vaisala/radarsw/webapp-proxy/
certificates/focus.acme.com.pem no-sslv3 ciphers EDH+aRSA+AESGCM:EDH
+aRSA+AES:DHE-RSA-AES256-SHA:EECDH+aRSA+AESGCM:EECDH+aRSA+AES:ECDHE-RSA-
AES256-SHA:ECDHE-RSA-AES128-SHA:RSA+AESGCM:RSA+AES+SHA:DES-CBC3-SHA:-
DHE-RSA-AES128-SHA:!aNULL:!eNULL:!LOW:!MD5:!EXP:!PSK:!DSS:!RC4:!SEED:!
ECDSA:!ADH:!IDEA
bind localhost:443 ssl crt /etc/vaisala/radarsw/webapp-proxy/
certificates/localhost.pem no-sslv3 ciphers EDH+aRSA+AESGCM:EDH+aRSA
+AES:DHE-RSA-AES256-SHA:EECDH+aRSA+AESGCM:EECDH+aRSA+AES:ECDHE-RSA-
AES256-SHA:ECDHE-RSA-AES128-SHA:RSA+AESGCM:RSA+AES+SHA:DES-CBC3-SHA:-
DHE-RSA-AES128-SHA:!aNULL:!eNULL:!LOW:!MD5:!EXP:!PSK:!DSS:!RC4:!SEED:!
ECDSA:!ADH:!IDEA
```

6. Save your changes and restart the haproxy service:

```
systemctl restart haproxy
```

The *haproxy.cfg* file contains security and service mappings specific to each release of IRIS Focus. When you upgrade IRIS Focus to a newer release, you will likely need repeat steps 5 and 6 to enable your certificate.

More information

- Web application (page 33)
- Certificates (page 227)

11.7 Backing-up system configuration

IRIS Focus is backed up automatically using a daily configuration and database backup job that are run at 02:30 AM server time. On factory settings, the server uses UTC as the time zone.

The backup script stores the server configuration and application settings database.

Automatic back-up

Backup is done by the cron job `/etc/cron.d/vaisala-radarsw-backup-cron` that launches the `/usr/vaisala/radarsw/backup/bin/do-backups` script.

The created backup files are zipped and stored in the following directories:

- `/srv/vaisala/radarsw/backup/configuration`
- `/srv/vaisala/radarsw/backup/database`

Backups are kept for 180 days, after which they are deleted.

Each backup file includes a timestamp in the format:

```
radarsw-configuration-2019-09-05T06-48-26.tar.gz
```

11.7.1 Making a manual back-up

1. Log in to the server as **root**.
2. Run: **/usr/vaisala/radarsw/backup/bin/do-backups**
3. Check that new files are created in the following directories:

```
/srv/vaisala/radarsw/backup/configuration/radarsw-configuration-
<timestamp>.tar.gz
```

```
/srv/vaisala/radarsw/backup/database/radarsw-database-wx-<timestamp>.gz
```

```
/srv/vaisala/radarsw/backup/database/radarsw-database-vsp-<timestamp>.gz
```

```
/srv/vaisala/radarsw/backup/database/radarsw-database-keycloak-
<timestamp>.gz
```

Each backup file includes a timestamp in the format:

```
radarsw-configuration-2019-09-05T06-48-26.tar.gz
```

11.8 Restoring from backup

If you have lost your configuration files, you will need to restore them before you will be able to restore databases. To restore your configuration files from backup, you can find a recent configuration backup under the */srv/vaisala/radarsw/backup* directory to restore and then run the following command:

```
bd=/srv/vaisala/radarsw/backup/configuration
(cd / && tar xzf ${bd}/radarsw-
configuration-2019-10-12T07-54-50.tar.gz)
```

1. Log in to the server as **root**.
2. Stop the Monit service:

```
systemctl stop monit.service
```

3. Stop the IRIS Focus web application:

```
systemctl stop vaisala-radarsw-webapp.service
```

4. Stop all services which might access the database.

```
kubectl delete -f /etc/vaisala/focus/k8s/vaisala-focus.yaml
```

5. (Optional) Run the backup script:

   /usr/vaisala/radarsw/backup/bin/do-backups
   Database backups for the wx and vsp databases will be in */srv/vaisala/radarsw/backup/database*. Move a copy to a remote host if reinstalling or otherwise reimaging the machine.

6. Drop the current database with the **rsw-db-tool** utility:

```
rsw-db-tool drop-db
```

7. Drop the current database with the **rsw-vsp-db-tool** utility:

```
rsw-vsp-db-tool drop-db
```

8. Drop the current keycloak database using the rsw-api-auth-tool:

```
rsw-api-auth-tool delete-db --no-prompt
```

9. Recreate an empty wx database:

```
rsw-db-tool create-db
```

10. Create an empty keycloak database:

```
rsw-api-auth-tool create-db
```

11. Recreate an empty vsp database:

```
rsw-vsp-db-tool create-db
```

12. Copy your database backup files back to the Focus server and restore the database contents by reading the file contents into the standard output stream and inserting them in the IRIS Focus databases:

```
ext=2019-10-12T07-54-50.gz
pre=radarsw-database
gzip -dc ${pre}-vsp-${ext} | psql -d vsp_v1 -U vsp_user -h localhost
gzip -dc ${pre}-wx-${ext} | psql -d wxdb2 -U wxuser -h localhost
gzip -dc ${pre}-keycloak-${ext} | psql -d keycloak -U keycloak -h localhost
```

13. Restart services which might use database.

```
kubectl apply -f /etc/vaisala/focus/k8s/vaisala-focus.yaml
```

14. Start the IRIS Focus web application:

```
systemctl start vaisala-radarsw-webapp.service
```

15. Start the Monit service:

```
systemctl start monit.service
```

11.9 Server management software

If you are running server management software on your IRIS Focus server, make sure the management software's settings do not interfere with your intended network settings.

For example, in Dell PowerEdge servers, the integrated Dell Remote Access Controller (iDrac) sets a default static IP address for the server when it is first deployed.

On Vaisala preconfigured IRIS Focus systems, iDrac is disabled by default.

11.10 Licensing on server restart

Active sessions and their licenses are not stored when the IRIS Focus server is shut down.

When the server restarts, the licensing seats are allocated from scratch to users who log in. The total number of seats in the license pool is unaffected.

More information

- IRIS Focus licensing (page 14)

11.11 Reactivating the license after server upgrade

The product key in the IRIS Focus license is server-specific. If you upgrade your server, you must request a new service key and activate the new license.

1. Contact Vaisala and request a new server key.
2. Install IRIS Focus using the instructions in this guide.
3. Reactivate the license.

   Depending upon whether or not your server is connected to the internet, see:
   - Activating license – online (page 52)
   - Activating license – offline (page 55)

12. API in IRIS Focus

With IRIS Focus, you can allow access to an alert API service outside the browser. This allows you to leverage some of the features making up IRIS Focus in your own custom applications. In general, all API access follows these rules:

- Access is exposed through a TLS secured port (https on port 443).
- API access is denied by default.
- API access requires the creation of separate API account(s). No API accounts are created by default.
- API access requires authentication via a token retrieved from the authentication service.

12.1 API authentication

IRIS Focus requires all API clients to retrieve an API access token from the Keycloak authentication service before they are permitted to retrieve data from the desired API endpoint.

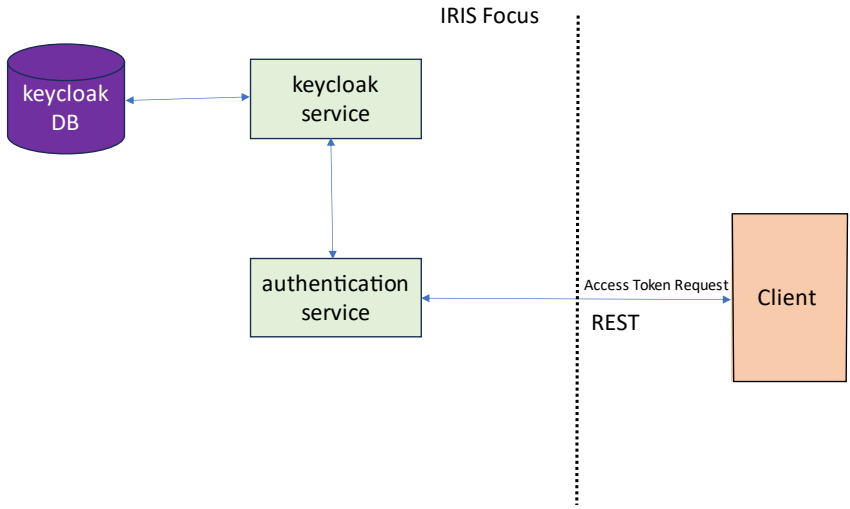


Figure 22 API authentication architecture in IRIS Focus

To retrieve an access token:

- An API account for the API client must first be created
- The API client must provide the correct credentials for the account
- The API client must be able to extract the API access token from the JSON response provided by the authentication service.

12.1.1 Managing API accounts

API accounts are managed from the command line of the IRIS Focus server using the **`rsw-api-auth-tool`** command. This command needs to be run as the **root** user to access protected files on the system.

Run **`rsw-api-auth-tool`** as the **root** user, or prefix it with **`sudo`**, as it needs elevated privileges to adjust the API accounts.

The **`rsw-api-auth-tool`** has a set of sub-commands. To see what subcommands are available, type:

```
rsw-api-auth-tool --help
```

To see additional information on the options available to any subcommand, specify the **`-help`** option after the subcommand:

```
rsw-api-auth-tool create-user --help
```

Some of the **`rsw-api-auth-tool`** commands produce JSON output, and most of the available API methods provide JSON output when returning information. When working with JSON output at the command line, the **jq** tool is indispensable. Its usage is often included in the examples in the following chapters. The following command will install the **jq** command onto your system if it is not installed:

```
[ -x /usr/bin/jq ] || dnf install -y /usr/bin/jq
```

12.1.1.1 Creating API accounts

To add a new API account, use the **`create-user`** command:

```
rsw-api-auth-tool create-user --realm alert --user testperson1
```

Using this method requires that you enter and then verify a password to be used for the API account. The API realm defaults to "**`alert`**", so you can omit that option when adding API accounts that need access to the alert API. If you do not want to be prompted to enter a password for the user, you can specify the password on the command line.

```
rsw-api-auth-tool create-user --user testperson1 --password 4Cpe-e6MB343yE25d
```

If the user account is created successfully, you will see a confirmation message:

```
Created user testperson1 under alert realm
```

If you accidentally attempt to create a user account that already exists, you will see a message that your request has been ignored as the user already exists.

```
User testperson1 under alert realm already exists, skipping
```

It is not possible to retrieve a forgotten password for an API account. If you need to reset an API account, first delete it and then create it.

If you want to keep a record of you API accounts and passwords, it is best to create a helper script with the file permission set to 700, so that only the **root** user can read the contents of the file. The following is an example of this approach to managing API accounts:

```
#!/bin/bash

alert_user() {
 rsw-api-auth-tool create-user --realm alert --user "${1}" --password "${2}"
}

alert_user testperson1 EY70-3a9c4XfaS02E
alert_user testperson2 rhWg-x7z9sSvFZw2J
alert_user testperson3 4Cpe-e6MB343yE25d
alert_user testperson4 1598-ET71WCXHo26d
```

If you save the above to a file named `create-api-accounts`, you can then set the permission and run the script to create all of your API accounts as needed. You can run the script multiple times as it will only create accounts that do not already exist.

```
chmod 700 create-api-accounts
./create-api-accounts
```

Running the above command produces an output like the following, indicating the accounts that were created and the accounts that had already existed.

```
User testperson1 under alert realm already exists, skipping
Created user testperson2 under alert realm
Created user testperson3 under alert realm
Created user testperson4 under alert realm
```

12.1.1.2 Deleting API accounts

Use the `delete-user` command to delete an API account:

```
rsw-api-auth-tool delete-user --realm alert --user testperson2
```

If the API account is deleted successfully, you will see a confirmation message:

```
Deleting existing user testperson2 with id:
e0d4b429-58ed-4091-9698-274efc7e53b0

Deleted user with id: e0d4b429-58ed-4091-9698-274efc7e53b0 from the alert realm
```

If you accidentally attempt to delete an API account that does not exist or was already deleted, you will see a message that your request has been ignored:

```
Did not find user testperson2 under alert realm, skipping delete
```

12.1.1.3 Listing API accounts

Use the **`get-users`** command to retrieve a list of users. The output returned is in JSON and will be easier to view on a terminal if you pipe it through the **`jq`** command.

```
rsw-api-auth-tool get-users --realm alert | jq
```

The output can be long. The following shows the first few lines of the output:

```
[
  {
    "id": "c1f8ce56-de6e-4228-a923-3a864f62889f",
    "createdTimestamp": 1692979498961,
    "username": "testperson1",
    ...
```

12.1.1.4 Listing alert state keys

An API account can get a list of all the keys that it has permission to monitor by making the following request to the **`alert-api`** service:

```
TOKEN_FILE=$HOME/alert-token.json
ALERT_API_URL="https://localhost/focus-alert/api/v1"

curl -D ~/headers.log --insecure -X 'POST' "${ALERT_API_URL}/alerts/keys" \
-H 'accept: application/json' -H 'Content-Type: application/json' \
-d '{
"token": "'$(jq -r '.access_token' ${TOKEN_FILE})'"
}' | jq
```

In this example, the following alert state keys are returned:

```
[
  {"user":"testperson1","area":"Helsinki Downtown","event":"Heavy Rain"},
  {"user":"testperson1","area":"Helsinki Downtown","event":"Lightning"},
  {"user":"testperson1","area":"Helsinki Downtown","event":"Lightning
Threat"},
  {"user":"testperson1","area":"Helsinki Suburban","event":"Heavy Rain"},
  {"user":"testperson1","area":"Helsinki Suburban","event":"Lightning
Threat"},
  {"user":"@global","area":"Helsinki Airport","event":"Heavy Rain"},
  {"user":"@global","area":"Helsinki Airport","event":"Lightning"}
]
```

The keys in the example above correspond to the following alert states that are being monitored:

- The **`Helsinki Downtown`** area, created by **`testperson1`**, is being monitored for the **`Heavy Rain`**, **`Lightning`**, and **`Lightning Threat`** events.

- The **Helsinki Suburban** area, created by **testperson1**, is being monitored for the **Heavy Rain** and **Lightning** events.
- The **Helsinki Airport** area, created by a Focus user with the **poweruser** role and therefore globally available to all API accounts, is being monitored for the **Heavy Rain** and **Lightning** events.

12.1.2 Clearing the Keycloak database

If you want to start with an empty set of API accounts, you can delete the Keycloak service, delete the Keycloak database, create a new empty Keycloak database, and then start the Keycloak service up.

All these commands must be run as the **root** user. Access to the exposed API services will be unavailable until you start the Keycloak service up again and add the necessary API accounts.

To stop the Keycloak service, type:

```
kubectl --namespace vaisala-focus delete deployment keycloak
```

To verify that the keycloak service is stopped, use the **kubectl get** command:

```
kubectl --namespace vaisala-focus get deployment keycloak
```

If the Keycloak service is still running, you see something like:

```
NAME       READY   UP-TO-DATE   AVAILABLE   AGE
keycloak   1/1     1            1           2d1h
```

If the Keycloak service is stopped, you see:

```
Error from server (NotFound): deployments.apps "keycloak" not found
```

After the Keycloak service has been stopped, run the following command to remove and then create a fresh Keycloak database:

```
rsw-api-auth-tool recreate-db --no-prompt
```

You can then apply the yaml configuration file that defines the Keycloak service to bring the keycloak service back up.

