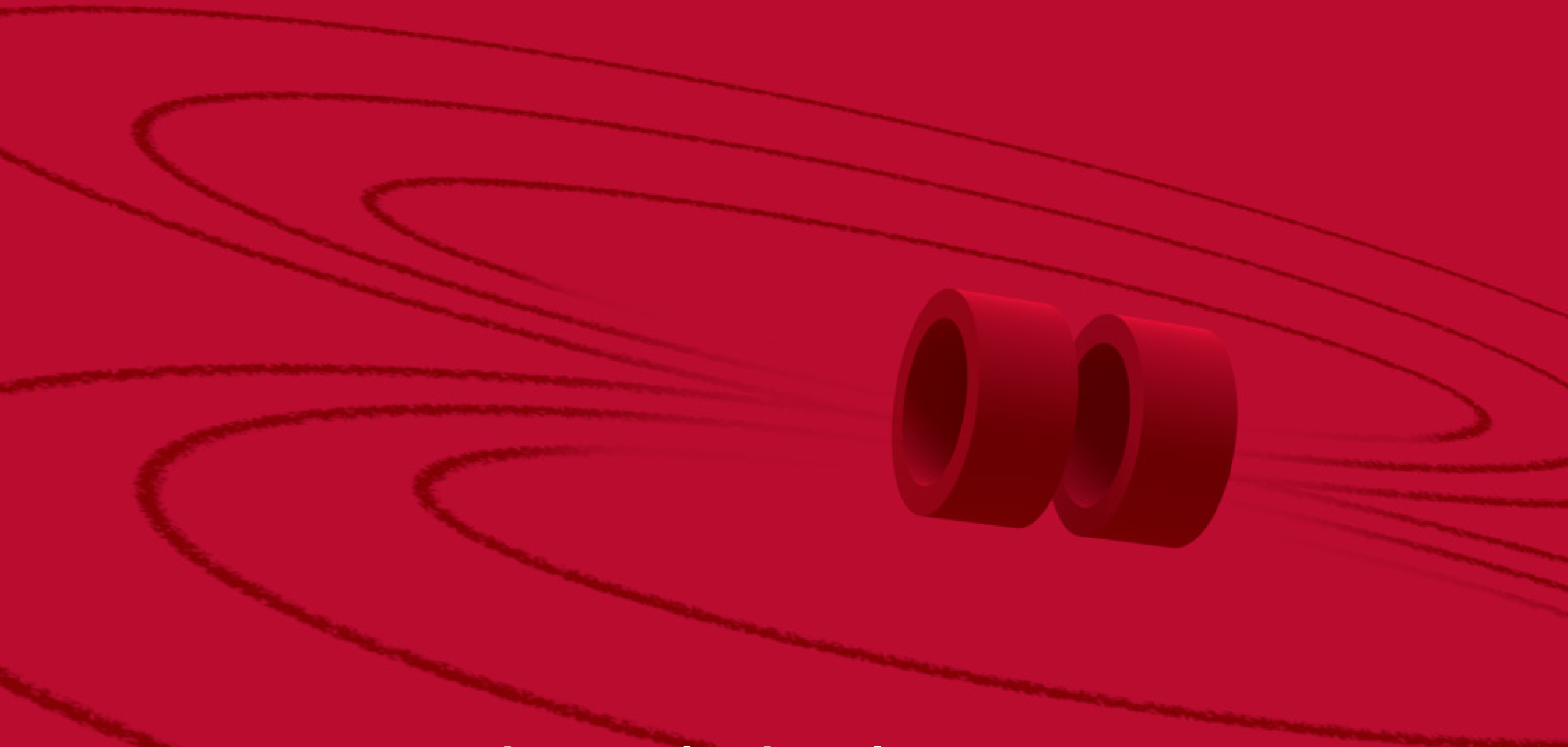
RUSKIN

USER GUIDE

RBR*coda*^{3} RBR*tridente*

RBR*coda* RBR*quadrante*



rbr-global.com

Table of contents

1 Ruskin software

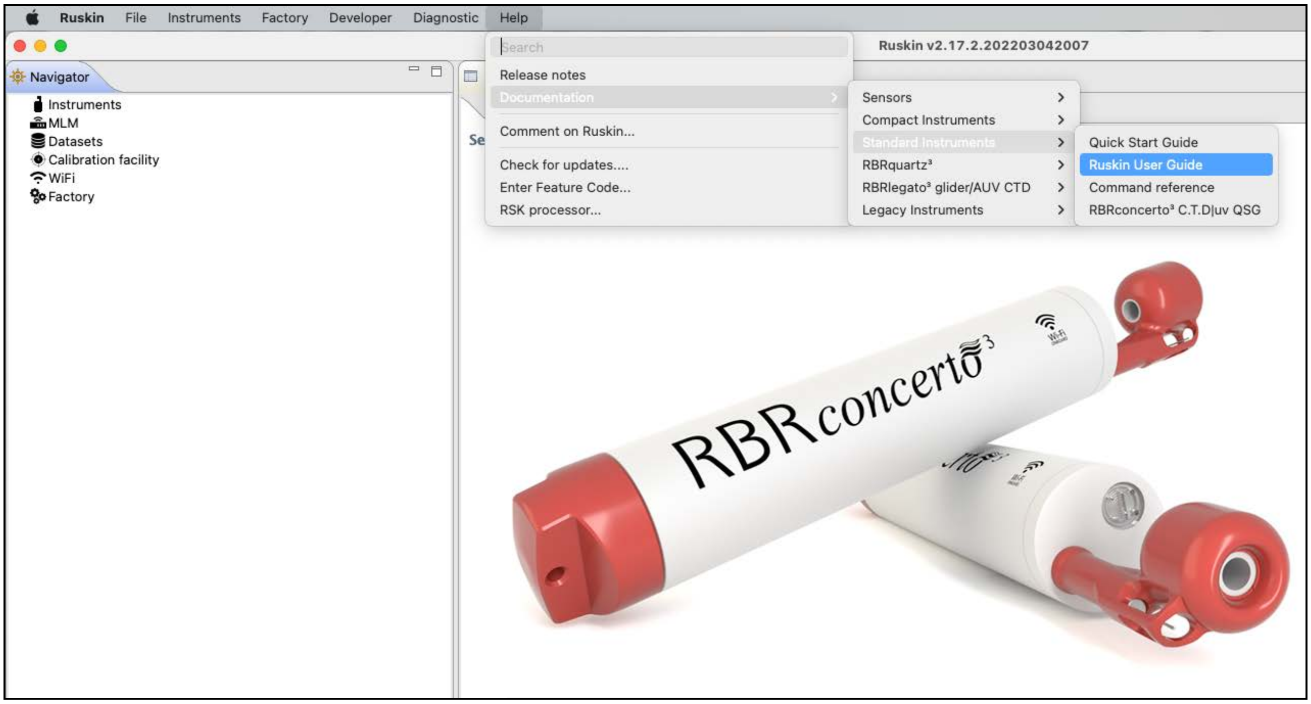
Use the Ruskin software to do the following:

- Configure and enable multiple sensors
- View your datasets graphically
- Export your data in various formats
- Analyse your data
- Change the calibration coefficients
- Perform user calibration
- Contact RBR

RSK files generated by Ruskin contain contain raw measurement data, firmware and software versions, dates and times, sampling settings, power settings, measured and derived parameters, errors and events, other diagnostic information. They can be read directly in Matlab, or exported to Excel, OceanDataView®, or text files.

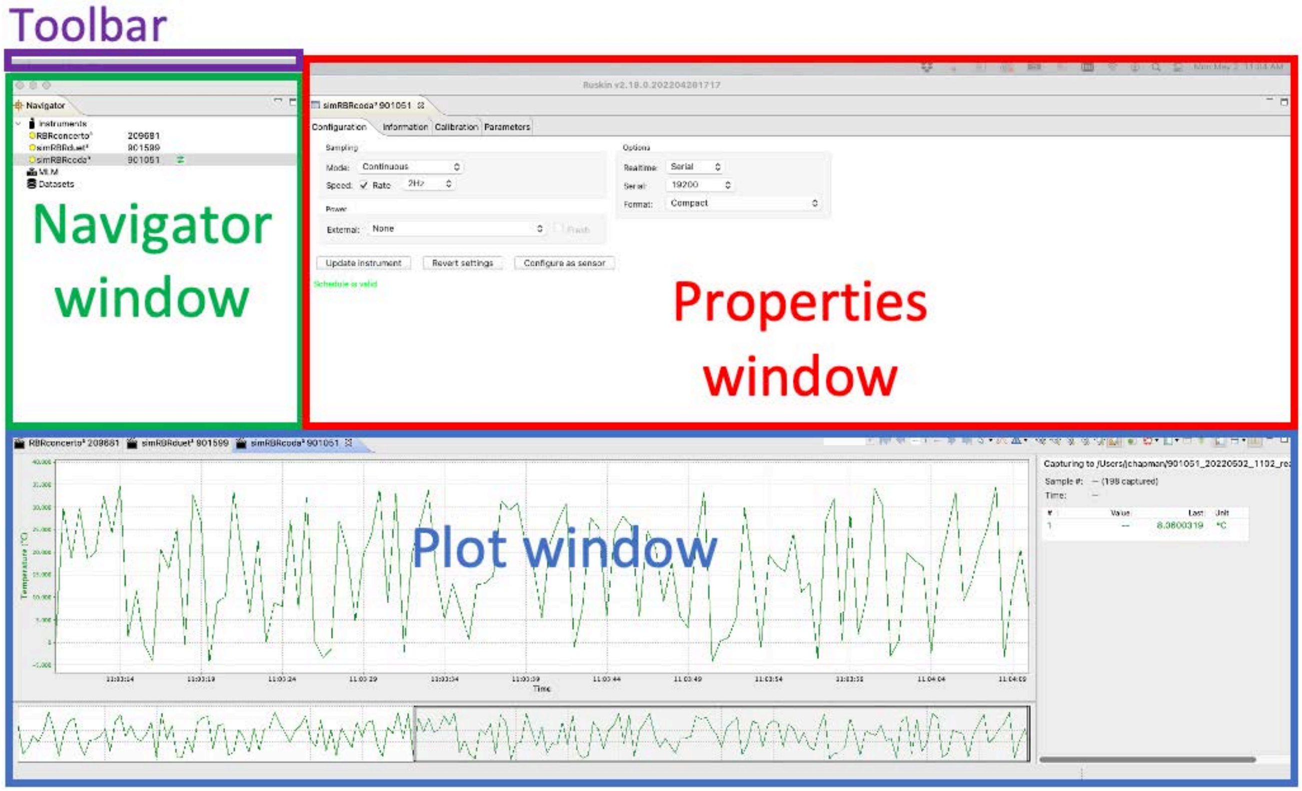
Find Ruskin software for Mac and PC on the USB stick included with your instrument, or visit the RBR website to download the latest version.

The most recent software updates, release notes, Ruskin user guides, instrument guides, and quick start guides are available from the **Help** menu in Ruskin.



2 Ruskin interface

Ruskin software has a user-friendly interface which consists of a **Toolbar** and three main windows: **Navigator**, **Properties**, and **Plot.**



Sampling

Mode: Continuous

Speed: ✓ Rate 2Hz

Power

External: None

Options

Realtime: Serial

Serial: 19200

Format: Compact

Schedule is valid

Capturing to /Users/jchapman/901051_20220502_1102_re

Sample #: — (198 captured)

Time: —

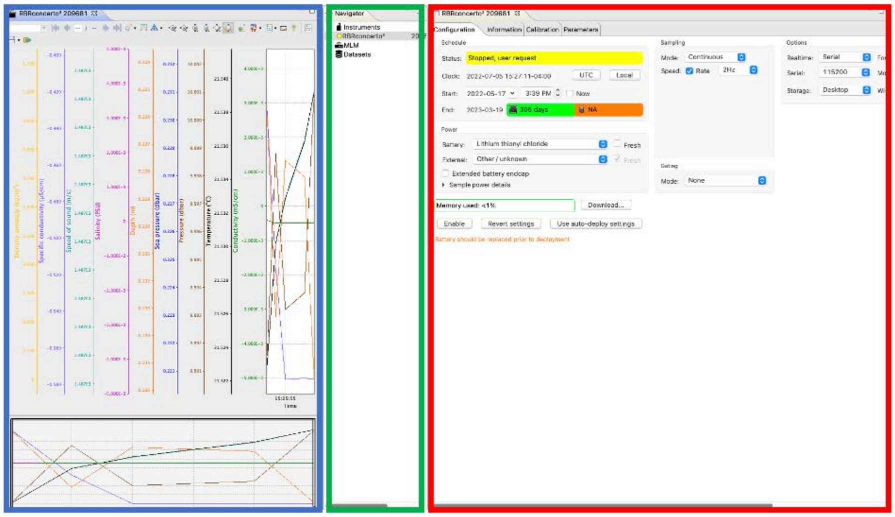
#	Value	Last	Unit
1	—	8.0800319	°C

In the **Navigator** window, you will find the list of instruments connected to Ruskin, as well as MLM (if applicable) and datasets.

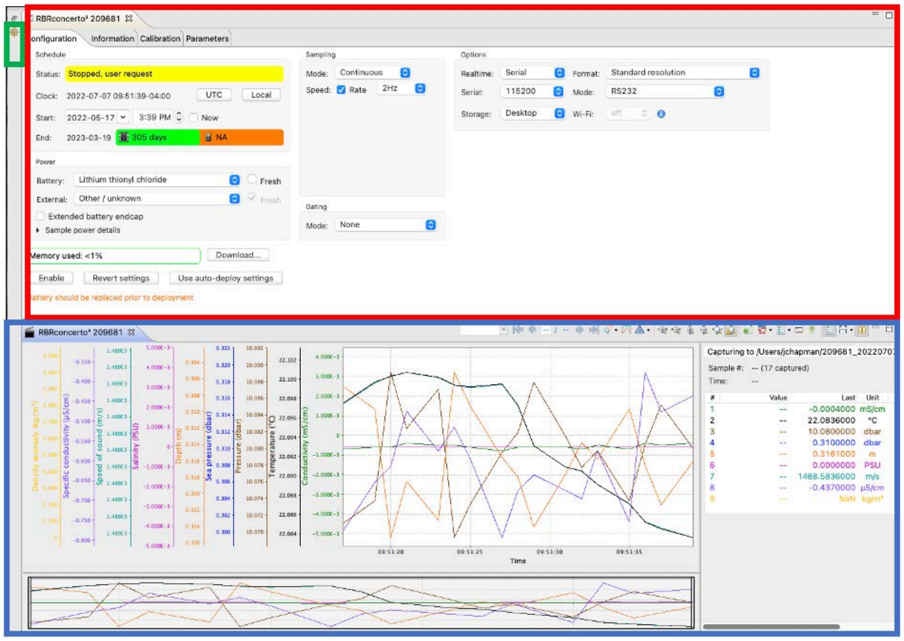
When an instrument on the list is selected, the **Properties** window displays four tabs with information related to this instrument: Configuration, Information, Calibration, and Parameters. Use these tabs to configure your deployment, check the calibration certificates, and verify your parameters.

The data plot will appear in the Plot window below if the realtime option is available and selected for this instrument, or if you click the **Fetch** button on the far right of the toolbar.

Click-and-drag window tabs to rearrange the windows within Ruskin, or move a window outside of Ruskin. Hover over a side of a window, and when the cursor becomes a two-way arrow ↔, move it to adjust the window size.



Click the **Close** ▭ button in the top right corner to close a window. For example, you can close the **Navigator** window to have a better view of the **Properties**. The closed window icon will appear on the sidebar (in this case, **Navigator** ✲). Click the **Multiple windows** 🗗 button above the closed window icon to re-open it.



Schedule

Status: Stopped, user request

Clock: 2022-07-07 09:51:39-04:00

Start: 2022-05-17 3:39 PM □ Now

End: 2023-03-19 305 days NA

Power

Battery: Lithium thionyl chloride □ Fresh

External: Other / unknown ☑ Fresh

□ Extended battery endcap

Sample power details

Memory used: <1%

attery should be replaced prior to deployment

Sampling

Mode: Continuous

Speed: ☑ Rate 2Hz

Gating

Mode: None

Options

Realtime: Serial Format: Standard resolution

Serial: 115200 Mode: RS232

Storage: Desktop Wi-Fi: off

Capturing to /Users/jchapman/209681_2022070

Sample #: -- (17 captured)

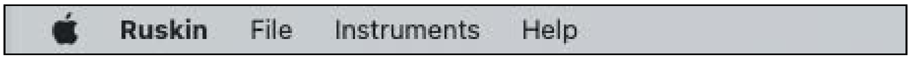
Time: --

#	Value	Last	Unit
1	--	-0.0004000	mS/cm
2	--	22.0836000	°C
3	--	10.0800000	dbar
4	--	0.3100000	dbar
5	--	0.3161000	m
6	--	0.0000000	PSU
7	--	1488.5836000	m/s
8	--	-0.4370000	µS/cm
9	--	NaN	kg/m³

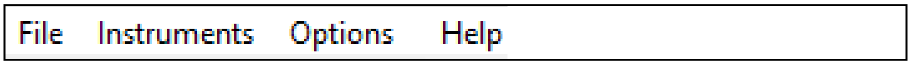
Double-click a window tab to maximise that window, and double-click the tab again to return it to the original size. Alternatively, use the **Single window** □ and **Multiple windows** 🗗 buttons.

ⓘ When you restart Ruskin, it returns to its original layout.

The **Toolbar** at the top left above the **Navigator** window has four dropdown menus. If you are using a Mac computer, it appears as follows:



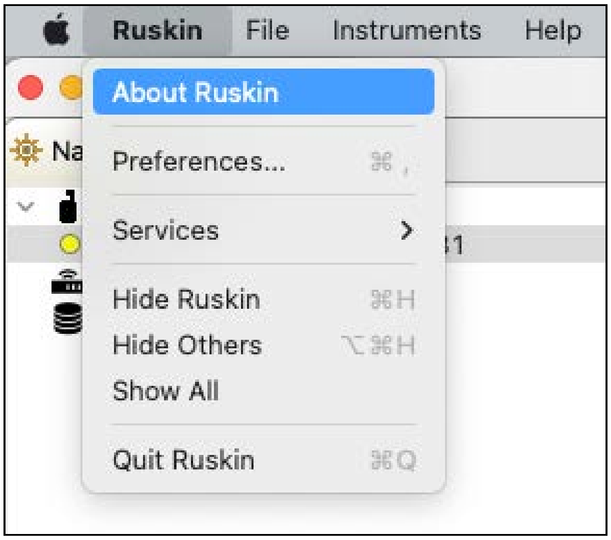
On a Windows PC, the **Toolbar** looks slightly differently:



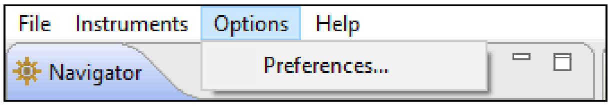
The rest of the interface is the same between the two systems.

2.1 Ruskin / Options

On your Mac, go to the **Ruskin** dropdown menu to access information on current software version, set up your preferences, or quit Ruskin. Note that the rest of the options on this dropdown menu depend on your operating system and are external to Ruskin.

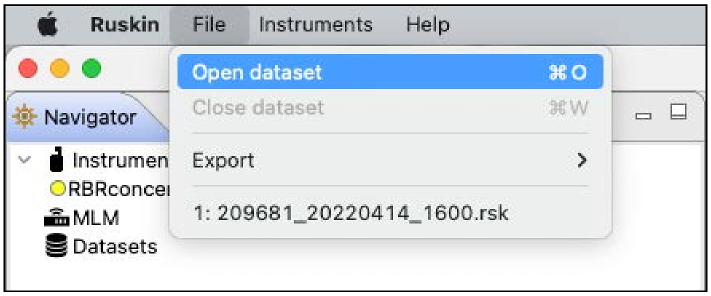


On your Windows PC, use the **Options** dropdown menu to set up your preferences.



2.2 File

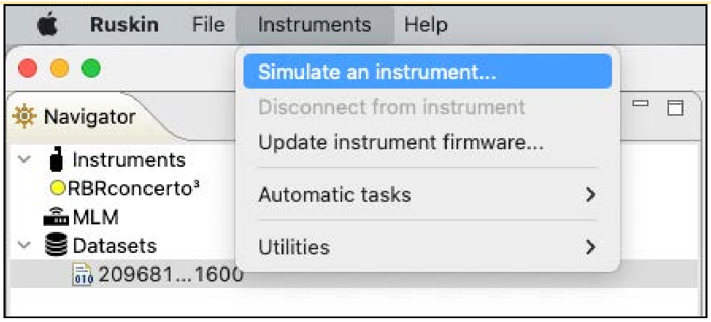
Go to the **File** dropdown menu to open or close a dataset, export your data, and see the list of available datasets. The **Datasets** menu in the **Navigator** windows duplicates some of these options for convenience.



2.3 Instruments

Go to the **Instruments** dropdown menu to simulate an instrument or update your firmware, or take advantage of the utilities.

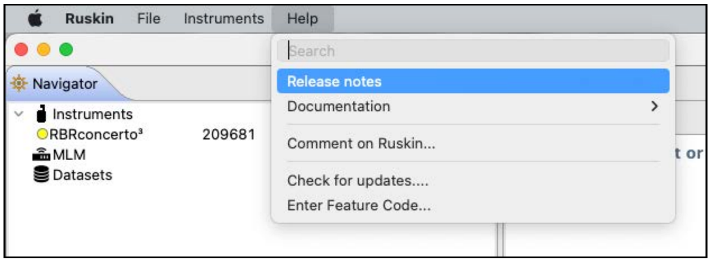
> **Automatic tasks** and **Utilities** are not applicable for RBR*coda*^{3} instruments.



The **Instruments** menu in the **Navigator** windows duplicates some of these options for convenience.

2.4 Help

Go to the **Help** dropdown menu to access the latest release notes and technical documentation, provide your feedback, check for updates, or to add a beta feature. On Mac, you can also use the Search option.



3 Installation

Ruskin is the only piece of software you need to operate your RBR instruments. RBR releases Ruskin software versions compatible with Mac and PC (Ruskin Desktop), as well as with iOS® and Android™ (Ruskin Mobile).

Install the Ruskin Mobile app on your mobile device by going to the Apple App store or Google Play.

Follow the steps below to install Ruskin Desktop on your computer.

3.1 Install Ruskin on a PC

You can install Ruskin on a PC that runs the Windows 7, 8, 8.1, or 10 operating system.

The minimum requirements for Ruskin are:

- OS = Windows 7
- Processor speed = 1.4GHz
- RAM required = 2GB
- Display resolution = 1024x768 recommended
- HDD space for installation = 500MB

The instruments use a USB interface to communicate with Ruskin.

Steps

1. Connect the USB stick included with your instrument to a port on your computer.
2. Navigate to the "Ruskin Installation" folder and doubleclick on the file `RuskinSetup.exe`.
3. Follow the installation wizard. By default, Ruskin will be installed to `C:\RBR`.
4. At the end of the installation, you will see a prompt asking, "Would you like to install the logger driver at this time?"

   ⚠ Realtime sensors will not require USB drivers to operate; they are only required for our standard and compact instruments.

5. Click **Yes** to install the drivers.

   ⚠ You may need to run the setup application as an administrator to install the driver correctly.

   A shortcut to Ruskin will appear on the desktop and in a **Start** menu folder.

ⓘ The most recent version of Ruskin software can be found at https://rbr-global.com/products/software.

3.2 Install Ruskin on a Mac

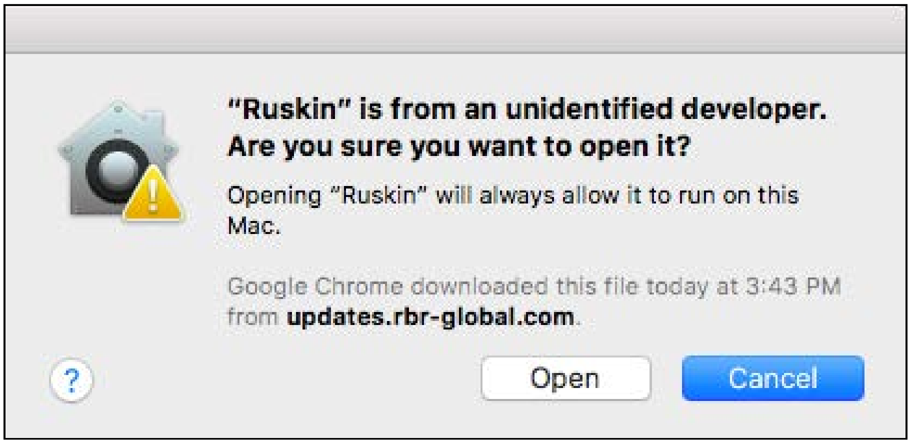
You can install Ruskin on a Mac running OS X 10.12 (Sierra) or later.

Steps

1. Connect the USB stick included with your instrument to a port on your adaptor.
2. Navigate to the "OSX" folder and double-click on the file `Ruskin.dmg`.
3. When the disk image window opens, drag the Ruskin icon into the **Applications** directory and wait for the copy to complete. An application named "Ruskin" will appear in the **Applications** folder.

> ⚠ Always use the default **Applications** folder as the working directory.

4. Right-click on the icon, and select **Open**.
5. The dialogue box shown below will prompt you to authorise the opening of Ruskin.



“Ruskin” is from an unidentified developer. Are you sure you want to open it?

Opening “Ruskin” will always allow it to run on this Mac.

Google Chrome downloaded this file today at 3:43 PM from **updates.rbr-global.com**.

> ⚠ You may need to navigate to **System Preferences** > **Security & privacy** to allow the apps downloaded from “**Anywhere**” to complete the installation.

6. Drag the `Ruskin.app` application to the **Dock**.

3.3 Update Ruskin

To take advantage of new features and bug fixes, ensure that you are using the most recent version of Ruskin.

- Find the most recent version of Ruskin at https://rbr-global.com/products/software.
- Receive an automatic notification about a new release when you start the application.
- Check from within Ruskin by navigating to **Help** > **Check for updates**.
- Send an email to support@rbr-global.com, subject: "Ruskin update request", and ask us to add you to the mailing list. You will receive an email as soon as the new version is released.

If you have a broadband connection, simply follow the installation instructions on your computer. Otherwise, request a USB stick from RBR.

> ⚠ It is not necessary to uninstall the older version of Ruskin before installing the newer version. The installation program deletes the older files before installing the newer ones. It does not delete any Ruskin data files or log files.

3.4 Uninstall Ruskin

Uninstall Ruskin only if you no longer need to manage RBR instruments from your computer.

It is not necessary to uninstall the older version of Ruskin before installing the newer version. The installation program deletes the older files before installing the newer ones. For more information, see Update Ruskin.

> ⚠ Removing Ruskin will not delete your data files or your diagnostic logs.

Uninstall Ruskin on a PC, Windows 7, 8, 8.1, or 10

1. Go to **Start** > **Control Panel** > **Programs**.
2. Under **Programs and Features**, click **Uninstall a program**.
3. Find **Ruskin** on the list.
4. Click **Ruskin** to highlight it.
5. Click **Uninstall**.

Uninstall Ruskin on a Mac, OS X 10.12 or later

Move the Ruskin application from **Applications** to the **Trash**.

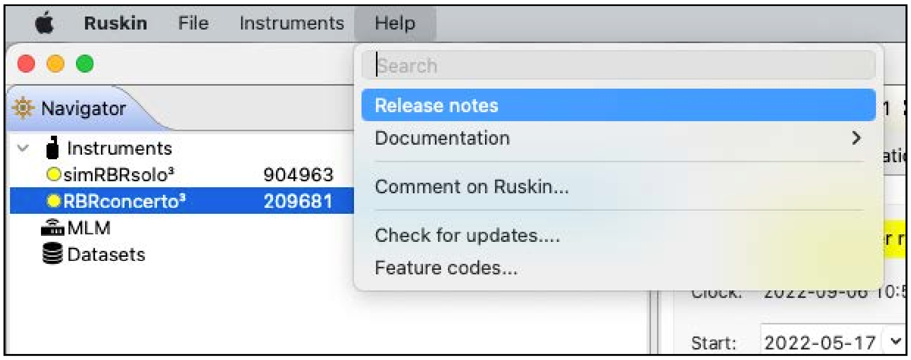
4 Support

You can get in touch with RBR in several different ways:

- Send us an email. For technical questions, use support@rbr-global.com, and for general inquiries, info@rbr-global.com.
- Contact us via the website at https://rbr-global.com/about-rbr/contact-rbr.
- Call us at +1 (613) 599-8900.
- Send us a bug report from within Ruskin itself.

⚠ When experiencing an issue, look for a solution under the **Help** menu before contacting RBR.

The **Help** menu on Ruskin offers access to technical documentation, feedback forms, and software updates. On Mac, you can also search for Ruskin menu options.

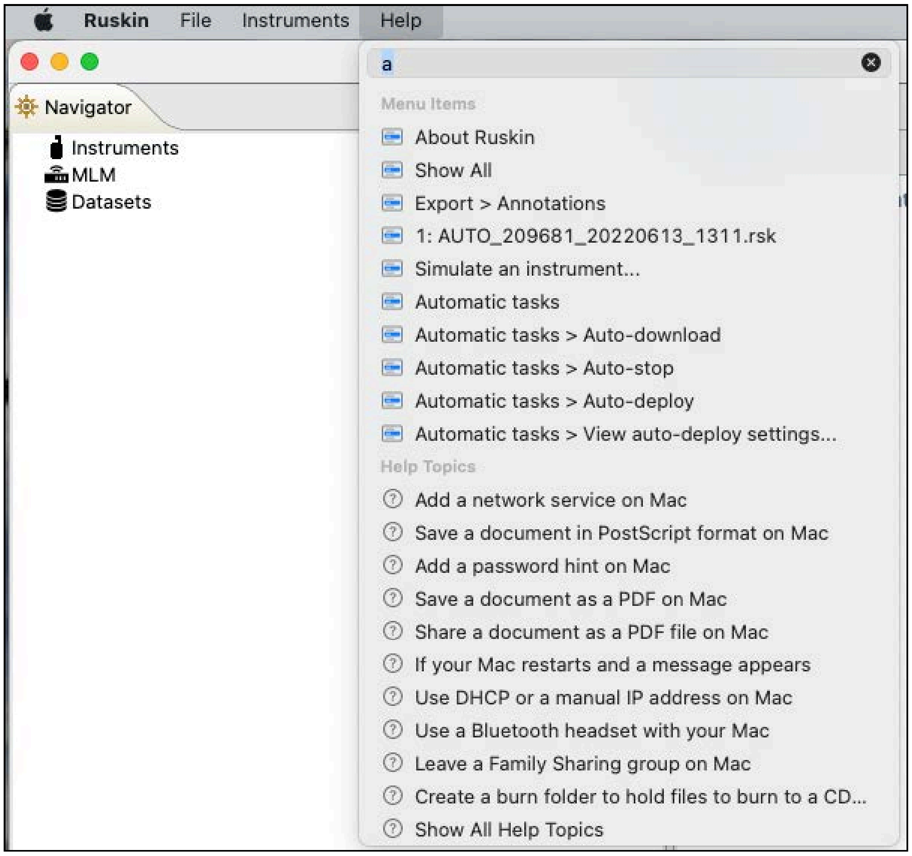


4.1 Search on Mac

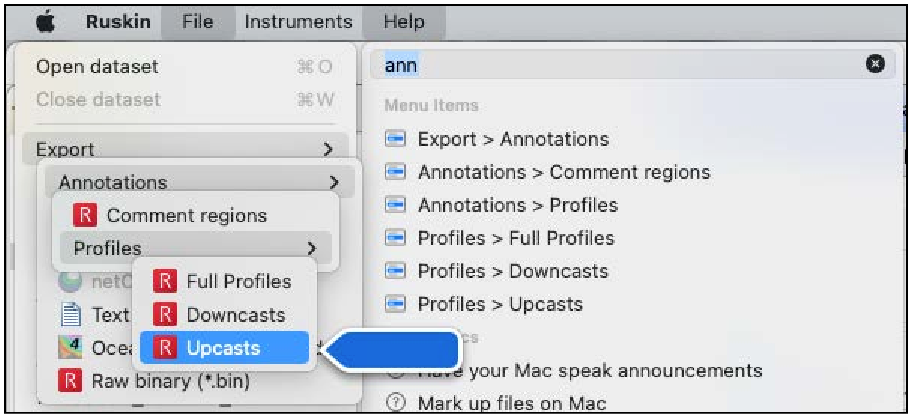
If using a Mac computer, you can take advantage of the **Search** option.