```
kubectl apply -f /etc/vaisala/focus/k8s/vaisala-focus.yaml
```

Once the Keycloak service is up and running, you should have an empty database. You can verify this by requesting the list of API accounts:

```
rsw-api-auth-tool get-users --realm alert | jq
```

It should return an empty JSON array:

```
[]
```

12.1.3 Keycloak system accounts

There are two system accounts associated with the Keycloak service. They are initialized with random passwords at the time of installation. These account names and random passwords are `base64` encoded in the *`vaisala-focus.yaml`* file, which is located in the */etc/vaisala/focus/k8s* directory. The *`vaisala-focus.yaml`* file is owned by the **root** user, and only the root user is able to read its contents.

You should never need to use these accounts directly. They are created for the Keycloak service and intended to be used by the keycloak service.

The following table describes these two Keycloak system accounts:

Account	Usage
keycloak	The account used by the Keycloak service to access the Keycloak database used to manage API accounts.
admin	The administrative Keycloak account used by the `rsw-api-auth-tool` when managing the IRIS Focus API accounts.

12.1.4 API login request and response

In order to securely access the API, the user needs to provide credentials. The username and password can be configured in the IRIS Focus application.

For login, you need to make a **POST** request to IRIS Focus.

The base url for the authentication service is: */focus-webapp/api/v2/alert-api/login*.

The parameters to be included with the request body are expected to be part of an *JSON* encoded format. The body encoding should be UTF-8.

Parameter name	Value type	Use	Description
Query parameters			
Do not send any query parameters to this resource.			
Request body			
username	String	mandatory	A valid application username
password	String	mandatory	A valid password for the provided *username*

Response

If the request is valid and access is granted, the response body will contain the access token and other useful metadata related to it as a *JSON* message.

Response body		
`access_token`	String	OAuth 2.0 compliant access token. Example: `"MTQ0NjJkZmQ5OTM2NDE1ZTZjNGZmZjI3"`
`token_type`	String	The type of the token. Example: `"Bearer"`
`expires_in`	Integer	Duration of time the access token is granted for (in seconds).
`refresh_token`	String	Token for refreshing the access token. Example: `"IwOGYzYTlmM2YxOTQ5MGE3YmNmMDFkNTVk"`)
`scope`	String	Scope the client is granted access to.

If the request is invalid and access is denied, the response body will be sent as a *JSON* message containing attributes related to the error.

Response body		
error	String	Error type (example: "invalid_request", "unauthorized_client")
error_description	String	A sentence or two (max), describing the circumstance of the error
error_uri	String	Link to online documentation (example: `"See the full API docs at..."`)

In the case of a failure the endpoint will respond with a **HTTP 400 error code**.

HTTPS is used for accessing this endpoint.

12.2 API access tokens

After an API account has been created for the API service, the API client will be able to:

- Request an access token from the authentication service that will grant the client access to the API service for a period.
- Optionally extend the life of an access token if the client needs it for a longer period.
- Release an access token when access is no longer needed. This is an optional step, as the access token will automatically be released after it expires.

The following sections provide examples that demonstrate how these tasks can be done from the command line using simple curl requests. These examples assume that you have the **jq** tool installed and created an API account named **testperson1** as shown below. It is also assumed that you run the example commands locally on your IRIS Focus server. You can copy and paste these commands to your command line from the PDF version of this document.

```
rsw-api-auth-tool create-user --user testperson1 --password 4Cpe-e6MB343yE25d
[ -x /usr/bin/jq ] || dnf install -y /usr/bin/jq
API_USER="testperson1"
API_PASS="4Cpe-e6MB343yE25d"
BASE_URL="https://localhost"
API_URL="${BASE_URL}/focus-alert/api/v1"
AUTH_URL="${BASE_URL}/focus-authentication/api/v1"
TOKEN_FILE="${API_TOKEN_FILE:-$HOME/alert-token.json}"
```

12.2.1 Requesting an access token

Before an API client can access the alert API service, it must request an access token from the authentication service. The following example uses the **curl** command to demonstrate how the JSON token file can be retrieved from the authentication service by the API client. The JSON response will be stored in *TOKEN_FILE* as well as formatted and displayed to the screen. For diagnostic purposes, the HTTP headers are stored in the *~/headers.log* file and the insecure option is specified with the assumption that you have not installed a valid TLS certificate on your IRIS Focus server. Both options can be removed when you have a fully configured and working IRIS Focus installation.

```
curl -D ~/headers.log --insecure --request POST \
  --url ${AUTH_URL}/auth/keycloak/token \
  --header 'Content-Type: application/json' \
  --data '{
"username":"'${API_USER}'",
"password":"'${API_PASS}'",
"applicationName":"alert",
"grantType":"password"
}' | tee ${TOKEN_FILE} | jq
```

If things are working properly, you should receive a JSON response that looks like the following:

```
{
  "access_token": "eyJh ... random characters",
  "expires_in": 300,
  "refresh_expires_in": 1800,
  "refresh_token": "eyJh ... random characters",
  "token_type": "Bearer",
  "not-before-policy": 0,
  "session_state": "6ec96a62-3af4-49be-92ac-04218b382f3b",
  "scope": "profile email"
}
```

The **access_token** from the JSON response needs to be passed along to each alert API request. The **access_token** can be used more than once, but it will expire after the number of seconds reported in the JSON response (the **expires_in** value is shown as 300 seconds in the above output).

You can extract the **access_token** from the **TOKEN_FILE** using the following **jq** command. This strategy is used throughout the document when demonstrating the use of tokens in **curl** requests.

```
jq -r '.access_token' ${TOKEN_FILE}
```

12.2.2 Extending the life of access token

An access token has a limited lifetime, indicated by the **expires_in** attribute (in seconds). After an access token expires, an API client will need to request a new access token.

Alternatively, an API client can extend the lifetime of an access token by passing the **refresh_token** value back to the authentication service. The following demonstrates the **HTTP POST** request that takes the **refresh_token** value from the original ***TOKEN_FILE***, and passes it back to the authentication service to request.

```
REFRESH_TOKEN="$(jq -r '.refresh_token' ${TOKEN_FILE})"
curl -D ~/headers.log --insecure --request POST \
  --url ${AUTH_URL}/auth/keycloak/token/refresh \
  --header 'Content-Type: application/json' \
  --data '{
    "applicationName":"alert",
    "token":"'"${REFRESH_TOKEN}"'"
}' | tee "${TOKEN_FILE}" | jq
```

Each token refresh returns a new value for both the access and refresh token. That is why the **refresh_token** is extracted from the ***TOKEN_FILE*** before making the request and writing the new values back to the ***TOKEN_FILE***.

You can extend the life of an access token significantly, but not indefinitely. Your API client needs to be ready to request a new access token if a refresh request fails.

Refreshing token values reduces the number of times that API clients need to provide the API account credentials, but adds complexity to the API client implementation.

12.2.3 Releasing an access token

When an API client no longer needs the access token, an **HTTP POST** request can be made to notify the authentication service that API access is no longer needed. After this, the access token is no longer usable. The API client will need to request a new access token before it can be permitted to access the API service. This is an optional step, as the access token will automatically be closed out after it expires. However, closing out an access token as soon as possible is a good security practice.

The following `curl` command demonstrates how to issue a HTTP POST request to close out and release an active access token.

```
curl -D ~/headers.log --insecure --request POST \
  --url ${AUTH_URL}/auth/keycloak/token/delete \
  --header 'Content-Type: application/json' \
  --data '{
    "applicationName":"alert",
    "token":"'"$(jq -r '.refresh_token' ${TOKEN_FILE})"'"
}' | jq
```

You receive a JSON response from the service indicating that you have been logged out and that the access token is no longer usable:

```
{
  "value": "logged out"
}
```

12.3 Alert API service

IRIS Focus supports sending alert state change updates from IRIS Focus to other systems and applications. The service can be accessed via a **WebSocket** request or a **REST POST** request for pulling the full summary. The implementation of the request is up to the client.

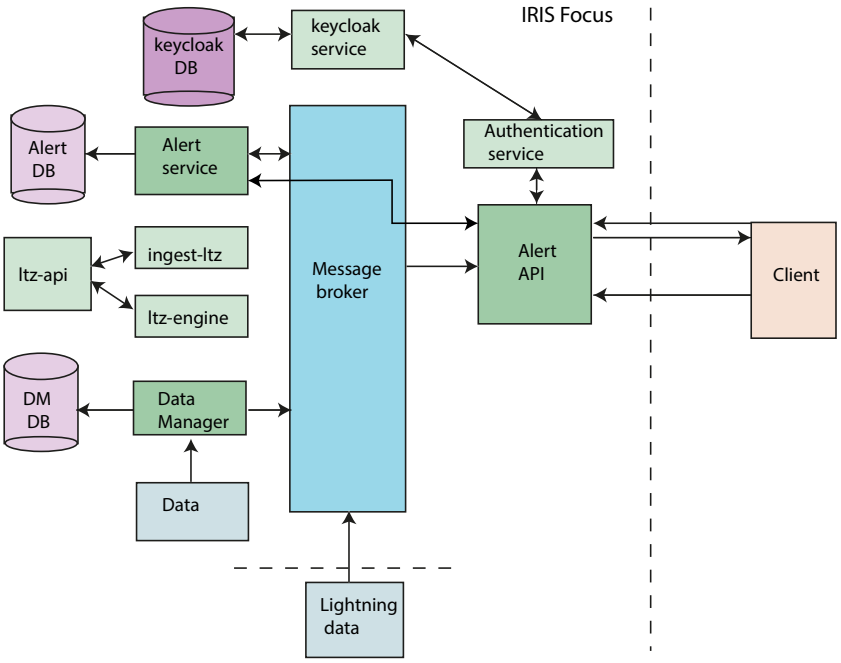


Figure 23 Alert API architecture in IRIS Focus

The response from the server is in *JSON* format. The status message is used by both the socket and **REST** connections. The difference is that for the **REST** status, the client receives a full list of messages at once, and on the real-time WebSocket connection, the initial alert states are sent to the client at initial connection, and then changes in the alert state are sent as they occur, one by one.

By default, the `alert-api` service is disabled, as it is only required when you want to expose access to the alert states to external connections.

12.3.1 HTTP POST request versus WebSocket application

The HTTP POST request is in some cases a useful solution for gathering information or polling the current alert states, but there are some disadvantages to using it:

- The API client does not know that an alert state has changed until its next poll cycle.
- The API client may miss alert state transitions if the alert state changes an even number of times between the API client's poll cycles. (For example, the state changes twice: from inactive to active, and then back to inactive.)
- Polling requires more effort to maintain access tokens.
- Polling tends to add more load to both the client and server implementations.

To provide a better alternative than polling, the **api-alert** service enables API clients to establish a standard WebSocket connection to the server. The WebSocket connection has the following features:

- The API client connects to the **alert-api** WebSocket service.
- The API client sends a **JSON** message including an access token and list of alert state keys (filters) to be monitored.
- The **api-client** service is active on the background. It first receives the current state of alerts from the server, and then any state changes that occur afterwards.

12.3.2 Filtering

Both the REST endpoint request and the WebSocket rely on a **filter** parameter for specifying which alert configurations are included in the alert state update messages. The filter can be a single definition or a list of definitions. The format for the filter is a JSON array of **user**, **area**, and **event** fields:

- **user** = username of the user that created the alert state configuration
  - **@global** for the global alert states configured by any poweruser
  - **@technical** for system alerts, such as the loss of communications from a radar site
  - * for everything the API account has access to
- **area** = name of the area of interest, or * for everything the API account has access to
- **event** = name of the event, or * for everything the API account has access to

API accounts are permitted access to all the **@global** and **@technical** alert states, but they are only permitted access to the private alert state of the IRIS Focus user account that matches the API account name. For example, an API account named "**person1**" permits access to the personal alert states created by the **person1** IRIS Focus user. But this account does not permit access to the alert states created by the **person2** IRIS Focus user.

Examples

The following example shows a JSON array that has a single filter key. With this filter, you can monitor all alert states the API account has access to. This is the most useful filter for most API clients.

```
[
  {"user": "*", "area": "*", "event": "*"}
]
```

The following example shows a JSON array that has two entries indicating that the API client wants information for all the personal alerts created by the IRIS Focus user "**person2**" and for Lightning alerts in all the global areas set up by the powerusers.

```
[
    {"user": "person2", "area": "*", "event": "*"},
    {"user": "@global", "area": "*", "event": "Lightning"}
]
```

12.4 WebSocket connection

The WebSocket solution is useful for notifying third parties who do not have access to IRIS Focus about alert state changes in real time.

The response message size is small, but the frequency can vary depending on the client configuration.

HTTPS is used for the subscription. After the initial connection, WSS is used for the socket.

Subscription

The request needs to contain a valid access token in its header. The client needs to first obtain the access token from the login endpoint of the API. The request also needs a filter parameter.

This endpoint requires a secure WebSocket (WSS) connection.

The base url for the alert API WebSocket connection is *`wss://localhost/focus-alert/ws/v1/monitor`*.

You will need to change **`localhost`** to the name of your IRIS Focus system when running the command on an external system.
Once connected, you will need to send a *`JSON`* message containing two attributes (a token for access and a list of 1 or more alert state keys).

See JSON messages used with the alert API (page 211) for details on the *`json`* messages exchanged between the client and the server.

12.4.1 Example Python implementation of API client code

You can find a sample program named *`focus-alert-api-monitor.py`* under the */srv/vaisala/focus/examples/alert-api* sub-directory of the IRIS Focus installation directory. If you choose to implement a custom alert API client in Python, this sample program provides the basic building blocks. The program demonstrates the following:

- How to request an access token from the authentication service.
- How to establish a WebSocket connection to the alert API service.
- How to send a message to the alert API service containing the access token and the list of alert state keys (filters) that are of interest to your client.
- How to use command line arguments to allow for adjustments and parameters to your API client.
- How to work around certificate issues while waiting for a valid certificate to be installed on your IRIS Focus server.

The following provides a minimal Python implementation that assumes the access token is provided as an environment variable. This minimal example demonstrates the following parts of using the alert API WebSocket endpoint:

- How to open a WebSocket connection.
- How to send a message providing an access token and enabling the monitoring of all alert states.

- Printing out the alert state key and state received from the server.

The examples included require the Python websocket library, which may not be installed by default. To install it on an AlmaLinux system, run the command: **`sudo dnf install python3-websocket-client`**.
On other systems, you may need to use the **`pip`** package installer.

You can copy and paste this minimal implementation to the ***`alert-api-websocket-client.py`*** file that is located under the ***`/srv/vaisala/focus/examples/alert-api`*** directory.

```
import datetime
import json
import math
import os
import ssl
import sys
import websocket

WS_URL = "ws://localhost:31000/focus-alert/ws/v1/monitor"
ACCESS_TOKEN = os.getenv("ACCESS_TOKEN", "export ACCESS_TOKEN")

  def on_message(ws, message):
    alert_state = json.loads(message)
    k = alert_state['key']
    a = 'ACTIVE  ' if alert_state['isActive'] else 'inactive'
    epoch_secs = math.floor(alert_state['lastChange'] / 1000)
    t = datetime.datetime.utcfromtimestamp(epoch_secs).isoformat() + 'Z'
    print(f"{t} {a} {k['user']}:{k['area']}:{k['event']}")

def on_error(ws, error):
    print(f'WebSocket error: {error}', file=sys.stderr)

def on_close(ws, close_status_code, close_msg):
    print(f'WebSocket closed ({close_status_code}: {close_msg}))', file=sys.stderr)
    sys.exit(0)

def on_open(ws):
    keys = [{"user":"*", "area":"*", "event": "*"}]
    message = json.dumps({"keys": keys, "token": ACCESS_TOKEN})
    ws.send(message)

if __name__ == "__main__":
    conn = websocket.WebSocketApp(WS_URL, on_open=on_open,
                                  on_message=on_message,
                                  on_error=on_error, on_close=on_close)
    conn.run_forever(sslopt={"cert_reqs": ssl.CERT_NONE},
                     ping_interval=60, ping_timeout=10)
```

You can run this code on your IRIS Focus server by first requesting an access token from the authentication service and storing the value returned in an **`ACCESS_TOKEN`** environment variable. These instructions assume that you have the **`API_USER`**, **`API_PASS`** and **`TOKEN_FILE`** variables set as explained in the earlier examples.

```
curl --insecure --request POST --url ${AUTH_URL}/auth/keycloak/token \
     --header 'Content-Type: application/json' \
     --data '{
  "username":"'${API_USER}'",
  "password":"'${API_PASS}'",
  "applicationName":"alert",
  "grantType":"password"
}' >| ${TOKEN_FILE}

  export ACCESS_TOKEN="$(jq -r '.access_token' ${TOKEN_FILE})"
```

If you saved the python codeto the *alert-api-websocket-client.py* file, you can then run the following command:

```
python3 alert-api-websocket-client.py
```

You will see the initial alert states for all the alert state keys that the **API_USER** has access to. If you leave the sample client code running, you will see new messages whenever there is a change in an alert state.

```
2023-08-28T16:32:43Z inactive testperson1:Downtown Helsinki:Heavy Rain
2023-08-28T16:44:00Z ACTIVE    testperson1:Helsinki Suburban:Lightning Threat
2023-08-28T15:26:07Z inactive  @global:Helsinki Airport:Heavy Rain
2023-08-28T16:53:08Z ACTIVE    testperson1:Downtown Helsinki:Lightning
2023-08-28T04:00:00Z inactive  @global:Helsinki Airport:Lightning
2023-08-28T04:00:00Z ACTIVE    testperson1:Helsinki Suburban:Heavy Rain
2023-08-28T16:44:00Z ACTIVE    testperson1:Downtown Helsinki:Lightning Threat
2023-08-28T17:02:46Z inactive  testperson1:Downtown Helsinki:Lightning
```

The example shows that the system returned the first seven lines immediately. The last line (that has the same key as the fourth line) appeared later when the **Lightning** alert state for the **Downtown Helsinki** area changed to the inactive state.

12.4.2 Example JavaScript implementation of API client code

A simple JavaScript/HTML example of using the authentication service to request an access token and the alert API WebSocket service is available under the */srv/vaisala/focus/examples/alert-api* sub-directory of the IRIS Focus installation directory.

Table 18

File	Description
alert-api-websocket.js	A JavaScript file that demonstrates how to fetch an access token and then initiate a WebSocket connection to the alert-api service and monitor alert states.
alert-api-websocket.html	An HTML file that loads the JavaScript into a web browser and provides some additional information regarding certificates.

12.5 REST endpoint

IRIS Focus provides a REST endpoint for one-time query operations. The endpoint can be used in order to get information about which alerts are active at any given time. The filtering parameter needs to be used.

HTTPS is used for accessing this endpoint.

Request

The request needs to contain a valid access token and a list of alert state keys in a *JSON* encoded message than it submits to the server as a HTTP POST request. The access token needs to be requested from the authentication service as described in the API authentication section found earlier in this document.

This alert API endpoint to POST the request to is: *https://localhost/focus-alert/api/v1/alerts/states*.

You will need to change *localhost* to the name of your IRIS Focus system when running the command on an external system.

See JSON messages used with the alert API (page 211) for details on the *JSON* messages exchanged between the client and server.

Response

The response from the server is in *JSON* format. The response contains a list of alert configurations and the last state change for each.

12.5.1 Variables for curl examples

In the following sections, you will come across several examples of using the curl command to retrieve information from the alert-api service.

The examples assume that the following commands have been run to setup a test user and test account:

```
[ -x /usr/bin/jq ] || dnf install -y /usr/bin/jq
API_USER="testperson1"
API_PASS="4Cpe-e6MB343yE25d"
BASE_URL="https://localhost"
ALERT_API_URL="${BASE_URL}/focus-alert/api/v1"
AUTH_URL="${BASE_URL}/focus-authentication/api/v1"
TOKEN_FILE="${$HOME/alert-token.json}"
rsw-api-auth-tool create-user --user "${API_USER}" --password "${API_PASS}"
```

The examples also assume that a valid access token can be found in the file specified by the **TOKEN_FILE** variable. After copying and pasting the variables above, you should be able to create the **TOKEN_FILE** with the following curl command:

```
curl -D ~/headers.log --insecure --request POST \
  --url ${AUTH_URL}/auth/keycloak/token \
  --header 'Content-Type: application/json' \
  --data '{
  "username":"'${API_USER}'",
  "password":"'${API_PASS}'",
  "applicationName":"alert",
  "grantType":"password"
}' | tee ${TOKEN_FILE} | jq
```

When your access token expires, you will need to repeat the curl request shown above in order to get a new access token.

12.5.2 Requesting a single alert state

To request the alert state associated with a specific key, you need to build a **HTTP POST** request that contains a single key with the corresponding matching fields. For example, to see if the **Heavy Rain** event, configured in IRIS Focus by a poweruser, is active at the **Helsinki Airport** area, use the following command to specify the exact key of the alert state:

```
curl -X 'POST' "${ALERT_API_REST}/alerts/states" \
  -H 'accept: application/json' -H 'Content-Type: application/json' \
  -d '{
  "token": "'$(jq -r '.access_token' ${TOKEN_FILE})'",
  "keys": [
    {"user": "@global", "area": "Auckland Airport", "event": "Heavy Rain"}
  ]
}' | jq
```

The **alert-api** will return at most one alert state to the request above. If the alert state key does match an alert state key that the API account is permitted to monitor, a JSON array containing a single alert state will be returned, such as the following:

```
[
  {
    "key": {"user":"@global","area":"Helsinki Airport","event":"Heavy Rain"},
    "lastChange": "2023-08-25T04:00:00.000+0000",
    "attributes": {"id":"1","poweruser":"true"},
    "isActive": false
  }
]
```

When the **isActive** field has the value **false**, this indicates that **Heavy Rain** is NOT currently detected at **Helsinki Airport**.

If the key you specify does not match an alert state key that the API account has access to, the system will return an empty list:

```
[ ]
```

12.5.3 Requesting a set of alert states

Multiple alert state keys may be included when creating a `HTTP POST` request. The following example shows three keys, two of which include wildcards (`*`).

```
curl -X 'POST' "${ALERT_API_REST}/alerts/states" \
  -H 'accept: application/json' -H 'Content-Type: application/json' \
  -d '{
  "token": "'$(jq -r '.access_token' ${TOKEN_FILE})'",
  "keys": [
    {"user": "@global", "area": "Helsinki Airport", "event": "*"},
    {"user": "@global", "area": "*", "event": "Heavy Rain"},
    {"user": "testperson1", "area": "Helsinki Suburban", "event": "Heavy
Rain"}
  ]
}' | jq
```

In this example, 3 alert states are returned, none of which are currently active.

```
[
  {
    "key": {"user":"@global","area":"Helsinki Airport","event":"Heavy Rain"},
    "lastChange": "2023-08-25T04:00:00.000+0000",
    "attributes": {"id":"1","poweruser":"true"},
    "isActive": false
  }, {
    "key": {"user":"@global","area":"Helsinki Airport","event":"Lightning"},
    "lastChange": "2023-08-28T09:19:25.942+0000",
    "attributes": {},
    "isActive": false
  }, {
    "key": {"user":"testperson1","area":"Helsinki Suburban","event":"Heavy
Rain"},
    "lastChange": "2023-08-25T04:00:00.000+0000",
    "attributes": {"id":"8","poweruser":"false"},
    "isActive": false
  }
]
```

12.5.4 Requesting all alert states

To request all alert states, include a single key where each field is set to match any string (`*`). Example:

```
curl -X 'POST' "${ALERT_API_REST}/alerts/states" \
  -H 'accept: application/json' -H 'Content-Type: application/json' \
  -d '{
  "token": "'$(jq -r '.access_token' ${TOKEN_FILE})'",
  "keys": [
    {"user": "*", "area": "*", "event": "*"}
  ]
}' | jq
```

Because the access token was issued for the **testperson1** API account, the **alert-api** service returns the alert state for all alert monitoring conditions created by the **testperson1** user, as well as all global alert conditions available to all API accounts. The example output below shows that:

- IRIS Focus has two global alert states
- **testperson1** account has five alert states.
- Most alert states are currently inactive
- The only active alert state is the **Lightning** occurring on the **Helsinki Downtown** area set up by **testperson1**.


```
[
  {
    "key": {"user":"@global","area":"Helsinki Airport","event":"Heavy Rain"},
    "lastChange": "2023-08-25T04:00:00.000+0000",
    "attributes": {"id":"1","poweruser":"true"},
    "isActive": false
  }, {
    "key": {"user":"@global","area":"Helsinki Airport","event":"Lightning"},
    "lastChange": "2023-08-28T09:19:25.942+0000",
    "attributes": {},
    "isActive": false
  }, {
    "key": {"user":"testperson1","area":"Helsinki Downtown","event":"Heavy
Rain"},
    "lastChange": "2023-08-25T04:00:00.000+0000",
    "attributes": {"id":"5","poweruser":"false"},
    "isActive": false
  }, {
    "key": {"user":"testperson1","area":"Helsinki
Downtown","event":"Lightning"},
    "lastChange": "2023-08-28T08:51:57.520+0000",
    "attributes": {},
    "isActive": true
  }, {
    "key": {"user":"testperson1","area":"Helsinki Downtown",
            "event":"Lightning Threat"},
    "lastChange": "2023-08-25T04:00:00.000+0000",
    "attributes": {"id":"7","poweruser":"false"},
    "isActive": false
  }, {
    "key": {"user":"testperson1","area":"Helsinki Suburban","event":"Heavy
Rain"},
    "lastChange": "2023-08-25T04:00:00.000+0000",
    "attributes": {"id":"8","poweruser":"false"},
    "isActive": false
  }, {
    "key": {"user":"testperson1","area":"Helsinki Suburban","event":"Lightning
Threat"},
    "lastChange": "2023-08-25T04:00:00.000+0000",
    "attributes": {"id":"9","poweruser":"false"},
    "isActive": false
  }
]
```

12.6 JSON messages used with the alert API

12.6.1 All keys: request and response

Request all keys

When requesting the list of all alert state keys from the **alert-api** service, you will need to POST a JSON message in the following form:

```
{
  "token": "ACCESS_TOKEN_FROM_AUTHENTICATION_SERVICE"
}
```

Attribute	Description
token	The access_token received from the authentication API service.

Response all keys

The **alert-api** service responds to an alert state keys requests with a JSON array of 0 or more alert state keys that the API account associated with the access token permits access to. The JSON message has the following form:

```
[
  {"user": "USER1", "area": "AREA1", "event": "EVENT1"},
  {"user": "USER2", "area": "AREA2", "event": "EVENT2"},
  ...
]
```

Table 19

Attribute	Description
user	The owner of the alert state key. • If this is a personal alert state, the value is the name of the IRIS Focus user account associated with the key. • If the alert state was created by an IRIS Focus user with the poweruser role, the value is **@global**. • If the alert state was created by a technical system event, such as a data outage, the value is **@technical**.
area	The place of interest or source associated with the alert state.
event	The event associated with the alert state.

12.6.2 Alert states: request and response

Request alert states

When requesting alert states, you must specify an access token and a list of one or more alert state keys. You will need to POST a JSON message in the following form:

```
{
  "token": "ACCESS_TOKEN_FROM_AUTHENTICATION_SERVICE",
  "keys": [
    {"user": "USER1", "area": "AREA1", "event": "EVENT1"},
    {"user": "USER2", "area": "AREA2", "event": "EVENT2"},
    ...
  ]
}
```

Table 20

Attribute	Description
`token`	The `access_token` received from the authentication API service.
`keys`	For specifying a list of one or more alert state keys to match.
`user`	The owner of the alert state key. • If this is a personal alert state, the value is the name of the IRIS Focus user account associated with the key. • If the alert state was created by an IRIS Focus user with the poweruser role, the value is `@global`. • If the alert state was created by a technical system event, such as a data outage, the value is `@technical`. You can use the wildcard * to match any user.
`area`	The place of interest or source associated with the alert state. You can use the wildcard * to match any area.
`event`	The event associated with the alert state. You can use the wildcard * to match any event.

Response alert states

When responding to an alert state HTTP request, the `alert-api` service returns a JSON message containing an array of zero or more alert states. The JSON alert state messages have the following form:

```
[
  {
    "key": {"user":"@global","area":"Auckland Airport","event":"Heavy Rain"},
    "lastChange": "2023-08-25T04:00:00.000+0000",
    "attributes": {"id":"1","poweruser":"true"},
    "isActive": false
  }, {
    "key": {"user":"@global","area":"Auckland Airport","event":"Lightning"},
    "lastChange": "2023-08-28T09:19:25.942+0000",
    "attributes": {},
    "isActive": false
  }, ...
]
```

More information

- WebSocket alert states: request and response (page 213)

12.6.3 WebSocket alert states: request and response

WebSocket alert state request

After opening a WebSocket connection to the **`alert-api`** service, a client must send a JSON message in the same format as the "Request alert states" **`JSON`** message. This will turn on the monitoring of alert states for the keys specified in the message.

WebSocket alert state responses

When the WebSocket client has provided an access token and a list of alert state keys to match the **`alert-api`** service, the client will receive alert state messages. The JSON alert state messages have the following form:

```
{
  "key": {
    "user": "testperson1",
    "area": "Downtown Helsinki",
    "event": "Lightning Threat"
  },
  "lastChange": 1693241040000,
  "attributes": {},
  "isActive": true
}
```

You will receive an individual JSON message for each alert state that is being monitored.

Attribute	Description
`key`	The unique key associated with the alert state.

Attribute	Description
`user`	The owner of the alert state key. • If this is a personal alert state, the value is the name of the IRIS Focus user account associated with the key. • If the alert state was created by an IRIS Focus user with the poweruser role, the value is **`@global`**. • If the alert state was created by a technical system event, such as a data outage, the value is **`@technical`**.
`area`	The place of interest or source associated with the alert state.
`event`	The event associated with the alert state.
`lastChange`	The count of milliseconds since January 1, 1970. This is a standard epoch time commonly used in JavaScript and Java. Divide by 1,000 to convert to seconds.
`attributes`	An optional dictionary of key/value pairs if the associated alert state has any additional meta data available. This can be empty (meta data is not required).
`isActive`	A boolean value indicating whether the alert state is currently active or not.

More information

- Alert states: request and response (page 212)

12.7 Technical alerts

Technical alerts can occur when there are failures in system processes. Technical alert states have the same structure as weather alert states associated with places of interest. You can identify technical alert states by the values set in the key associated with an alert state report. The following conventions are used when setting the key fields in technical alert state keys:

Attribute	Value	Description
`user`	**`@technical`**	The user attribute is always **`@technical`** to indicate a technical alert state.
`area`	varies	This value is the source associated with the alert state. This will be **`Alert Repository`** for the alert state of the alert tracking database. For data outage alert states, this is the name of the radar or lidar site.
`event`	varies	This value is **`Housekeeping`** for the alert state of the alert tracking database. It will have the form of **`DATAFLOW:task_name`** for radar or lidar sites that have data flow alerts enabled. The **`task_name`** shown is replaced by the actual task name that the data outage monitoring is enabled on.

Example alert repository alert state message

The following technical alert relates to the alert tracking database alert state. When there are a very large number of alerts logged to the database, this alert state will become **`true`**. This technical alert state is always enabled in IRIS Focus and always available to be selected by API clients for monitoring. If the alert state has never occurred (which is common) the "**`lastChange`**" time remains at its initial value of **`0`** (January 1, 1970).

```
{
  "key": {
    "user": "@technical",
    "area": " Alert Repository",
    "event": "Housekeeping"
  },
  "lastChange": 0,
  "attributes": {},
  "isActive": false
}
```

Example dataflow alert state message

The following example indicates that the alert state for the PPI task from the CHC Lidar is currently active, meaning that there is a data outage occurring. In other words, IRIS Focus stopped receiving data from the PPI task running at the CHC Lidar.

```
{
  "key": {
    "user": "@technical",
    "area": "CHC Lidar",
    "event": "DATAFLOW:PPI"
  },
  "lastChange": 1693339764470,
  "attributes": {},
  "isActive": true
}
```

By default, data flow alerts are disabled. The data flow alert states can be enabled by the IRIS Focus administrator using the IRIS Focus web interface.

13. IRIS Focus services and users

The following tables list the IRIS Focus users and IRIS Focus services running on `systemd`, `Docker`, and `Kubernetes`.

Table 21 IRIS Focus users

User	Description
`radardminput`	Restricted user account for running the Data Manager input service.
`radarop`	Non-root user account typically included.
`radarweb`	Restricted user account for running the IRIS Focus web application.
`warnreader`	Restricted user account for running the warn reader service.

Table 22 IRIS Focus `systemd` services

Service	Description
`chronyd`	Maintains time synchronization.
`containerd`	Service required to run container based services.
`docker`	Engine for running services in Docker compatible images.
`microk8s`	Collection of systemd services to run a Kubernetes cluster.
`monit`	Monitoring tool for Unix systems and processes.
`HAProxy`	Encodes outgoing traffic with HTTPS encryption.
`vaisala-radarsw-webapp`	IRIS Focus web application.
`vaisala-radarsw-usbdaemon`	System service to read Sentinel license key on systems using the USB license key.

Table 23 IRIS Focus `Docker` services

Service	Description
`postgis`	Postgresql database server with GIS extensions.
`redis`	A data structure store for shared information.

Service	Description
`kafka`	Kafka data broker service for lightning.
`zookeeper`	A manager service required by kafka data brokers.
`postgis95`	Database service required by the geoserver container.
`geoserver`	GeoServer service that provides map tile images for IRIS Focus.
`ltz-db`	Database used by the **Lightning Threat Zone** engine, geoserver, and API services.
`ltz-geoserver`	A **Lightning Threat Zone** specific geoserver that provides WMS overlays.

Table 24 IRIS Focus `Kubernetes` services

Namespace	Name	Description
`vaisala-focus-api`	`alert-api`	An exposed API service that provides external client applications the ability to monitor IRIS Focus alert states.
`vaisala-focus`	`authentication-service`	Authenticates requests for services.
`vaisala-focus`	`data-manager-service`	Handles requests for radar data.
`vaisala-focus`	`documentation-service`	Handles requests for static documents.
`vaisala-focus`	`keycloak`	Used by the authentication service to manage API access tokens for API clients.
`vaisala-focus`	`licensing-service`	Determines whether a feature is licensed or not.
`vaisala-focus`	`notification-service`	Provides external notifications via email and SMS.
`vaisala-focus`	`nowcast-service`	Provides nowcasting information to IRIS Focus.
`vaisala-focus`	`router-service`	Used to route traffic between outside world and Kubernetes services.
`vaisala-focus-algorithms`	`turbulence-service`	Computes turbulence reports from data pulled out of Data Manager.

Namespace	Name	Description
vaisala-focus-data-access	input-service	Injects radar data from IRIS Analysis into Data Manager.
vaisala-focus-data-access	warn-reader	Injects warn products from IRIS Analysis into IRIS Focus.
vaisala-focus-data-access	lidar-input-service	Imports data from NetCDF files into Data Manager.
vaisala-focus-lightning	lightning-websocket	Provides WebSocket service for external browsers displaying real-time lightning data.
vaisala-focus-logging	alert-logger	Optional. Records JSON alert records posted to Kafka broker to rolling files.
vaisala-focus-logging	grafana-service	Provides a tool to view Kubernetes metrics and logs.
vaisala-focus-logging	loki-service	Stores logs and provides viewer.
vaisala-focus-logging	prometheus-service	Event monitoring end alert tool.
vaisala-focus-logging	promtail-daemonset	Provides log information to the grafana service.
vaisala-focus-logging	zipkin-service	Distributed tracing system used for troubleshooting latency issues.
vaisala-focus-ltz	ingest-ltz	A service that publishes new Lightning Threat Zone reports to Kafka.
vaisala-focus-ltz	ltz-api	An internal API service that provides access to **Lightning Threat Zone** reports.
vaisala-focus-ltz	ltz-engine	A service that consumes lightning data and creates**Lightning Threat Zone** reports.

13.1 systemd

systemd and is an AlmaLinux component that manages system services.

Several of the services that were running under **systemd** in earlier IRIS Focus releases, are now run as Docker or Kurbernetes services.

More information

- Installing IRIS Focus components (page 51)

13.1.1 GeoServer

GeoServer is used for caching and generating the base map layers.

The GeoServer service is provided by the **geoserver** and **postgis95** docker containers.

13.1.2 IRIS Focus web application

This is the main web UI of the IRIS Focus system.

In the command line, the IRIS Focus web application service is called **vaisala-radarsw-webapp**.

13.1.3 HAProxy

HAProxy is a proxying tool that IRIS Focus uses for traffic forwarding within the system and HTTPS encryption for outgoing traffic.

In the command line, the HAProxy service is called **haproxy**.

More information

- Encryption (page 227)

13.1.4 Monit

Monit is a watchdog tool for monitoring Unix systems and processes. IRIS Focus uses Monit to automatically restart the application or a related process or service if it becomes unstable.

If you do maintenance work that requires you to take the application down, you must first stop Monit before proceeding further, and restart it after maintenance.

In the command line, the Monit service is called **monit**.

13.2 Kubernetes

Starting from IRIS Focus 7.0, several services in IRIS Focus run on Kubernetes.

13.2.1 Managing Kubernetes services

These are the common use cases when managing Kubernetes services in IRIS Focus:

- Viewing service state (**k9s** or **kubectl**)

- Restart services (`k9s` or `kubectl`)
- Configuring services (`kubectl`)
- Remove and install services (`kubectl`)
- Viewing service logs (`k9s` or `kubectl`)

There are several command line tools that are used to manage Kubernetes services. These tools can be used in a local terminal window or remotely over an SSH connection.

Table 25 Command line tools

Command line tool	Purpose
`k9s`	An interactive command line tool used to manage the containers running in a Kubernetes cluster
`kubectl`	A command line tool to manage the containers running in a Kubernetes cluster
`microk8s`	A command line tool specific to managing the microk8s implementation of Kubernetes

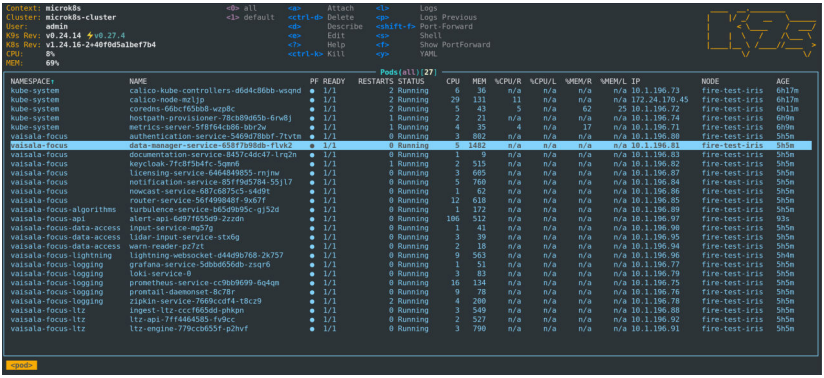
13.2.1.1 Viewing service state in Kubernetes

The k9s utility can be used to quickly show the status of services running in the Kubernetes cluster.

1. To start the k9s utility, log in as the **`root`** user and run this command:

```
k9s
```

You see a screen that lists the IRIS Focus containers running in the Kubernetes cluster. Normally, these are all written in blue font and in the **Running** state. You can navigate the screen with the arrow keys.



2. To exit k9s, press **CTRL+C**.

13.2.1.2 Restarting a service running in Kubernetes

If you need to restart a service that runs on Kubernetes, do the following:

1. Log in as the **root** user.
2. Type **k9s** to open the status overview.
3. If the terminal did not open in the **Pods** view, open the **Pods** view.
4. Type **0** to show all the containers.
5. Use the up and down arrow keys to highlight the service you want to restart.
6. Press **CTRL+D** to delete the current instance of the service.

   As soon as you delete the service, the Kubernetes cluster detects that it is missing one of the required services, and starts up a new instance for you.
7. Alternatively, if you know the namespace and name of the service you want to restart, you can use the **kubectl** command to restart the service.

   For example, if you want to restart the **nowcast-service** that runs in the **vaisala-focus** namespace, you can run the following commands to determine the full address of the Kubernetes pod running the **nowcast-service**:

   ```
   kubectl get --namespace vaisala-focus pods | grep nowcast-service
   ```

   You will see the following output:

   ```
   nowcast-service-748d9fdfd4-wg8ld         1/1     Running
   0          2m51s
   ```

8. Once you know the full address of the pod (*nowcast-service- 748d9fdfd4-wg8ld*), you can restart it by using this command:

   ```
   kubectl delete --namespace vaisala-focus pod/nowcast-service-748d9fdfd4-
   wg8ld
   ```

   You will see the following output:

   ```
   pod "nowcast-service-748d9fdfd4-wg8ld" deleted
   ```

9. You can verify that a new instance was created using the **kubectl** command.

   Sometimes, the creation of the new instance may take some seconds, and you see this process when verifying the command. For example, if the user runs the following **kubectl** command quickly enough:

   ```
   kubectl get --namespace vaisala-focus pods | grep nowcast-service
   ```

   then the output will show that Kubernetes has started a new instance of the **nowcast-service** (**nowcast-service-748d9fdfd4-r8lph**) and is terminating the old instance (**nowcast-service-748d9fdfd4-wg8ld**):

   ```
   nowcast-service-748d9fdfd4-wg8ld          1/1     Terminating
   0          4m12s
   nowcast-service-748d9fdfd4-r8lph          1/1     Running
   0          23s
   ```

13.2.1.3 Configuring Kubernetes services

There are several YAML configuration files found under the */etc/vaisala/focus/k8s* directory that are used to configure groups of services that run on an IRIS Focus server. You do not typically need to modify the configuration found in these files.

1. If you receive instructions from Vaisala to make changes, use the **kubectl** command to apply your changes to the running Kubernetes cluster.

   For example, if you have made modifications to the *vaisala-focus-lightning.yaml* file that configures the services related to sending lightning data to the web browser, you would run the following command to apply your changes to the Kubernetes cluster:

   ```
   kubectl apply -f /etc/vaisala/focus/k8s/vaisala-focus-lightning.yaml
   ```

   Applying changes to the Kubernetes cluster will often only update the configmap objects in the cluster. The services that read their configuration values from these Kubernetes configmap objects will need to be restarted.

13.2.1.4 Removing and installing Kubernetes services

There are several YAML configuration files stored in the */etc/vaisala/focus/k8s* directory that are used to configure groups of services that run on an IRIS Focus server.

1. For example, to remove the services related to sending lightning data to the web browser, you can run this command:

   ```
   kubectl delete -f /etc/vaisala/focus/k8s/vaisala-focus-lightning.yaml
   ```

2. To restore the services related to sending lightning data to the web browser, you can run this command:

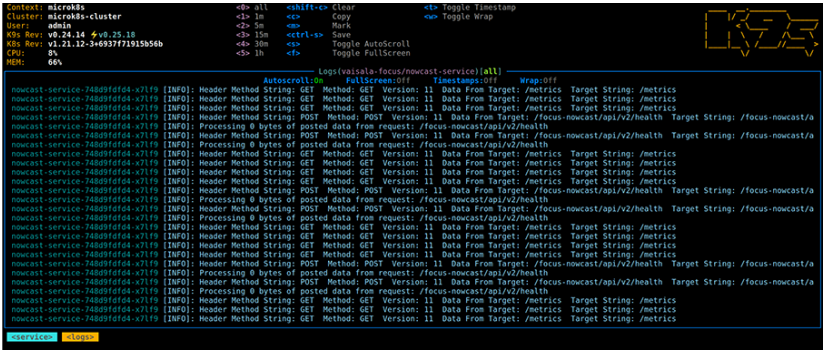
```
kubectl apply -f /etc/vaisala/focus/k8s/vaisala-focus-lightning.yaml
```

Typically, Vaisala does not recommend doing this in normal operations, as it is more severe than restarting an individual service. However, this may be necessary when troubleshooting or when major changes have been made to one of the YAML configuration files.

13.2.1.5 Viewing logs from Kubernetes services

The k9s tool makes it easy to view the latest logs from Kubernetes services.

1. Use the arrow keys to highlight the service you are interested in and then press the **l** key to view the logs. This Figure shows k9s in log viewing mode:



2. While k9s is very handy for a quick look, you can also use the **`kubectl`** command.

   The **`kubectl`** command is particularly useful when you want to post process the logs with a grep. To use the **`kubectl`** command, you need to know the namespace of the service deployment.
   As an example, the following command will monitor the log output of the nowcast-service running in the vaisala-focus namespace:

   ```
   kubectl logs --tail=20 -f --namespace vaisala-focus deployment/nowcast-
   service
   ```

   You will see the following output:

   ```
   [INFO]: Header Method String: POST  Method: POST  Version: 11  Data From
   Target: /focus-nowcast/api/v2/health  Target String: /focus-nowcast/api/v2/
   health
   [INFO]: Processing 0 bytes of posted data from request: /focus-
   nowcast/api/v2/health
   [INFO]: Header Method String: GET  Method: GET  Version: 11  Data From
   Target: /metrics  Target String: /metrics
   [INFO]: Header Method String: GET  Method: GET  Version: 11  Data From
   Target: /metrics  Target String: /metrics
   [INFO]: Header Method String: GET  Method: GET  Version: 11  Data From
   Target: /metrics  Target String: /metrics
   [INFO]: Header Method String: GET  Method: GET  Version: 11  Data From
   Target: /metrics  Target String: /metrics
   [INFO]: Header Method String: POST  Method: POST  Version: 11  Data From
   Target: /focus-nowcast/api/v2/health  Target String: /focus-nowcast/api/v2/
   health
   [INFO]: Processing 0 bytes of posted data from request: /focus-
   nowcast/api/v2/health
   [INFO]: Header Method String: POST  Method: POST  Version: 11  Data From
   Target: /focus-nowcast/api/v2/health  Target String: /focus-nowcast/api/v2/
   health
   [INFO]: Processing 0 bytes of posted data from request: /focus-
   nowcast/api/v2/health
   [INFO]: Header Method String: GET  Method: GET  Version: 11  Data From
   Target: /metrics  Target String: /metrics
   [INFO]: Header Method String: GET  Method: GET  Version: 11  Data From
   Target: /metrics  Target String: /metrics
   [INFO]: Header Method String: GET  Method: GET  Version: 11  Data From
   Target: /metrics  Target String: /metrics
   [INFO]: Header Method String: GET  Method: GET  Version: 11  Data From
   Target: /metrics  Target String: /metrics
   [INFO]: Header Method String: POST  Method: POST  Version: 11  Data From
   Target: /focus-nowcast/api/v2/health  Target String: /focus-nowcast/api/v2/
   ```

```
health
[INFO]: Processing 0 bytes of posted data from request: /focus-
nowcast/api/v2/health
[INFO]: Header Method String: POST  Method: POST  Version: 11  Data From
Target: /focus-nowcast/api/v2/health  Target String: /focus-nowcast/api/v2/
health
[INFO]: Processing 0 bytes of posted data from request: /focus-
nowcast/api/v2/health
[INFO]: Header Method String: GET  Method: GET  Version: 11  Data From
Target: /metrics  Target String: /metrics
[INFO]: Header Method String: GET  Method: GET  Version: 11  Data From
Target: /metrics  Target String: /metrics
[INFO]: Header Method String: GET  Method: GET  Version: 11  Data From
Target: /metrics  Target String: /metrics
^C
```

3. To get a listing of the many **`kubectl logs`** command options, you can run it using the **`–help`** parameter:

```
kubectl logs --help
```

13.2.2 Lightning WebSocket service

Lightning WebSocket service is responsible for pushing lightning data to the user's browsers when connected to the IRIS Focus web application.

The service runs on Kubernetes and is called **`lightning-websocket`**.

13.2.3 Nowcasting service

The radar-based nowcasting performs advection calculations on motion data from radar products to predict weather movement and severity up to 2 hours into the future.

Starting from IRIS Focus release 7.0, the nowcasting service runs under Kubernetes.

13.3 Docker

Starting from IRIS Focus 7.0, several services in IRIS Focus run on Docker.

13.3.1 Kafka data broker

The Kafka data broker is used by the external **Total Lightning Processor** system to push lightning data into the IRIS Focus system so that local services (for example, **`lightning-websocket`**) can access it.

The Kafka data broker service is provided by the **`kafka`** docker container.

13.3.2 Kafka manager

The Kafka data broker supports running in a cluster configuration where multiple systems are interconnected. The Kafka manager service is used to manage all of the Kafka data broker service instances in a cluster. This service is required even if you are running a single instance of the Kafka data broker, which is typical for IRIS Focus.

The Kafka manager service is provided by the **zookeeper** docker container.

13.4 Stopping, starting, and restarting services

You should only need to start or stop a service during certain troubleshooting cases. These cases are described step-by-step in the *Troubleshooting* section. In normal circumstances the services are always running.

In AlmaLinux, services are stopped, started, and restarted with the **systemctl stop / start / restart [servicename]** command.

To use the **systemctl** command, you must be a logged in as the **root** user.

The following example shows how to stop, start and restart the IRIS Focus web application service. Note that the **monit** service starts along with the web application.

Stopping the service

- **systemctl stop monit**
- **systemctl stop vaisala-radarsw-webapp**

Starting the service

- **systemctl start vaisala-radarsw-webapp**
- **systemctl start monit**

Restarting the service

- **systemctl restart vaisala-radarsw-webapp**

14. Security

14.1 Encryption

Communication between the browser and the web application is encrypted.

Other data traffic within the IRIS Focus application server is unencrypted.

IRIS Focus uses Jetty as web server software, and HAProxy for handling HTTPS encryption. SSL encryption has been disabled in HAProxy, and only TLS encryption is supported.

More information

- Web application (page 33)
- HAProxy (page 219)

14.2 Certificates

The web application comes with a temporary, self-signed SSL certificate that secures the connection between the IRIS Focus server and the user's web browser.

Although the browser displays a security warning in the browser when you try to access the web application, you can use the application normally even with the warning.

Consider acquiring and using a trusted certificate from a certificate authority (CA), especially if you plan to offer access to IRIS Focus outside your organization.

More information

- Web application (page 33)
- Installing a CA certificate (page 184)

14.3 Security settings

Please follow industry security standards while deploying IRIS Focus into an internal network. Care should be taken to allow access to only ports 80 and 443 from the internet.

The IRIS Focus server has a pre-configured firewall.

Ports for SSH access (22), HTTP (80), HTTPS (443), and Kafka (9094) are intentionally open.

- Use SSH for configuration.
- HTTP port is for redirecting to HTTPS.
  The application is always used over HTTPS.

The server requires access to HTTP and HTTPS for end users. If the system is accessed through the internet, you should restrict internet access to the SSH port from the internet to improve system security.

The firewall is configured through the AlmaLinux firewall system.

Port 9094 is only opened if the Kafka service is running. The **Total Lightning Processor** uses this port when pushing lightning data into the Kafka data broker running on the IRIS Focus Server. See Connecting the TLP system (page 90) for details on configuring the `firewalld` rule so that only the TLP system is allowed access to this port.

More information

- Installing IRIS Focus components (page 51)

14.4 Removal of X Window System

For customer convenience, Vaisala ships IRIS Focus with a graphical desktop environment installed. IRIS Focus does not require a graphical desktop environment to run. Having a graphical desktop environment and the X server in particular is sometimes regarded as a security concern.

Use the following commands to configure the system to run in console mode and remove the X server and graphical desktop environment:

```
systemctl set-default multi-user
systemctl isolate multi-user
dnf remove --noauto xorg-x11*
```

CAUTION! Do not do this if you are running applications other than IRIS Focus on the same system that do require a graphical environment such as IRIS Analysis.

14.5 SELinux

If IRIS Analysis is not required to be installed on the same Focus server, then SELinux may be left enabled (as is the default behaviour in AlmaLinux).

14.6 Running OS hardening scripts

IRIS Focus includes a small set of example scripts to help secure the AlmaLinux operating system. You can find them in the *`security-scripts`* directory. However, the example OS hardening scripts are not meant to be used as such, they are included for reference only. If you want to use these scripts, review and edit them for your own needs before running.

Do not disable IPv6.

Table 26 Hardened areas included in the example scripts

Hardened area
Install AIDE (Advanced Intrusion Detection Environment)
Restrict core dumps
Set permissions for `grub` configuration
Set default Message of the Day
Configure Chrony NTP
Configure TCP Wrappers
Strengthen log file permissions
Strengthen `Cron` configuration
Lockout for failed login attempts
Password suffiency
Strengthen file permissions
Enable SSH issue banner
Remove support for unneeded file system types: `cramfs`, `freevxfs`, `jffs2`, `hfs`, `hfsplus`, `squashfs`, `udf`, `vfat`, `dccp`,`sctp`, `rds`, `tipc`, `cups`, `avahi-daemon`

14.7 Installation security notes

- *CVE-2022-40735 and CVE-2002-20001*
  To address security issues CVE-2022-40735 and CVE-2002-20001, you can run the ***CVE-2022-40735.sh*** script found under the ***security-scripts*** directory. The script disables support for the out-dated Diffie-Hellman key exchange algorithms in the SSH client and the SSHD server connections.
  Run the following command from the release directory as the root user to apply this security configuration change to IRIS Focus:

  ```
  ./security-scripts/CVE-2022-40735.sh
  ```

  Once applied, you will be unable to establish SSH connections between IRIS Focus and older systems that only support Diffie-Hellman algorithms.

- IRIS Focus supports installation when SELinux is set to **Enforcing**.
- IRIS Focus supports installation when **firewalld** is enabled.

Follow the industry security standards while deploying IRIS Focus into an internal network. Only allow access to ports 80 and 443 from the Internet.

15. Troubleshooting

15.1 Sending logs to Technical support

When you contact Technical support (page 285), be ready to send IRIS Focus logs to the technical support personnel. For retrieving logs, do the following steps:

1. Log into the IRIS Focus server as **root**.
2. Run the command:

```
rsw-tar-logs
```

3. Send the resulting tar file in /tmp to Vaisala for analysis.
   The file should be in the form *rsw-tar-logs-<date>-<time>.tar*, for example: *rsw-tar-logs-2022-04-28-16-28-51.tar*.

15.2 IRIS Focus fails to resolve host names

If IRIS Focus is unable to resolve host names, you need to specify the IP address when communicating with external systems. The inability to resolve host names can be caused by one of the following reasons:

- DNS has been turned off during the installation using the **--disable-dns** option. If you are able to access DNS servers, check the */etc/resolv.conf* file and then enable DNS again by using the **rsw-manage-network** command.
- DNS has been turned off using the **rsw-manage-network** command after the installation.
- DNS has been turned off in the */etc/nsswitch.conf* file.
- DNS has been improperly configured. Check the DNS servers shown in the */etc/resolv.conf* file.
- A network configuration change that is preventing IRIS Focus from reaching the servers in the */etc/resolv.conf* file.

More information

- Configuring DNS (page 145)

15.3 Configuring server after changing IP address

If the IP address of the IRIS Focus server is changed, and after the change you experience problems using kubectl, microk8s, or k9s commands from the command line, do the following:

1. Log in to the server as **root**.

2. Run the following command:

```
microk8s config | tee ~/.kube/config
chmod 600 ~/.kube/config
```

15.4 Notification sound is not played when an alert is triggered

Some web browsers (for example, Mozilla Firefox and Google Chrome) by default block sounds on web pages until the user interacts with the page. Thus, in some cases, the web page may not play the alert notification sounds in IRIS Focus. This may occur, for example, if a user automatically logs into IRIS Focus by clicking the browser reload button while logged in. When the user has logged into IRIS Focus through the normal login, this issue should not occur.

To make sure users hear the sound notifications right away when alerts are triggered, enable the web browser to play sounds by default.

15.5 Slowness in system with a high volume of lightning data

When heavy lightning occurs for a long period of time, and it is visualized with the **TimeSpan**product, the number of lightning icons on the map can increase dramatically. This may cause a performance degradation in the client (browser) of IRIS Focus. This may happen, for example, when the time frame for **TimeSpan** is very long.

To fix the performance issue, shorten the time frame for viewing the data, or zoom in on the map to show fewer icons.

15.6 Data Manager does not work as expected

When running correctly, Data Manager and the socket server run continuously.

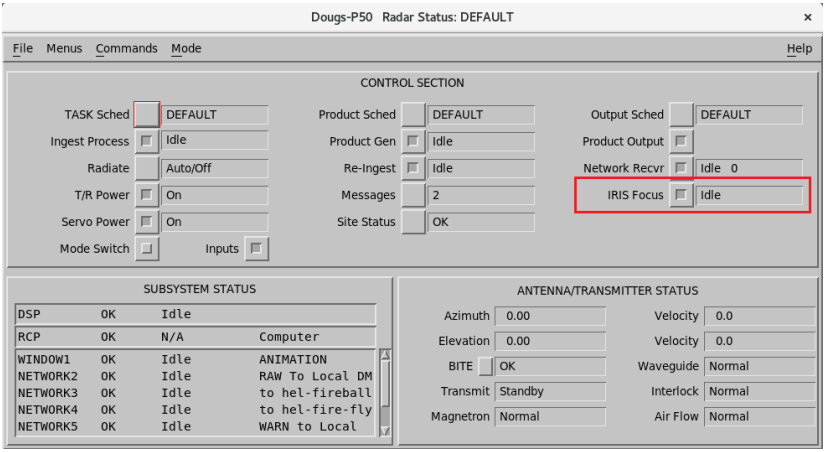
In some cases, IRIS Focus may be unable to request IRIS Analysis products from the IRIS Analysis server through the socket server or IRIS Analysis may be unable to send **RAW** products to IRIS Focus. In such cases, try the following solutions.

1. Try shutting down the firewall on the socket server machine:

```
service firewall stop
```

2. Check the product configuration in IRIS Analysis and consider the following:
   - To generate correct centers in IRIS Focus for IRIS Analysis products, IRIS Analysis must create 1 product for each site on the IRIS Analysis server.
   - The IRIS socket server has a limit of 1000 products that can be received by IRIS Focus, so the socket server provides only the 1000 most recent products.
     For example, if IRIS Analysis creates a new product every 15 minutes, IRIS Focus visualizes only the last 10 days of data. (*4 products/hour * 24 hours * 10 days*).
   - **RAW** products are only needed for the IRIS Focus Data Manager
3. In the IRIS Analysis **Radar Status** menu, make sure **IRIS Focus** is toggled on.

   The toggle button turns on/off the socket server.



4. If IRIS Focus was installed in a system that was already running IRIS Analysis, and the IRIS Analysis did not have a license that supports connectivity to IRIS Focus, you may need a new IRIS Analysis license. Request a new license from your Vaisala representative.
5. To check the delivery of **RAW** files, make sure Data Manager **radarinput** is setup correctly on the IRIS Focus server:
   a. On the IRIS Analysis server, login as **root**.
   b. Type: **-- ssh radardminput@the-focus-machine-hostname.com date**
   c. Make sure the data and time are returned from the IRIS Focus machine without having to type a password.
   d. Check the security keys and permissions:
      - */var/lib/radardminput/.ssh/authorized_keys* must be correct
      - Permissions must be set to **chmod 644 ./authorized_keys**

6. Reboot the IRIS Analysis and/or the IRIS Focus servers.

More information

- Setting up Data Manager (page 60)

15.7 Data Manager housekeeping not working as expected

If the data is corrupted, the application crashes. If Focus is not able to display data, even though you know that data should be available, the data is probably corrupted. The logs may also indicate that there has been an error in processing the files.

Use the `rsw-data-manager-clear-data` script if the Data Manager data storage becomes corrupt or if there is a need to remove all data from Data Manager.

CAUTION! Running the script deletes all radar data from IRIS Focus, including Nowcasting configurations, pre-defined composite configurations, and RAW radar data.

1. Run the script:

```
DM_RESET=yes rsw-data-manager-clear-data
```

If there is a lot of RAW radar data in Data Manager, it may take some time to run the script.

CAUTION! Do not interrupt the script execution.

15.8 Nowcasting is unavailable

If you cannot see nowcasting features on your display, it is likely because you either do not have a license or because nowcasting is disabled.

1. Check that you have a nowcasting license.
   a. You must have an IRIS Focus seat to use nowcasting.

      If no seats are available, wait until an IRIS Focus seat is available and try again.
   b. Login to the IRIS Focus web application as **`administrator`**.
   c. Select **Admin > System > Licensing Management**.

2. Check that MVF is configured for your site.
3. Log in to the server as **root**.
4. Go to */etc/vaisala/radarsw/configuration/vsoweb-override.ini*.
5. In the **[NOWCAST]** section of the **vsoweb-override.ini** file, check that MVF creation is enabled in IRIS Focus:

```
[NOWCAST]
nowcast.mvf.run = true
```

By default, MVF generation is enabled (**true**).

6. Restart the **vaisala-radarsw-webapp** service by typing:

```
systemctl restart vaisala-radarsw-webapp
```

7. Restart the Nowcasting service using the instruction in Restarting a service running in Kubernetes (page 221).
8. Start the nowcast server by typing:

```
systemctl start vaisala-radarsw-nowcast-server
```

   a. To verify that the server starts, type:

```
systemctl status vaisala-radarsw-nowcast-server.service
```

   b. Check for the status:

```
Active: active (running)
```

More information

- IRIS Focus licensing (page 14)

15.9 No connection/data from the TLP

If there are problems in the TLP data connection, try the following troubleshooting procedures.

1. Check the status of the IRIS Focus related services.
   a. Log in to IRIS Focus as the **root** user.
   b. Check the status of the services related to the incoming TLP lightning data with the following commands:

```
kubectl get --namespace vaisala-focus-lightning deployments/lightning-
websocket
docker ps --filter name=kafka --filter name=zookeeper
```

2. Check the status of the TLP related services:
   a. Log into the TLP system as the **vops** user.
   b. Use the **lpstart** command to verify that the **tlp-to-kafka** service is running:

```
lpstart details tlp-to-kafka
```

3. Check services and processes with the **netstat** command:
   a. Use the **netstat** command on the IRIS Focus system and **grep** on port 9094:

```
netstat -tnap | grep 9094
```

   You should see the Kafka process listening on port 9094, and an established connection to port 9094 with the IP address of your TLP system.
   b. If you do not see an established connection from the TLP system, verify that the **tlp-to-kafka** service is running on the TLP system, and that the ***kafka-producer.properties*** file in the */opt/vai/tlp/etc* directory has the correct IP address for your IRIS Focus server set in the **bootstrap.servers** parameter.
   c. Use the **netstat** command on the IRIS Focus system and **grep** on port 30100.

```
netstat -tnap | grep 30100
```

   You should see the **vaisala-iris-lightning-ws** service listening on port 30100, and an established connection to port 30100 with the proxied IP address **127.0.0.1** for each user connected to the IRIS Focus web application.

15.10 Network Health updates missing

If you are getting infrequent updates of the **Network Health** product, or no updates at all, try the following troubleshooting procedures.

1. Check that the **regstatd2** service is running on the TLP system.
2. Check that the ***regstatd2.cfg*** configuration file in the */opt/vai/tlp/etc* directory has the **updateIntervalMinutes** parameter set to 10 minutes.

15.11 Check disk space usage of Kafka

The Kafka service keeps an archive of historical data in the */var/lib/kafka* directory. Use the**df** command to check that the partition has space left.

```
df -h /srv/container/mnt/kafka
```

15.12 GLD360 lightning layer empty

If you have subscribed to Vaisala GLD360 lightning detection service, and the layer exists in your IRIS Focus application, but you do not see any lightning strikes, verify the following requirements:

1. Check that lightning strikes have occurred at the time of observation.
2. Check that the configuration file */etc/vaisala/radarsw/configuration/vsoweb-override.ini* contains the following line:

   ```
   lightning.wms.url = [URL received from Vaisala]
   ```

3. Check that your subscription to Vaisala GLD360 service is active.

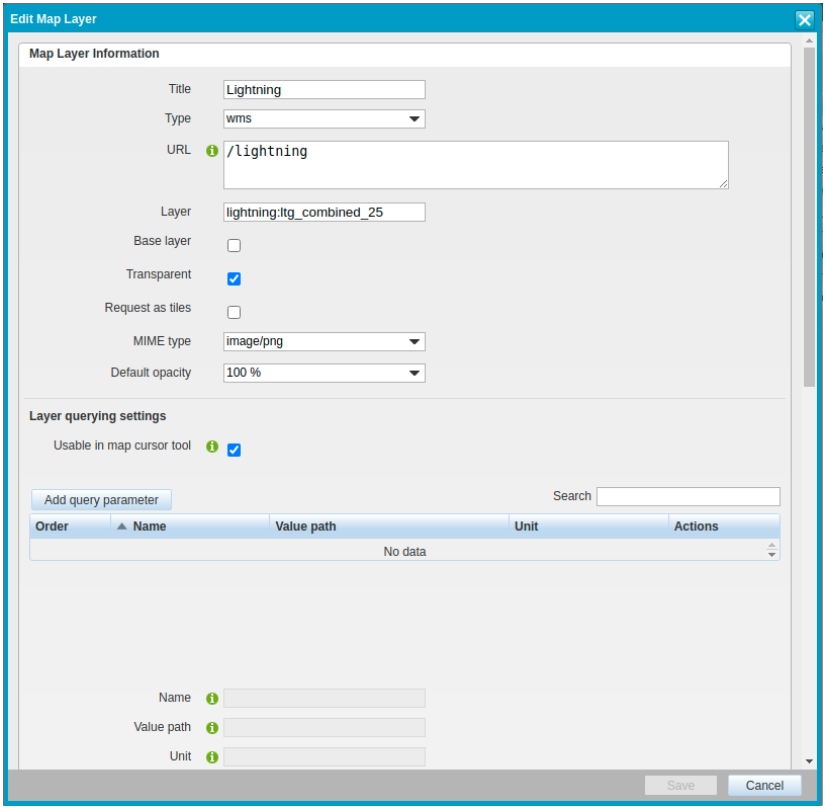
If you modify the configuration file, you must restart the vaisala-radarsw-webapp service with the **service vaisala-radarsw-webapp restart** command.

15.13 GLD360 lightning layer missing

If you have subscribed to Vaisala GLD360 lightning detection service, and you do not see the lightning layer in the IRIS Focus user interface after running the **rsw-lightning-configure** script, add the lightning layer manually.

1. Log in to IRIS Focus with an administrator account and select **Admin**.
2. Select **Map > Map Layers**.
3. Select **Add New Layer**.

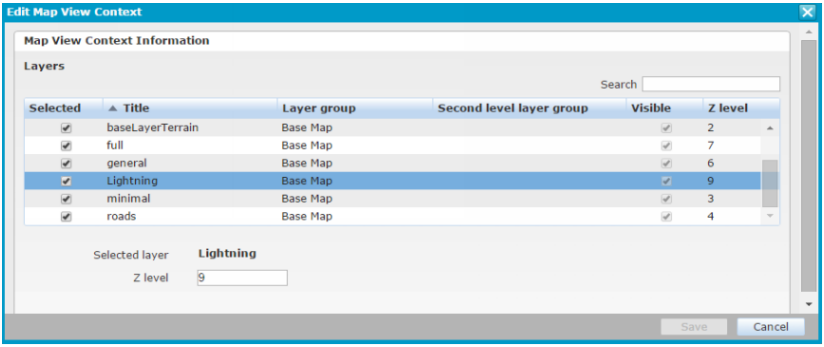
4. In **Map Layer Information**, enter the following values on the layer properties:



   a. **URL**: `/lightning`
   b. **Layer**: `lightning:ltg_combined_25`
   c. **Transparent**: Checkbox selected
   d. **SLD URL**: https://tsm.vaisala.com/geolegends/ltg_combined_25.sld
   e. **Name**: `ltg_combined_25.ltg_type`
5. Select **Save**.
6. Select **Map > Map View Contexts**
7. Edit the default map context **TheMap**.

8. Select the newly created lightning layer and set its **Z level** higher than all base map layers in the map context.

   Z level defines the drawing order of the layers on the map. Higher values are always drawn on top.



In the web application, the new layer is listed at the bottom of the radar product selection list.

15.14 Taking a snapshot gives server error

If, when taking a snapshot or requesting an image via URL, the server times out or gives server error, there may be a problem with the **`image-export`** user account.

1. Check that the application log shows the error:

   ```
   Login failed for username image-export
   ```

2. Log in to the IRIS Focus web application as **`administrator`**.
3. Check that the **`image-export`** user password matches the password listed in ***`vsoweb-override.ini`***.

15.15 "Issue with loading OnScreen struct" when connecting to the socket server

The webapp complains during the connection about an "Issue with loading OnScreen struct", when connecting to IRIS Analysis during the install or otherwise.

This issue is caused by both having an older version of IRIS Analysis, for example 8.13.6, and having a RAIN1 product created using a 3d CAPPI with R (rain intensity) as an input to the RAIN1. As a result, there are multiple layers in the RAIN1, while RAIN1 is supposed to have just one layer.

IRIS Analysis versions prior to 9.1.0 cannot properly handle this kind of multi-layer RAIN1 products.

The problem can be solved by user either;
- updating to IRIS Analysis version 9.1.0 and beyond
- or removing the extra RAIN1 product from Analysis.

15.16 Identifying IRIS Focus software version

Before contacting Vaisala technical support about an issue, check which version of IRIS Focus you have on your system.

1. In the terminal window, run:

```
rpm -qa --qf '%{NAME} %{VERSION}\n' | grep 'vaisala-radarsw-webapp'
```

More information

- Technical support (page 285)

15.17 Uninstalling IRIS Focus

Use this procedure to recover from a failed installation that is stuck in a state where it cannot be resumed.

CAUTION! The `rsw-uninstaller` script completely removes IRIS Focus, including all data and configurations.

CAUTION! The `rsw-uninstaller` script removes `Postgres` and all databases.
If you share the system with other software that uses `PostgreSQL`, do not run the script -- it also removes `PostgreSQL` databases not related to IRIS Focus.

1. Navigate to the directory containing the IRIS Focus installation files.
2. Run: **`./rsw-uninstaller`**

   When prompted, confirm that you want to run the script.
   The script removes all users, configurations, and data from the system so that you can rerun the installation.

More information

- Installing IRIS Focus components (page 51)

Appendix A. High-end server installation requirements

Table 27 Requirements for a high-end server for IRIS Focus installation

IRIS Focus server (T/R version)	Specifications
PowerEdge 640	Intel® Xeon® Gold 5118 2.3 G, 12C/24T, 10.4 GT/s 2UPI 16 M cache, Turbo, HT (105W) DDR4-2400 12 x 16 B RDIMM, 2667 MT/s, Dual Rank 10 x 10 TB 7.2 K RPM NLSAS 12 Gbps 512e 3.5 in hot-plug hard drive PERC H740P adapter RAID contoller without bracket 8x DVD-ROM, USB, External Dual, hot-plug, redundant power supply (1+1), 750 W

Appendix B. File locations

Table 28 IRIS Focus application and configuration files

File or directory	Description
/etc/vaisala/radarsw/configuration • *gis-override.ini* GeoServer database settings. • *logback.xml* Logging level settings. • *radar_centers.properties* List of stored radar site center points.	Configuration files for IRIS Focus module settings. The files listed here are the most important. **CAUTION!** Some settings have a default config file and an override file. For example: • *gis-config.ini* • *gis-override.ini* When needed, edit the override file.
/etc/vaisala/radarsw/configuration/vsoweb-override.ini	Connection settings for socket server, lightning layers, nowcasting and so on.
/etc/vaisala/radarsw/nowcast/nowcast.ini	Configuration files for the nowcast server.
/etc/vaisala/lightning/iris-lightning-ws.properties	Kafka configuration file for the **vaisala-radarsw-webapp systemd** service.
/etc/vaisala/lightning/iris-lightning-ws.kafka.properties	The configuration file used by the **vaisala-radarsw-webapp** service when configured to connect to a Kafka data broker for its lightning data. The default configuration will connect to the Kafka data broker running on the IRIS Focus system. Normally, the user does not need to adjust this.
/etc/vaisala/lightning/lightning.simu.properties	Used to configure how simulated lightning events are produced when the **lightning-websocket**service is configured in simulation mode.
/etc/vaisala/lightning/regionstatus.simu.properties	Used to configure how simulated lightning events are produced when the **vaisala-radarsw-webapp** service is configured in simulation mode.
/etc/vaisala/lightning/regionstatus.template.json	Template network health report used when the **vaisala-radarsw-webapp** service is configured in simulation mode.
/usr/vaisala/radarsw/configuration	Configuration files for helper applications used in IRIS Focus maintenance.
/var/lib/radarweb	Home directory of the **radarweb** user. The IRIS Focus Web Application is deployed here.

File or directory	Description
/var/lib/radardm	Home directory of the **radardm** user.
/var/lib/radardminput	Home directory of the **radardminput** user.
/srv/vaisala/radarsw/datamanager/input	Files sent from an IRIS Analysis server are copied here. The data manager input service processes files copied here.
/srv/vaisala/radarsw/datamanager/storage	This is where data manager stores polar or **RAW** data.
/var/lib/warnreader	Configuration files for events and alerts.
/var/log/vaisala/radarsw	Log files from IRIS Focus web application

Appendix C. Map layer configuration options

Table 29 Map layer configuration options

Option	Description	WMS layer only
Map Layer Information	Defines basic map settings, such as the title and the URL address of the Web Map Service (WMS).	--
Title	Title of the layer. Visible in the layer selection list.	--
Type	• `wms`: generic GIS services such as base maps or raster-type forecast data • `google`: Google base maps • `marker`: observations from stations configured using the JX source service on the map.	--
URL	Address of the WMS service.	✔
Layer	Name of the layer in the map server. If using GeoServer, usually `workspace:layer`.	--
Base layer	Select if the layer is a base map.	--
Transparent	Select for WMS to request a transparent background for the layer.	✔
Request as Tiles	Use if the map layer should be requested as tiles. Usually selected for base maps.	✔
MIME type	Map image type. Change if the service does not support the default *`image/png`*.	✔
Default opacity	Not used in IRIS Focus.	--
Layer query settings		--
Supported Coordinate Reference Systems	Select supported coordinate reference systems for the layer.	--
Time Support	Configure for layers using time dimensions.	✔
Coverage	Maximum bounding box for the layer.	✔
Layer style	For advanced configurations, add SLD (Styled Layer Descriptor) parameters.	--

Option	Description	WMS layer only
Realtime offset	Defines the offset from the current time in which to make the request for the latest data. Sometimes, when requesting the latest time from a WMS service, there is no data available because the WMS service provider is collecting and processing the data for the latest time, so it is useful to set an offset. Supported values are 0...3600 seconds. To use this parameter, set the system to always use time parameter support.	
Refresh rate	Defines the interval of the time ticks on the histogram. This defines how often the system makes data requests. The interval always starts on the hour. Supported values are 10...86400 seconds. To use this parameter, set the system to always use time parameter support.	
Request width	Controls the legend graphic request parameters.	✔
Request height	Controls the legend graphic request parameters.	✔
Display height	Defines the size of the color legend graphic on the display in case the original graphic is too large.	✔

More information

- Adding and editing map layers (page 172)

Appendix D. Nowcasting configuration files

D.1. nowcast.ini

The following example shows the *nowcast.ini* configuration file for configuring the nowcasting HTTP server.

```
; Algorithm to use.
correlator=trec
```

TREC

```
[trec]
; Number of decimals to keep in data when converting to integers.
; Range:  [0 ; 3]. Default: 2.
input_precision=2
```

```
; The value in image that declares a missing/invalid value.
; Default: -999.0.
missing_value=-999.0
```

```
; The value in image that declares a not-scanned pixel, outside the aperture
area.
; Default: -900.0.
not_scanned_value=-900.0
```

```
; Minimum measurement aperture coverage (%) in correlation region.
; Range:  [0.0 ; 1.0]. Default: 0.60.
aperture_coverage_threshold=0.60
```

```
; Minimum signal value for the pixel to be 'active' and used.
; Default: 10.0.
signal_threshold=10.0
```

```
; Feature box size.
; Range:  > 0 Default: 14
field_feature_box_width=14
```

```
; Amount of skip when calculating field values.
; Range:  > 0.  Default: 1 (no skip).
field_feature_box_spacing=1
```

```
; Minimum fraction (%) of active pixels in feature box needed to trigger
correlation analysis.
; Range:  [0.0 ; 1.0] Default: 0.10
field_signal_coverage_threshold=0.10
```

```
; Minimum allowable cross-correlation coefficient.
; Range:  [0.0 ; 1.0] Default: 0.55
correlation_threshold=0.55
```

```
; Maximum storm movement between images, search region radius.
; Range:  > 0  Default: 15
speed_limit=15
```

```
; Spatial smoothing factor, exp(-d/decay). Used for spreading effect
; of local motion vector to its surroundings.
; Range:  >= 0 (0 == no spatial smoothing) Default: 6
field_spatial_decay=6
```

```
; Spatial filtering flag. Whether to discard points that differ from global
average.
; Range:  0 == NO; 1 == GLOBAL; 2 == LOCAL . Default: 1(GLOBAL)
field_use_spatial_filtering=1
```

```
; Feature box size for local spatial thresholding (applied only when using
local spatial thresholding).
; Range:  > 0   Default: 9
field_spatial_filtering_box_width=9
```

```
; Maximum allowed direction difference from mean motion (applied only when
using spatial filtering).
; Range: [0 ; 180]  Default: 90
field_spatial_direction_threshold=90
```

```
; Maximum allowed speed (mgt*mean_motion) above mean motion (applied only when
using global spatial filtering).
; Range:  >= 1.0  Default: 3.0
field_spatial_magnitude_threshold=3.0
```

```
; Global vector weight applied to local values.
; Range:  [0.0 ; 1.0] (0.0 = no global weighting). Default: 0.25
field_global_weight=0.25
```

```
; Method for temporal smoothing.
; Range: 0 == NO_TEMPORAL_SMOOTHING; 1 == HISTORY_WEIGHTING; 2 ==
CHANGE_WEIGHTING.
; Default: 1(HISTORY_WEIGHTING)
temporal_smoothing_method=1
```

```
; History weight factor (applied when temporal smoothing is made by using
HISTORY_WEIGHTING).
; Range:  ]0.0 ; 1.0]  Default: 0.25
temporal_smoothing_history_weight=0.25
```

```
; Change weight factor (applied when temporal smoothing is made by using
CHANGE_WEIGHTING).
; Range:  ]0.0 ; 1.0] Default: 0.33
temporal_smoothing_change_weight=0.33
```

D.2. vsoweb-override.ini

The *vsoweb-override.ini* configuration file contains setting for managing the **MVF** (motion vector field) product and advection used in nowcasting.

Vaisala has carefully chosen good defaults for the nowcasting configuration. The raster product, such as **PPI**, **CAPPI**, of any intensity parameters like **Z**, **R**, **KDP**, or **rhoHV** that is used as an input for MVF generation should have:

- As little as possible of ground clutter and the near-radar clear air or particulates (such as dust) returns.
- The bounding box not smaller than any other raster product produced from this site's data.

Because the two conditions are contradictory, the easiest way to satisfy first condition is to use a true (not pseudo) **CAPPI** product with a height of 1.5 ... 2km , but the longest range (biggest bounding box) product is a raster product generated from the survey scans, which usually consist of just one **PPI** scan and cannot be used to generate true **CAPPI** products. You must balance these two conditions.

If there are not enough valid products to generate an MVF request, the iteration is skipped and the system waits for the next product to arrive from IRIS.

Basic settings

nowcast.mvf.run defines if MVF generation is enabled in IRIS Focus. By default, MVF generation is enabled (**true**).

```
[NOWCAST]
nowcast.mvf.run = true
```

The nowcast server URL dentifies where the nowcast HTTP server runs. The default value is for a fully local installation, which is the default installation configuration.

```
nowcast.http.server.url = http://localhost:31000/focus-nowcast/api/v2/mvf/
```

The *netCDF* directory stores MVF generation requests and responses to the Nowcast HTTP Server in netCDF format as well as internal representations of MVF serialized to disk. This directory is cleaned periodically by default.

```
nowcast.netcdf.dir = /srv/vaisala/radarsw/product/nowcast/
```

Advanced settings

`nowcast.mvf.request.num.rasters` defines the number of products sent to the nowcast server for generating the MVF. Default is 2.

```
nowcast.mvf.request.num.rasters = 2
```

`nowcast.mvf.product.age.limit.minutes` defines the maximum number of minutes (5 ... 1000) the system goes back in time to find valid products (of the type used to define MVF generation for a site) to use in generating the MVF. Default is 100.

```
nowcast.mvf.product.age.limit.minutes = 100
```

`nowcast.mvf.max.gap.minutes` defines the maximum acceptable gap in minutes (1 ... 1000) between products for MVF generation. Default is 30.

MVF is a shift in pixels per time interval between frames of the product which was used to generate MVF. The interval between advected products may be different from the interval between advected frames. For example, if MVF was generated from the product which was available every 5 minutes but the interval between advected frames has to be 10 minutes, the MVF shift should be doubled. That MVF scaling is taken into account by a scaling shift in every iteration.

```
nowcast.mvf.max.gap.minutes = 30
```

`nowcast.product.times.age.limit.minutes` defines the time range for calculating advected product times (2 ... 2880 mininutes. 2880 is the entire two-day range). Default is 100

Advected product times must be evenly spaced (due to the calculation). The time is derived by dividing the last number of minutes defined in this property by **n** products found in that period.

The spacing is used as the time gap between advected products. In most cases, set this value to match the value in `nowcast.mvf.product.age.limit.minutes`.

```
nowcast.product.times.age.limit.minutes = 100
```

`nowcast.advection.mvf.age.limit.minutes` is the maximum number of minutes to go back in time to find an MVF when generating advected products. If an MVF is not found in the time span given, the iteration is skipped and Focus waits for the next product to arrive from IRIS. Range: 5 ... 1000 minutes. Default is 30.

```
nowcast.advection.mvf.age.limit.minutes=30
```

`nowcast.advection.time.span.minutes` defines the time limit when extending nowcasted products into the future (minutes). The normal range is 1 ... 3 hours. Default is 120.

You can raise the time span to up to 6 hours but this is not recommended as accuracy decreases as time extends into the future.