In the **Search** field under **Help**, type in a word, part of the word, or even a single letter to open a list of matching items.

The top half of the list pertains to Ruskin, while the bottom part comes from the Mac environment.


Ruskin
File
Instruments
Help
Navigator
Instruments
MLM
Datasets
a
Menu Items
About Ruskin
Show All
Export > Annotations
1: AUTO_209681_20220613_1311.rsk
Simulate an instrument...
Automatic tasks
Automatic tasks > Auto-download
Automatic tasks > Auto-stop
Automatic tasks > Auto-deploy
Automatic tasks > View auto-deploy settings...
Help Topics
Add a network service on Mac
Save a document in PostScript format on Mac
Add a password hint on Mac
Save a document as a PDF on Mac
Share a document as a PDF file on Mac
If your Mac restarts and a message appears
Use DHCP or a manual IP address on Mac
Use a Bluetooth headset with your Mac
Leave a Family Sharing group on Mac
Create a burn folder to hold files to burn to a CD...
Show All Help Topics

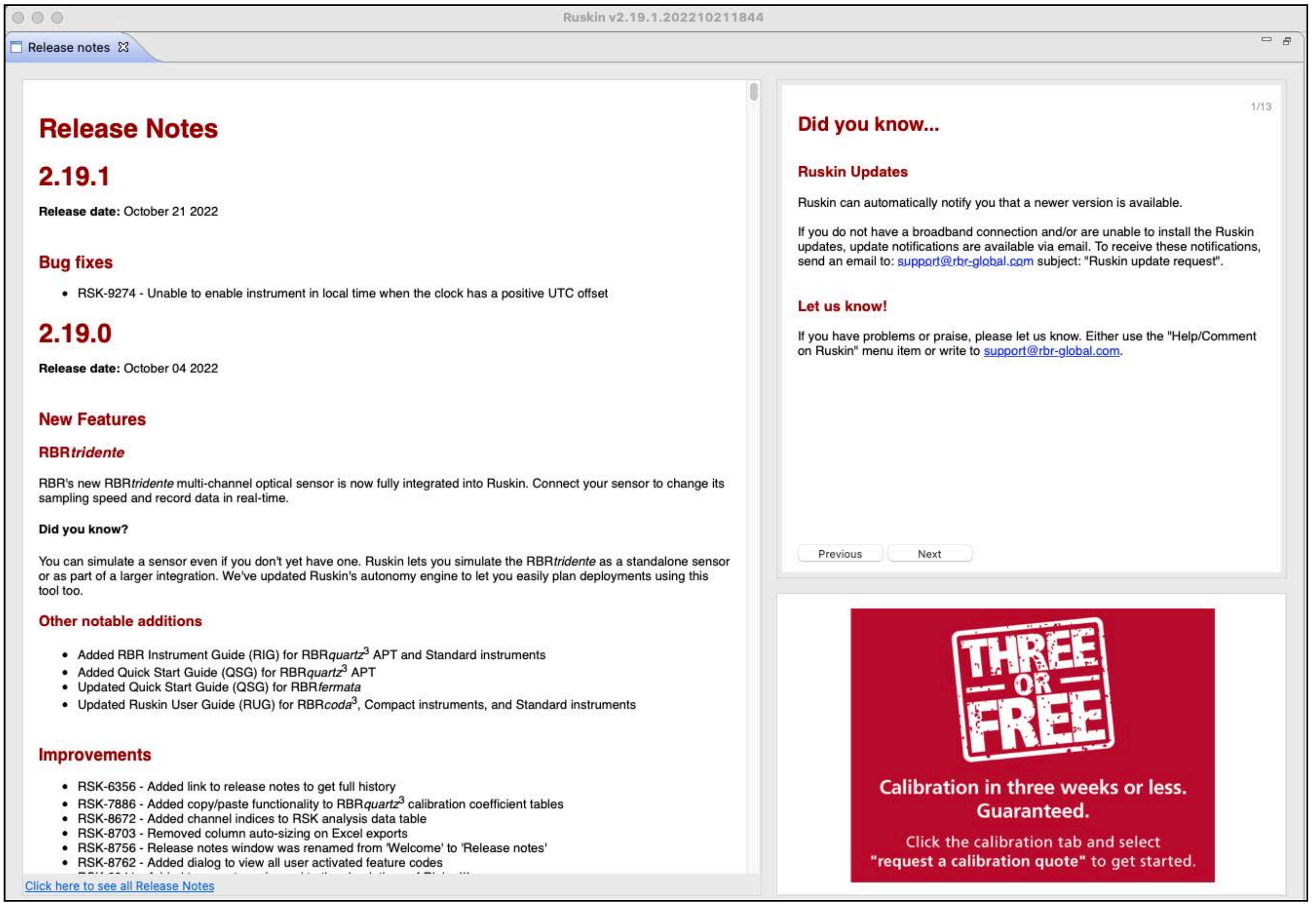
Scroll down the Ruskin list and select an item to locate. Ruskin will open the toolbar menu and point to the option in the dropdown menu.



You can execute the located option right away.

4.2 Release notes

Click on **Help > Release notes** to open the **Welcome** window. It will outline new features, improvements, and bug fixes associated with each release version.



Release Notes

2.19.1

Release date: October 21 2022

Bug fixes

- RSK-9274 - Unable to enable instrument in local time when the clock has a positive UTC offset

2.19.0

Release date: October 04 2022

New Features

RBR*tridente*

RBR's new RBR*tridente* multi-channel optical sensor is now fully integrated into Ruskin. Connect your sensor to change its sampling speed and record data in real-time.

Did you know?

You can simulate a sensor even if you don't yet have one. Ruskin lets you simulate the RBR*tridente* as a standalone sensor or as part of a larger integration. We've updated Ruskin's autonomy engine to let you easily plan deployments using this tool too.

Other notable additions

- Added RBR Instrument Guide (RIG) for RBR*quartz*^{3} APT and Standard instruments
- Added Quick Start Guide (QSG) for RBR*quartz*^{3} APT
- Updated Quick Start Guide (QSG) for RBR*fermata*
- Updated Ruskin User Guide (RUG) for RBR*coda*^{3}, Compact instruments, and Standard instruments

Improvements

- RSK-6356 - Added link to release notes to get full history
- RSK-7886 - Added copy/paste functionality to RBR*quartz*^{3} calibration coefficient tables
- RSK-8672 - Added channel indices to RSK analysis data table
- RSK-8703 - Removed column auto-sizing on Excel exports
- RSK-8756 - Release notes window was renamed from 'Welcome' to 'Release notes'
- RSK-8762 - Added dialog to view all user activated feature codes

Click here to see all Release Notes

1/13

Did you know...

Ruskin Updates

Ruskin can automatically notify you that a newer version is available.

If you do not have a broadband connection and/or are unable to install the Ruskin updates, update notifications are available via email. To receive these notifications, send an email to: support@rbr-global.com subject: "Ruskin update request".

Let us know!

If you have problems or praise, please let us know. Either use the "Help/Comment on Ruskin" menu item or write to support@rbr-global.com.

Calibration in three weeks or less. Guaranteed.

Click the calibration tab and select "request a calibration quote" to get started.

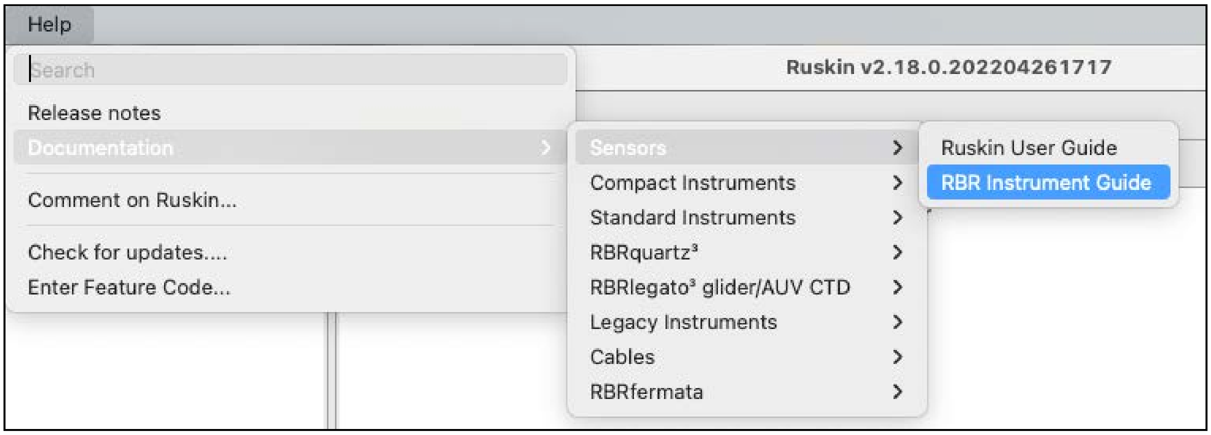
Click the link at the bottom left to see the full history of the release notes. The link will take you to the RBR support page.

Alternatively, scroll down to see brief recaps of the release notes from all previous versions.

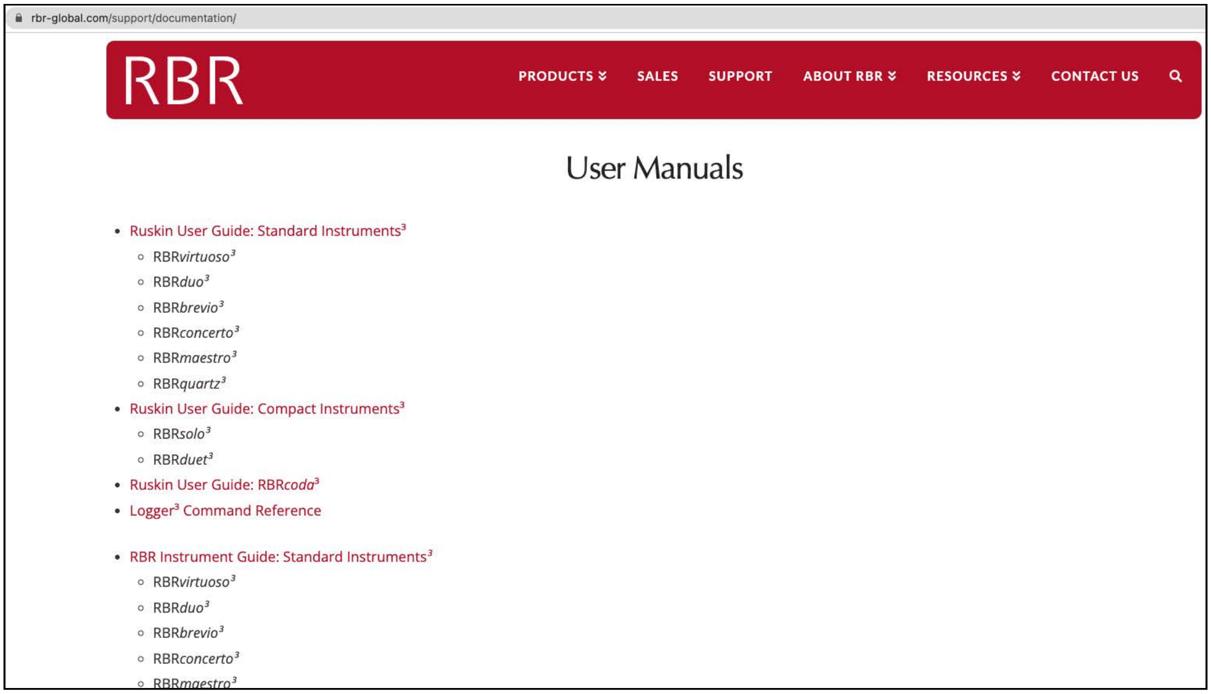
Click the **x** to close the **Release notes** window and return to the **Navigator** mode.

4.3 Documentation

Click on **Help > Documentation** to open the menu tree. Browse through a variety of guides available for instruments and accessories.



Please note that you can always access the most current revisions of RBR documents on the RBR website.



rbr-global.com/support/documentation/

RBR   PRODUCTS   SALES   SUPPORT   ABOUT RBR   RESOURCES   CONTACT US

User Manuals

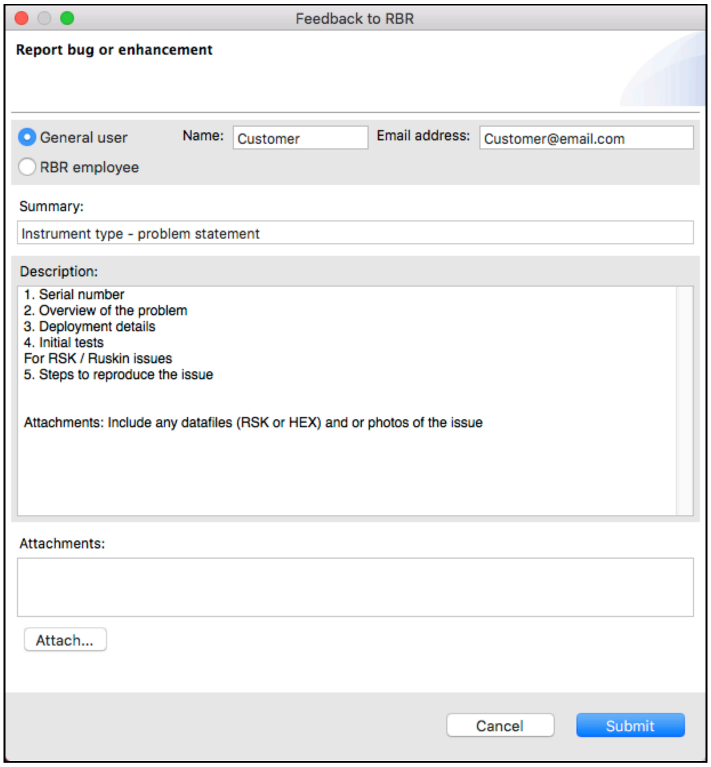
- Ruskin User Guide: Standard Instruments^3
    - RBR*virtuoso*^{3}
    - RBR*duo*^{3}
    - RBR*brevio*^{3}
    - RBR*concerto*^{3}
    - RBR*maestro*^{3}
    - RBR*quartz*^{3}
- Ruskin User Guide: Compact Instruments^3
    - RBR*solo*^{3}
    - RBR*duet*^{3}
- Ruskin User Guide: RBR*coda*^{3}
- Logger^3 Command Reference
- RBR Instrument Guide: Standard Instruments^3
    - RBR*virtuoso*^{3}
    - RBR*duo*^{3}
    - RBR*brevio*^{3}
    - RBR*concerto*^{3}
    - RBR*maestro*^{3}

4.4 Comment on Ruskin

There is an advantage to sending a bug report from within Ruskin because this allows you to include the diagnostic logs, RSK datasets, screenshots, and any other files that could help us reproduce the problem and find the solution as quickly as possible. Follow the steps below.

Steps

1. Click on **Help > Comment on Ruskin...** . The **Feedback to RBR** form will appear.
2. Enter your identification information.
3. Enter a brief summary of the problem.
4. Provide a detailed description (serial number for the instrument, overview of the problem, deployment details, initial tests, and, for software bugs, steps to reproduce the issue)
5. Include any attachments, if required.
6. Verify all the information entered and click **Submit**.



● General user
○ RBR employee

Name: Customer

Email address: Customer@email.com

Summary:
Instrument type - problem statement

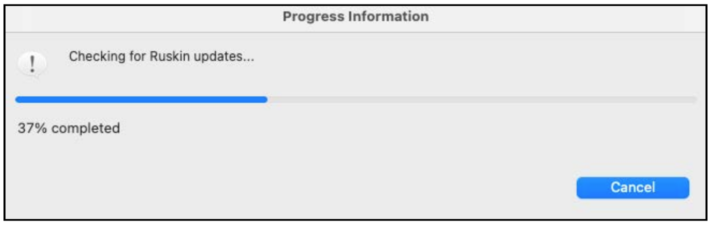
Description:
1. Serial number
2. Overview of the problem
3. Deployment details
4. Initial tests
For RSK / Ruskin issues
5. Steps to reproduce the issue

Attachments: Include any datafiles (RSK or HEX) and or photos of the issue

Attachments:

4.5 Check for updates

Click on **Help > Check for updates** to search for the latest software version.



Checking for Ruskin updates...

37% completed

If your software is up-to-date, no further action will be required.

If an update is found, Ruskin will offer to install it. Follow the prompts. Restart Ruskin for the updates to take effect.

4.6 Feature codes

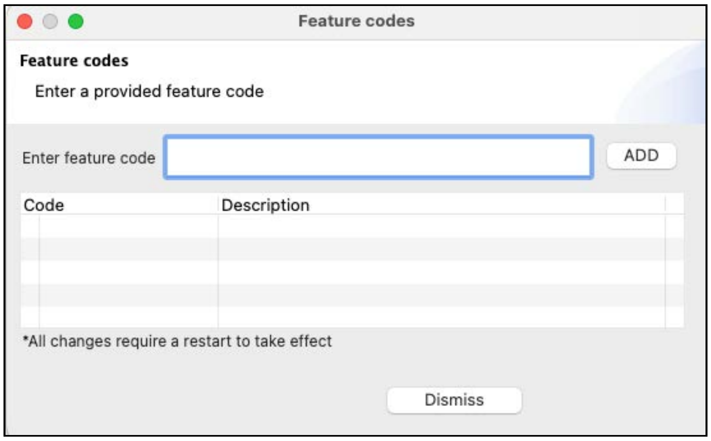
Sometimes, RBR may release new features to certain users only, as a beta trial. This includes:

- Features in early development
- Features supporting a new product, in preparation for the product launch
- Features requested by a specific customer

Usually, the goal is to get feedback and improve the feature before its official release. Contact your sales representative or RBR support team at support@rbr-global.com if you have questions about your feature code.

Steps

1. Click on **Help > Feature codes...**.
   The **Feature codes** window will appear.



Feature codes

Feature codes

Enter a provided feature code

Enter feature code

Code	Description

*All changes require a restart to take effect

2. Enter the feature code provided by RBR.
3. Click **ADD**.

If the code is correct, Ruskin will save the feature in the local database and enable it. In most cases, you will have access to the new feature immediately, without restarting Ruskin. Otherwise, you will receive any additional instructions at the time when you receive the feature code.

Errors

1. The feature code is not valid.
   This error may indicate the code was entered incorrectly or it is not compatible with the current version of Ruskin.
2. The feature code is already enabled.
   This error may occur when another user has already activated this code.

5 Quick start

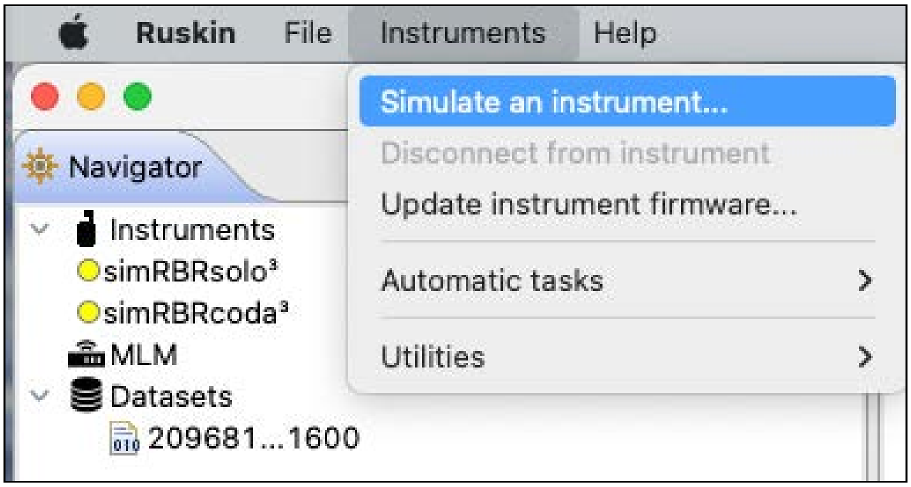
Ruskin Desktop version is designed for computers and laptops. It generates dataset files which include raw measurement data, firmware and software versions, dates and times, sampling settings, power settings, measured and derived parameters, errors and events, calibration coefficients, and diagnostic information, offering the fullest Ruskin experience.

5.1 Simulate an instrument

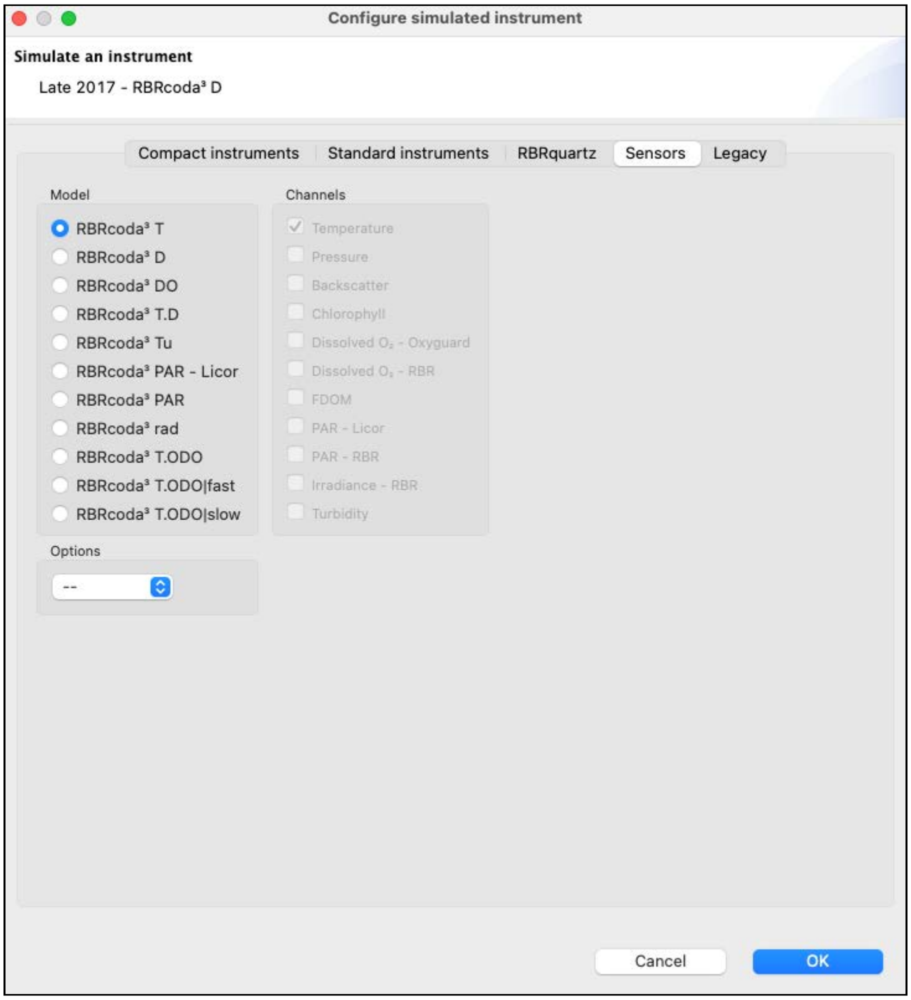
Ruskin can simulate most instruments and sensors that RBR produces. We recommend that you experiment with different simulated options before enabling a schedule for your actual deployment. This practice will also help you become more familiar with how an RBR instrument interacts with Ruskin and how to configure its settings and features. Work with simulated instruments the same way as you work with a real RBR instrument, configuring, calibrating, and fetching data.

Steps

1. From the **Instruments** menu, click **Simulate an instrument**.



The **Configure simulated instrument** dialog box will appear.



Configure simulated instrument

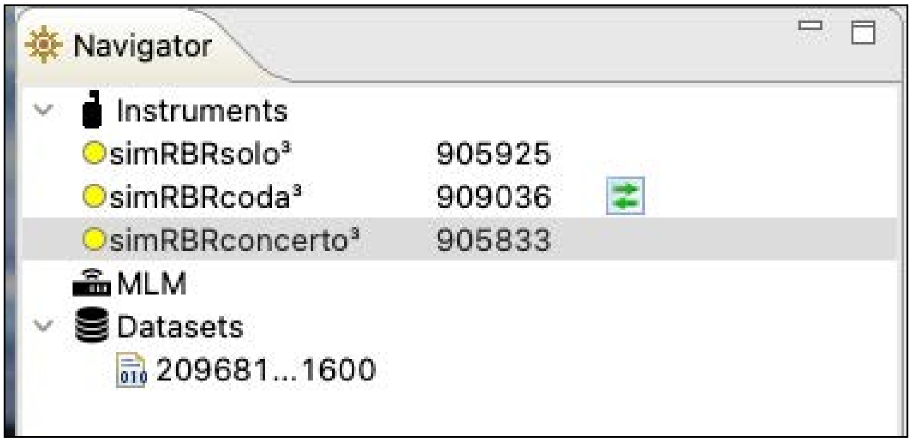
Simulate an instrument

Late 2017 - RBRcoda³ D

Model	Channels
● RBRcoda³ T	☑ Temperature
○ RBRcoda³ D	☐ Pressure
○ RBRcoda³ DO	☐ Backscatter
○ RBRcoda³ T.D	☐ Chlorophyll
○ RBRcoda³ Tu	☐ Dissolved O_2 - Oxyguard
○ RBRcoda³ PAR - Licor	☐ Dissolved O_2 - RBR
○ RBRcoda³ PAR	☐ FDOM
○ RBRcoda³ rad	☐ PAR - Licor
○ RBRcoda³ T.ODO	☐ PAR - RBR
○ RBRcoda³ T.ODO|fast	☐ Irradiance - RBR
○ RBRcoda³ T.ODO|slow	☐ Turbidity

Options: --

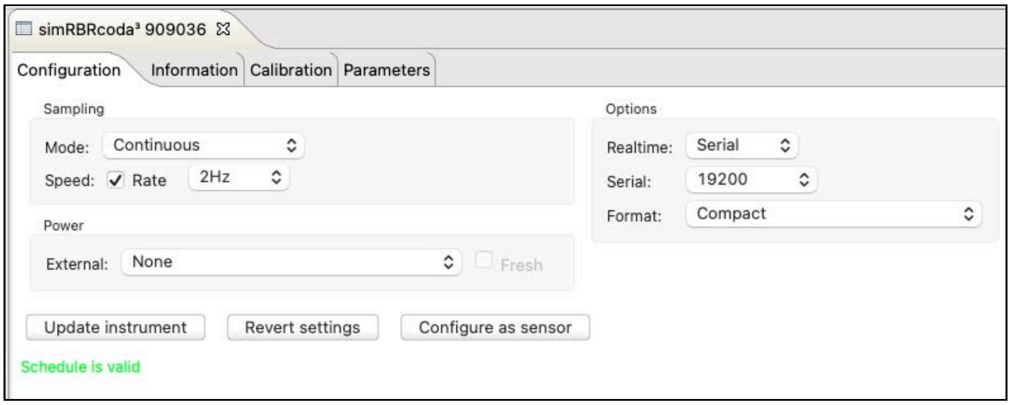
2. Click your instrument family and make your selections.
3. Click **OK** to proceed.
4. Your instrument will appear under **Instruments** in the **Navigator** view. Simulated instruments will have a prefix "sim" in the front of their name.



Navigator

- Instruments
  - simRBRsolo³ 905925
  - simRBRcoda³ 909036
  - simRBRconcerto³ 905833
- MLM
- Datasets
  - 209681...1600

The instrument properties will be displayed on the right, under four tabs.



Sampling

Mode: Continuous

Speed: ☑ Rate 2Hz

Power

External: None □ Fresh

Options

Realtime: Serial

Serial: 19200

Format: Compact

Schedule is valid

- The **Configuration** tab displays sampling mode and speed, serial connection details, and external power options
- The **Information** tab includes general details, such as the instrument model, generation, serial number, firmware version, power source, and the channels
- The **Calibration** tab allows access to the current calibrations for each channel in the instrument
- The **Parameters** tab outlines default values used by the sensor when performing computations and derivations

5. Familiarise yourself with the instrument by changing configuration options and comparing deployment estimates. Select the desired configuration using the dropdown options or entering the values manually in the fields provided. The data plot will appear at the bottom of the screen.
6. Manage multiple instruments by switching between them in the **Navigator** window.

To remove a simulated instrument from Ruskin, right-click it in the **Navigator** window, and select **Disconnect from instrument**. You can also use the **Instruments** menu.
Alternatively, simply close its **Plot** tab by clicking the **X.** The instrument will disappear from the list.

5.2 Deploy an RBR sensor

Before using your RBR sensors, you may want to experiment with different options by simulating your instrument on Ruskin. When you are ready to use your physical RBR instrument, follow the steps below to ensure that you measure exactly what you want on the first attempt.

The only two parameters needed to operate your sensor in streaming mode are the streaming speed, and the serial baudrate. The sensor will begin streaming data as soon as power is provided.

- The factory streaming speed is the fastest that the sensor is rated for, e.g. | fast32 will be set to 32Hz.
- The factory baudrate is 9600.

ⓘ Ruskin supports connection baudrate options from 1200 to 115200.

Steps

1. Connect your RBR sensor to a computer using a serial interface or a serial to USB adaptor.
   The instrument will appear in the **Navigator** view after a few seconds.
2. Click on the instrument.
   Its properties will appear to the right, grouped under four tabs:
   - The **Configuration** tab displays sampling mode and speed, serial connection details, and external power options
   - The **Information** tab includes general details, such as the instrument model, generation, serial number, firmware version, power source, and the channels
   - The **Calibration** tab allows access to the current calibrations for each channel in the instrument
   - The **Parameters** tab outlines default values used by the sensor when performing computations and derivations

   The **Plot** window located at the bottom of the screen will display realtime data streaming from the sensor.

Change the parameters using the on-screen controls and they take effect immediately. All settings persist over power cycles.

RBR realtime sensors will begin streaming data as soon as power is provided.

ⓘ The RBR realtime sensors can operate in streaming or polled modes. To set up the polled mode, you will need to communicate with your instrument using a serial terminal. See the OEM commands reference section for details.

Ruskin automatically saves the data streamed or fetched from your sensor.

Go to Preferences to change your default location for RSK files.

See RSK files for the file naming conventions.

6 Configuration

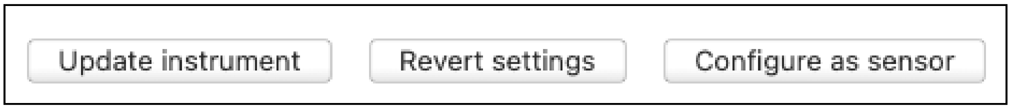
Use the **Configuration** tab to set up your deployment parameters.

After adjusting any parameters during deployment, verify that they meet your requirements. To return to previous settings, click **Revert settings**.

To save new settings, click **Update instrument.** New values will replace all previously set deployment parameters.

> ⚠ You cannot use **Revert settings** after clicking **Update instrument.** Once the settings are saved to the instrument, you will need to restore old values manually and update the instrument again.

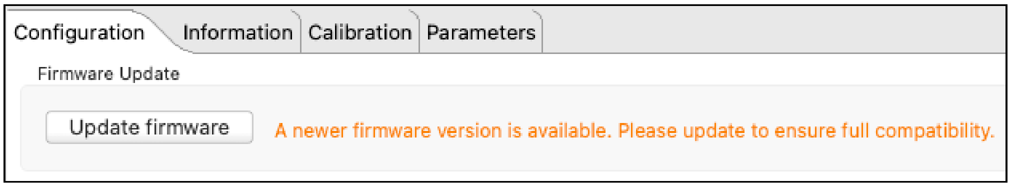
If using your RBR*coda*^{3} as a sensor attached to a standard instrument, click Configure as sensor to set up compatible baudrate and sampling speed. To return to previous settings, click **Revert settings.**



6.1 Update firmware

Ruskin automatically checks if your instrument firmware is up-to-date. Updates may be critical, routine, or customised.

If a newer version is available and this update is critical, the **Update firmware** button and a warning message will appear at the top of the **Configuration** tab.

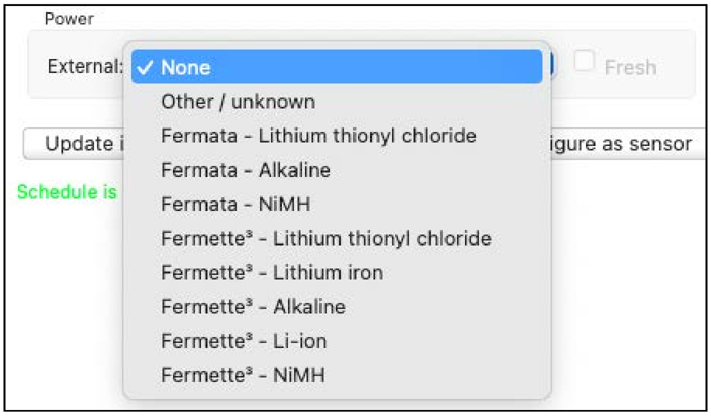


A newer firmware version is available. Please update to ensure full compatibility.

See Critical firmware updates for details.

6.2 Power

The **Power** section includes external power options for the RBR*coda*^{3} instruments. These include RBR*fermata* and RBR*fermette*^{3} underwater battery canisters with various battery chemistries available. The RBR*fermata* uses D-type batteries, and the RBR*fermette*^{3} uses AA cells.



AA-type batteries (RBR*fermette*^{3}*)*

Name	Chemistry	Voltage (V)	mAh*
Lithium thionyl chloride	Li-SOCl$_2$	3.6	2700
Lithium iron	Li-FeS$_2$	1.5	3500
Alkaline	Zn-MnO$_2$	1.5	2500
Li-ion	Li-NiMnCo	3.6	750
Nickel metal hydride	Ni-MH	1.2	2400

*The nominal capacity of each type of battery is given at room temperature.

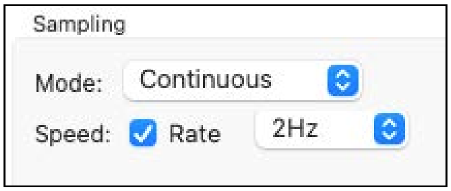
D-type batteries (RBR*fermata)*

Name	Chemistry	Voltage (V)	mAh*
Lithium thionyl chloride	Li-SOCL $_2$	3.6	18400
Alkaline	Zn-MnO$_2$	1.5	16000
Nickel metal hydride	Ni-MH	1.2	10000

*The nominal capacity of each type of battery is given at room temperature.

6.3 Sampling

The **Sampling** section includes three fields: **Mode**, **Speed**, and **Rate**, as well as several optional fields available with certain sampling modes.



Sampling

Mode: Continuous

Speed: ☑ Rate 2Hz

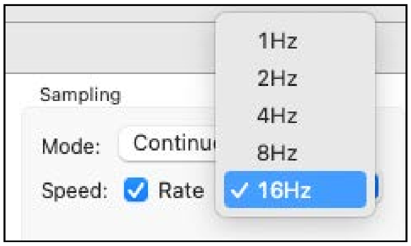
Mode

Most RBR*coda*^{3} realtime sensors can only operate in **Continuous** sampling mode. The RBR*coda*^{3} D | tide16 and RBR*coda*^{3} T.D | tide16 also support **Tide** sampling mode.

Speed and Rate

These are the only two parameters you need to configure **Continuous** sampling mode.

Check **Speed** to select the sampling **Rate** in Hz from the dropdown menu. The default option is always the highest instrument rating; for example, | fast16 will default to 16Hz.

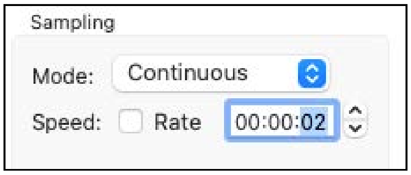


Sampling

Mode: Continu

Speed: ☑ Rate ✓ 16Hz

Uncheck **Speed** to enter the sampling **Rate** manually in hours, minutes, and seconds (HH:mm:ss format). The fastest rate to enter manually is 1s. Type the numbers in, or use the up and down arrows.



Sampling

Mode: Continuous

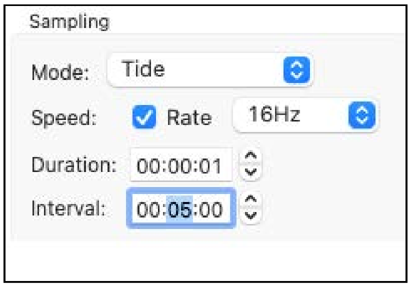
Speed: □ Rate 00:00:02

When operating in **Tide** sampling mode, you need to configure two more parameters: **Duration** and **Interval**.

The RBR*coda*^{3} D | tide16 and RBR*coda*^{3} T.D | tide16 sensors average a burst of pressure readings and stream only the resulting average.

Duration specifies the time range over which to average. It must always be longer than **Speed**.

Interval is the repetition period for performing the averaging. It must always be longer than **Duration**.



Sampling

Mode: Tide

Speed: ☑ Rate 16Hz

Duration: 00:00:01

Interval: 00:05:00

In the example shown, the sampling rate selected is 16Hz (16 samples per second). The instrument will be sampling for a duration of one second, then pause and average the sixteen samples collected.

The sensor will stream only one sample, averaged over the sixteen samples collected during the period of the **Duration** (one second) and remain paused for the time of the **Interval** (five minutes).

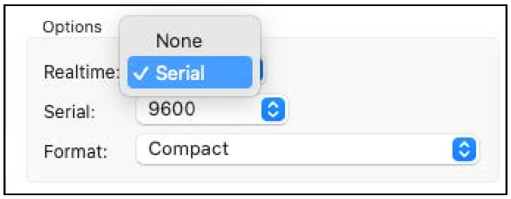
After the interval ends, the sensor will start sampling again, repeating the same cycle.

> ⓘ At every **Interval**, the instrument records only one sample, averaged over all samples collected during the period of **Duration**. The rest of the samples are discarded after averaging is performed.

6.4 Options

Three dropdown menus are available for the RBR*coda*³ in the **Options** section: **Realtime**, **Serial**, and **Format**.

Realtime specifies your external streaming connection. Select **Serial** to stream live data over RS-232. Sensors will begin streaming data as soon as power is provided.



Options

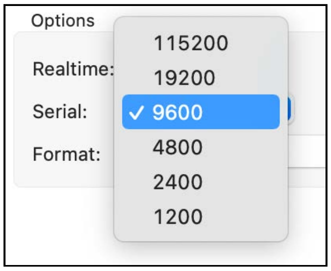
Realtime: Serial

Serial: 9600

Format: Compact

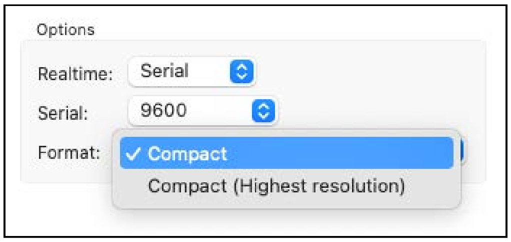
Otherwise, select **None** and use the fetching feature.

Serial offers possible baudrate options for your serial connection. The factory baudrate is 9600.



Select the output format for the realtime data from two dropdown options.

- **Compact**
  All values are shown to four decimal places, with no units displayed.
- **Compact (Highest resolution)**
  All values are shown to the number of decimal places required to adequately represent them (not limited to four decimal places), with no units displayed.



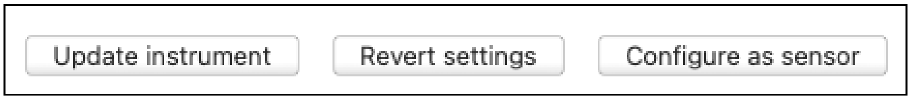
Ruskin automatically saves the data streamed from your sensor. Go to **Preferences >** General to change your default location for RSK files.

See RSK files for the file naming conventions.

6.5 Configure as sensor

All RBR*coda*^{3} sensors may be integrated with a standard RBR instrument, such as the RBR*concerto*^{3} or RBR*maestro*^{3}.

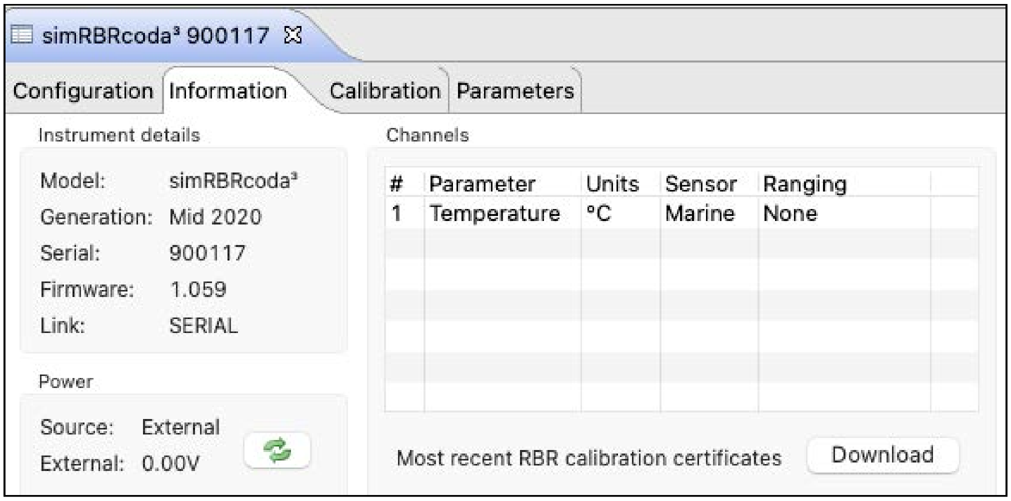
Click the **Configure as sensor** button before connecting to the host instrument to set up correct the baudrate and sampling speed.



Failing to perform this operation prior to attaching the sensor to the instrument may yield poor results as the host instrument will be unable to communicate with the sensor.

7 Information

Use the **Information** tab to view general information for each of your instruments, such as as model, generation, serial number, firmware version, power source, and the channels.



Model: simRBRcoda³
Generation: Mid 2020
Serial: 900117
Firmware: 1.059
Link: SERIAL

Source: External
External: 0.00V

#	Parameter	Units	Sensor	Ranging
1	Temperature	°C	Marine	None

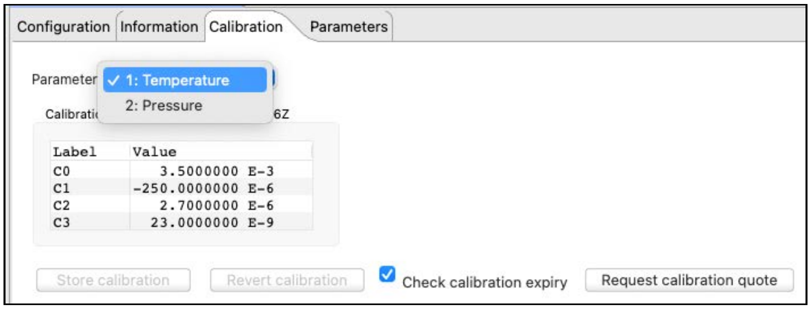
In the **Instrument details** section, find the model, generation, and serial number for your instrument, as well as the firmware version installed and link used.

Refer to the **Power** section for the power source, power type, and voltage.

Under **Channels**, confirm measured and derived parameters for your instrument.

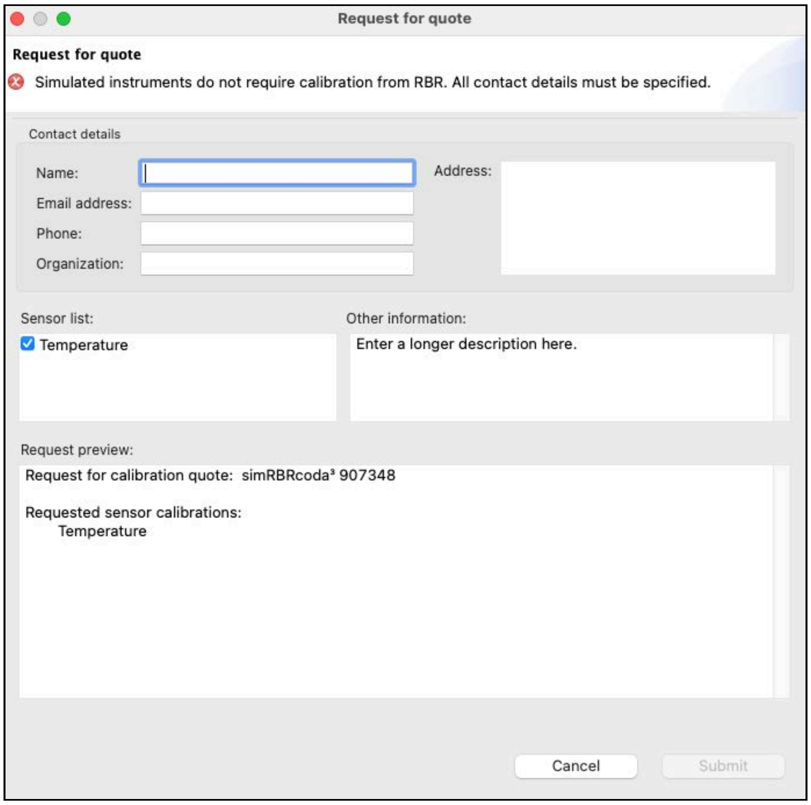
8 Calibration

The **Calibration** tab displays calibration coefficients for each parameter, and the date and time of the last calibration. Select a sensor from the **Parameter** dropdown menu to display its calibrations.



Label	Value
C0	3.5000000 E-3
C1	-250.0000000 E-6
C2	2.7000000 E-6
C3	23.0000000 E-9

RBR calibration certificates contain calibration equations, coefficients, and residuals for each sensor.
RBR recommends calibrating your instrument before any critical deployment, periodically once a year, or if you suspect the readings to be out of specifications.
Discuss your calibration requirements with RBR. In some cases, the instrument may need to be returned to RBR to have it checked and re-calibrated.
To contact RBR for our current calibration fees, click **Request calibration quote**. A **Request for quote** dialog box will appear.



Request for quote

Simulated instruments do not require calibration from RBR. All contact details must be specified.

☑ Temperature

Enter a longer description here.

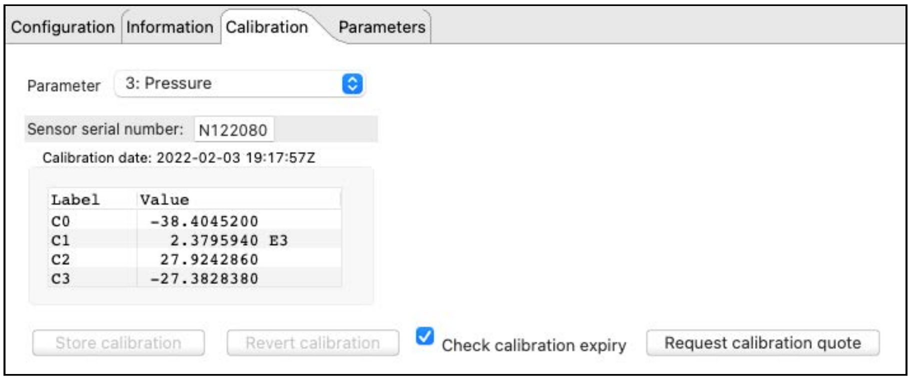
Request for calibration quote: simRBRcoda³ 907348

Requested sensor calibrations:
Temperature

Verify that all information is correct and click **Submit**.

Simple calibration controls

When using the simple version, you will be able to view and edit only the main calibration coefficients, displayed as "C" values: C0, C1, C2, C3.



Parameter: 3: Pressure

Sensor serial number: N122080

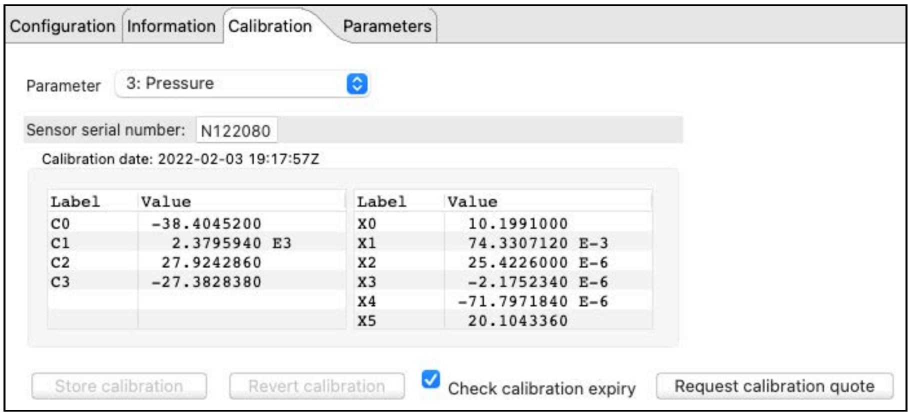
Calibration date: 2022-02-03 19:17:57Z

Label	Value
C0	-38.4045200
C1	2.3795940 E3
C2	27.9242860
C3	-27.3828380

☑ Check calibration expiry

Advanced calibration controls

When advanced calibration controls are enabled in **Preferences >** General, you will be able to view and edit both main and dependent coefficients. Dependent coefficients are displayed as "X" values: X0, X1, X2, etc.



Parameter: 3: Pressure

Sensor serial number: N122080

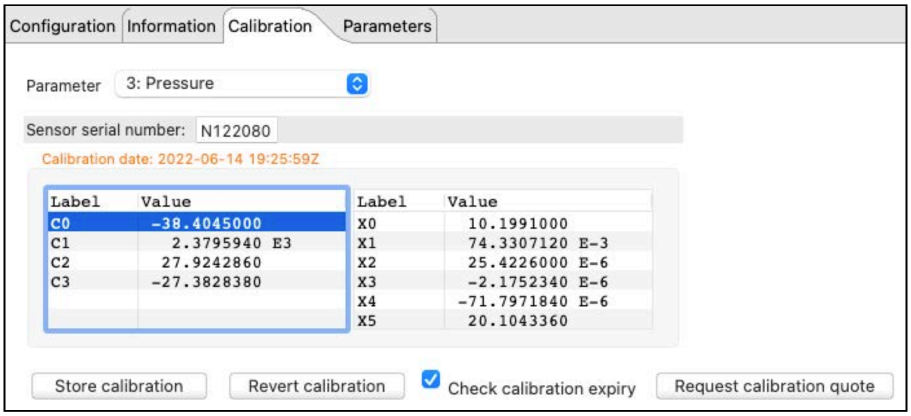
Calibration date: 2022-02-03 19:17:57Z

Label	Value
C0	-38.4045200
C1	2.3795940 E3
C2	27.9242860
C3	-27.3828380

Label	Value
X0	10.1991000
X1	74.3307120 E-3
X2	25.4226000 E-6
X3	-2.1752340 E-6
X4	-71.7971840 E-6
X5	20.1043360

☑ Check calibration expiry

You can edit the coefficients. As soon as you make a change and move to another cell, a new calibration date will be displayed. Also, **Store calibration** and **Revert calibration** buttons will become active.



Parameter: 3: Pressure

Sensor serial number: N122080

Calibration date: 2022-06-14 19:25:59Z

Label	Value
C0	-38.4045000
C1	2.3795940 E3
C2	27.9242860
C3	-27.3828380

Label	Value
X0	10.1991000
X1	74.3307120 E-3
X2	25.4226000 E-6
X3	-2.1752340 E-6
X4	-71.7971840 E-6
X5	20.1043360

Use **Store calibration** to save the new values.

⚠ Always click **Store calibration** to save new coefficients on the instrument. Otherwise, all changes will be lost once your session is closed.

Use **Revert calibration** to recover to the original coefficients if you have not already saved your changes to the instrument.

⚠ You cannot use **Revert calibration** after clicking **Store calibration**. Once the coefficients are saved to the instrument, you will need to restore old values manually and save them again.

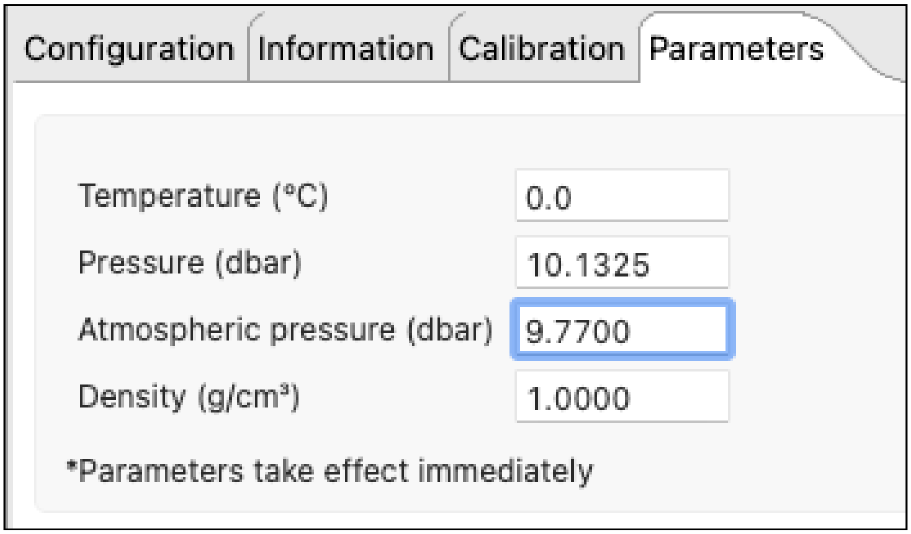
9 Parameters

The **Parameters** tab displays all the parameters the RBR instrument may need for calculating derived channels.

If your instrument can measure a parameter, it will use the measured value. Otherwise, you can enter the necessary values manually, and they will take effect immediately.

For example, you always need atmospheric pressure and water density for derivation of depth. Default values in Ruskin correspond to sea level and sea water. Therefore, it is especially important to adjust these parameters when working at high altitudes in fresh water.

Enter atmospheric pressure (in dbar) and water density (in g/cm^3) manually in the table under the **Parameter** tab. If not entered, default values of 10.1325dbar and 1.0281g/cm^3 will be used.

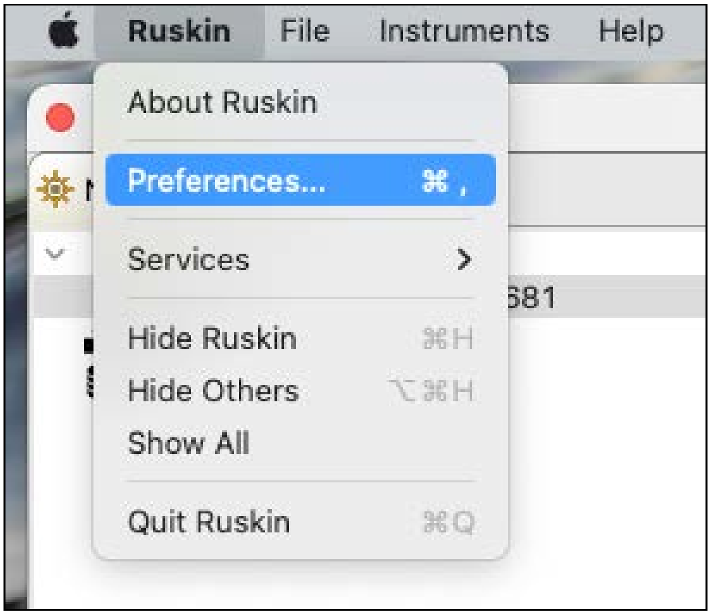


Temperature (°C)	0.0
Pressure (dbar)	10.1325
Atmospheric pressure (dbar)	9.7700
Density (g/cm³)	1.0000

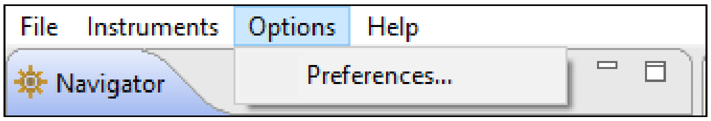
*Parameters take effect immediately

10 Preferences

On your Mac, access **Preferences** from the **Ruskin** dropdown menu in the top toolbar.



On your Windows PC, use the **Options** dropdown menu to set up your preferences.

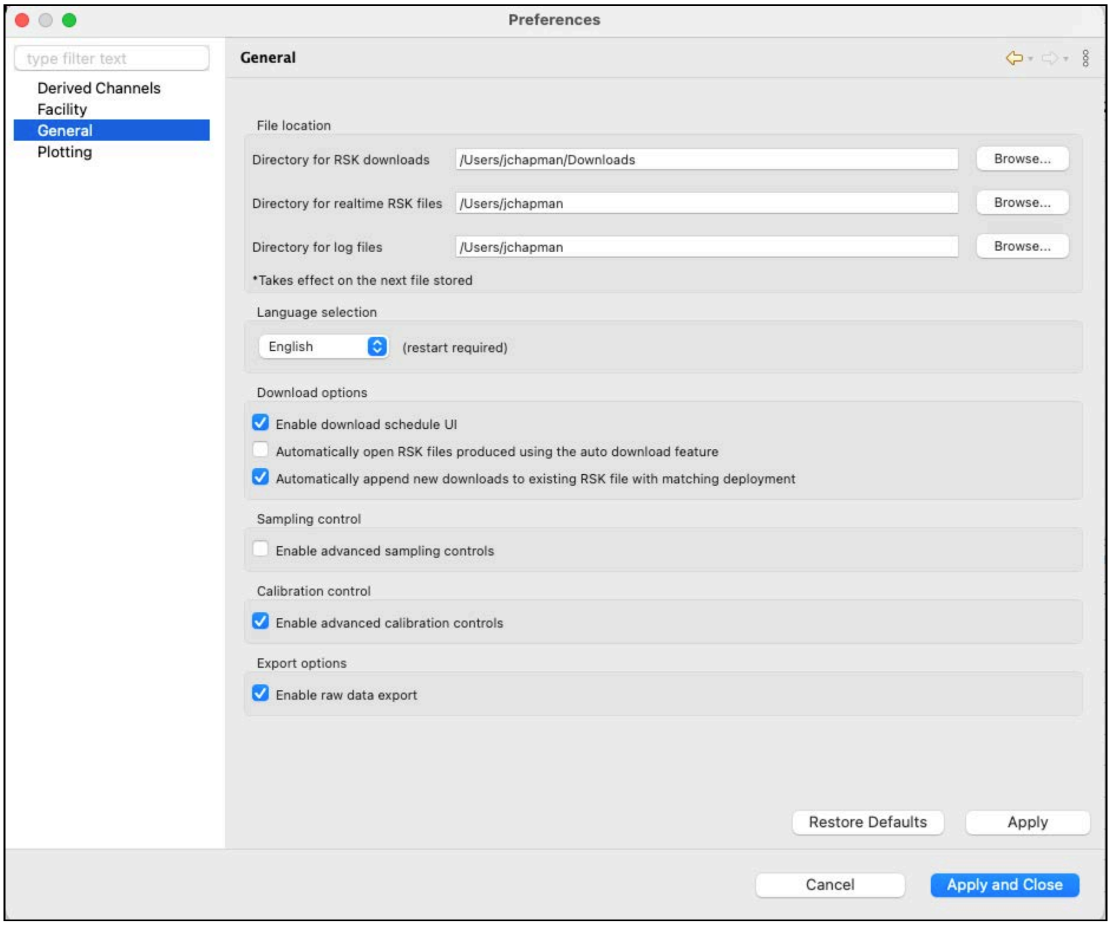


Factory-set global defaults require no changes in most situations. However, as you learn more about what Ruskin can do, you may want to change some of these settings to better suit your requirements. The global preferences include:

- Derived channels (applies to some legacy instruments)
- Facility (authorised users only, requires login)
- General (download locations and options, sampling and calibration controls, export options)
- Plotting (channel visibility and channel colours)

10.1 General

Use the **General** preference page to set up file locations, language, download options, sampling and calibration controls, and export options.



General

File location

Directory for RSK downloads: /Users/jchapman/Downloads — Browse...

Directory for realtime RSK files: /Users/jchapman — Browse...

Directory for log files: /Users/jchapman — Browse...

*Takes effect on the next file stored

Language selection

English (restart required)

Download options

☑ Enable download schedule UI
☐ Automatically open RSK files produced using the auto download feature
☑ Automatically append new downloads to existing RSK file with matching deployment

Sampling control

☐ Enable advanced sampling controls

Calibration control

☑ Enable advanced calibration controls

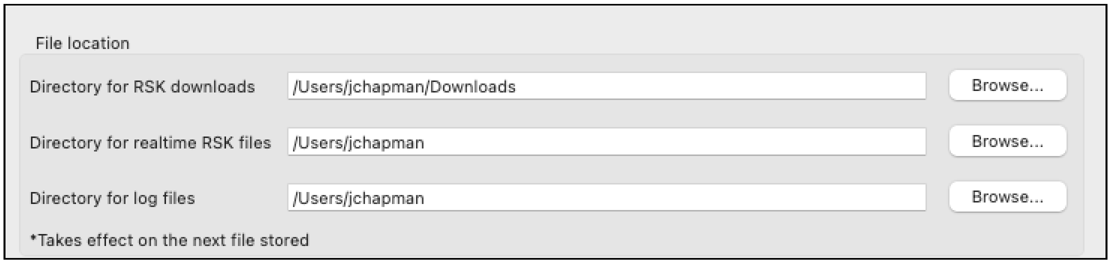
Export options

☑ Enable raw data export

✓ You can click **Apply** to save your changes without closing the dialogue box.