```
nowcast.advection.time.span.minutes=120
```

Appendix E. NetCDF file format

Many resources describing the **NetCDF** format are available on Internet. The interested user will easily find more information, especially on the website of UCAR (University Corporation for Atmospheric Research) that maintains the format: http://www.unidata.ucar.edu/software/netcdf/.

NetCDF (Network Common Data Form) is a set of interfaces for array-oriented data access and a freely distributed collection of data access libraries for `C`, `Fortran`, `C++`, `Java`, and other languages. The **NetCDF** libraries support a machine-independent format for representing scientific data. Together, the interfaces, libraries, and format support the creation, access, and sharing of scientific data.

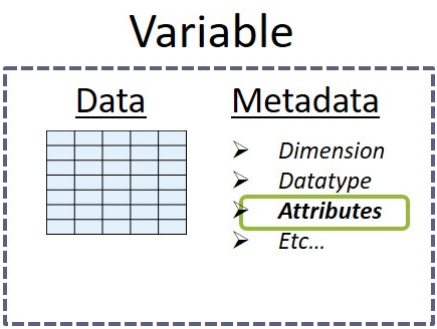
The Vaisala **NetCDF** format is built on **NetCDF-4**, which is in turn built on **HDF5**. Consequently, you can use any **HDF5** or **NetCDF-4** reader to open Vaisala **NetCDF** files. In this document, the term **NetCDF** refers to **NetCDF-4**.

This file format allows grouping all data types (radial, reconstruction, beta, structure, meta-data, and so on) into a single file. This new file format was established using different conventions. These conventions are mentioned in the files. It is an auto-documented format (that is, it is auto-sufficient and does not require meta-data files).

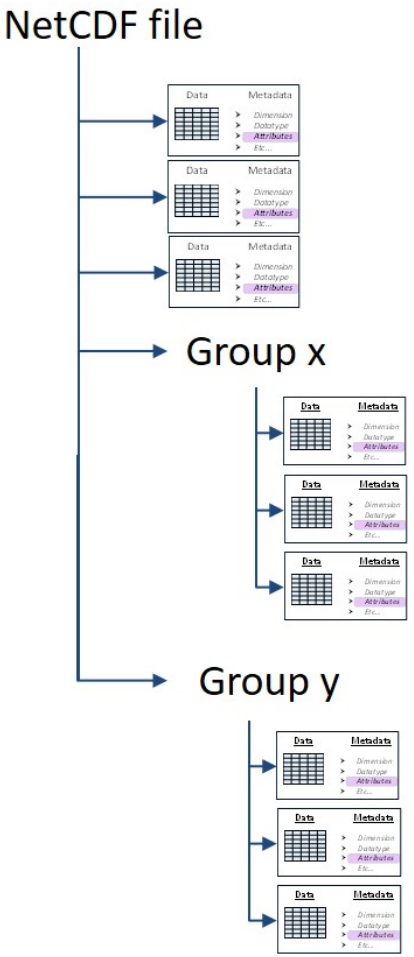
A **NetCDF** file is made of one or several variables. Each variable is made of:

- The data, which is a multidimensional table or a value
- Several metadata that characterize the data.

Figure 24 NetCDF file format

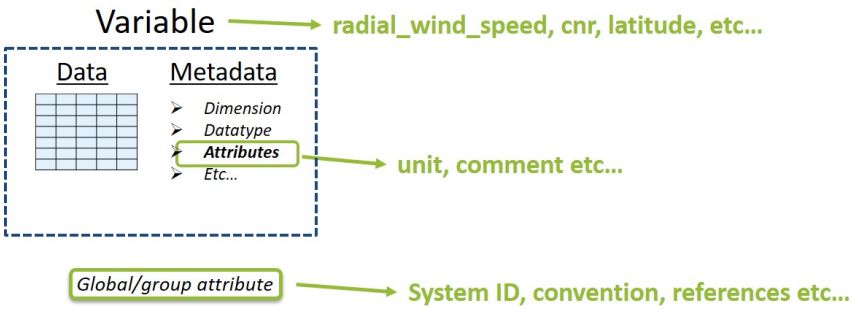


Variables can be organized in a tree structure like below:



Variables are not necessarily organized in a tree-structure. If attached at the root level, variables are called “global” variables.

Similarly, an attribute is not necessarily attached to a variable. In that case, they are called “global” or “group” attributes.



E.1. NetCDF conventions

The **NetCDF** format does not define a mandatory architecture. Users can choose any architecture fitting their purpose. However, several groups of users have introduced proposed conventions to homogenize the content of **NetCDF** files for their community. A list of conventions is given in UCAR website: https://www.unidata.ucar.edu/software/netcdf/conventions.html.

As far as Lidar or RADAR data is concerned, several conventions can be applied. For instance: Cf Convention, CfRadial2, ODIM_H5 (Opera), HD(CP)^2, WindprofNetCDF. Some are still on development. Generally speaking, the conventions are:

- Generic conventions: defining only best practices and principles. The counterpart is that, two files built according to these conventions won't have the same architecture, even if they share common rules. This is, for instance, the case of the Cf Convention.
- Dedicated to a project or a sensor (or both): much stricter about the file content. They are in general based on generic conventions (the Cf Convention most of the time). The main advantage is to have a greater homogeneity among the output files. The counter part is that proposed rules, don't necessary apply to your own system and applications. This is for instance the case of ODIM_H5 (Opera), very radar oriented, HD(CP)^2 or WindprofNetCDF, dedicated to specific observation networks and type of measurements.

The CfRadial2 convention is somewhere in the middle of these two categories: not too generic, but not too rigid and well adapted to Lidar measurements, whatever the application. Consequently, Vaisala has chosen to base its **NetCDF** files architecture on this convention. Even if very advanced, the CfRadial2 convention is still evolving and being improved. Furthermore some details do not fit to Scanning Windcube data.

Consequently:**Whenever applicable, Vaisala NetCDF architecture is based on CfRadial2 convention. Otherwise, it relies on the Cf Convention**.

E.2. Vaisala NetCDF files architecture

As a radar or Lidar scans (or points), the data **fields** (commonly known as 'moments') are computed over limits specified by a time interval or angular interval. This entity is commonly designed as a **ray**, beam, line-of-sight or dwell.

A ray contains a number of **fields**, with a value for each **field** at each **gate**. In the **ray** abstraction, **fields** are represented as 1-D arrays, with length **range**.

In Vaisala **NetCDF** files the term ray is used such that a **ray** = **Line of Sight** (**LOS**).

A **sweep** is a collection of **rays**, for which certain properties remain constant. For a given **ray**, the field's data (or moments) are computed for a sequence of ranges increasing radially away from the instrument. These are referred to as **range gates**.

In the data model adopted by CfRadial2, the **sweeps** contain the field (moments) data directly, stored as 2-D arrays. This requires that the number of gates be constant for all rays in a **sweep**, which is always the case with a Scanning WindCube.

The following *always* remain constant for all **rays** in a **sweep**:

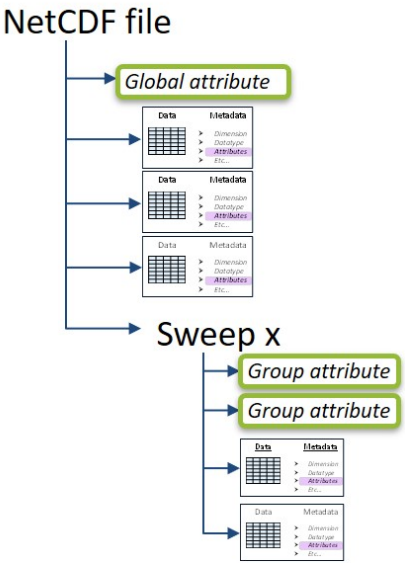
- number of gates
- range geometry (range to each gate)
- **sweep** mode (***PPI***, ***RHI***, etc.)
- target angle(s)

Only Volume Scan sequence contains more than 1 **sweep**.

Here are described only the basics and the specificity of Vaisala **NetCDF** format.

The used convention proposes to classify the files by sequences. For one sequence you have one **NetCDF** file.

The convention uses the term of **sweep**. In this case, a **sweep** corresponds to a sequence and we have 1 **sweep** per scan (for example: 1 **sweep** = 1 **PPI**).



Example of Vaisala NetCDF architecture

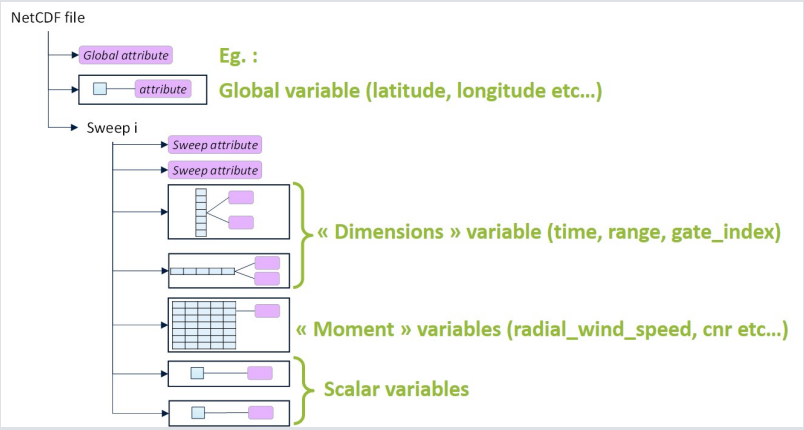
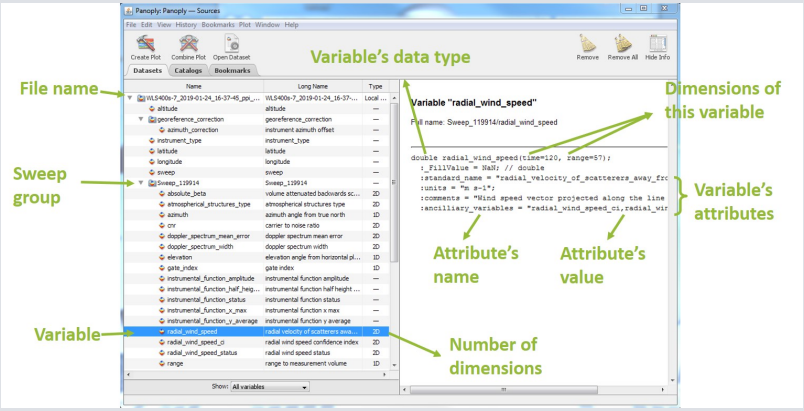


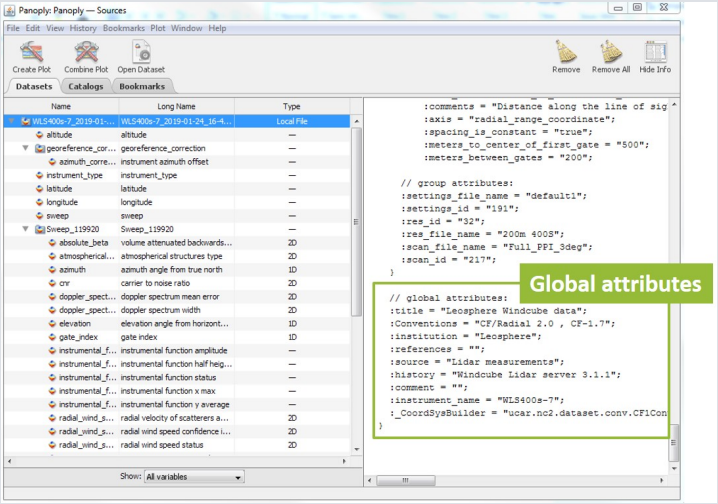
Figure 25 Vaisala NetCDF architecture

Example of a PPI file, opened with Panoply

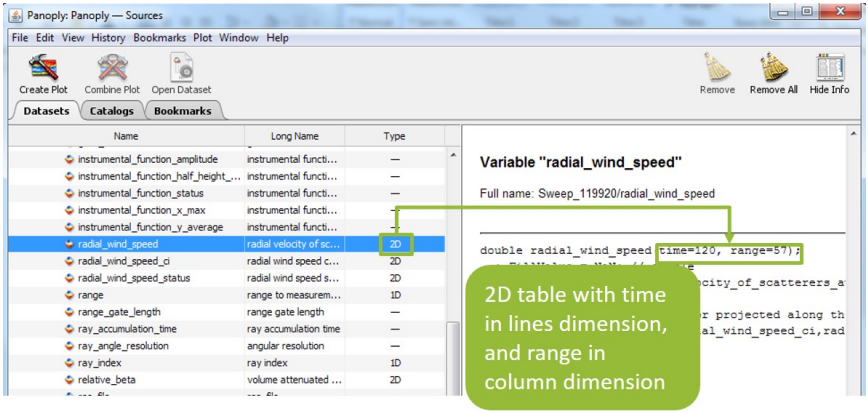


Click on the file name to see the full architecture of the file summarized in the meta-data window. In particular the “global attributes” can be read by scrolling down to the bottom of this meta-data window.

If a field has been selected but there is no available data, this field will still be visible in the **NetCDF** files with values to “NaN” except in vortex mode where there will be no exported files (except the spectra always in **CSV**).



All variables are either scalar values or linked to dimension variables. For instance, in a **PPI** file, `radial_wind_speed` is a 2D table with time in lines dimension, and range in column dimension. Time and range are dimensions.



All wind and aerosol variables (`radial_wind_speed`, cnr etc.) are **2 dimensional**:
- The first dimension is always the time (that identifies each ray).
- The second dimension is either range or `gate_index`. The gate index is an identification number of each range.

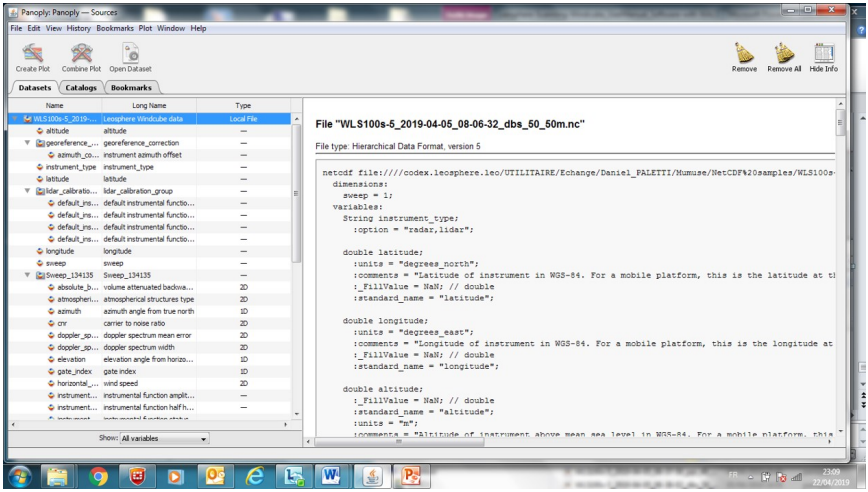
CAUTION!
- Range is used whenever it is constant along the time (**PPI**, **RHI**, **FIXED**). In that case, gate_index is a simple 1D variable depending on range.
- `Gate_index` is used whenever range is not constant (**DBS**). In that case, range is a simple 2D variable depending on time and `gate_index`.

The following table indicates which variables are written in the **NetCDF** files, according to the group chosen:

Table 30 Groups and variables in NetCDF files

Group	Variable
`radial_wind_data`	radial_wind_speed doppler_spectrum_width doppler_spectrum_mean_error cnr radial_wind_speed_ci radial_wind_speed_status
`wind_reconstruction_data`	horizontal_wind_speed vertical_wind_speed wind_direction wind_speed_ci wind_speed_status
`radial_beta_data` (optional)	relative_beta radial_wind_speed_status instrumental_function_x_max instrumental_function_y_average instrumental_function_amplitude instrumental_function_half_height_width instrumental_function_status

Group	Variable
`radial_absolute_beta_data` (optional)	absolute_beta radial_wind_speed_status instrumental_function_x_max instrumental_function_y_average instrumental_function_amplitude instrumental_function_half_height_width instrumental_function_status
`atmospherical_structure_data` (optional)	atmospherical_structures_type



The **NetCDF** file generation timeout is 30 minutes. Beyond this time, the file export will be interrupted.

Several powerful freewares are available on internet to read, explore and plot **NetCDF** file. We recommend you to use Panoply software or/and HDFview software:

- https://www.giss.nasa.gov/tools/panoply/download/
- https://www.hdfgroup.org/downloads/hdfview/

E.3. Global and group attributes description

The following table gives a description and the type of all global or group attributes.

Table 31 Global attributes description and type

Global Attributes	Description	Type
title	“Vaisala Windcube data”	String
scan_file_name	Name of the scan file integrated in the **NetCDF**	String
scan_id	**ID** of the scan that was used to generate this file (stored as a string)	String
Conventions	Gives a coma separated list of convention on which is based the **NetCDF** architecture	String
institution	“Vaisala”	String
references	Empty attribute. Could be used in the future	String
source	“Lidar measurement”	String
history	Specifies the Windcube Lidar server version used to generate the **NetCDF** file. Characterizes the data format architecture revision.	String
comment	Empty attribute. Could be used in the future	String
instrument_name	Lidar’s serial number	String
settings_file_name	Name of the settings file integrated in the **NetCDF**	String
settings_id	**ID** of the settings that was used to generate this file (stored as a string)	String

Table 32 Sweep description and type

(Sweep) group attributes	Description	Type
res_file_name	Name of the **Resolution files** integrated in the **NetCDF**	String
res_id	**ID** of the resolution that was used to generate this file (stored as a string)	String

E.4. Variables list and definition

The following table summarizes the key parameters of all variables and attributes and reminds the **comments** attribute when available. Of course this information is contained in the **NetCDF** files themselves.

Presence column indicates if the variable/attribute is added in the **NetCDF** files according to the choice made by the user in the **DATABASE** tab or in the **FTP** configuration:

- Always = this variable/attribute is always contained in the file whatever the group(s) chosen

- RWD = Radial wind data group
- WR = Wind reconstruction data group
- RB = Radial beta data group
- AB = Radial absolute beta data group
- ATM = Atmospherical structure data group

Table 33 Parameters for variables/attributes and comment field

Group	Type	Variable/ attribute Name	Dimension(s)	Presence	Comments
root	string (attribute)	title	-	always	
root	string (attribute)	Conventions	-	always	
root	string (attribute)	institution	-	always	
root	string (attribute)	references	-	always	
root	string (attribute)	source	-	always	
root	string (attribute)	history	-	always	
root	string (attribute)	comment	-	always	
root	string (attribute)	instrument_name	-	always	
root	int (dimension)	sweep	-	always	Number of sweeps in the dataset.
root	string (variable)	instrument_type	-	always	
root	double (variable)	latitude	-	always	Latitude of instrument in WGS-84. For a mobile platform, this is the latitude at the start of the volume.

Group	Type	Variable/ attribute Name	Dimension(s)	Presence	Comments
root	double (variable)	longitude	-	always	Longitude of instrument in WGS-84. For a mobile platform, this is the longitude at the start of the volume.
root	double (variable)	altitude	-	always	Altitude of instrument above mean sea level in WGS-84. For a mobile platform, this is the altitude at the start of the volume.
root	double (variable)	default_altitude	-	always	Default Altitude as configured in the software.
root	double (variable)	default_longitude	-	always	Default Longitude as configured in the software.
root	double (variable)	default_latitude	-	always	Default Latitude as configured in the software.
root	string (variable)	sweep_group_name	[sweep]	always	Array of names of each sweep group in file.
root	int (variable)	sequence_index	-	always	Stores the sequence id
root	double (variable)	sweep_fixed_angle	[sweep]	always	Array of angles of each sweep in file. Azimuth(s) for RHI, elevation(s) for other modes including FIXED line of sight.
root	int (variable)	time_zone	-	always	Contains the user selected time zone information
root	string (attribute)	scan_file_name	-	always	
root	attribute	scan_id	-	always	
root	char (variable)	scan_file	-	always	Binary content of scan file.
root	string (attribute)	settings_file_name	-	always	

Group	Type	Variable/ attribute Name	Dimension(s)	Presence	Comments
root	attribute	settings_id	-	always	
root	char (variable)	settings_file	-	always	Binary content of settings file.
sweep i	string (variable)	sweep_mode	-	always	
sweep i	int (variable)	sweep_index	-	always	Identification number of the current sweep.
sweep i	string (attribute)	res_file_name	-	always	
sweep i	attribute	res_id	-	always	
sweep i	char (variable)	res_file	-	always	Binary content of res file.
sweep i	string (variable)	rotation_direction	-	always in PPI/RHI	
sweep i	double (variable)	ray_angle_resolution	-	always	Angle between the center of two consecutive rays when scanning head's angular speed, and accumulation time are constants.
sweep i	string (variable)	time_reference	-	always	UTC reference date. Format follows ISO 8601 standard.
sweep i	double (dimension)	time	[time]	always	Number of seconds between time_reference and the end of each ray measurement.
sweep i	double (variable)	ray_index	[time]	always	Identification number of each ray.
sweep i	string (variable)	timestamp_local	[time]	always	
sweep i	string (variable)	timestamp	[time]	always	Timestamp at the end of each ray measurement following ISO8601 standard

Group	Type	Variable/ attribute Name	Dimension(s)	Presence	Comments
sweep i	int (dimension /variable)	range	[range] or [time] [gate_index]	always	Distance along the line of sight, between the instrument and the center of each range gate. Either a dimension or a variable. When this vector is a dimension, gate_index is a variable and vice versa.
sweep i	double (variable)	measurement_height	[range] or [time] [gate_index]	always, in DBS	Vertical distance normal to the ground, between the instrument and the center of each range gate.
sweep i	int (dimension /variable)	gate_index	-	always	Identification number of each range gate. Either a dimension or a variable. When this vector is a dimension, range is a variable and vice versa.
sweep i	double (variable)	azimuth	[time]	always	Scanning head's azimuth angle relative to true north when each measurement finished. 0 to 360. 0 is the North, 90 is the East. This angle only incorporates azimuth_correction. The Lidar is not supposed to be moving.
sweep i	double (variable)	elevation	[time]	always	Scanning head's elevation angle relative to horizontal plane when each measurement finished. -90 to 90. 90 is the zenith. This angle does not incorporate any automatic corrections. The Lidar is not supposed to be moving.
sweep i	double (variable)	range_gate_length	-	always	Radial dimension of range gates
sweep i	double (variable)	radial_wind_speed	[time] [range/ gate_index]	RWD	Wind speed vector projected along the line of sights.

Group	Type	Variable/ attribute Name	Dimension(s)	Presence	Comments
sweep i	double (variable)	cnr	[time] [range/ gate_index]	RWD	
sweep i	double (variable)	doppler_spectrum_width	[time] [range/ gate_index]	RWD	Full width at half maximum of the spectrum. Representative of particules speed dispersion in the range gate.
sweep i	double (variable)	doppler_spectrum_mean_error	[time] [range/ gate_index]	RWD	Root Mean Square Error between the measured Doppler spectrum and the estimated Doppler spectrum.
sweep i	double (variable)	radial_wind_speed_ci	[time] [range/ gate_index]	RWD	Quality indicator between 0 and 100.
sweep i	ubyte (variable)	radial_wind_speed_status	[time] [range/ gate_index]	RWD	0 for rejected data and 1 for accepted data. A data is rejected if the confidence index is below a threshold calibrated in factory or when radial wind speed is out of the accepted range.
sweep i	double (variable)	horizontal_wind_speed	[time] [gate_index]	WR	Norm of the wind projection on local horizontal plane.
sweep i	double (variable)	vertical_wind_speed	[time] [gate_index]	WR	Vertical component of the wind. Positive towards zenith.
sweep i	double (variable)	wind_direction	[time] [gate_index]	WR	Wind direction with respect to true north, (0=wind coming from the north, 90=east, 180=south, 270=west)

Group	Type	Variable/ attribute Name	Dimension(s)	Presence	Comments
sweep i	double (variable)	wind_speed_ci	[time] [gate_index]	WR	For inclined lines of sight this figure is equal to 0, 75 or 100 depending on the number of line of sight used for the reconstruction (maximum 4 lines of sight are used). For vertical lines of sight this figure is equal to 100 when the status of the radial wind speed is equal to 1.