File location

You can control where RSK files, realtime RSK file, and log files are stored by default.
Specify these locations before downloading any data by typing in a directory name or by browsing.



File location

Directory for RSK downloads: /Users/jchapman/Downloads — Browse...

Directory for realtime RSK files: /Users/jchapman — Browse...

Directory for log files: /Users/jchapman — Browse...

*Takes effect on the next file stored

RSK files are the datasets obtained from your instrument.

Realtime RSK files are generated when a connected instrument is streaming or when you enable fetching mode on the instrument.

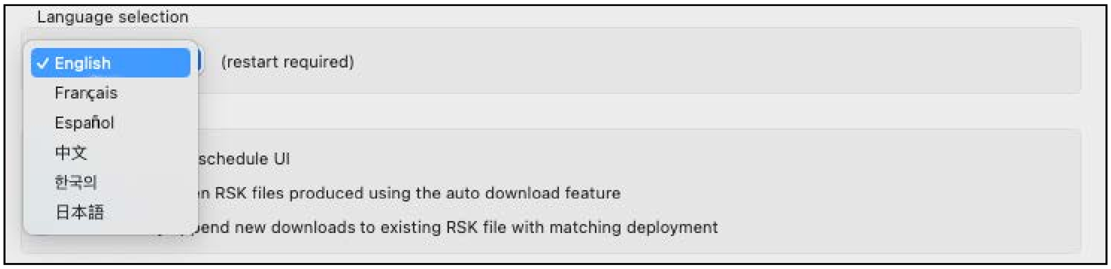
Log files are generated by Ruskin and help when diagnosing issues. These files are automatically attached to issues generated using the **Comment on Ruskin** feature.

⚠ RBR recommends storing your datasets on a local drive rather than a network storage drive. Cloud storage options such as GoogleDrive and Dropbox may extend download times.

You can change where future files are stored by default at any time. When done, click **Apply** or **Apply and close**. The changes will take effect immediately.

Language selection

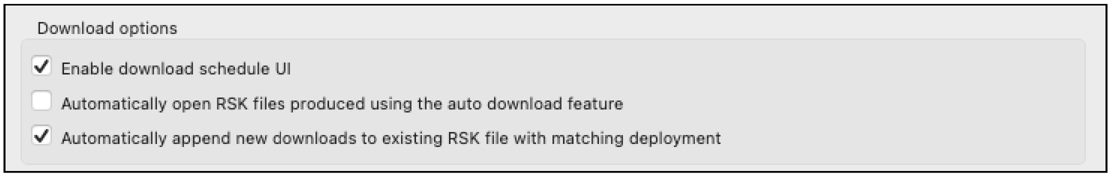
Ruskin is available in six languages: English, French, Spanish, Simplified Chinese, Korean, and Japanese.


Language selection
✓ English
Français
Español
中文
한국의
日本語
(restart required)
schedule UI
n RSK files produced using the auto download feature
end new downloads to existing RSK file with matching deployment

Select your language from the dropdown menu and click **Apply** or **Apply and close**. Ruskin will need to restart for the changes to take effect.

Download options

Select the appropriate checkboxes to change your download preferences.



Download options

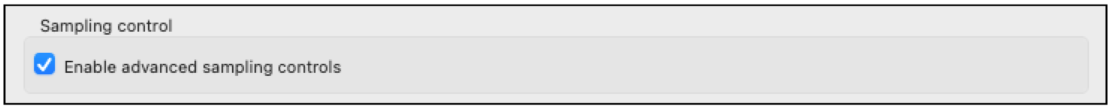
☑ Enable download schedule UI
□ Automatically open RSK files produced using the auto download feature
☑ Automatically append new downloads to existing RSK file with matching deployment

⚠ The **Enable download schedule UI** checkbox is only for legacy instruments.

When done, click **Apply** or **Apply and close**. The changes will take effect immediately.

Sampling control

Select the checkbox to enable advanced controls when setting up your sampling options.



Sampling control

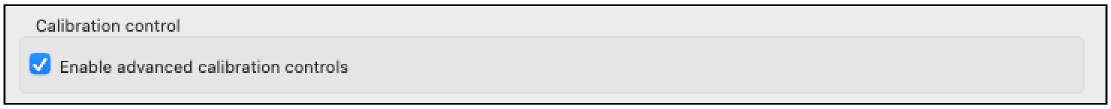
☑ Enable advanced sampling controls

With simple controls, you will select **Speed**, **Duration**, and **Interval** from pre-defined options in three dropdown menus. When advanced controls are enabled, you will be able to enter these parameters manually and achieve a more precise setting.

When done, click **Apply** or **Apply and close**. The changes will take effect immediately.

Calibration control

Select the checkbox to enable advanced controls when setting up your calibration options.



Calibration control

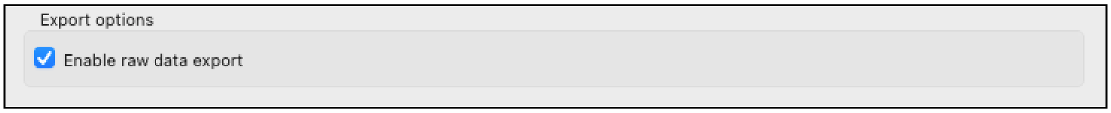
☑ Enable advanced calibration controls

With simple controls, you will be able to view and edit only the main coefficients (C). When advanced controls are enabled, you will be able to view and edit all main and dependent calibration coefficients (C, X) for any channel.

When done, click **Apply** or **Apply and close**. The changes will take effect immediately.

Export options

Select the checkbox to enable raw data export. Use this option for downloading your RSK file when storage space is limited or expensive. The dataset will be exported in the most compact raw binary format.



Export options

☑ Enable raw data export

When done, click **Apply** or **Apply and close**. The changes will take effect immediately.

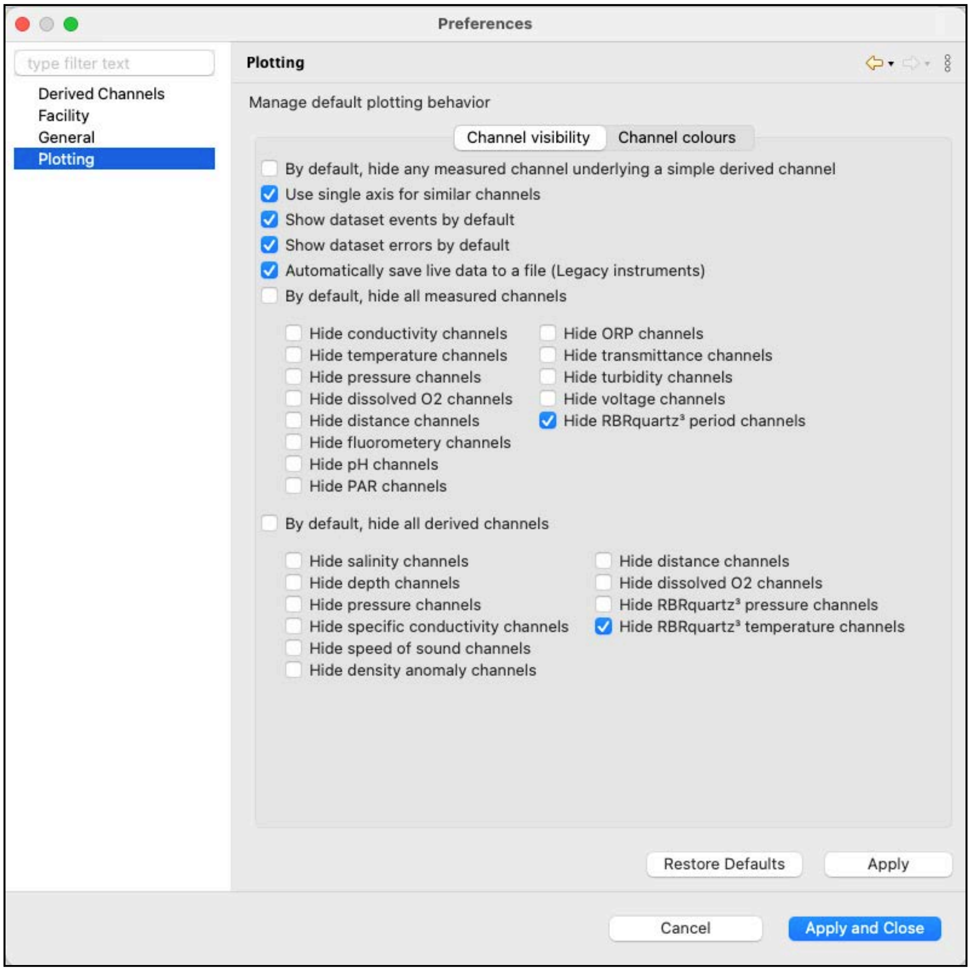
10.2 Plotting

Adjust default settings for the graphical display in the **Plot** view. Manage selections in **Channel visibility** and **Channel colours**, and apply them globally.

✓ You can click **Apply** to save your changes without closing the dialogue box.

Channel visibility

The **Channel visibility** tab displays many options to hide or show specific information.



Manage default plotting behavior

☐ By default, hide any measured channel underlying a simple derived channel
☑ Use single axis for similar channels
☑ Show dataset events by default
☑ Show dataset errors by default
☑ Automatically save live data to a file (Legacy instruments)
☐ By default, hide all measured channels

☐ Hide conductivity channels
☐ Hide temperature channels
☐ Hide pressure channels
☐ Hide dissolved O2 channels
☐ Hide distance channels
☐ Hide fluorometery channels
☐ Hide pH channels
☐ Hide PAR channels
☐ Hide ORP channels
☐ Hide transmittance channels
☐ Hide turbidity channels
☐ Hide voltage channels
☑ Hide RBRquartz^3 period channels

☐ By default, hide all derived channels

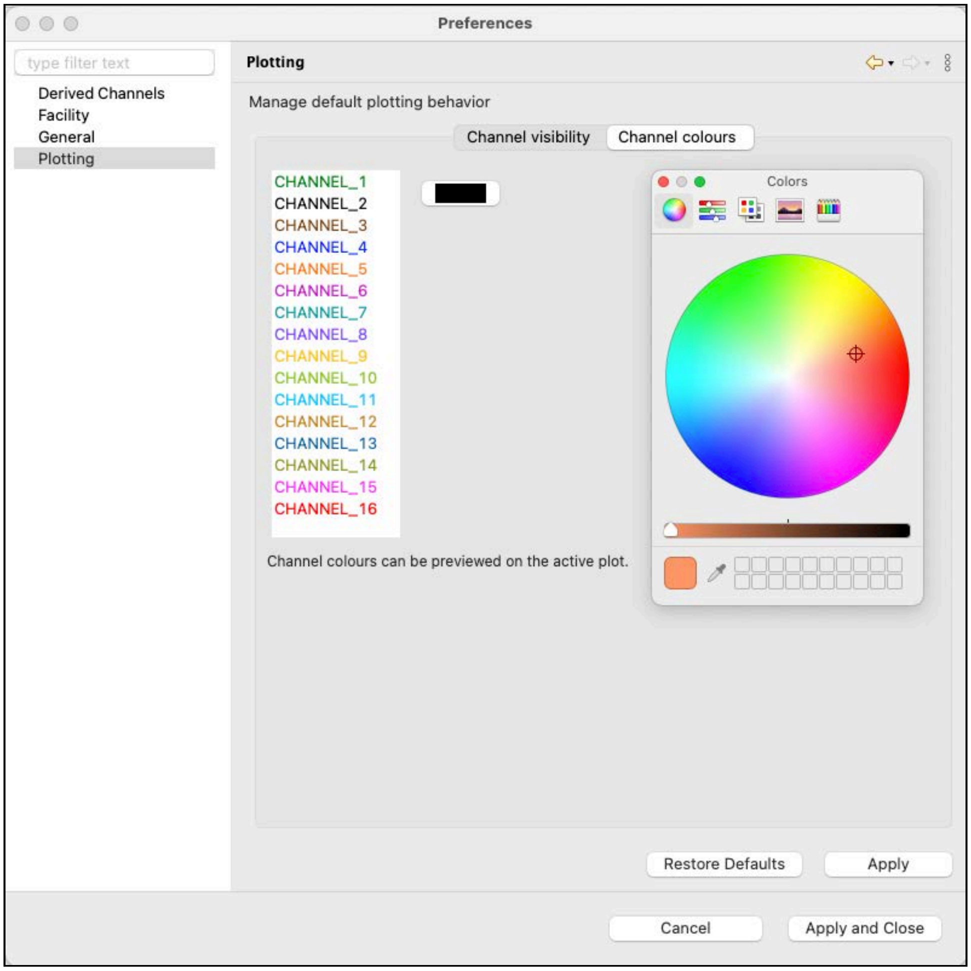
☐ Hide salinity channels
☐ Hide depth channels
☐ Hide pressure channels
☐ Hide specific conductivity channels
☐ Hide speed of sound channels
☐ Hide density anomaly channels
☐ Hide distance channels
☐ Hide dissolved O2 channels
☐ Hide RBRquartz^3 pressure channels
☑ Hide RBRquartz^3 temperature channels

Opt for displaying errors and events for a more informative graph, or hide them for a cleaner view.
Select the **By default, hide all measured channels** checkbox to display only derived channels.
Select the **By default, hide all derived channels** checkbox to display only measured channels.
Otherwise, pick and choose channels to display or hide.
When done, click **Apply** or **Apply and close**. The changes will take effect immediately.

⚠ You can override visibility preferences in the **Plot** view for any particular graphical display.

Channel colours

The **Channel colours** tab allows you to choose a different colour for each channel that is shown for information currently running in **Plot** view, or from an opened dataset. Select from the list of pre-defined options or create your own colour.



Manage default plotting behavior

CHANNEL_1
CHANNEL_2
CHANNEL_3
CHANNEL_4
CHANNEL_5
CHANNEL_6
CHANNEL_7
CHANNEL_8
CHANNEL_9
CHANNEL_10
CHANNEL_11
CHANNEL_12
CHANNEL_13
CHANNEL_14
CHANNEL_15
CHANNEL_16

Channel colours can be previewed on the active plot.

When done, click **Apply** or **Apply and close**. The changes will take effect immediately.

11 RSK files

The RSK file is a single file database based on SQLite that allows us to have large files with high-speed access to any part of the dataset. RSK files have different types and versions, which reflect historical changes to the underlying dataset structure. To upgrade the version of an older RSK file, go to **File info** tab.

RBR provides open-source RSKtools toolboxes for both Matlab (RSKtools) and Python (pyRSKtools) to read, post-process, visualise, and export RSK files.

RSK files can be exported to Excel, OceanDataView®, or text files.

RSK files contain raw measurement data, firmware and software versions, dates and times, sampling settings, power settings, measured and derived parameters, errors and events, other diagnostic information.

- Datasets in the **Desktop** storage format allow changes to calibration coefficients. GPS mapping for full RSK files relies on manually added location points.
  This file format is available for RBR standard and compact instruments.
- Datasets in the **Mobile** storage format support charting and GPS tracking. They do not allow changes to calibration coefficients.
  This file format is only available for RBR standard instruments.
- For datasets obtained via streaming or fetching, the RSK format is **Realtime**. Their names contain the "realtime" label.

  This file format is available for RBR standard instruments and RBR*coda*^{3} when in streaming mode, and for all instruments when fetching.

To identify the type of the RSK file, go to the File info tab.

File naming convention for realtime datasets

By default, realtime dataset names are composed of the following information:

- The instrument serial number: six digits
- The date of the download (YYYY-MM-DD): eight digits
- The time of the download (HH-MM in the 24-hour format): four digits
- The "realtime" label, indicating the data were streamed or fetched from the instrument

For example, the file named 209682_20220531_1408_realtime.rsk contains the dataset for a streaming instrument with a serial number of 209682; the data were saved on May 31, 2022 at 14:08 (02:08pm), local time.

The data streamed or fetched from your instrument are automatically saved to your computer. Go to **Preferences >** General to change your default location for realtime RSK files.

⚠ The file extension is .rsk; it indicates the Ruskin file format and must not be edited. If you change it, your new file extension will become part of the dataset name, and the .rsk extension will be appended.

12 Realtime data format

Sensors will begin streaming data as soon as power is provided. Below are example strings of data from each variety of the RBR sensors:

Instrument	Example data	Channels list
RBR*coda*^{3} T.ODO	`29000, 23.2868, 200.4000, 93.0000, 245.0000, 29.6900`	Timestamp (ms), Temperature (°C), concentration compensated for salinity (µmol/L), saturation (%), concentration uncompensated for salinity (µmol/L), phase (°)
RBR*coda*^{3} T	`29000, 23.2868`	Timestamp (ms), Temperature (°C)
RBR*coda*^{3} D	`29000, 10.2484`	Timestamp (ms), Pressure (dbar)
RBR*coda*^{3} DO	`29000, 98.8754`	Timestamp (ms), Saturation (%)
RBR*coda*^{3} T.D	`29000, 23.2868, 10.2484`	Timestamp (ms), Temperature (°C), Pressure (dbar)
RBR*coda*^{3} PAR, RBR*quadrante*	`1250, 38.6671142`	Timestamp (ms), PAR (µmol/m^2/s)
RBR*coda*^{3} rad, RBR*quadrante*	`1250, 22.0217241`	Timestamp (ms), Irradiance (µW/cm^2/nm)
RBR*tridente*	`3550, 2.6534132, 22.0217241, 1.9596633`	Timestamp (ms), backscatter (m^{-1}), chlorophyll *a* (µg/L), fDOM (ppb)
RBR*coda* Tu	29000, 35.6674542	Timestamp (ms), Turbidity (FTU)

The timestamp is the number of milliseconds since the time of the first sample, and so the first sample is always timestamped as zero. The timestamp will restart any time the CPU resets or if the following sample parameters are changed:

- Changing the sampling period
- Changing the sampling mode
- Changing the burst length or burst interval
- Turning a channel on/off (advanced users only)

The values following the timestamp are all the enabled channels. To identify which channels are enabled and their order in the string you can use the command **outputformat channelslist**.

Example

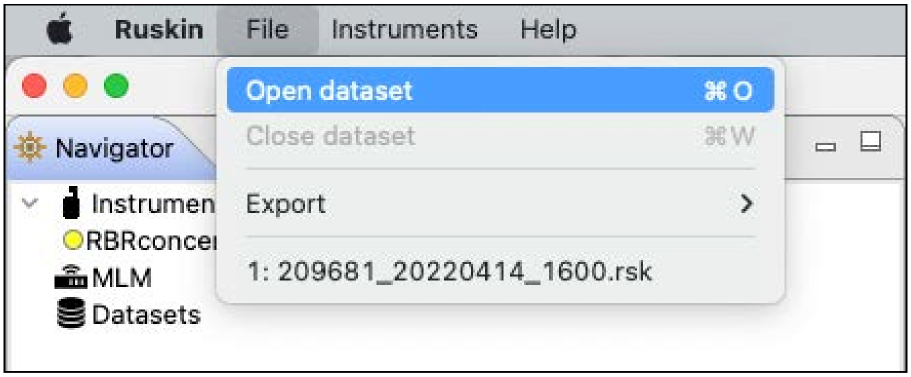
```
>> outputformat channelslist

<< outputformat channelslist = temperature (C), O2_concentration (umol/L), O2_air_saturation (%),
uncompensated_O2_concentration (umol/L)
```

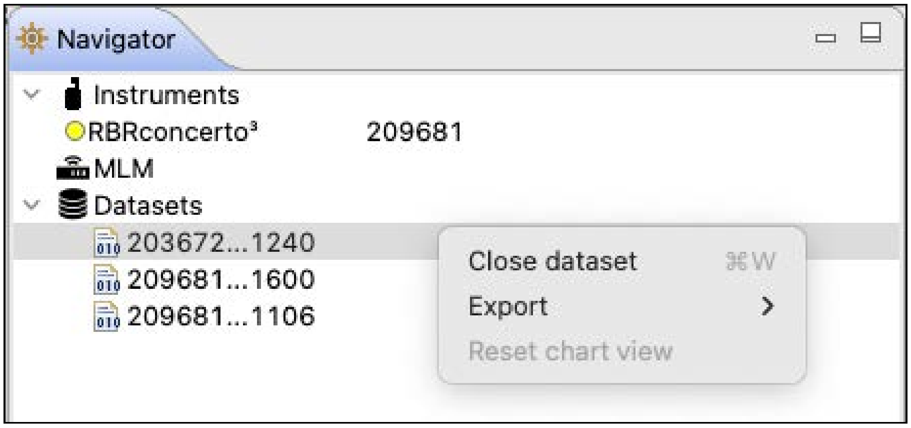
13 Datasets

Datasets are RSK files generated by Ruskin and have an .rsk extension. You can view and analyse your data using the options below.

Go to the **File** dropdown menu to open or close a dataset, export your data, and see the list of available datasets.

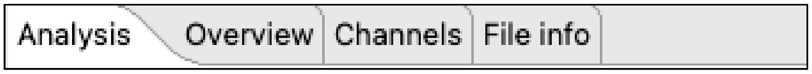


The **Datasets** context menu in the **Navigator** windows duplicates some of these options for convenience.



Explore the contents of your datasets in the **Properties** window.

When a dataset on the list is selected, the **Properties** window displays four tabs with information related to this dataset.



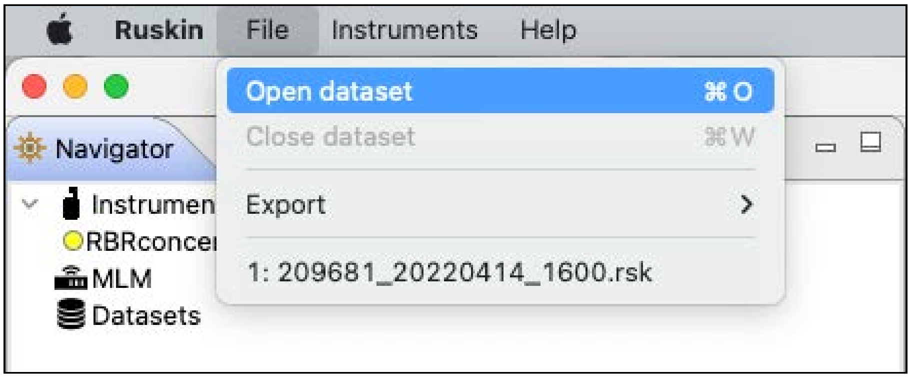
Use these tabs to analyse your data, check the calibration certificates, and verify your parameters.

13.1 Open dataset

Open any stored dataset from the **File** menu in the top toolbar or from the **Datasets** menu in the **Navigator** view.

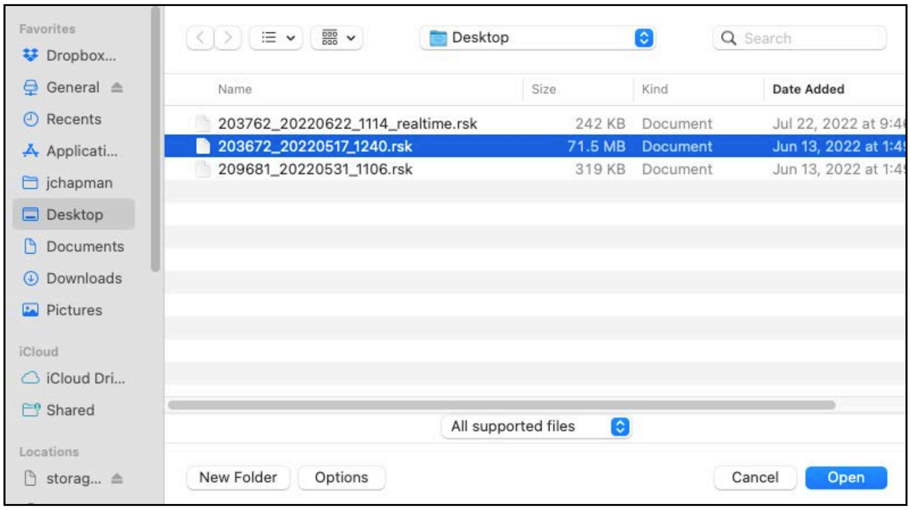
Steps

1. Click **File** > **Open dataset**.



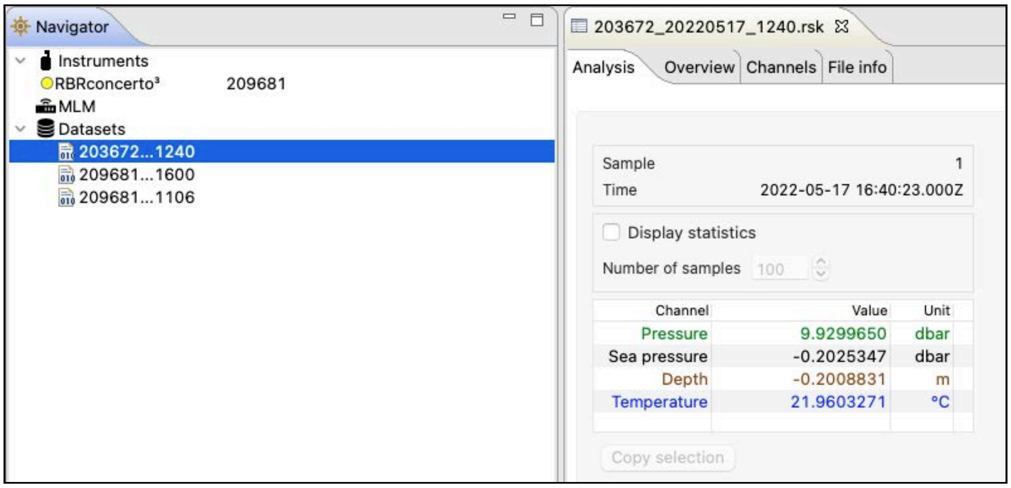
   Alternatively, right-click **Datasets** in the **Navigator** view and select **Open dataset**.

2. A browsing window will appear. Double-click the .rsk file you want to view, or select the file and click **Open**.



Name	Size	Kind	Date Added
203762_20220622_1114_realtime.rsk	242 KB	Document	Jul 22, 2022 at 9:4
203672_20220517_1240.rsk	71.5 MB	Document	Jun 13, 2022 at 1:4
209681_20220531_1106.rsk	319 KB	Document	Jun 13, 2022 at 1:4

3. The name of the dataset appears and is highlighted automatically in the **Navigator** view. The **Properties** view changes to the dataset, with four tabs: **Analysis**, **Overview**, **Channels**, and **File info**.



Navigator

- Instruments
  - RBRconcerto³ 209681
  - MLM
- Datasets
  - 203672...1240
  - 209681...1600
  - 209681...1106

203672_20220517_1240.rsk

Analysis | Overview | Channels | File info

Sample 1

Time 2022-05-17 16:40:23.000Z

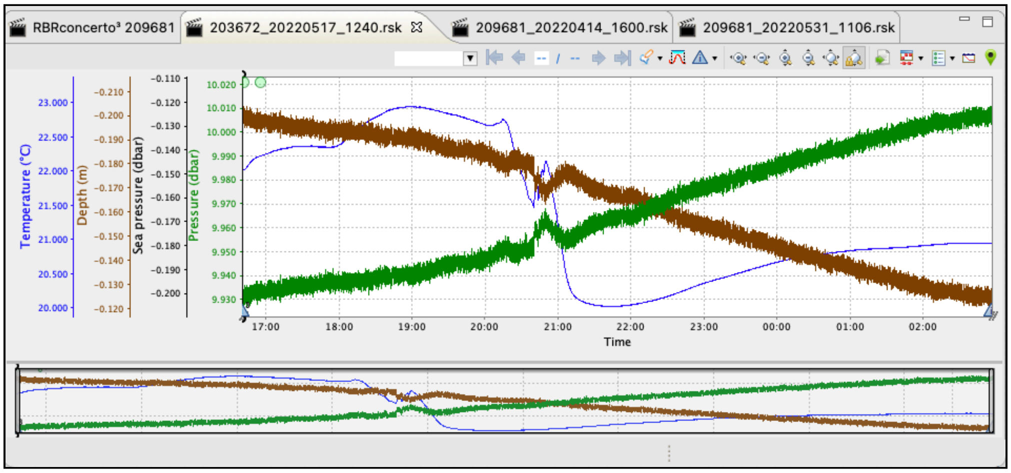
☐ Display statistics

Number of samples 100

Channel	Value	Unit
Pressure	9.9299650	dbar
Sea pressure	-0.2025347	dbar
Depth	-0.2008831	m
Temperature	21.9603271	°C

Copy selection

4. The data in the file will automatically appear in the **Plot** view.



RBRconcerto³ 209681 | 203672_20220517_1240.rsk | 209681_20220414_1600.rsk | 209681_20220531_1106.rsk

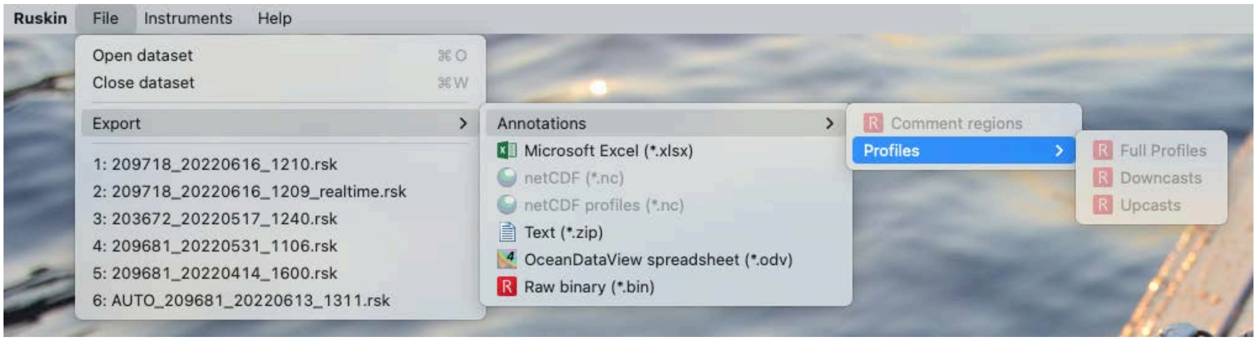
To close a dataset, go to **File** > **Close dataset**. Alternatively, right-click the dataset name in the **Navigator** view and select **Close dataset**.
Alternatively, simply close its **Plot** tab by clicking the **X.** The dataset will disappear from the list.

13.2 Export

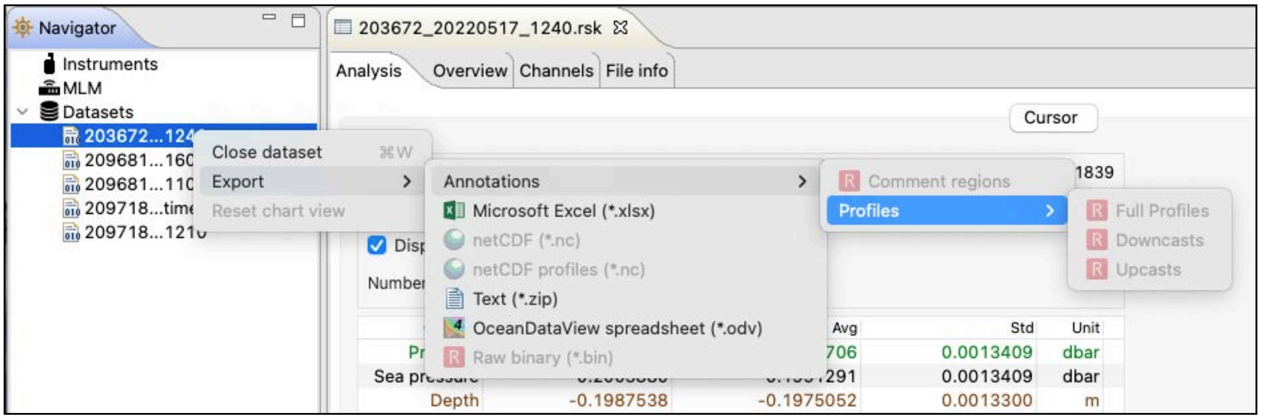
Use the **Export** option to share your data with others or analyse your data using other software. Datasets can be read directly in Matlab with our toolkit RSKtools, or exported to Excel, OceanDataView®, or text files.

Steps

1. **File** menu > **Export**, then hover to open a set of submenus.



or right-click the dataset in the **Navigator** window, then hover to open a set of submenus.



2. Select the appropriate export format (**Microsoft Excel**, **netCDF**/**netCDF profiles**, **Text**, **OceanDataView spreadsheet**, or **Raw binary**).

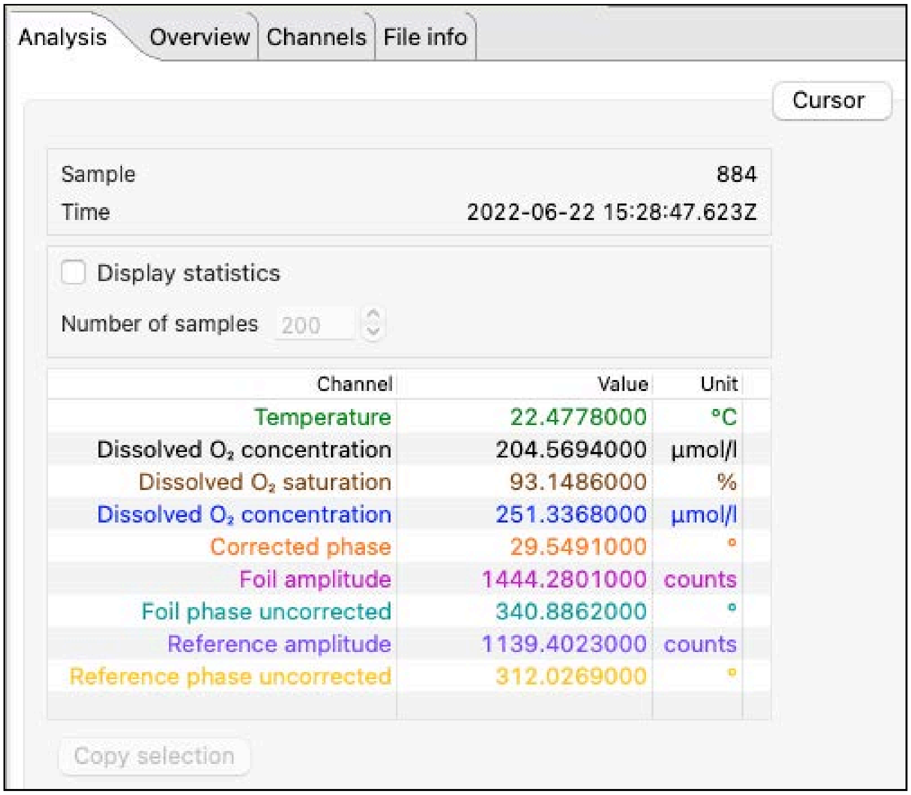
   ⚠ For realtime RSK files, the **Raw binary** export format is not available.

3. If exporting annotations, select the category: **Full profiles**, **Downcasts**, or **Upcasts**.
4. Specify a location and a name for the data file.
5. Click **Save**.

13.3 Analysis

Select a dataset in the **Navigator** view. The **Properties** view will change to dataset properties, with four tabs: **Analysis**, **Overview**, **Channels**, and **File Info**.

For most realtime datasets, the **Analysis** tab will only have the **Cursor** option. For Oxyguard DO sensors, it will also include **Calibration**.



Sample 884

Time 2022-06-22 15:28:47.623Z

□ Display statistics

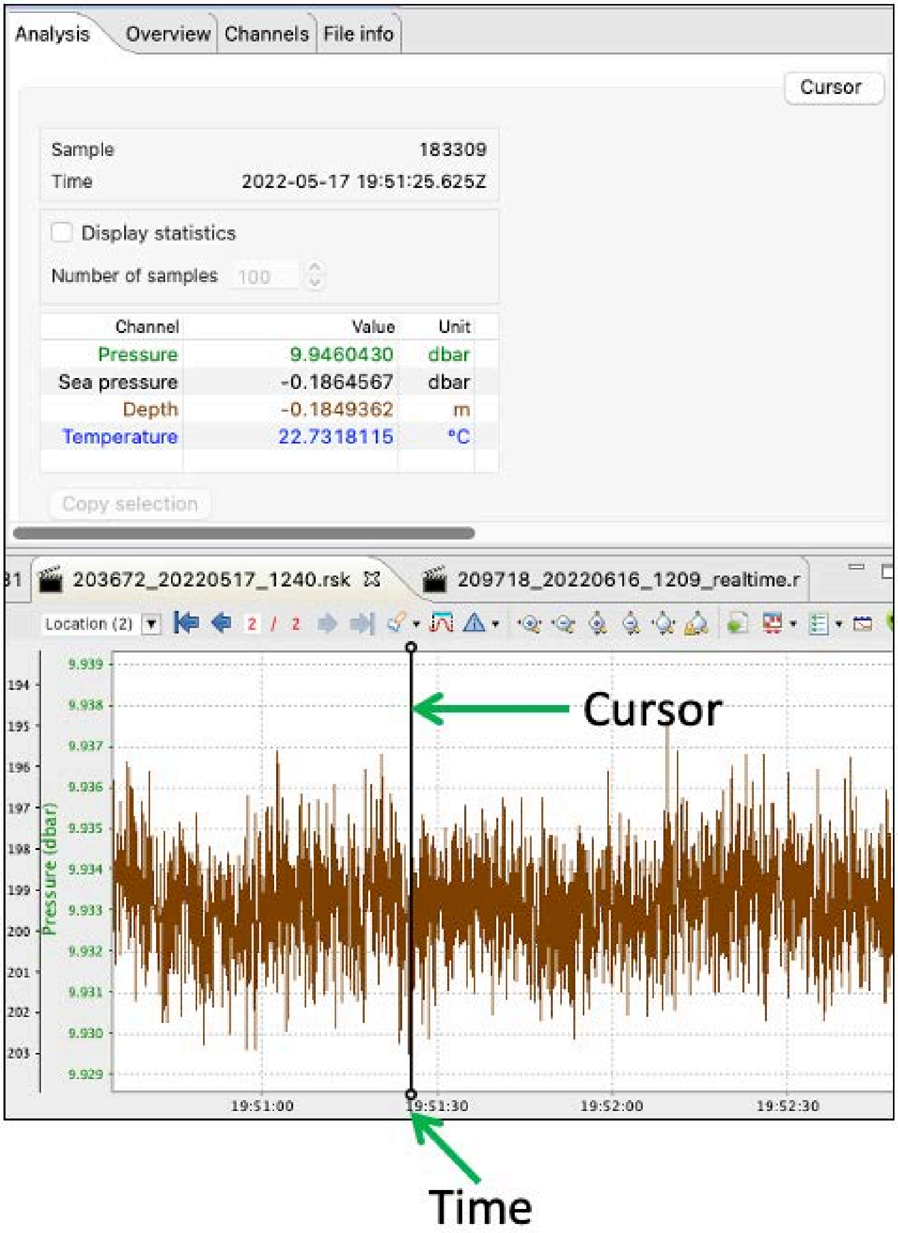
Number of samples 200

Channel	Value	Unit
Temperature	22.4778000	°C
Dissolved O_2 concentration	204.5694000	µmol/l
Dissolved O_2 saturation	93.1486000	%
Dissolved O_2 concentration	251.3368000	µmol/l
Corrected phase	29.5491000	°
Foil amplitude	1444.2801000	counts
Foil phase uncorrected	340.8862000	°
Reference amplitude	1139.4023000	counts
Reference phase uncorrected	312.0269000	°

The **Cursor** view contains statistical information about deployment, as well as information about the dataset channels and their values.

Move the black cursor ("double lollipop") in the **Plot** window to display parameter values for each sample selected.

Every time you move the black cursor, going form sample to sample, the table above the plot will display the information for this particular sample.



Sample	183309
Time	2022-05-17 19:51:25.625Z

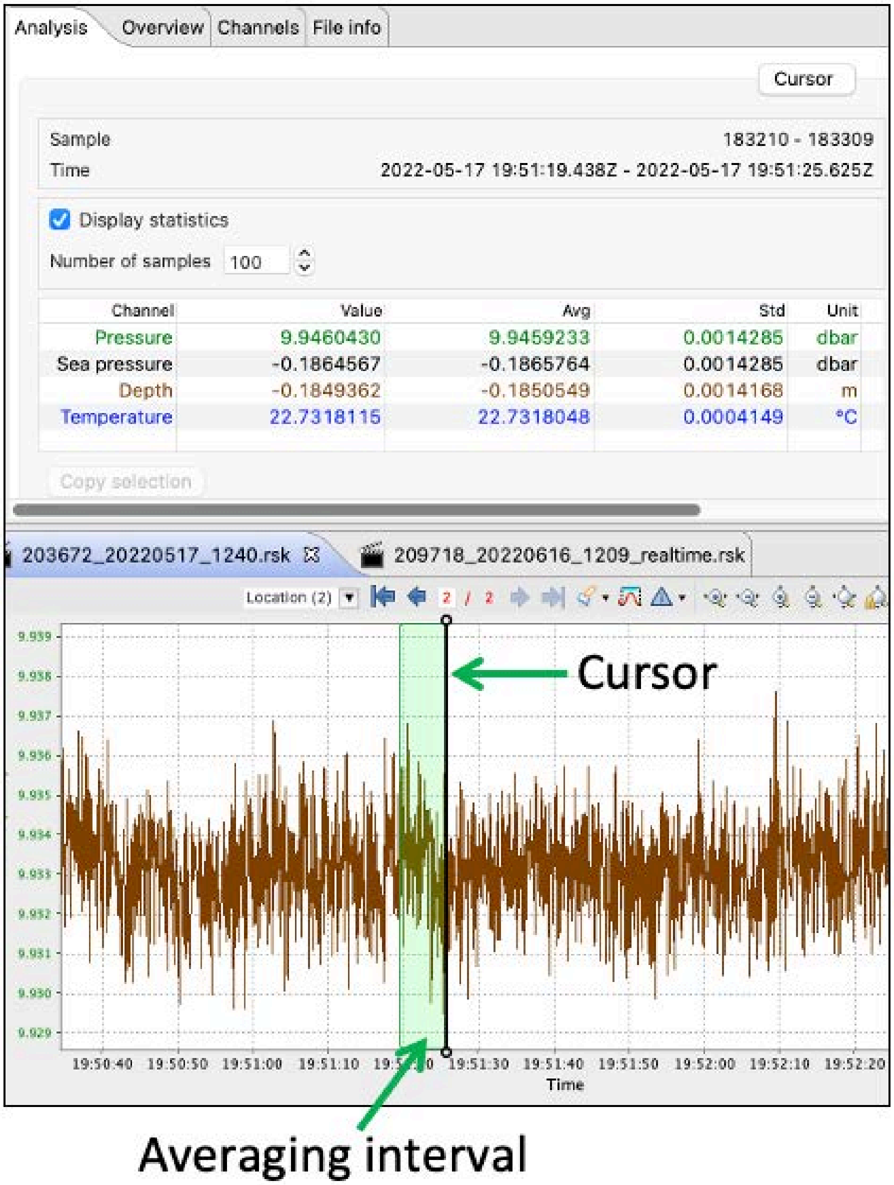
□ Display statistics

Number of samples 100

Channel	Value	Unit
Pressure	9.9460430	dbar
Sea pressure	-0.1864567	dbar
Depth	-0.1849362	m
Temperature	22.7318115	°C

Select **Display statistics** checkbox to determine the average value and standard deviation over a range of samples. Change the number of samples to average using the up and down arrows.

The table will display the averaging data. The averaging interval will be highlighted in the plot.



Sample	183210 - 183309
Time	2022-05-17 19:51:19.438Z - 2022-05-17 19:51:25.625Z

☑ Display statistics

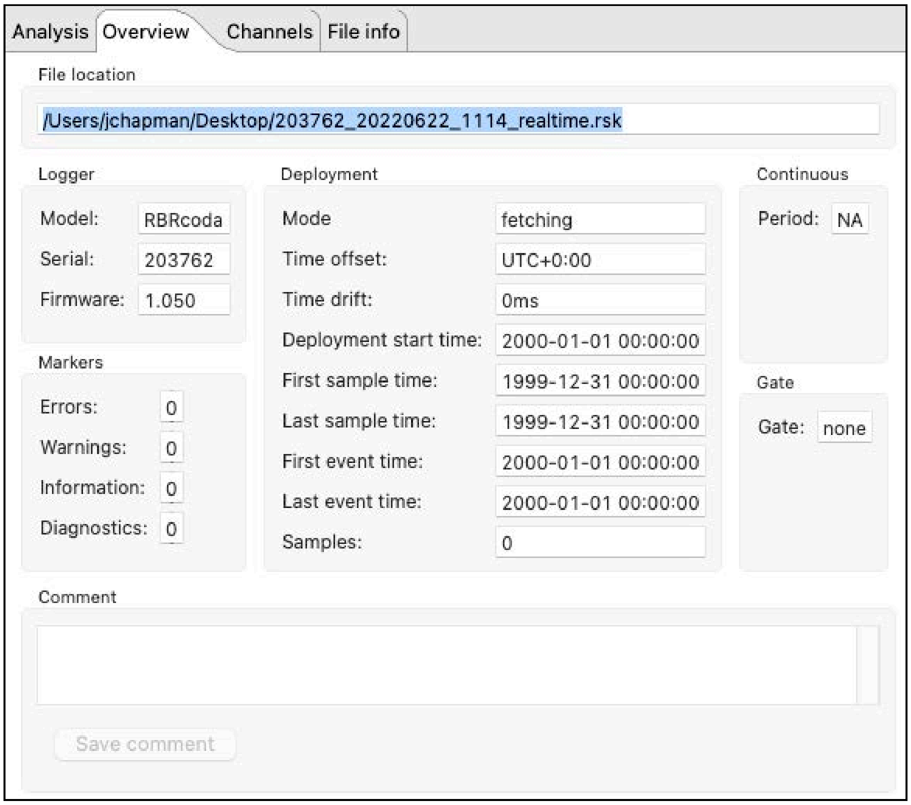
Number of samples 100

Channel	Value	Avg	Std	Unit
Pressure	9.9460430	9.9459233	0.0014285	dbar
Sea pressure	-0.1864567	-0.1865764	0.0014285	dbar
Depth	-0.1849362	-0.1850549	0.0014168	m
Temperature	22.7318115	22.7318048	0.0004149	°C

When the **Calibration** view is available, you can recalculate your coefficients for the DO channel. See User calibration of DO sensors.

13.4 Overview

Use the **Overview** tab to view general information for each dataset, such as instrument details and deployment events.



File location

/Users/jchapman/Desktop/203762_20220622_1114_realtime.rsk

Logger

Model: RBRcoda
Serial: 203762
Firmware: 1.050

Markers

Errors: 0
Warnings: 0
Information: 0
Diagnostics: 0

Deployment

Mode: fetching
Time offset: UTC+0:00
Time drift: 0ms
Deployment start time: 2000-01-01 00:00:00
First sample time: 1999-12-31 00:00:00
Last sample time: 1999-12-31 00:00:00
First event time: 2000-01-01 00:00:00
Last event time: 2000-01-01 00:00:00
Samples: 0

Continuous

Period: NA

Gate

Gate: none

Comment

To make a note about this dataset, enter your comment in the field provided and click **Save comment**.

13.5 Channels

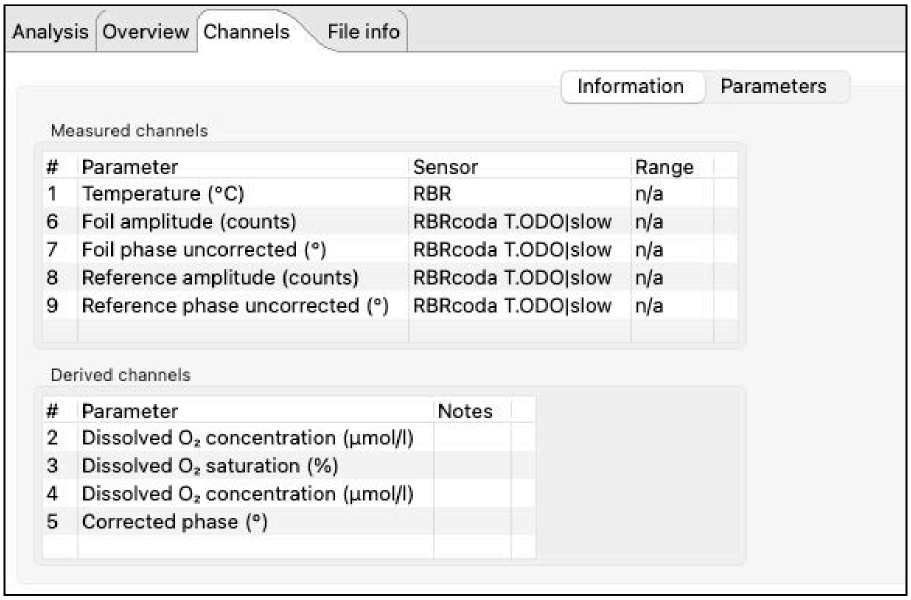
The **Channels** tab has two options: **Information** and **Parameters**.

Information

The **Information** page displays two tables: **Measured channels** and **Derived channels**.

Measured channels include measured parameters with their units of measurement, the sensor manufacturer, and the range setting (if applicable).

Derived channels include all available derived parameters with their units of measurement and the methodology used to calculate them.



Measured channels

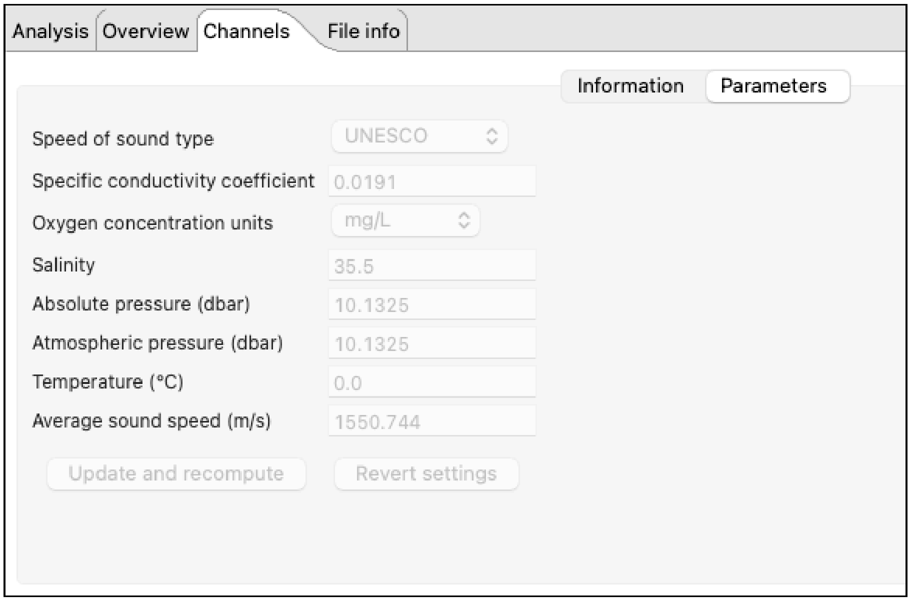
#	Parameter	Sensor	Range
1	Temperature (°C)	RBR	n/a
6	Foil amplitude (counts)	RBRcoda T.ODO|slow	n/a
7	Foil phase uncorrected (°)	RBRcoda T.ODO|slow	n/a
8	Reference amplitude (counts)	RBRcoda T.ODO|slow	n/a
9	Reference phase uncorrected (°)	RBRcoda T.ODO|slow	n/a

Derived channels

#	Parameter	Notes
2	Dissolved O_2 concentration (µmol/l)	
3	Dissolved O_2 saturation (%)	
4	Dissolved O_2 concentration (µmol/l)	
5	Corrected phase (°)	

Parameters

The **Parameters** tab displays parameters necessary to calculate derived channels and the method used for calculation. This page is not editable for realtime datasets.



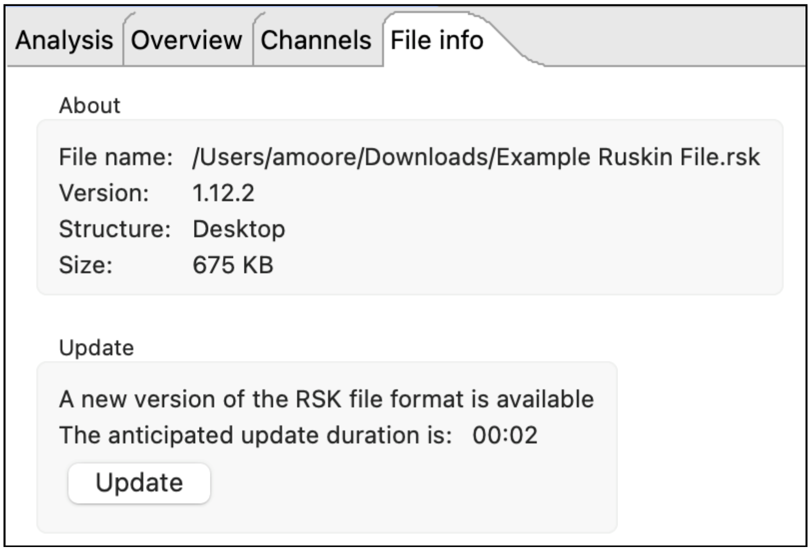
Speed of sound type	UNESCO
Specific conductivity coefficient	0.0191
Oxygen concentration units	mg/L
Salinity	35.5
Absolute pressure (dbar)	10.1325
Atmospheric pressure (dbar)	10.1325
Temperature (°C)	0.0
Average sound speed (m/s)	1550.744

13.6 File info

The **File info** page displays the dataset name and download location, Ruskin version used when it was obtained, file structure, and size. The **Structure** label reflects the RSK file format, which can be **Desktop**, **Mobile**, or **Realtime**.

⚠ The RBR smart streaming sensors generate RSK files in the **Realtime** storage format only.

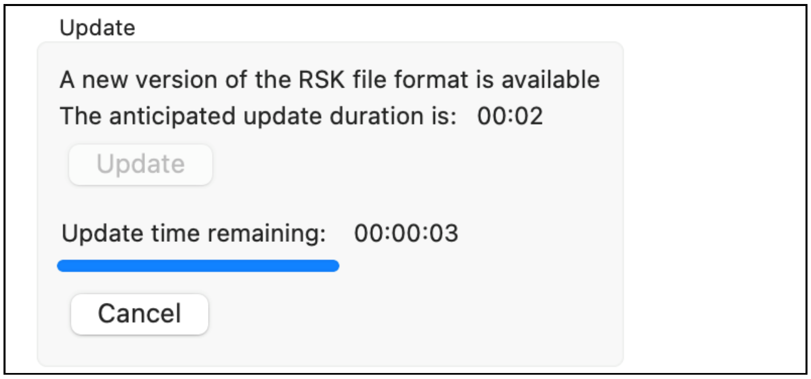
Ruskin checks the dataset for compatibility with the newest version of Ruskin software. If the file requires updating, a warning message and an **Update** button will be displayed.



File name: /Users/amoore/Downloads/Example Ruskin File.rsk
Version: 1.12.2
Structure: Desktop
Size: 675 KB

A new version of the RSK file format is available
The anticipated update duration is: 00:02

Click **Update**. It may take several minutes for it to complete. RBR strongly recommends that all datasets are updated and meet compatibility requirements.



A new version of the RSK file format is available
The anticipated update duration is: 00:02

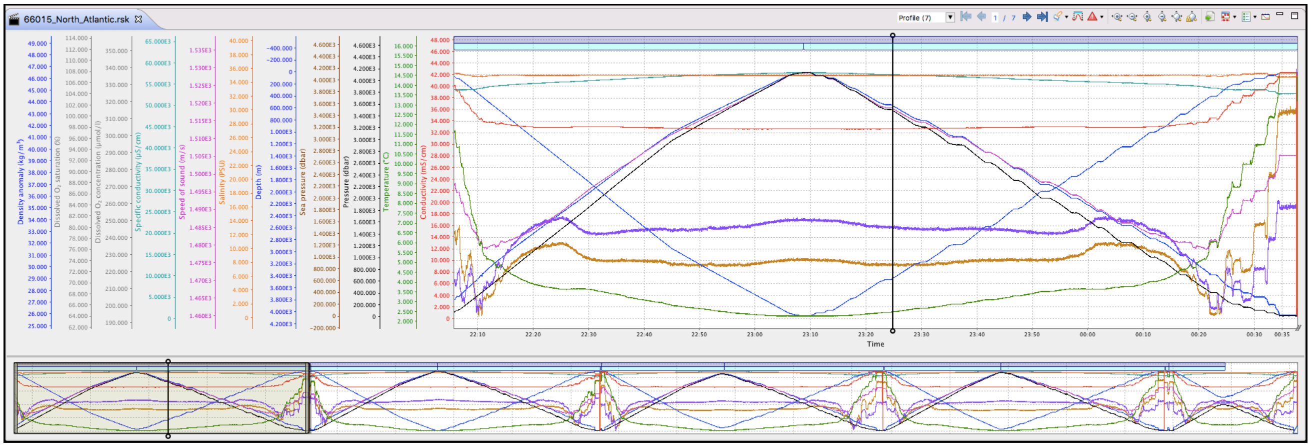
Update time remaining: 00:00:03

❗ Failing to update your dataset may result in odd graphical displays and incorrect reporting of the sample values.

14 Plot window

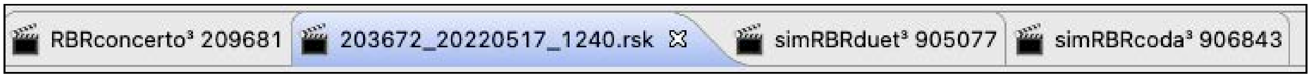
Use the **Plot** window to view graphical displays of your datasets or live data. When viewing live data, the instrument must be attached.

Each channel appears as a different colour in the plot. These colours are specified under **Preferences** > Plotting and can be changed only there.



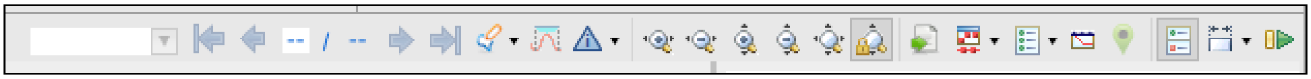
Move the black "double lollipop" cursor from one sample to the next to display parameter values for each sample selected. For the datasets, these values will appear in the **Properties** window above (see Analysis). For live data, they will be on the side of the plot.

The graphical display for each dataset or instrument will be in its own tab. The dataset file name or the instrument model and serial number will appear at the top of the tab. Simulated instruments will have "sim" in front of their name.

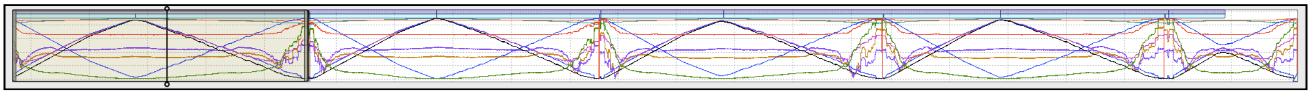


RBRconcerto³ 209681 | 203672_20220517_1240.rsk ☒ | simRBRduet³ 905077 | simRBRcoda³ 906843

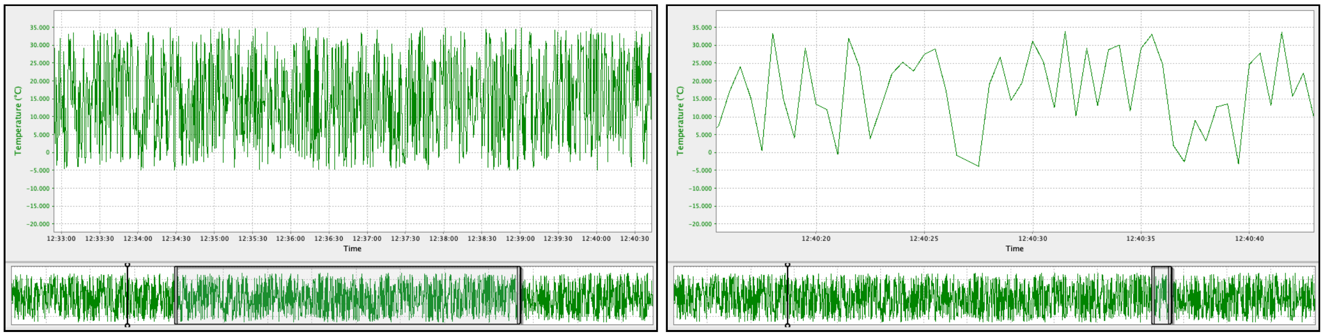
A toolbar at the top right of the **Plot** window contains various buttons to help you customise, explore, and export the graphical display. The buttons selection changes depending on what is displayed in the tab. Hover over a button to display its name and function.



A downsampled representation of the dataset appears in the thumbnail view, along with an indicator of the current plotted time range.



Position the indicator by clicking on the thumbnail view and dragging the indicator. Resize the indicator by dragging the edges left and right. It will zoom the main plot along the time domain to match the new indicator size.



⚠ The thumbnail view always shows the plot-by-time data, even if the plot is switched to the plot-by-depth mode.

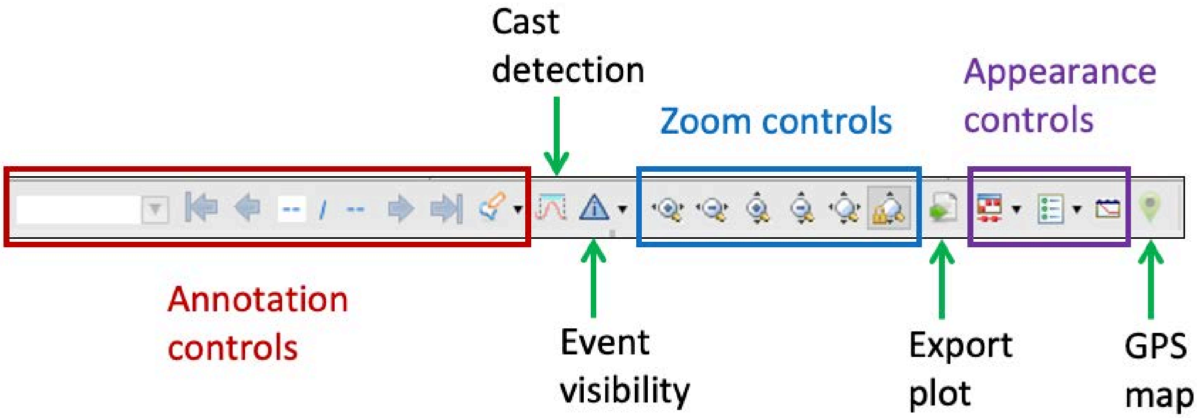
When you close and re-open a dataset, it will retain your latest settings (channel visibility, zoom, plot style (plot-by-time or plot-by-depth), annotation selection, and display style (lines or markers).