sweep i	ubyte (variable)	wind_speed_status	[time] [gate_index]	WR	0 for rejected data and 1 for accepted data. A data is rejected if its confidence index is lower than 100.
sweep i	double (variable)	relative_beta	[time] [range/ gate_index]	RB	Attenuated relative backscatter coefficient. Processed from the CNR.
sweep i	double (variable)	absolute_beta	[time] [range/ gate_index]	AB	Attenuated absolute backscatter coefficient. Processed from the CNR.
sweep i	int (variable)	atmospherical_ structures_type	[time] [range/ gate_index]	ATM	Atmospherical structures detected out of the planetary boundary layer.
sweep i	int (variable)	ray_accumulation_time	-	always	Time during which the detector collects light. A ray is defined by this duration.
sweep i	double (variable)	instrumental_function_x_max	-	RB,AB	Maximum horizontal axis of the Lorentz distribution obtained in the last calibration.
sweep i	double (variable)	instrumental_function_y_average	-	RB,AB	Average value of the y-axis of the Lorentz distribution obtained in the last calibration.
sweep i	variable	instrumental_function_amplitude	-	RB,AB	Amplitude of variations of the Lorentz distribution obtained in the last calibration.

Group	Type	Variable/ attribute Name	Dimension(s)	Presence	Comments
sweep i	double (variable)	instrumental_function_half_height_width	-	RB,AB	Scale parameter specifying the half height width of the Lorentz distribution obtained in the last calibration.
sweep i	ubyte (variable)	instrumental_function_status	-	RB,AB	0 for rejected data and 1 for accepted data. Data is rejected if the beta calibration is not successful.
lidar_calibration group	double (variable)	default_instrumental_function_x_max	-	RB,AB	Default maximum horizontal axis of the Lorentz distribution used for beta computation.
lidar_calibration group	double (variable)	default_instrumental_function_y_average	-	RB,AB	Default average value of the y-axis of the Lorentz distribution used for beta computation.
lidar_calibration group	double (variable)	default_instrumental_function_amplitude	-	RB,AB	Default amplitude of variations of the Lorentz distribution used for beta computation.
lidar_calibration group	double (variable)	default_instrumental_function_half_height_width	-	RB,AB	Default scale parameter specifying the half height width of the Lorentz distribution used for beta computation.
georeference_correction	double (variable)	azimuth_correction	-	always	Azimuth offset angle used if the Lidar cannot be physically oriented to the North.

E.5. Turbulence NetCDF file content (product data)

The following list summarizes the key parameters of the NetCDF file containing the product data.

All items in the list belong to the group **`root`**.

Type	Name	Type	Units	Comments	Notes
global attributes	title	string	-	-	
global attributes	conventions	string	-	-	
global attributes	comment	string	-	-	
global attributes	history	string	-	-	Version of IRIS Focus that generated the data file
global attributes	institution	string	-	-	
global attributes	source	string	-	-	
Variable	instrument_latitude	double	degrees_north	Latitude of instrument in WGS-84	
Variable	instrument_longitude	double	degrees_east	Longitude of instrument in WGS-84	
Variable	instrument_altitude	double	m	Altitude of instrument above mean sea level in WGS-84	
Dimension	time[time]	double	seconds since 1970-01-01T00:00:00Z	Number of seconds between time_reference and the end of each time averaging period	
Variable	timestamp	string	-	Timestamp at the end of each time averaging period. Format follows ISO 8601 standard.	All times must be UTC (for the moment)
Variable	averaging_time	int	s	Time averaging period	

Type	Name	Type	Units	Comments	Notes
Variable	sequence_index[number_of_sequences]	string	-	List of input files names used for the processing	
Dimension	number_of_sequences	int	-	Number of input sequences used for the processing	
Variable	instrument_name	string	-	Serial number of the instrument providing the input data.	
Variable	instrument_type	string	-	Type of instrument providing the input data.	
Dimension/ variable	range[range]	int	m	Distance along the line of sight, between the instrument and the center of each range gate.	
Dimension	direction_index[direction_index]	int	m	Identification number of the direction of each ray. Increments only if the scanner moves in azimuth and/or in elevation.	
Variable	azimuth[direction_index]	double	degrees	Azimuth at the middle of each ray, identified by the direction_index, relative to true north. 0 to 360. 0 is the North, 90 is the East.	

Type	Name	Type	Units	Comments	Notes
Variable	elevation[direction_index}	double	degrees	Elevation angle at the middle of each ray, identified by the direction_index, relative to horizontal plane. Varies between -90 and 90 degrees. 90 is the zenith.	
Variable	turbulence_variance [time, range, direction_index]	double	m^2/s^2	Turbulence calculated as the variance of the radial wind speed, over the averaging period.	anciliary_variable_ availability
Variable	availability [time, range, direction_index]	double	percent	Number of valid radial wind speed data used during the averaging time, divided by the theoretical maximum number of radial wind speed data. The theoretical maximum number of radial wind speed data is the averaging time divided by the accumulation time of one radial wind speed.	turbulence_variance

E.6. Variable's attributes description

The following table gives a description and the type of all attributes that can be used to characterize variables.

All variables are not necessarily qualified by all attributes.

Table 34 Variable's attributes description

Attribute	Description	Type
`_FillValue`	Indicates what default value is used if no data is available.	Same as variable to which it is attached
`ancilliary_variables`	Indicates what variables are used to characterize the current one. For instance **`radial_wind_speed`** variable has **`radial_wind_speed_ci`** and **`radial_wind_speed_status`** as ancillary variables.	String (coma separated)
`axis`	Defines the axis of coordinate variables	String
`Calendar`	Defines the calendar used for variable time.	String
`comments`	Defines the variable.	String
`flag_masks`	Describes a number of independent Boolean conditions using unique bits in each **`flag_masks`** value. This attribute is systematically associated with the **`flag_meanings`** attribute. Example : in the **`atmospherical_structures_type`**, a 2 in the tens digit signifies "residual layer" and a 3 in the tens digit signifies "mixed layer"	Same as variable to which it is attached
`flag_meanings`	String whose value is a coma separated list of descriptive words or phrases, one for each **`flag_values`** or **`flag_masks`**.	String (coma separated)
`flag_values`	Contains a list of the possible flag values. This attribute is systematically associated with the **`flag_meanings`** attribute.	Same as variable to which it is attached
`is_quality_field`	Indicates if this variable qualifies another.	String: "true" or "false"
`long_name`	Used : • instead of **`standard_name`** when no standard_name has been defined for a given quantity. • or additionally to the **`standard_name`** to give additional information on the variable content	String

Attribute	Description	Type
`meters_between_gates`	Indicates the distance between the centers of 2 consecutive range gates when `spacing_is_constant` is true.	String
`meters_to_center_of first_gate`	Indicates the distance to the center of the first range gate.	String
`option`	Gives all possible options when a variable can take only pre-determined values. For instance, options are “direct” or “indirect” for the variable `rotation_direction`.	String (coma separated)
`qualified_variables`	Indicates what variables are characterized by the current (ancillary) one. For instance `radial_wind_speed_status` qualifies `radial_wind_speed`.	String (coma separated)
`spacing_is_constant`	Indicates if spacing between range gates is constant	String: “true” or “false”
`standard_name`	Describes the physical quantity of a variable. The Cf convention standardized a list of standard_name http://cfconventions.org/Data/cf-standard-names/65/build/cf-standard-name-table.html. We used the values given by the Cf convention when available. Otherwise, this field has been left empty and long_name attribute was used instead.	String
`units`	Unit of the variable to which it is attached. This attribute is not implemented if a variable has no unit. Possible values are: `degrees_north`, `degrees_east`, m, degrees, seconds since `time_reference`, m s-1, dB, percent, m-1 sr-1, ms	String

E.7. Atmospherical structures variable description

Some of the variables in the **NetCDF** format are flag variables. In addition to the raw flag data, these variables contain attributes that describe how the flag values are interpreted. This is the case for atmospherical structures; the structures are defined according to the flags below:

Table 35 Structure types according to flags

Flag	Structure type
0000	No data or no detection
0020	Residual layer
0030	Mixed layer

Flag	Structure type
0200	Unclassified cloud
0300	Ice cloud
0400	Water cloud
2000	Unclassified aerosol
3000	Spherical aerosol
4000	Aspherical aerosol

Glossary

advection
The transfer of a property of the atmosphere, such as heat, cold, or humidity, by the horizontal movement of an air mass. Advection calculations are used to perform some of the nowcasting calculations.

alarm
An alarm is an alert of highest severity.

alert
Alert is a state that requires user intervention or recognition. Different types of alerts include alarms, warning, and informational alerts.

area of interest
An area of interest is a geographical area that you can monitor for weather events. If the system detects a weather event within an area of interest, it generates an alert.

bin
A single sample of weather data detected at a known direction, altitude, and distance from the radar site.

composite
Composites combine data (for example, a group of **CAPPI**, **VIL**, **PPI**, or **TOPS** products) from many radars or lidars in one image.

Data Manager
The raw volume data from the radar signal processor and for wind lidars is stored in Data Manager, which makes the data available to the IRIS Focus user interface. Through Data Manager, IRIS Focus can read raw volume data and generate on-demand radar products in real time.

dynamic composite
A radar or lidar composite of on-demand products created by selecting multiple radar sites on the fly. The combining criteria are based on standardized settings.

event
See weather event.

hybrid task
A group of up to 3 tasks with the same scan type which are scheduled together and used together to make products. This allows flexibility of volume scanning schemes.

hydrometeor
A particle of condensed water vapor in the atmosphere. Rain, snow, and hail are examples of hydrometeors.

k9s
An easy to use tool for exploring and controlling a Kubernetes cluster.

Kubernetes (k8s)
General name for managing a collection of containers (services) running on a computer (conductor of the programs running on the computer).

lightning strike
In IRIS Focus, a *lightning strike* refers to either a flash or a lightning stroke, depending on the configuration of the TLP.

Max Time Span
Max Time Span is the maximum time (minutes) allowed between the newest and oldest points of data. When new data is processed, points that are older than the specified time span are removed. Used in, for example, composites of radar data.

microk8s
The implementation of Kubernetes run on IRIS Focus.

MSL
Mean sea level. An average level for the surface of the sea or ocean.

NDOP product
Dual-Doppler velocity product. Combines the velocity measurements from 2 or more radars to get the wind direction and speed.

nowcasting
Weather forecasting up to the next 2 hours.

NWP
Numerical weather prediction

on-demand product
On-demand products are based on raw data from the IRIS back-end. IRIS Focus reads raw volume data and generate weather products in real-time. Users can manipulate product criteria in the user interface in real time.

pin
Pins on a map indicate points of interest with reference points and labels.

place of interest
A location on the map that is either a single point (pin) or a larger area. See area of interest and pin.

pre-configured products
Pre-configured products are products with default settings used for advanced data visualization such as nowcasting, warnings, or multilayer products.

pre-defined composite
A pre-defined radar or lidar composite with customized settings, such as the combining algorithm.

PRF
See pulse repetition frequency (PRF).

pulse
A short burst transmission signal sent by the radar, used to measure the weather activity in atmosphere. The reflection measurements from a pulse are sorted into bins.

pulse repetition frequency (PRF)
Number of pulses transmitted per second. When measuring PRF, a *pulse* contains transmit, receive, and dead time phases. PRF affects *range folding* and *velocity folding* detection. In Vaisala IRIS products, PRF limits the area displayed in radar images and the maximum measurable wind speed.

radar product
Radar products are raw signal data from a radar receiver processed to provide information about current weather conditions. Radar products are calculated from ingest files that are collected during the execution of radar tasks. Products may be data, pictures, or text. For example, **PPI** and **RHI**.

range folding
Detection of the 2nd trip echoes, which are radar signal echoes from outside the radar maximum range. Range folding causes them to be incorrectly displayed within the radar measurement area. Also called range aliasing.

RAW product
Spherical coordinate data product obtained directly from the raw ingest data. The data are stored in compressed format so they can be recorded on tape or sent to a workstation for further processing.

ray
A group of pulses processed together according to configuration rules. See also pulse.

signal processor
A programmable device for digitizing and processing video signals from the radar receiver.

sweep
A collection of pulses or light at a constant elevation as the device rotates around its axis 360°. After a sweep, the device usually changes its elevation and starts a new sweep. Each sweep typically contains the same number of bins or range gates independent of the elevation.

task
A set of instructions to the lidar or radar and signal processing systems including, but not limited to, the scan type (PPI or RHI), PRF, pulse width, signal processing data types, time and range averaging criteria. For example, a PPI volume scan at multiple elevation angles or an RHI at a single azimuth. Also called radar task/ lidar task.

TLP
See **Total Lightning Processor**.

Total Lightning Processor
Total Lightning Processor (TLP) is the central processor of a Vaisala Lightning Detection System, which uses multiple, remote sensors to detect lightning. Each sensor sends its data to the central processor.

velocity folding
Erroneous readings due to particles in the measurement area exceeding the maximum velocity detection threshold of the radar system. The measured velocity "wraps around" to the other end of the scale, resulting in discontinuous readings. Also called velocity aliasing.

volume
Complete set of raw measurement data collected from sweeps, that is used to calculate a model of the atmosphere. The maximum volume is half of a sphere (from 0° elevation upwards), but other shapes are more typical.

warning
A warning is an alert of medium severity.

weather event
A user-defined set of weather-related criteria. When an event occurs on the map, it is shown as an icon. When an event occurs within an area of interest, it triggers an alert.

weather product
Weather products are raw signal data from the TLP or from a radar receiver that are processed to provide information about current weather conditions. Weather products are displayed as layers in IRIS Focus.

WMS
Web Map Service protocol

Index

Warranty

For standard warranty terms and conditions, see www.vaisala.com/warranty.
Please observe that any such warranty may not be valid in case of damage due to normal wear and tear, exceptional operating conditions, negligent handling or installation, or unauthorized modifications. Please see the applicable supply contract or Conditions of Sale for details of the warranty for each product.

Technical support

Contact Vaisala technical support at helpdesk@vaisala.com. Provide at least the following supporting information as applicable:

- Product name, model, and serial number
- Software/Firmware version
- Name and location of the installation site
- Name and contact information of a technical person who can provide further information on the problem

For more information, see www.vaisala.com/support.

Recycling

Recycle all applicable material according to local regulations.

VAISALA