Double-click anywhere on the main plot to return to the full-chart view.

Go to Appearance controls to change your selection of lines and markers.

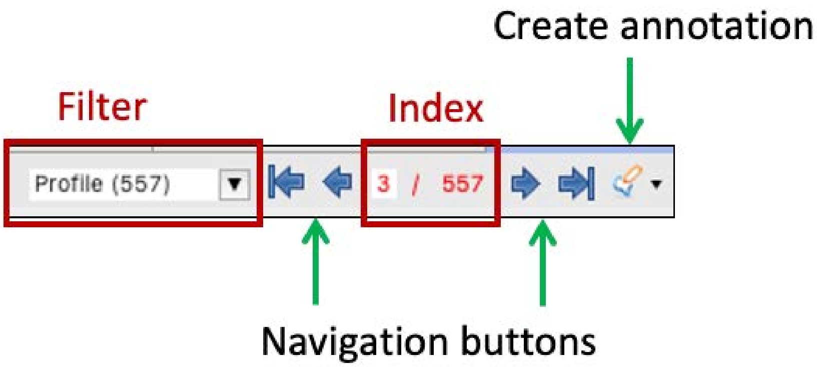
14.1 Dataset plot toolbar

Use the plot toolbar to adjust the display of an open dataset.



14.1.1 Annotation controls

Annotations are information messages added either by users or by the instrument.

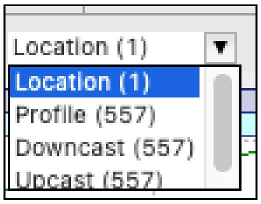


Annotation types

Annotation types are divided in two categories: profiling (**Profile**, **Downcast**, or **Upcast**) or non-profiling (**Comment**, **Exclusion**, **Location**, or **Calibration**). Use annotation controls to manage this information and to navigate the plot.

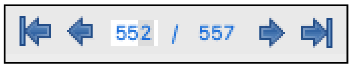
Type	Function
Profile, Downcast, Upcast	Identify profiling
Comment	Add a note to your data
Exclusion	Remove a non-representative interval from analysis
Location	Add in latitude/longitude coordinates for RSK files to track locations of data logging
Calibration	Calibrate the ODO sensor (available only for RBR standard instruments using the RBR*coda*^{3} T.ODO as an integrated sensor)

Open the **Filter** dropdown menu to select the annotation type to display. The number of annotations for each type will be indicated.



Select an annotation type. The plot will centre on the first item in the selected category. For example, when you select **Profile**, the plot will move to the first profile.

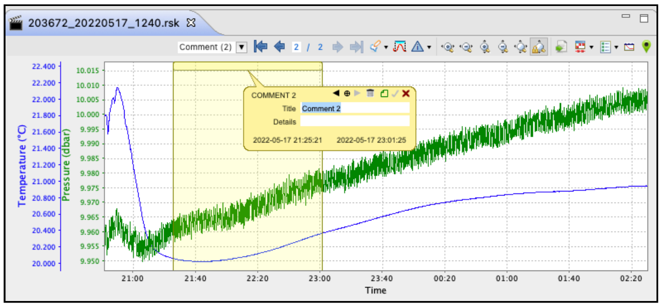
If you switch the filter to another annotation type within the same category (for example, **Profile** to **Upcast**), the plot will not change. If you switch to a type from another category (for example, from **Profile** to **Comment**), the plot will move to the first annotation in the new category (for example, to the first comment).

Use the **Navigation** buttons  to navigate between items within the selected category.

The navigation buttons will be activated if there is more than one item in the selected annotation type. Otherwise, the plot will centre on the only item available.

Navigation controls

- Move from one annotation to another by clicking the **left** and **right** arrows.
- Jump all the way to the beginning or end of the list by using the **end-left** and **end-right** arrows.
- Alternatively, manually enter the number in the **Index** field to go to that particular annotation.
- Double-click an annotation to centre and zoom in the plot at that particular area.

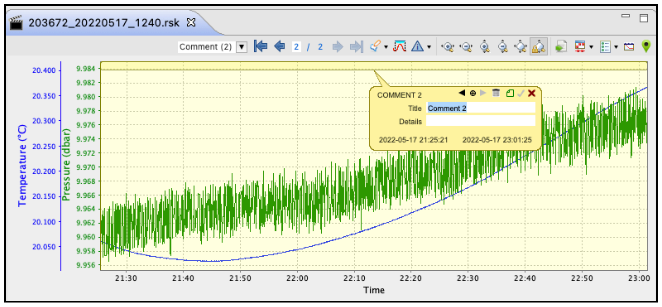


COMMENT 2

Title: Comment 2

Details

2022-05-17 21:25:21 2022-05-17 23:01:25



COMMENT 2

Title: Comment 2

Details

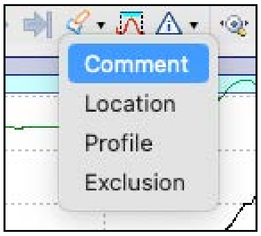
2022-05-17 21:25:21 2022-05-17 23:01:25

Double-click anywhere on the main plot to return to the full-chart view.

Click the **Create annotation** button to enter a new comment, location, profile, or exclusion.

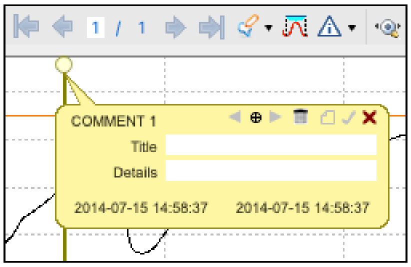
Steps to create an annotation

1. Select the type of annotation to create, for example, **Comment**.



The cursor in the **Plot** window will become a two-way arrow ✥ .

2. Click on the main plot to identify a single point to comment, or click and drag the cursor left or right to select an interval. When you release the click, an editable tooltip will open.



COMMENT 1

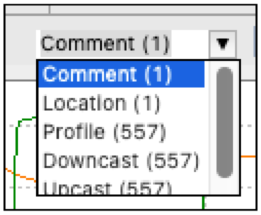
Title

Details

2014-07-15 14:58:37 2014-07-15 14:58:37

The tooltip controls include left and right **arrows** ◀ ⊕ ▶, **Delete** 🗑, **Save** ✓, and **Close** ✗.

3. Enter the necessary information into the fields and save by pressing **Enter** on your keyboard or by clicking **Save** ✓ on the tooltip. Your new comment will appear in the dropdown list.



Comment (1)

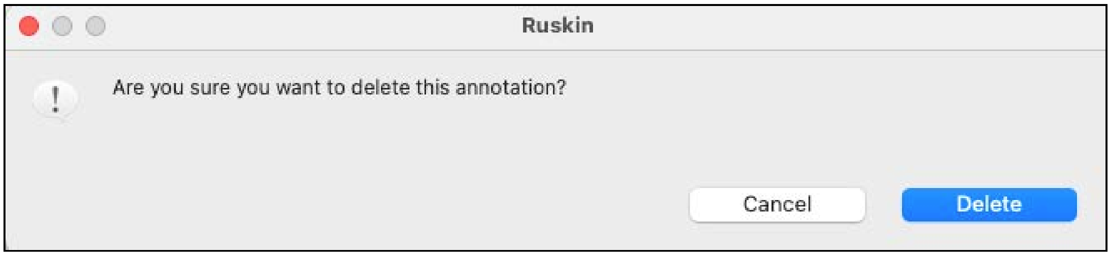
Comment (1)
Location (1)
Profile (557)
Downcast (557)
Upcast (557)

Clicking **Close** ✗ or pressing **Esc** on the keyboard will close the tooltip without saving.

⚠ Edit at least one detail in the tooltip before saving. Otherwise, the annotation will be deleted when deselected. Moving or resizing the annotation counts as editing and saving.

4. Click the **arrows** ◀ ⊕ ▶ on the tooltip to move to the next annotation within the same category.

5. Click **Delete** 🗑 on the tooltip to delete the annotation.
A warning message will appear.



Ruskin

Are you sure you want to delete this annotation?

Click **Delete** to proceed. This action cannot be undone.

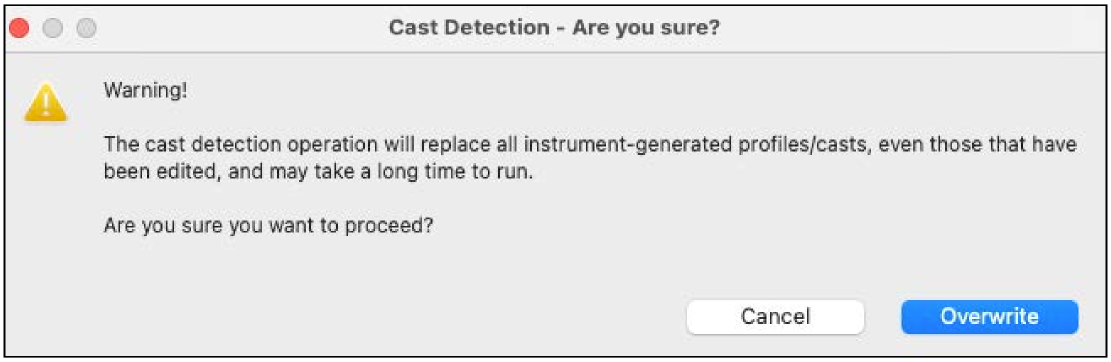
14.1.2 Cast detection

Ruskin automatically enables profile detection on instruments when a deployment is enabled. There are several criteria for enabling this feature.

- The instrument must support the `castdetection` setting
- The instrument must have a pressure sensor
- The deployment must be configured for | fast sampling (2Hz and up)
- The deployment mode must be configured as either continuous or directional

It is possible to recalculate the casts in a dataset using the cast detection feature in the **Plot** view.

Click the **Cast detection** button to run a cast detection algorithm based on depth and conductivity to automatically generate profiles and casts. A warning message will appear.


Cast Detection - Are you sure?

Warning!

The cast detection operation will replace all instrument-generated profiles/casts, even those that have been edited, and may take a long time to run.

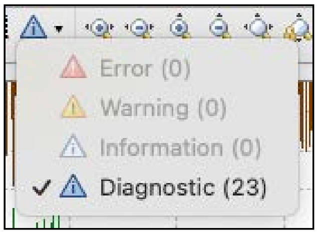
Are you sure you want to proceed?

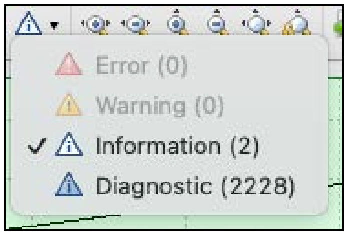
If you proceed, Ruskin will find peaks, collect events, and overwrite profiles recorded by the instrument.

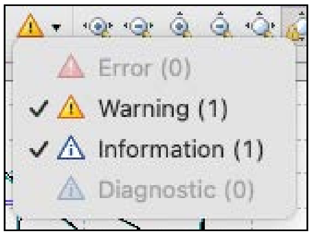
✓ Double-click anywhere on the main plot to return to the full-chart view.

14.1.3 Event visibility

Click the **Event visibility** button to turn on and off the visibility of the various levels of events: **Error**, **Warning**, **Information**, and **Diagnostic**. Its colour will reflect the highest level of event severity reported.







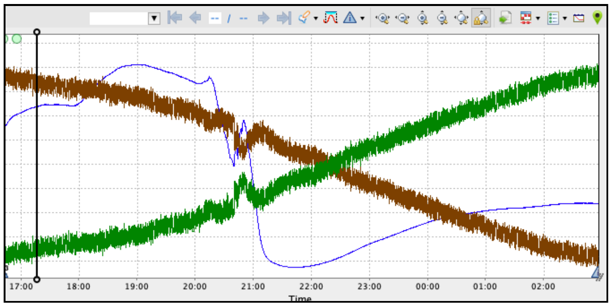
14.1.4 Zoom controls

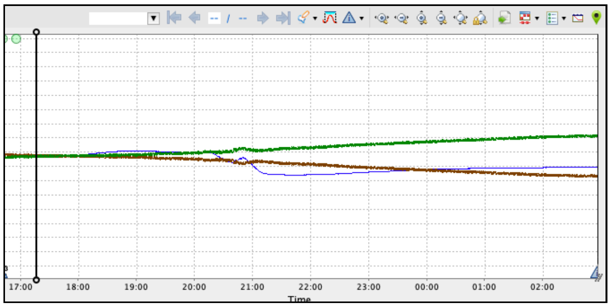
Default scaling of the plot is based on the minimum and maximum values sampled for each channel.

Use the **Zoom in horizontally** and **Zoom out horizontally** buttons to increase or decrease the the time range displayed.

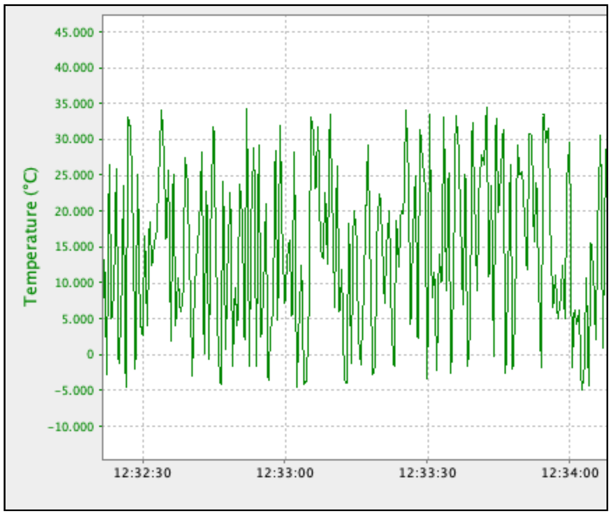
In the same way, use the **Zoom in vertically** and **Zoom out vertically** buttons to increase or decrease the resolution of the data displayed. Even more conveniently, hover the cursor and scroll your mouse wheel up and down to change the scale. Alternatively, hold **Shift** + **left click** and drag where you want to zoom in or out.

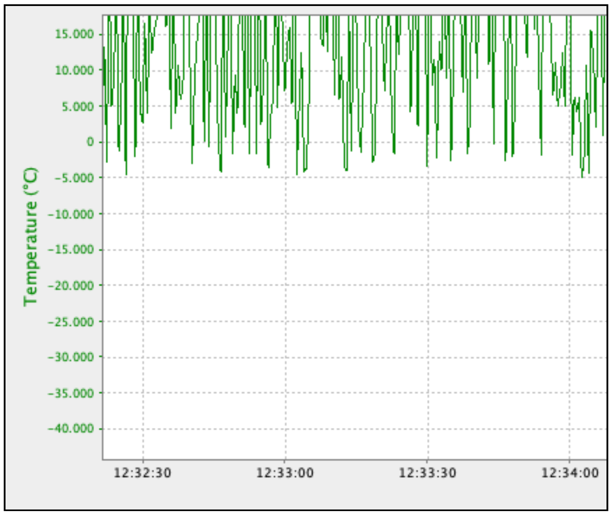
This function may be helpful when there are erroneously high or low values that make the real data appear incorrect.





Hold **Ctrl** + **left click** and move your mouse left and right to pan your chart horizontally.
Click a value on the vertical axis and move your mouse up and down to pan your chart vertically to centre as desired.





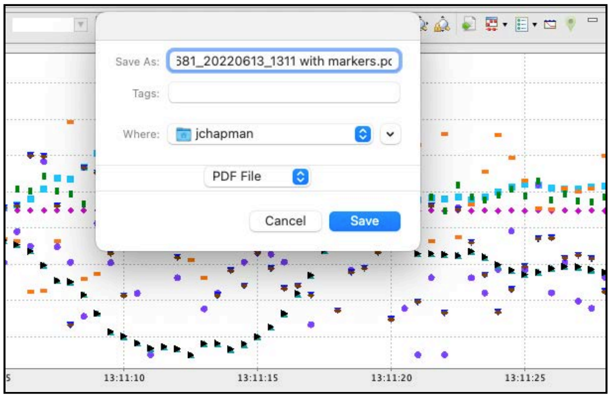
Click **Reset zoom to maximum bounds** or **double-click** anywhere on the chart to restore the original appearance.

Use the **Toggle auto-ranging** button to turn on and off the auto-ranging of the channel axes. Disabling the auto-ranging of the axis can be useful for looking at trends in data.

✓ Double-click anywhere on the main plot to return to the full-chart view.

14.1.5 Export plot

Click the **Export plot** button and select file name and location to save the current view of the main plot to a PDF or PNG file. The exported file will contain the graph only and retain all the changes to display active at the moment of export: zoom, lines and markers, visible events and annotations, and plotting by depth or time.



Save As: 381_20220613_1311 with markers.pc

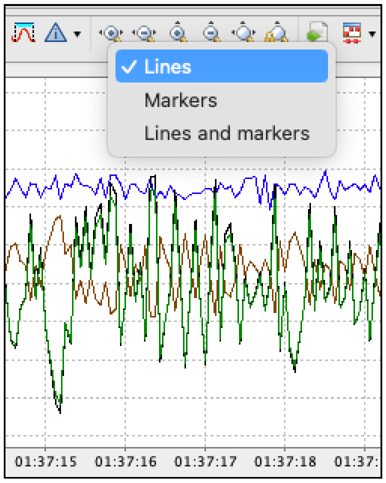
Tags:

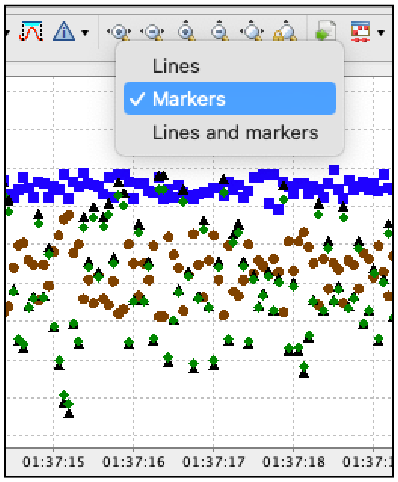
Where: jchapman

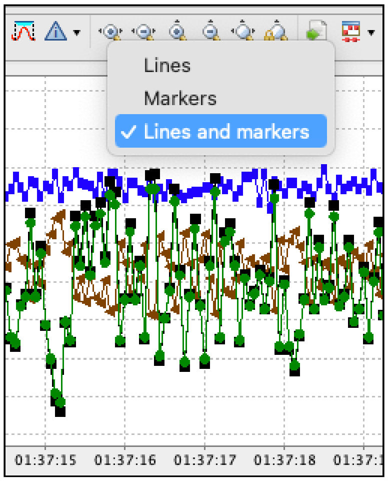
PDF File

14.1.6 Appearance controls

Select your data rendering type from the **Change line type** dropdown menu : **Lines**, **Markers**, or **Lines and markers**.

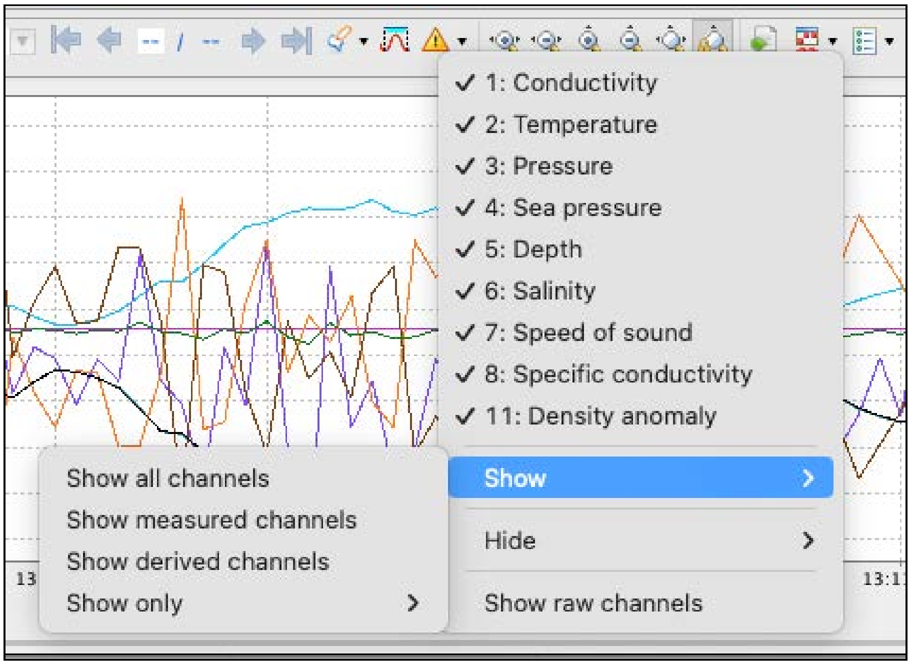




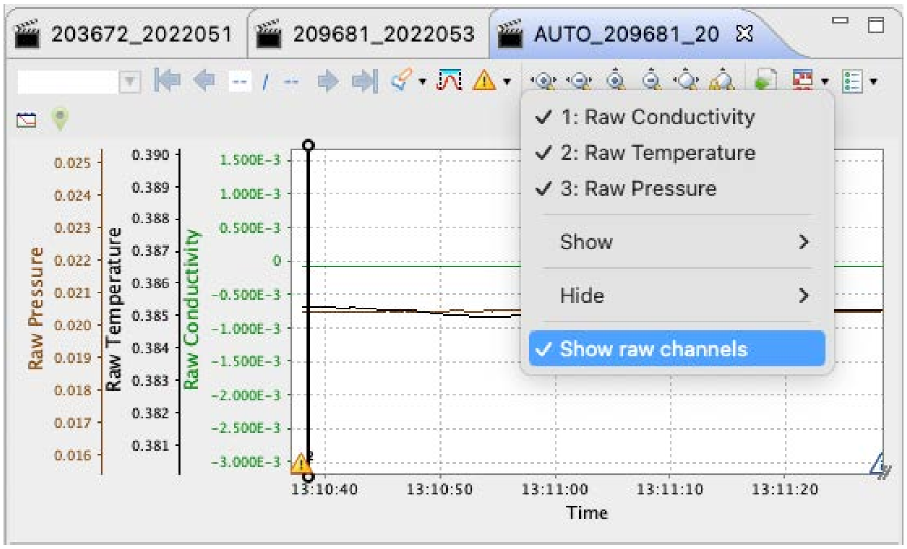


✓ If no data appears in the **Plot** view, try changing the rendering to **Markers**. Time spans that are larger than expected will cause breaks between data points and lines will not be drawn connecting those points, whereas markers draw at each individual point.

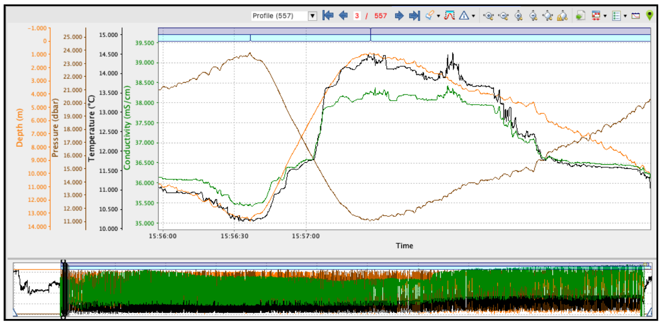
Use the **Toggle channel visibility** button to open the dropdown menu with the list of available channels and options. Check and uncheck channels to display as desired, or select groups to show and hide.

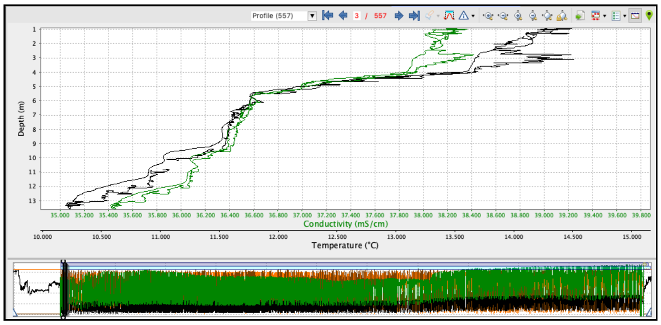


Select **Show raw channels** to display measured data with no calibration equations applied.



By default, the data are plotted by time. Use the **Toggle depth plot** button to switch to the plot-by-depth mode.



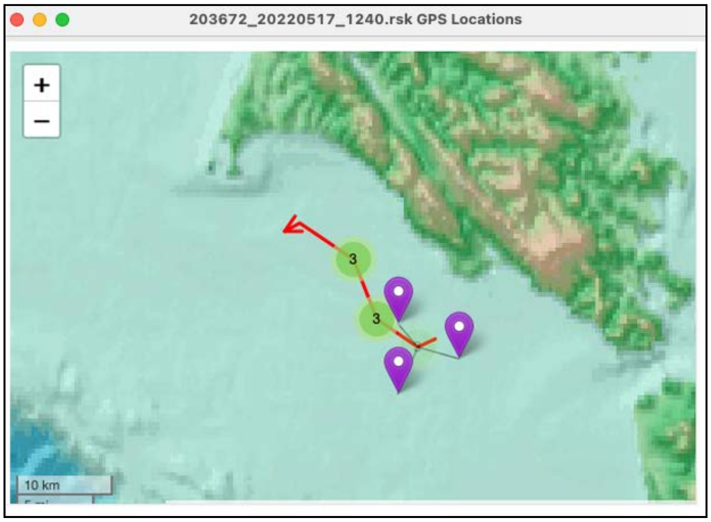


The thumbnail view will continue to display the plot-by-time data.

ⓘ If there is no depth channel, the plot will use the next pressure or pressure-derived channel starting from the bottom of the channel list.

14.1.7 GPS map

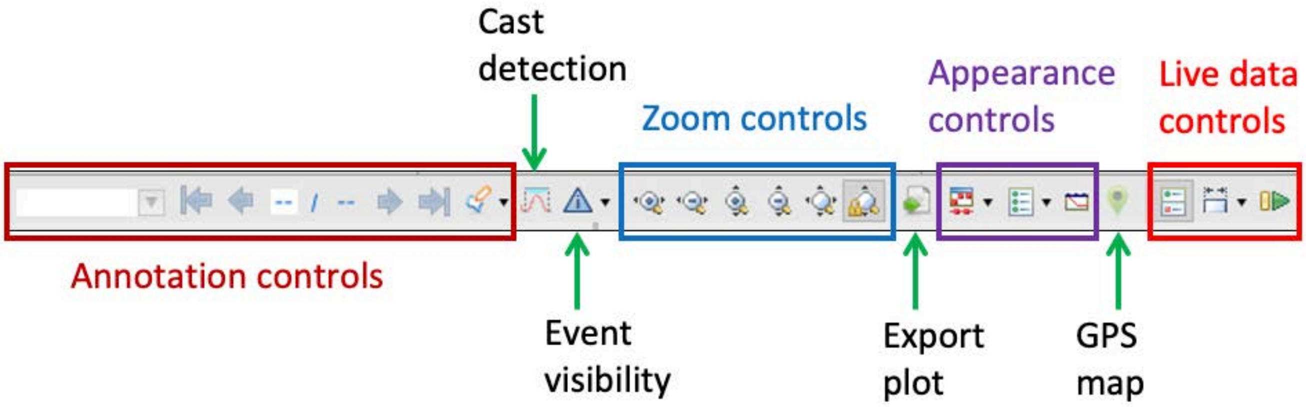
The **Show GPS map** button opens a pop-up window with the GPS map for the plot. Click on the map to display all the GPS location annotations recorded in the dataset.


203672_20220517_1240.rsk GPS Locations

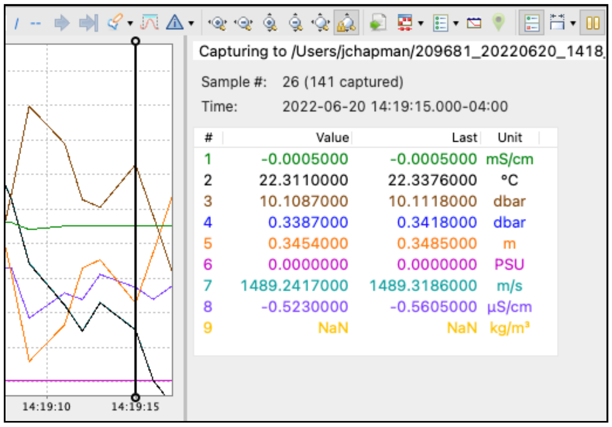
GPS mapping for **Desktop** RSK files relies on manually added location points (see **Annotation controls**). All location points you added using the **Annotations** feature are stored in the dataset file and can be displayed in Ruskin via the mapping functionality.

14.2 Live data plot toolbar

You can use the **Plot** window to view a live graphical display of your instrument sampling. All options described in the **Dataset plot toolbar** section apply to live data display. Additionally, three live data buttons appear at the right end of the toolbar: **Side data panel**, **Live plot time range**, and **Fetch**.



Click the **Side data panel** button to toggle on and off the data panel to the right of the plot. It shows the file location that will be used to capture samples, the number of samples collected, the values of the last received sample, and the currently selected sample.

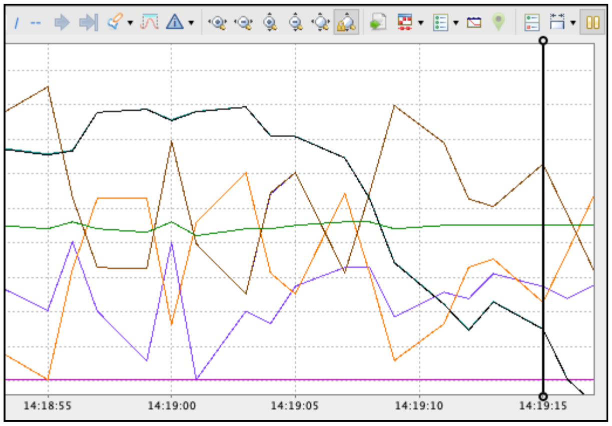


Capturing to /Users/jchapman/209681_20220620_1418

Sample #: 26 (141 captured)

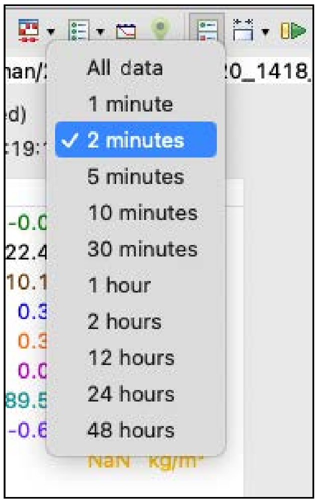
Time: 2022-06-20 14:19:15.000-04:00

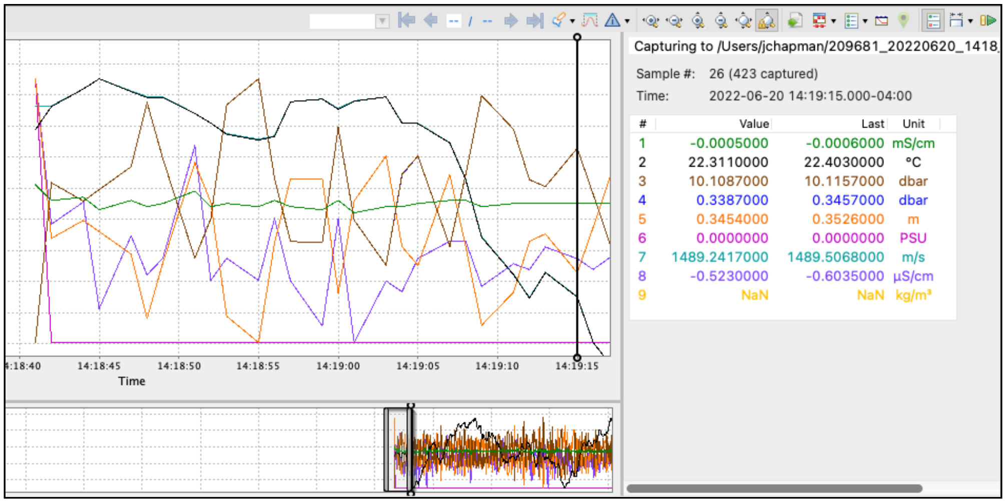
#	Value	Last	Unit
1	-0.0005000	-0.0005000	mS/cm
2	22.3110000	22.3376000	°C
3	10.1087000	10.1118000	dbar
4	0.3387000	0.3418000	dbar
5	0.3454000	0.3485000	m
6	0.0000000	0.0000000	PSU
7	1489.2417000	1489.3186000	m/s
8	-0.5230000	-0.5605000	µS/cm
9	NaN	NaN	kg/m³



Click the **Live plot time range** button to open a dropdown list of time span options.

For example, select **2 minutes** to display the data in the thumbnail view and on the side panel by two-minute time periods.





Capturing to /Users/jchapman/209681_20220620_1418

Sample #: 26 (423 captured)

Time: 2022-06-20 14:19:15.000-04:00

#	Value	Last	Unit
1	-0.0005000	-0.0006000	mS/cm
2	22.3110000	22.4030000	°C
3	10.1087000	10.1157000	dbar
4	0.3387000	0.3457000	dbar
5	0.3454000	0.3526000	m
6	0.0000000	0.0000000	PSU
7	1489.2417000	1489.5068000	m/s
8	-0.5230000	-0.6035000	µS/cm
9	NaN	NaN	kg/m³

You can also choose to display all of the data collected.

Fetching

Live data can come from one of two sources: streaming or fetching.

Fetching is the action of asking the instrument to take a reading and report it.

To start fetching, disable the streaming option. Go to Options under the **Configurations** tab and set **Realtime** to **None**.

Click the **Fetch** button to start fetching data. The button will change to **Pause**. Click the **Pause** button to stop fetching.

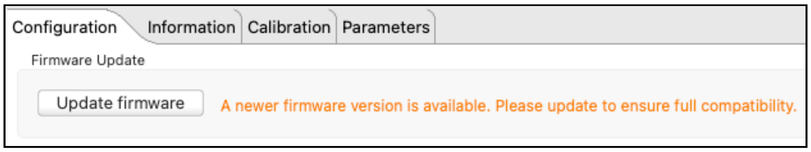
⚠ If streaming is selected (**Realtime** is set to **Serial**), clicking the **Fetch** button will have no effect on the plot.

15 Firmware updates

Ruskin automatically checks if your instrument firmware is up-to-date. Updates may be critical, routine, or customised.

15.1 Critical firmware updates

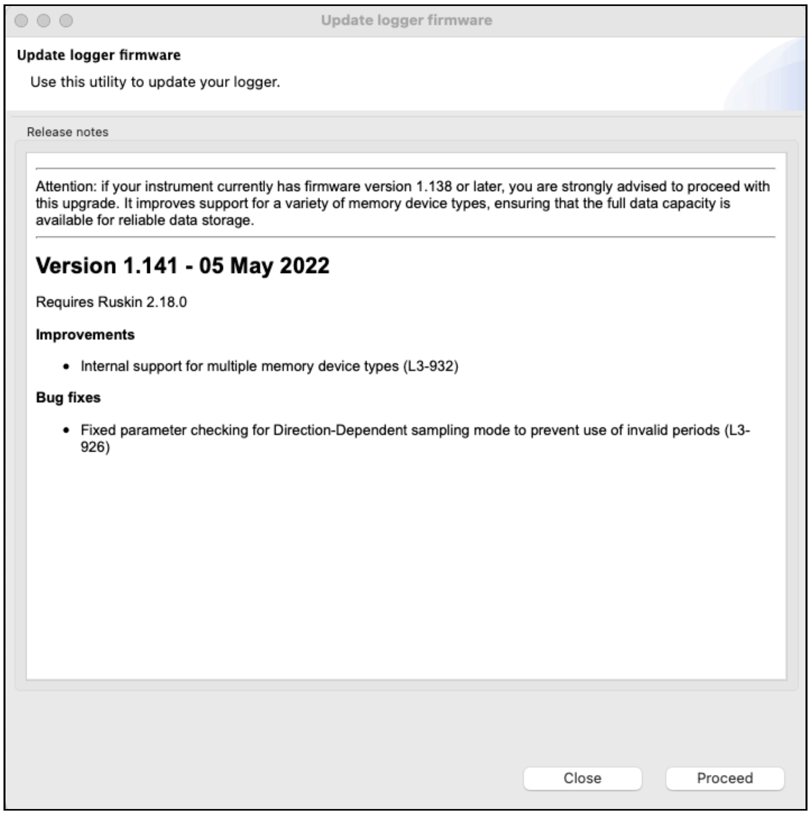
If a newer version is available and this update is critical, the **Update firmware** button and a warning message will appear at the top of the **Configuration** tab.



A newer firmware version is available. Please update to ensure full compatibility.

Steps

1. Click **Update firmware** .
   The **Update logger firmware** window will appear. It will contain **Release notes** , outlining the new version number, its release date, improvements, and bug fixes associated with this release, and two standard buttons at the bottom of the window: **Close** and **Proceed** .



Update logger firmware

Use this utility to update your logger.

Release notes

Attention: if your instrument currently has firmware version 1.138 or later, you are strongly advised to proceed with this upgrade. It improves support for a variety of memory device types, ensuring that the full data capacity is available for reliable data storage.

Version 1.141 - 05 May 2022

Requires Ruskin 2.18.0

Improvements

- Internal support for multiple memory device types (L3-932)

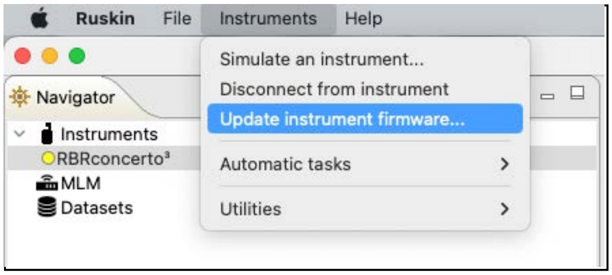
Bug fixes

- Fixed parameter checking for Direction-Dependent sampling mode to prevent use of invalid periods (L3-926)

2. Click **Proceed** to initialise the update.
   The instrument will disappear from the **Navigator** window during the upgrade. Watch the progress bar at the bottom of the window for the status.

> Do not disconnect the instrument until a confirmation message is displayed. Disconnecting during the update process may render the instrument inoperable.

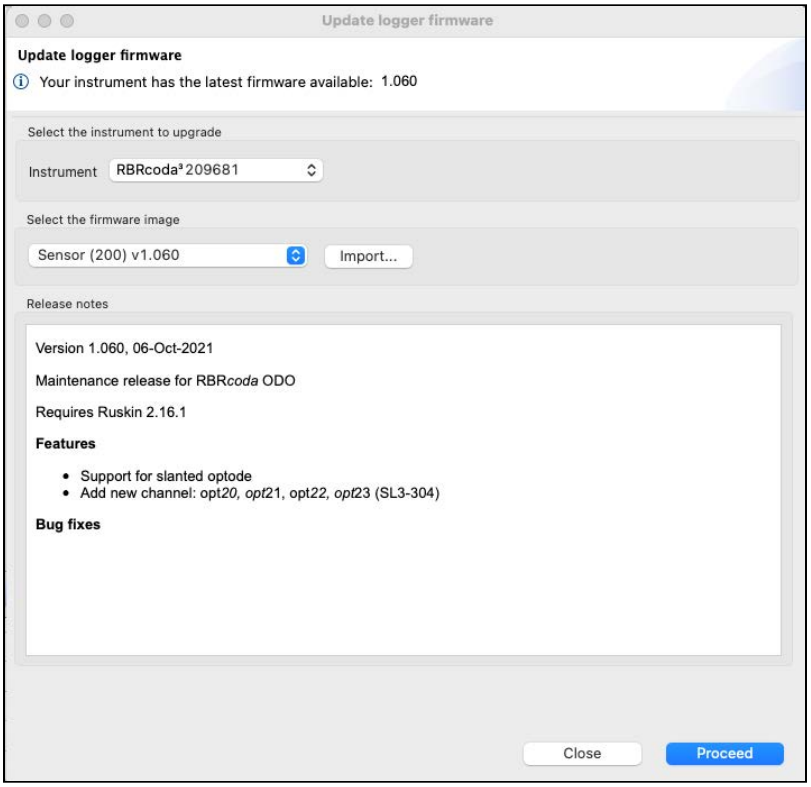
If the upgrade does not complete successfully, go to the **Instruments** menu and launch the update dialogue.



The **Update logger firmware** window will appear. Click **Proceed** again to retry.
If the instrument does not reappear in the **Navigator** window after a firmware update, disconnect your instrument from the USB port and then reconnect to establish communication.

15.2 Routine firmware updates

If the latest update is not critical, Ruskin will not display any warning message. However, you can always check manually if a newer firmware warning is available. Go to **Instruments** > **Update instrument firmware.**
The **Update logger firmware** dialog box will appear.



Update logger firmware

ⓘ Your instrument has the latest firmware available: 1.060

Select the instrument to upgrade

Instrument: RBRcoda³ 209681

Select the firmware image

Sensor (200) v1.060

Release notes

Version 1.060, 06-Oct-2021

Maintenance release for RBR*coda* ODO

Requires Ruskin 2.16.1

Features

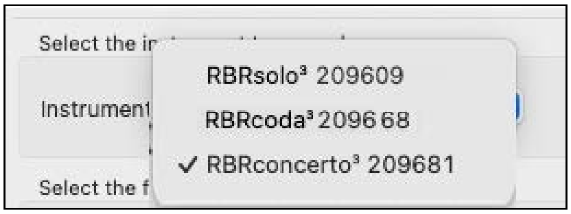
- Support for slanted optode
- Add new channel: opt*20*, *opt*21, opt*22*, *opt*23 (SL3-304)

Bug fixes

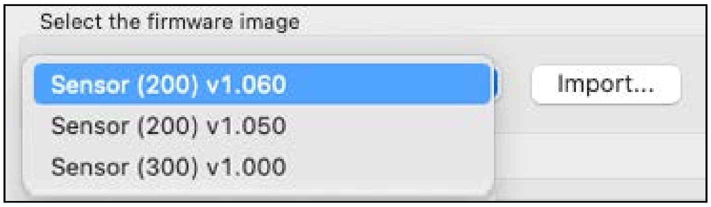
In addition to the new version number, its release date, improvements, and bug fixes, there will be two additional sections: **Select the instrument to update** and **Select the firmware image**.

Steps

1. From the **Instrument** dropdown, select the instrument you wish to upgrade.



2. From the next dropdown, select the latest firmware image.



3. Click **Proceed** to initialise the update.
   The instrument will disappear from the **Navigator** window during the upgrade. Watch the progress bar at the bottom of the window for the status.

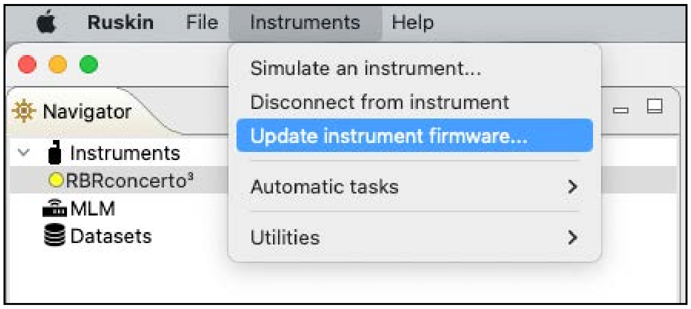
> Do not disconnect the instrument until a confirmation message is displayed.
> Disconnecting during the update process may render the instrument inoperable.

If the upgrade does not complete successfully, click **Proceed** again to retry.
If the instrument does not reappear in the **Navigator** window after a firmware update, disconnect your instrument from the USB port and then reconnect to re-establish communication.

15.3 Manual firmware updates

In certain cases, RBR may supply a customised firmware version, which will not be on the dropdown list.

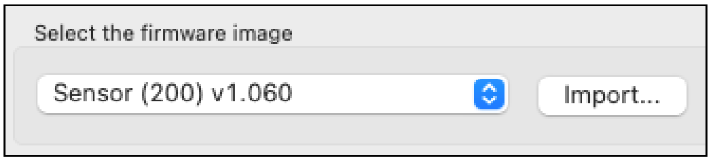
Go to the **Instruments** menu and launch the update dialogue.



The **Update logger firmware** dialog box will appear. Follow the steps below to import the firmware image provided by RBR and update your firmware.

Steps

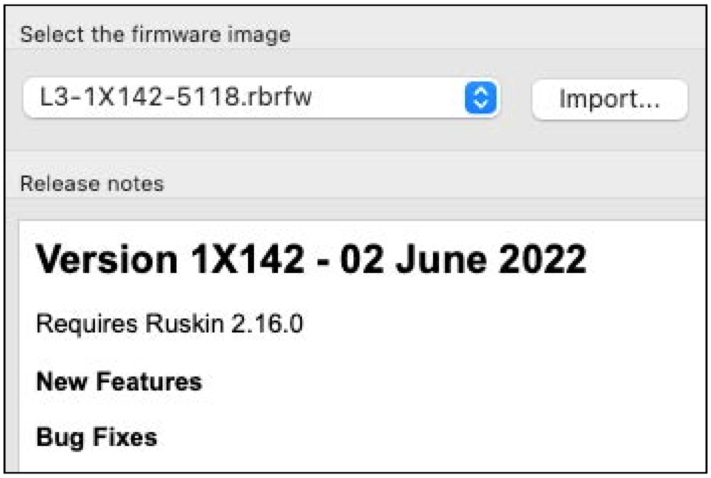
1. In the **Update logger firmware** dialog box, click **Import**.



Select the firmware image

2. Browse to the location of the firmware update file. The file will have the extension .rbrfw.

3. Select the file and click **Open.**
   The file will appear on the dropdown list.



Select the firmware image

Release notes

Version 1X142 - 02 June 2022

Requires Ruskin 2.16.0

New Features

Bug Fixes

4. lick **Proceed** to initialise the update.
   The instrument will disappear from the **Navigator** window during the upgrade. Watch the progress bar at the bottom of the window for the status.

> Do not disconnect the instrument until a confirmation message is displayed. Disconnecting during the update process may render the instrument inoperable.

16 User calibration of DO sensors

Factory calibration coefficients are calculated for each sensor, and the coefficients are stored on the instrument.

RBR calibration certificates contain calibration equations, coefficients, and residuals for each sensor. Hard copies are provided with each shipment. RBR can replace lost or misplaced calibration certificates upon request.

RBR recommends calibrating your instrument before any critical deployment, periodically once a year, or if you suspect the readings to be out of specifications.

Discuss your calibration requirements with RBR. In some cases, the instrument will need to be returned to RBR to have it checked and re-calibrated.

Please contact RBR for our current calibration fees.

View current calibration coefficients for your sensors under the Calibration tab in the **Properties** window.

Dissolved oxygen sensors (OxyGuard®) can be field-calibrated. Update the calibration coefficients for the DO channel periodically.

To calibrate a sensor, you need to let it sample in a known environment and then compare the result to the established value. The difference will be used to generate calibration coefficients. With the **N**-point method, the known environment is changed **N** times. The higher the **N**, the more accurate are calibration coefficients.

For DO sensors, RBR recommends a two-point user calibration, which includes a **zero reference point** and a **non-zero reference point**. It is, however, acceptable to use a one-point calibration.

16.1 Data collection

Follow the steps below to obtain the calibration dataset for your Oxyguard DO sensor.

Equipment and materials

- A wide-mouth, black-walled container with a cover
- A small container or beaker
- Air pump (for example, an aquarium air pump with a stone bubbler)
- Magnetic stirrer
- Sodium sulphite Na_2SO_3
- Distilled water

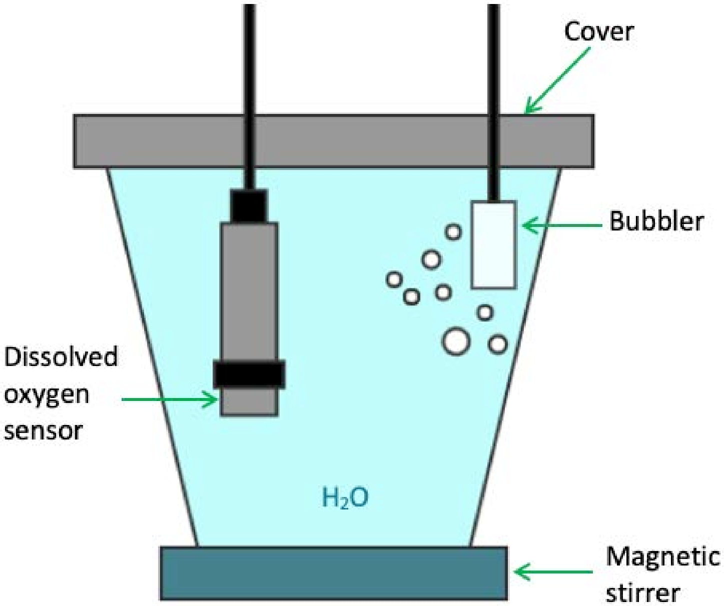
Ruskin configuration

1. Go to **Configuration**.
2. Select **UTC** or **Local**, depending on your preferences, to synchronise the instrument clock with your computer.
3. Set the sampling rate at 1s (1Hz).
4. Enable your schedule.

⚠ The instrument must be logging throughout the calibration procedure.

Non-zero reference point

1. Fill the black-walled container with distilled water.
2. Install the magnetic stirrer.
3. Install the air pump.
4. Switch on both the air pump and the magnetic stirrer.
5. Leave them on overnight, or for at least 3-4 hours, to achieve saturation.
6. Submerge the dissolved oxygen sensor in the solution, going in at an angle and stirring to avoid bubble buildup on the sensor window.
   Leave some space between the sensor and the beaker walls.
7. Position the sensor near the stirrer but not directly in the bubbles.
   The bubbler should be shallower than the sensor.



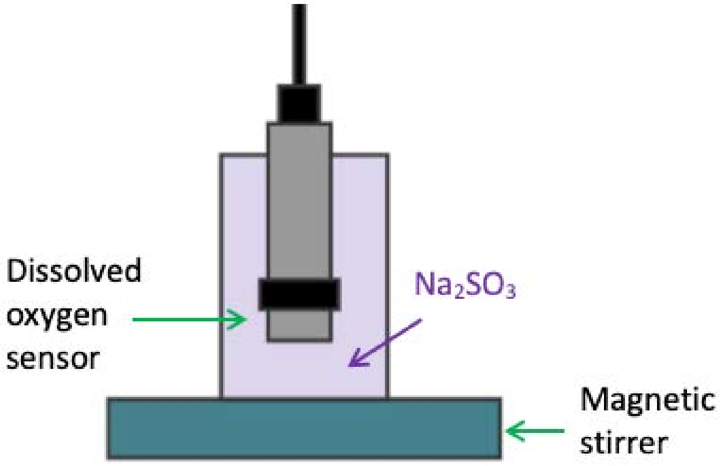
8. Record the data for at least 30 minutes.
   This is the 100% oxygen concentration calibration point.

Zero reference point

> ⚠ Omit this step when performing the one-point calibration.
> If you calibrate your dissolved oxygen sensor using the one-point calibration method at 100% oxygen concentration only, calibration should be performed at the expected temperature and salinity of the deployment environment.

1. In a small container or beaker, prepare a 0.1% solution of sodium sulphite (0.25g of Na_2SO_3 for 250mL tap water).
2. Mix the solution thoroughly with a magnetic stirrer. The solution will be oxygen-free after 15 minutes.

3. Submerge the dissolved oxygen sensor in the solution, going in at an angle and stirring to avoid bubble buildup on the sensor window.
   Leave some space between the sensor and the beaker walls.



4. Record the data for at least 30 minutes.
   This is the 0% oxygen concentration calibration point.

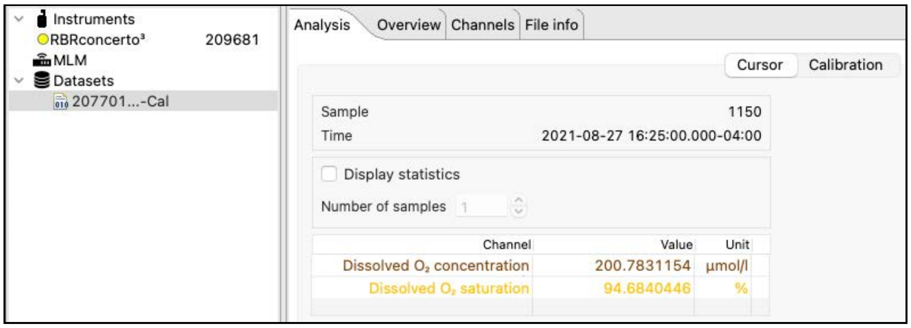
⚠ This Na_2SO_3 solution has a limited lifespan and should not be used for more than a day.

16.2 Calibration on Ruskin

For DO sensors, RBR recommends a two-point user calibration, which includes a **zero reference point** and a **non-zero reference point**. It is, however, acceptable to use a one-point calibration.

Two-point calibration steps on Ruskin

1. Stop the deployment.
2. Download the dataset.
3. Open the dataset on Ruskin. The **Properties** windows will change to dataset properties, with the **Analysis** tab displayed.



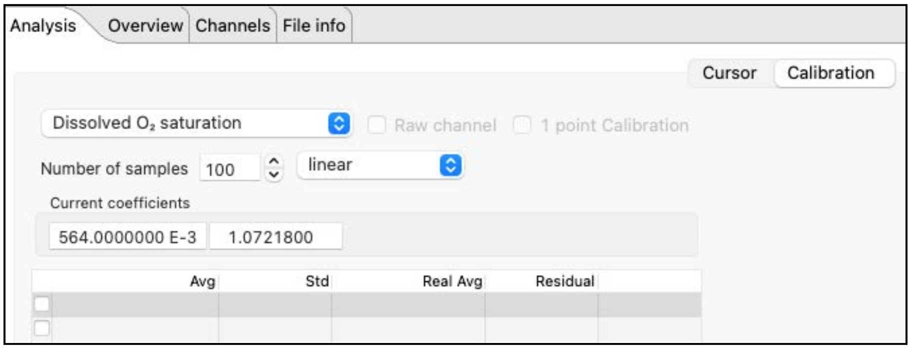
Sample 1150
Time 2021-08-27 16:25:00.000-04:00

□ Display statistics
Number of samples 1

Channel	Value	Unit
Dissolved O_2 concentration	200.7831154	µmol/l
Dissolved O_2 saturation	94.6840446	%

4. Click **Calibration**.

5. Select **Dissolved O_2 saturation** from the dropdown list.
   By default, the **Number of samples** is set to **100**, which means Ruskin will average the calibration coefficients over a hundred sample points. Typically, this value should be in the range from 50 to 100 samples.
   The **Raw channel** box is unchecked, and thus the true DO values will be displayed in the plot below. To see the raw voltage values, check this box.



Dissolved O_2 saturation □ Raw channel □ 1 point Calibration

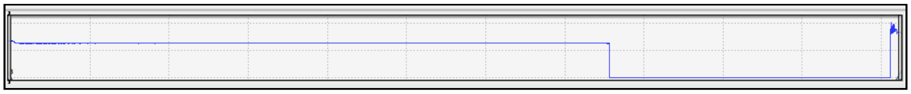
Number of samples 100 linear

Current coefficients

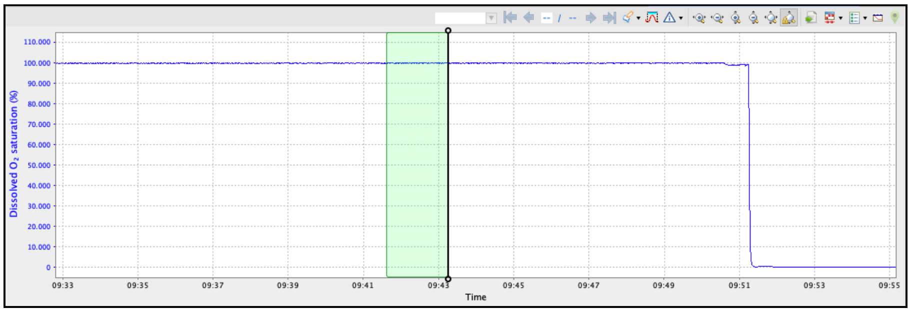
564.0000000 E-3 1.0721800

	Avg	Std	Real Avg	Residual
□				
□				

A downsampled representation of the dataset will appear in the thumbnail view, with two distinct plateaus corresponding to 100% and 0%. Irregular readings in-between mark the times when you changed from distilled water to sodium sulphite solution and reassembled the setup.

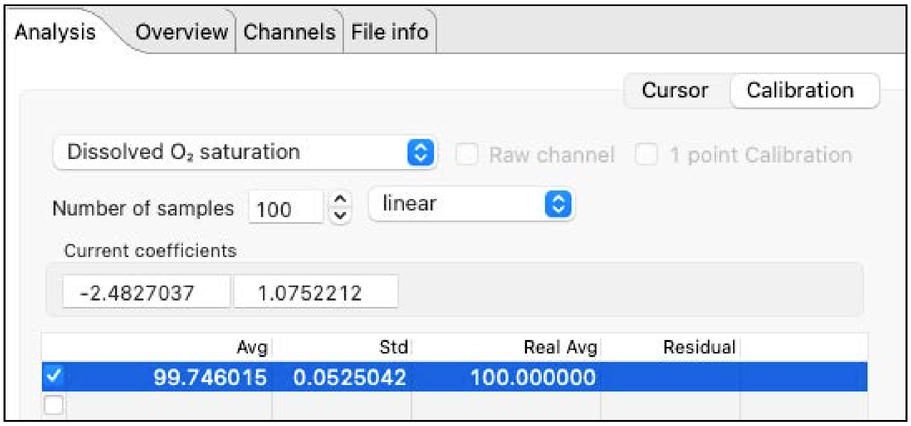


6. Click on a stable point in the **Plot** view corresponding to 100% oxygen.



The interval of 100 samples will be highlighted. The average value (**Avg**) and standard error (**Std**) for this interval will appear in the table above.

7. Under **Real Avg** in the first row, enter **100**.



Dissolved O_2 saturation □ Raw channel □ 1 point Calibration

Number of samples 100 linear

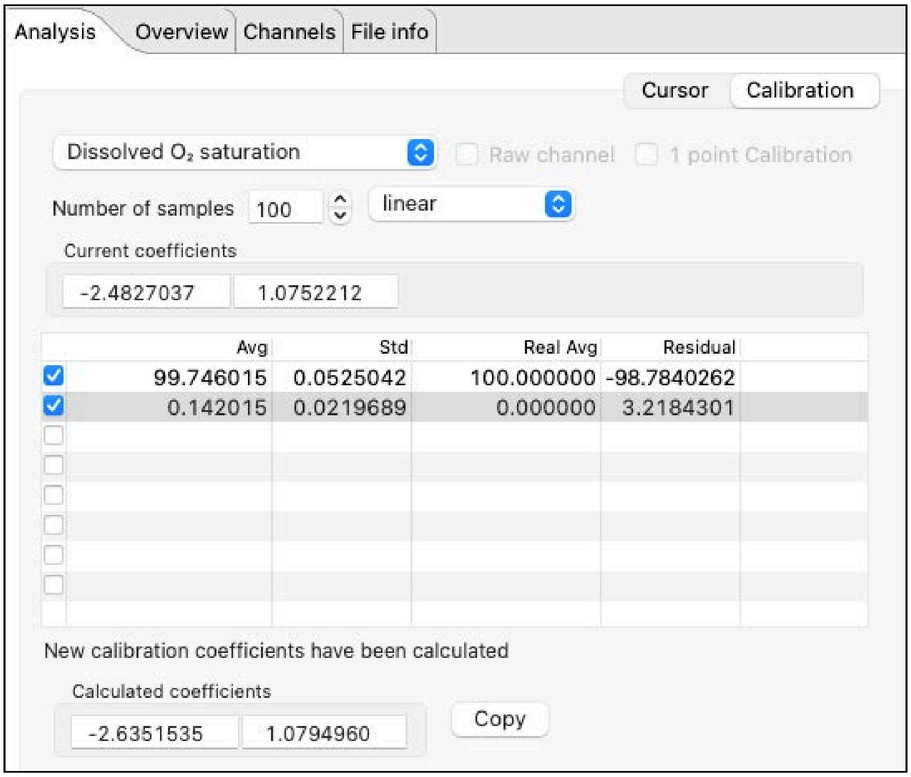
Current coefficients

-2.4827037 1.0752212

	Avg	Std	Real Avg	Residual
☑	99.746015	0.0525042	100.000000	
□				

8. Repeat the same steps for the second plateau, entering **0** under **Real Avg.**

9. Ruskin automatically calculates the calibration coefficients, and the values appear in **Calculated coefficients**.



Dissolved O_2 saturation □ Raw channel □ 1 point Calibration

Number of samples 100 linear

Current coefficients

-2.4827037 1.0752212

	Avg	Std	Real Avg	Residual
☑	99.746015	0.0525042	100.000000	-98.7840262
☑	0.142015	0.0219689	0.000000	3.2184301
□				
□				
□				
□				
□				
□				

New calibration coefficients have been calculated

Calculated coefficients

-2.6351535 1.0794960

10. Click **Copy** to save the new calibration coefficients to the clipboard, or write them down.

11. In the **Navigator** view, find your instrument.
12. Click Calibration and select **Dissolved O_2 saturation** from the dropdown list.
13. Click **C0** to select the entry field and type in the new value (in this example, **-2.6351535**).
14. Click **C1** to select the entry field and type in the new value (in this example, **1.0794960**).
15. Click **Store calibration** to save the new coefficients.

> ⚠ Always click **Store calibration** to save new coefficients on the instrument. Otherwise, all changes will be lost once your session is closed.

One-point calibration steps on Ruskin

> ⚠ If you calibrate your dissolved oxygen sensor using a one-point calibration method at 100% oxygen concentration only, calibration should be performed at the expected temperature and salinity of the deployment environment.

1. Stop the deployment.
2. Download the dataset.
3. Open the dataset on Ruskin. The **Properties** windows will change to dataset properties, with the **Analysis** tab displayed.
4. Click **Calibration**.
5. Select **Dissolved O_2 saturation** from the dropdown list.
6. Check the **1 point calibration** box.
7. Click on a stable point in the **Plot** view corresponding to 100% oxygen.
   The interval of 100 samples will be highlighted. The average value (**Avg**) and standard error (**Std**) for this interval will appear in the table above.
8. Click the **RealAvg** column and select **100** from the dropdown.
9. Click **Copy** to save the new calibration coefficients to the clipboard, or write them down.
10. In the **Navigator** view, find your instrument.
11. Click Calibration and select **Dissolved O_2 saturation** from the dropdown list.
12. Click **C0** to select the entry field and type in the new value.
13. Click **C1** to select the entry field and type in the new value.
14. Click **Store calibration** to save the new coefficients.

> ⚠ Always click **Store calibration** to save new coefficients on the instrument. Otherwise, all changes will be lost once your session is closed.

17 OEM command reference

OEM commands use the following formatting conventions:

1. Examples of literal input to and output from the sensor are shown in **bold** type.
2. In examples of dialogue between the sensor and a host, input to the sensor is preceded by **>>**, while output from the sensor is preceded by **<<**. These characters must not actually be included in commands or expected in responses.
3. Some examples of command dialogues contain descriptive comments which are not part of the command or response. These start with a percent character, **%**.
4. When an item or group of items is optional, it is enclosed in `[square brackets]`.
5. Where an item can be only one of several options, options are separated by vertical bars, **|** .
6. Placeholders for variable fields are in **<***italics enclosed in angle brackets***>.**
7. Lists are used for unknown or variable numbers of items, or to abbreviate large numbers of options. These are specified by giving a first example of an item, followed by a comma and ellipsis: <*example-value*>, … .

17.1 Quick start - streaming and polled modes

> ⚠ You must have a serial terminal to set up your RBR*coda*^{3} in the polled mode. The streaming mode can also be set via Ruskin.

17.1.1 Streaming

The only two parameters needed to operate your sensor in streaming mode are the streaming speed (period), and the serial baudrate. The sensor will begin streaming data as soon as power is provided.

The factory streaming speed is the fastest that the sensor is rated for, e.g. |fast 32 will be set to 32Hz.

The factory baudrate is 9600.

- **sampling** reports or sets various sampling parameters.
  - period [=<*period*>] reports or sets the time between measurements.
    The <*period*> is specified in milliseconds. Values for 1Hz sampling or slower are supported by all sensors and must be given in multiples of 1000 (1000ms, 2000ms, etc.). If the sensor is configured to support fast (sub-second) measurement speeds, the <*period*> must be one of the following: 500 (2Hz), 250 (4Hz), 125 (8Hz), 63 (16Hz), and 31 (32Hz).

Examples

```
>> sampling
<< sampling mode = continuous, period = 250, userperiodlimit = 63
```

The sensor has been programmed for continuous 4Hz sampling. The shortest sampling period allowed is 16Hz.

```
>> sampling period = 5000
<< sampling period = 5000
```

The sensor has been programmed to sample once every 5 seconds.

```
>> sampling
<< sampling mode = continuous, period = 125, burstlength = 60, burstinterval = 300000
```

The sensor has been programmed for continuous 8Hz sampling. The programmed values of the burst parameters are shown/reported only when the sensor supports sampling in **Tide** mode. They do not apply to continuous sampling.

- **serial baudrate** [=<*rate*>] reports or sets the parameters which apply to the serial link.
  The following baudrates are supported: 115200, 19200, 9600, 4800, 2400, and 1200. The default shipped from the factory is 9600.

Example

```
>> serial baudrate = 4800
<< serial baudrate = 4800
```

The sensor has been programmed with a baudrate of 4800.

17.1.2 Polling

In the polled mode, the data transfer from the RBR*coda*^{3} occurs only when you request it. Disable streaming first, and then use the `fetch` command to retrieve a sample from the sensor.

- **`stream`** turns data streaming on or off. If the command is given with no parameters, the value of the `state` parameter is reported.
  - `state [= on | off]` reports the state of the streamed data feature and optionally turns it on or off. When the feature is on, acquired data is sent over the serial link. When the feature is off, data is not sent.

Example

```
>> stream
<< stream state = off
>> stream state = on
<< stream state = on
```

- **`fetch`** requests an "on-demand" sample set from the sensor.

Example

```
>> fetch
<< 1000, 18.1745, 12.7052
```

The example is for a two-channel sensor. Single-channel sensors will only report one value.

Go to the Commands section for the complete list of commands.

17.2 Supported channel types

Channel types are short, pre-defined names for the installed channels. These names are used by the channel command.

Key	Description
`doxy26`	Dissolved oxygen saturation, RBR*coda*^{3} DO (OxyGuard)
`doxy20, doxy21`	Dissolved oxygen concentration, RBR*coda*^{3} T.ODO
`doxy22`	Dissolved oxygen saturation, derived, RBR*coda*^{3} T.ODO
`fluo45`	Chlorophyll *a*, RBR*tridente*
`fluo46`	fDOM, RBR*tridente*
`irr_03`	Narrow-band irradiance, RBR*coda*^{3} rad
`irr_07`	Narrow-band irradiance, RBR*quadrante*
`opt_05`	Phase, RBR*coda*^{3} T.ODO
`par_02`	Photosynthetically active radiation, RBR*coda*^{3} PAR (LI-COR)
`par_04`	Photosynthetically active radiation, RBR*coda*^{3} PAR
`par_07`	Photosynthetically active radiation, RBR*quadrante*
`pres25`	Pressure, with temperature correction, RBR*coda*^{3} D
`pres26`	Pressure, with temperature correction, RBR*coda*^{3} T.D
`temp18, temp21`	Temperature, RBR*coda*^{3} T
`temp12, temp20`	Temperature, RBR*coda*^{3} T.D
`temp15`	Temperature, RBR *coda*^{3} T.ODO
`turb05`	Turbidity, RBR*coda*^{3} Tu (Seapoint)
`turb15`	Turbidity, RBR*tridente*
`turb16`	Backscatter, RBR*tridente*
`turb17, turb18`	Turbidity, RBR*coda* Tu

17.3 Command entry

Start and end of a command

A *potential* command is considered to begin when its first character is received. The potential command has been received once the instrument sees a termination character; either one of <**CR**> (0x0D) or <**LF**> (0x0A). Combinations of the two characters are dealt with as follows:

```
>> <CR><LF>
<< Ready:
```

```
>> <LF><CR>
<< Ready:
```

```
>> <CR><CR>
<< Ready: Ready:
```

```
>> <LF><LF>
<< Ready: Ready:
```

In the first two cases, the second character is considered redundant and is discarded; only one **`Ready:`** prompt is sent. For the last two cases, the second character is treated as a second empty command, so it also provokes the instrument's prompt, and a total of two prompts are sent (see also the `prompt` command).

Uppercase and lowercase

In general, the sensor is not sensitive to the case of the input; for example, **`ID`**, **`Id`**, **`iD`**, and **`id`** are all acceptable forms for the **`id`** command. Any exceptions to this rule are highlighted when necessary. The case of the output sensor response may or may not match the case of the input.

Common error messages

The received message may or may not form a valid command; errors detectable by the sensor will vary from one command to another, but some of the common, general errors include:

1. `E0102 invalid command` *`<unknown-command-name >`*
2. `E0107 expected argument missing`
3. `E0108 invalid argument to command:` *`<invalid-argument>`*

See Error messages .

17.4 Error messages

Refer to the list below for the most common error messages.

Each error message begins with *Ennnn*, where *nnnn* is a four-digit decimal number, padded with leading zeroes if necessary.

The number allows host software to interpret the error code correctly even if the rather terse messages from the sensor are unsuitable for any reason.

Note that some messages may be followed by variable elements not shown here.

Key	Description
E0102	`invalid command` *<unknown-command-name>*
E0107	`expected argument missing`
E0108	`invalid argument to command:` *<invalid-argument>*
E0109	`feature not available`
E0111	`command failed`
E0114	`feature not supported by hardware`
E0410	`no sampling channels active`
E0501	`item is not configured`
E0505	`no channels configured`

⚠ You may encounter errors not included in the list above. In that case, please contact RBR for help.

17.5 Commands

17.5.1 sampling

Usage

`>> sampling [ mode | period | availablefastperiods | userperiodlimit | burstinterval | burstlength]`

Description

Allows various parameters to be reported.

The following list includes currently supported parameters:

- **`mode`** `[=` *`<mode>`*`]` reports the sampling mode for the current schedule:
  - **continuous** mode is supported by all sensors; measurements are taken at the specified period between the start and end times.
  - **tide** mode is available for instruments which are so configured in the factory. Measurements are taken in bursts between the start and end times. The average value of all measurements in a burst is computed and stored as a single sample value at the end of the burst. The time between the start of two consecutive bursts is given by the **burstinterval**, the number of measurements in the burst is given by the **burstlength**, and the time between measurements within the burst is given by the **period**. This mode allows host software to process the data for tide monitoring applications.

- **`period`** `[=` *`<period>`*`]` reports or sets the time between measurements.
  This is straightforward in **continuous** mode; in **tide** mode it is the time between continuous measurements *within* the burst. The *<period>* is specified in milliseconds.
  Values for 1Hz sampling or slower are supported by all instruments. In **continuous** mode the permitted range is typically 1000 (one second) to 86400000 (24 hours), in increments of 1000. Some instruments may have a lower limit which is more than 1000 if an attached sensor has a very slow data acquisition time.
  For all other modes the upper limit is 255000 (about 4 minutes); this is unlikely to be a constraint in practice, as the period is the time between measurements *within* the burst.
  If the instrument is configured to support fast (sub-second) measurement periods, the *`<period>`* must correspond to an exact frequency in Hz, to the nearest millisecond. Permitted values are further constrained to a supported subset of sample rates, which may be determined via the **availablefastperiods** parameter described below.

- **`availablefastperiods`** reports a list of the fast measurement periods available for the instrument for sampling rates faster than 1Hz. Each available period is reported to the nearest millisecond, and the values are separated by a vertical bar, **|**. If there are no fast periods available, the word **none** is returned instead of a list of values. When a list is reported, the **period** for sampling rates faster than 1Hz can be set to any value in the list, but no others.
  Note that the periods on the list may be supported in some modes but not others, depending on the instrument's configuration.

- **`userperiodlimit`** reports the minimum period which can be used in 'fast' sampling modes, in milliseconds; the **period** can not be set to a value less than this.

The following parameters are available only if the sensor supports **tide** mode.

- **burstinterval** [= <*burstinterval*>] reports or sets the time between the first measurement of two consecutive bursts; it is not the gap between the end of one burst and the start of the next. The <*burstinterval*> is specified in milliseconds.
  The absolute limits of the permitted range are 1000 (one second) to 86400000 (24 hours), and the <*burstinterval*> must be set to a multiple of 1000. However, before the instrument can be enabled for sampling the value set must also be consistent with the measurement **period** and **burstlength**.

- **burstlength** [= <*burstlength*>] reports or sets the number of measurements taken in each burst.
  The permitted range is 2 to 65535, but before the instrument can be enabled for sampling the value set must also be consistent with the measurement **period** and **burstinterval**.
  The constraining relationship between the burst parameters is: **burstinterval** > (**burstlength** * **period**).

Examples

```
>> sampling
<< sampling mode = continuous, period = 250, userperiodlimit = 63
```

The instrument has been programmed for continuous 4Hz sampling. This instrument does not support any of the other modes, so the parameters are not reported.

```
>> sampling
<< sampling mode = continuous, period = 63, userperiodlimit = 63, burstlength = 60,
burstinterval = 300000
```

The instrument has been programmed for continuous 16Hz sampling. The programmed values of the burst parameters are reported but do not apply to **continuous** mode.

```
>> sampling mode = tide
<< sampling mode = tide
>> sampling
<< sampling mode = tide, period = 63, userperiodlimit = 63, burstlength = 60,
burstinterval = 300000
```

Tide mode is enabled without changing the other parameters; the instrument is now programmed to take a burst of 60 measurements at 16Hz every five minutes, and store the average of the 60 measurements.

```
>> sampling availablefastperiods
<< sampling availablefastperiods = 500|250|125|63

>> sampling availablefastperiods
<< sampling availablefastperiods = none
```

The first example shows support for sampling at 2, 4, 8, and 16Hz in at least one of the available modes; the second example shows that no sampling faster than 1Hz is supported.

Errors

`Error E0108 invalid argument to command:` ***`<invalid-argument>`***
The command was given with an argument which is unrecognised or misplaced.
`Error E0109 feature not available`
An attempt was made to use a feature which the instrument is not configured to support; for example "sampling mode = tide" if the instrument does not support tidal averaging.

17.5.2 outputformat

Usage

`>> outputformat [ type | availabletypes | channelslist | labelslist ]`

Description

Reports or sets the format used to transmit data over the communications link.

This format applies to both fetched data and live streamed data, if available. If no arguments are given, the current setting of the **`type`** parameter is reported.

The parameters currently supported are:

- **`type`** `[= <type>]` sets and/or reports the current output format type. Not all instruments support all the formats; query the **`availabletypes`** parameter to check which formats are available.
- **`availabletypes`** reports the formats which are available.
  At present, the supported formats are:
  - **`caltext06`** , calibrated ASCII output.
    The output starts with a duration (milliseconds) since power up. A value for each channel is then sent; this is the measured parameter after conversion to physical units according to the instrument's current calibration. All values are shown to four decimal places. A comma and space separate the timestamp and values. The timestamp is in milliseconds.
    Format: `timestamp, `*`<value1>, <value2>`*`, ...`
    Examples for a two-channel instrument: `10000, 38.6664, 21.5183`
  - **`caltext08`** , calibrated ASCII output.
    The output starts with a duration (milliseconds) since power up. A value for each channel is then sent; this is the measured parameter after conversion to physical units according to the instrument's current calibration. All values are shown to nine significant digits, ensuring no loss of resolution. A comma and space separate the timestamp and values. The timestamp is in milliseconds.
    Format: `timestamp, <value1>, <value2>, ...`
    Examples for PAR and rad sensors: `1250, 38.6671142, 22.0217241, 10.9596633`

- **`channelslist`** reports a list of names and units for the active channels, in order.
  This list is helpful in identifying the channel corresponding to each value in the transmitted data. Any channels which have been turned **`off`** are excluded. The list expands with support added for more sensor types over time, so not all firmware versions will support every sensor type listed. If there is more than one channel of a particular type, the same information is reported for all of them.
  The list of all possible names is given below. Some are quite generic, and others are very sensor-specific :
  - `amplitude`
  - `depth`
  - `dissolved-O2`
  - `irradiance`
  - `O2_air_saturation`
  - `O2_concentration`
  - `uncompensated_O2_concentration`
  - `phase`
  - `PAR`
  - `period`
  - `pressure`
  - `temperature`
  - `turbidity`
  - `voltage`

- **`labelslist`** reports the list of active channel labels, in order.
  This list is helpful in identifying the channel corresponding to each value in the transmitted data. Any channels which have been turned **off** are excluded. A channel's label is set upon request for OEM customers (see the `channel` command).

Note that any individual channel value may be replaced by one of the following:

- **`Error-`** <*EC*>: an identifiable error occurred on this channel, represented by one of the following two-digit error codes:
  00 - generic or unknown error
  09 - no sample was started
  10 - sample acquisition still in progress
  11 - sample process failed somewhere
  14 - supporting channel value not valid, or unknown equation
  16 - channel value is outside reasonable range
  21 - channel is not correctly calibrated

- **nan** : the value is Not A Number in IEEE floating point format, which indicates an internal problem with calculating the value

- **`inf / -inf`** : attempting to calculate the value produced a result outside the range which can be represented

- **`###`** : the channel is not calibrated, so an output value could not be calculated

Examples

```
>> outputformat
<< outputformat type = caltext06, labelslist = temperature_00|pressure_00
```

```
>> outputformat type
<< outputformat type = caltext06
```

```
>> outputformat availabletypes
<< outputformat availabletypes = caltext06|caltext08
```

```
>> outputformat channelslist
<< outputformat channelslist = temperature(C)|pressure(dbar)
```

```
>> outputformat labelslist
<< outputformat labelslist = temperature_00|pressure_00
```

Errors

`Error E0108 invalid argument to command:` *`<invalid-argument>`*
The command was given with an argument which is unrecognised or misplaced.

17.5.3 stream

```
>>stream [state]
```

Description

Turns data streaming on or off.

If the command is given with no parameters, the value of the **`state`** parameter is reported. The operating parameters of the serial link, such as baudrate and mode, are accessed using the serial command.

- **`state`** `[= on | off]` reports the state of the streamed data feature, and optionally turns it on or off. When the feature is on, acquired data is sent over the serial link. When the feature is off, data are not sent.

Examples

```
>> stream
<< stream state = off
```

```
>> stream state = on
<< stream state = on
```

Errors

`Error E0108 invalid argument to command:` ***`<invalid-argument>`***
The command was given with an argument which is unrecognised or misplaced.

17.5.4 fetch

Usage

`>> fetch [ channels ]`

Description

Requests an "on-demand" sample set from the sensor.

If a recent scheduled sample set is available, those values may be returned to satisfy the **`fetch`** request. "Recent" in this context means less than 500ms old. Otherwise, a sample set is explicitly acquired for the benefit of the **`fetch`**.

The sensor simply responds with the *`<sample-data>`*. Depending on the configured settling time (or power-on settling delay) for the attached sensors, there may be a noticeable delay before the *`<sample-data>`* appears. Refer to the channels command for further discussion of settling time.

- **`channels`** `=` *`<listofchannels>`* is the list of channels to be acquired, using either indices or labels of channels (see `channel`). At least one valid channel must be specified.
  Without **`channels`** `=` *`<listofchannels>`*, the **`fetch`** command reports *`<sample-data>`* with the channels in the expected order, as revealed by either `outputformat` **`channelslist`** or `outputformat` **`labelslist`**. If the *`<listofchannels>`* specifies the channels in a different order than this default, then they are reported in the same order as given by the *`<listofchannels>`*.

> If a channel or one of its supporting channels has its status set to **off**, requesting that channel will return an error as *`<sample-data>`* for this channel. See `channel` for details.

Examples

```
>> outputformat labelslist
<< outputformat labelslist= temperature_00|pressure_00
>> fetch
<< 29000, 23.2868, 10.2484
```

The instrument will fetch a sample from all channels.

```
>> channel 1 type
<< channel 1 type = temp12
>> channel 2 type
<< channel 2 type = pres26
>> >> outputformat
<< outputformat type = caltext06, labelslist = temperature_00|pressure_00
>> fetch
<< 29000, 23.2868, 10.2484
>> fetch channels = 2|1
<< 31000, 23.2870, 10.2502
>> fetch channels = pressure_00
<< 35000, 10.2495
```

The instrument will fetch a sample from specific channels only.

Errors

Error E0108 invalid argument to command: *<invalid-argument>*
The command was given with an argument which is unrecognised or misplaced.

Error E0410 no sampling channels active
The instrument has no channels activated for sampling.

17.5.5 calibration

Usage

>> calibration *<indexorlabel>* [datetime | c0 | ... | c<Nc> | x0 | ... | x<Nx> | n0 | ... | n<Nn>]

Description

Reports information regarding the most recent calibration for the channel specified by *<indexorlabel>*.

<indexorlabel> is a required parameter in all cases. The number and types of coefficients reported will vary depending on the channel **type** (see **channel**).
Some sensor types have complicated equations with many coefficients, and the equation may also use the output of one or more of the other channels in the instrument for correction or compensation purposes. This is a powerful facility, but requires a lot of information, and the **calibration** command helps to manage that information.

If given with only the `<indexorlabel>`, the command reports all information applicable to the channel. Requesting just the *`<label>`* or just the *`<index>`* will return an error message (E0108); it must always be *`<indexorlabel>.`* However, you can request **`datetime`** on its own.

Coefficients in each group may be requested all together by using the group name: **c**, **x**, or **n**. Requesting an item which does not exist (eg. **c3** for a linear sensor) may result in either an error message, or a response such as **c3 = n/a**.

A special value **`allindices`** may be given for the channel <*index*>, causing the requested parameters to be reported for all channels. The output for each channel is terminated by the double vertical bar **||**, except for the last channel which is terminated by a <*cr*><*lf*> pair as normal.

A special value **`alllabels`** may be given for the channel *`<label>`*, causing the requested parameters to be reported for all channels. The output for each channel is terminated by **||**, except for the last channel which is terminated by a *`<cr><lf>`* pair as normal.

Additionally, **`datetime`** is reported using a <*YYYYMMDDhhmmss*> format. It is the date and time of the most recent calibration change for the channel.

Coefficients are arranged in three groups, **`c0…`**, **`x0…`**, and there is a further group **`n0…`** of cross-channel reference indices. The purpose and function of each are described below. The groups may be referred to by name: **c**, **x** or **n**.

- **`c0, c1…`** are the primary coefficient values, reported as floating point numbers using a format with a mantissa and exponent; for example, 3.3910000e+003.
  These coefficients apply to a 'core' equation which yields a basic value for the parameter. In many cases this is all that is needed, and the **x** and **n** groups are not required. The exact function of each coefficient depends on the equation used.
- **`x0, x1…`** are reported for only some equation types, namely those which employ cross-channel compensation or correction of the primary value using one or more inputs from other channels in the logger. **`x0, x1`**… are also coefficient values which follow the same rules as the **c** group. The exact function of each coefficient depends on the equation used.
- **`n0, n1…`** apply only to some equation types, those using cross-channel compensation or correction. They are not coefficients, but (in general) the indices of other instrument channels whose data are also inputs to the equation for channel <*`indexorlabel`*>. This permits output data to depend on more than one channel; for example, to be corrected for temperature dependencies. The values of **n0, n1,** etc. are simple integer numbers. Remember that the index of the first channel is `1`; zero is not valid.

Equations which use the **`x0, x1…`** coefficients require at least one **n** index. The instrument may also have "derived parameter" channels, which are output values computed from measured channels: a good example would be pressure, which requires correction by a separately measured temperature channel.. In such cases **`n0`** is required to tell the instrument to use the input from the supporting channel.

When setting new values in either the **c** or the **x** group of coefficients, two rules apply.

1. A value for **`datetime = `***`<YYYYMMDDhhmmss>`* *must* also be specified, as the **`datetime`** must reflect the most recent coefficient change.
2. Only one group at a time can be accessed. You cannot set both **c** and **x** coefficient values in a single command. Within either group, one, all, or any number of the coefficients in between may be set.

⚠ Any value for any **n** coefficient can be set only at the factory.

There is one special case when the value of an **n** index may be the text field "**value**". Like all other **n** coefficient settings, this can be set only at the factory, and applies when an equation requires a correction term using a parameter which the instrument does not measure. In this case, the default parameter set by the `settings` command will be used. An example would be the derived channel for depth, which depends on measured pressure but also requires a value for atmospheric pressure, available via the `settings` command.

Examples

```
>> calibration 2
<< calibration 2 label = pressuretemperature_00, datetime = 20201204213302, c0 =
3.3878490e-003, c1 = -275.76637e-006, c2 = 3.9983520e-006, c3 = -221.88040e-009

>> calibration pressuretemperature_00
<< calibration pressuretemperature_00 index = 2, datetime = 20201204213302, c0 =
3.3878490e-003, c1 = -275.76637e-006, c2 = 3.9983520e-006, c3 = -221.88040e-009
```

The instrument is responding with the calibration for a single channel by index and by label.

```
>> calibration allindices
<< calibration 1 label = voltage_01, datetime = 20171203134201, c0 = 9.9873456e+000, c1
= 7.5640000e+000 || calibration 2 label = voltage_02, datetime = 20171203134201, c0 =
9.9873456e+000, c1 = 7.5640000e+000

>> calibration alllabels
<< calibration voltage_01 index = 1, datetime = 20171203134201, c0 = 9.9873456e+000, c1
= 7.5640000e+000 || calibration voltage_02 index = 2, datetime = 20171203134201, c0 =
9.9873456e+000, c1 = 7.5640000e+000
```

The instrument is responding with the calibration for all channels by indices and by labels.

```
>>calibration 1 datetime = 20171203134201, c0 = 3.5000000e-003, c1 = -250.00002e-006,
c2 = 2.7000000e-006, c3 = 23.000000e-009

<<calibration 1 label = temperature_00, datetime = 20000401000000, c0 = 3.5000000e-003,
c1 = -250.00002e-006, c2 = 2.7000000e-006, c3 = 23.000000e-009
```

The instrument is setting the calibration coefficients and time.

Errors

`Error E0107 expected argument missing`
An argument expected by the instrument was not given with the command; for example, there must always be an `<index>` argument, and if setting coefficients then all fields required must be supplied.

`Error E0108 invalid argument to command:` ***`<invalid-argument>`***

The command was given with an argument which is unrecognised or misplaced, for example:

- *`<index|label>`* is out of range; channels are numbered from 1 to N; zero is not valid, or no channel exists with this label
- improperly formatted or invalid *`<datetime>`* argument
- invalid coefficient name
- improperly specified value for a coefficient

`Error E0501 item is not configured`
There is a problem with the configuration of the specified channel. Please contact RBR for help.

`Error E0505 no channels configured`
There is a serious fault with the instrument. Please contact RBR for help.

17.5.6 channel

Usage

`>> channel` ***`<indexorlabel>`*** **`[ type | module | status | settlingtime | readtime | equation | userunits | gain | availablegains | derived | label | index ]`**

Description

Returns information about the channel at the specified <*indexorlabel*>.

Channels can be identified either by an index (an integer) or by their label (a meaningful string, for example temperature_00). The first channel has an index of 1.

Special values of **`allindices`** or **`alllabels`** may be given for the channel <*indexorlabel*>, causing the requested parameters to be reported for all channels. The output for each channel is terminated by the double vertical bar **`||`**, except for the last channel which is terminated by a *`<cr><lf>`* pair. The following parameters give the basic information available for all channels. None of these values can be modified by OEM customers.

- **`type`** is a short, pre-defined name for the installed channel, for example, **temp12** for temperature, **pres25** for pressure, or **irr_03** for irradiance. See Supported channel types for the full list.

- **`module`** is the internal address to which this channel responds; it is normally of no interest to OEM customers.

- **`status`** `[=` *`<status>`*`]` is another basic parameter which applies to all channels. It allows any individual channel to be turned off or on for the duration of a deployment, and can be modified by OEM customers.
  - **`on:`** the channel is activated for sampling; its data will appear in streamed output if streaming is enabled.
  - **`off:`** the channel is not sampled, no data will be streamed for this channel.

- **`settlingtime`** is the minimum power-on settling delay in milliseconds required by this channel, taking into account both the sensor and the interface electronics.

- **`readtime`** is the typical data acquisition time in milliseconds required by this channel, again taking into account both the sensor and the interface electronics.
  Most channels have a fixed, pre-determined **`readtime`**, but for some it may be variable. An example would be a channel which supports, and is configured to use, the auto-ranging feature: the **`readtime`** is longer when the channel is in auto-ranging mode than when operated in fixed-gain mode. The instrument adjusts the reported **`readtime`** value to reflect the operating mode and status of the channel.

- **`equation`** is the type of formula used to convert raw data readings to physical measurement units. The values for the core equations are shown below as examples.
  - **`tmp,`** temperature
  - **`lin,`** linear
  - **`qad,`** quadratic polynomial
  - **`cub,`** cubic polynomial

- **`userunits`** is a short text string giving the units in which processed data is normally reported from the instrument; for example **`C`** for Celsius, **`V`** for Volts, **`dbar`** for decibars, etc.

- **`derived`** is a flag which is either **`on`** or **`off`** to indicate whether the channel is a derived channel (**`on`**) or a measured channel (**`off`**). This is an intrinsic property of the channel **`type`**, and cannot be modified; it is for information only.

- **`gain`** reports the gain setting currently in use by the channel referenced by *`<indexorlabel>`*. In addition to one of the fixed values from the list reported by the **`availablegains`** option, the response may indicate **`auto`** for auto-ranging. In this mode the channel will select the most appropriate gain setting depending on the value of the parameter being measured. If the channel does not support multiple gain settings, the response is **none**.
  The **`gain`** option may also be used to set the gain used. For a fixed gain setting, the value supplied must be from the list reported by the **`availablegains`** option. For auto-ranging, use the word **`auto`**. Although they are typically whole numbers, gains are reported in a floating point format, and may be specified as such, as long as the value appears in the list of available gains.
- **`availablegains`** reports the gain settings supported by the sensor at channel *`<indexorlabel>`*. The settings are given as a list of numerical values in order of increasing gain, with a vertical bar **`|`** separating the values. If the channel at *`<indexorlabel>`* does not support multiple gain settings, the response is **none**.

⚠ The **`availablegains`** and **`gain`** parameters are only available for channel types which support sensors having variable gain, or multiple ranges. For the RBR*coda*^{3} instruments, this only includes the Seapoint turbidity sensor.

- **label** is a short text string without white spaces, comma, equal symbol, or special character, describing the physical parameter measured (example: **`temperature_00`**). It is reported when the channel requested is referred by its index.

- **index** is the index of the channel. It is reported when the channel requested is referred by its label.

Examples

```
>> channel 1
<< channel 1 type = temp09, module = 1, status = on, settlingtime = 140, readtime =
150, equation = tmp, userunits = C, derived = off, label = temperature_00
```

```
>> channel 2 equation userunits
<< channel 2 equation = cub, userunits = dbar
```

```
>> channel allindices type
<< channel 1 type = temp12 || 2 type = pres26
```

```
>> channel 4 availablegains
<< channel 4 availablegains = 1.0|5.0|20.0|100.0
```

```
>> channel 4 gain
<< channel 4 gain = auto
```

```
>> channel 4 gain = 20
<< channel 4 gain = 20.0
```

Errors

`Error E0107 expected argument missing`
An argument expected by the logger was not given with the command; for example, there must always be an *<indexorlabel>* argument.

`Error E0108 invalid argument to command:` ***<invalid-argument>***
This error will occur if the <index> is out of range, or if an unknown parameter is requested. Logger channels are numbered from 1 to N; zero is not valid.

`Error E0501 item is not configured`
There is a problem with the configuration of the specified channel. Contact RBR for help.

`Error E0505 no channels configured`
There was a problem reading or modifying some configuration data for the specified channel; typically in response to accessing gain control information for those channels which support it. Please contact RBR for help.

17.5.7 channels

Usage

>> channels [count | on | settlingtime | readtime | minperiod]

Description

Returns general channel information for the sensor.

It is a read-only command. The parameters currently supported are:

- **count** is the number of channels installed and configured in the sensor.
- **on** is the number of active channels, which excludes any channels turned **off** by the user. See the **status** argument to the channel command.
- **settlingtime** is the power-on settling delay in milliseconds; this is the time the instrument will wait after waking from its quiescent state, before attempting to take measurements. It allows all sensors and channel electronics to reach a stable condition. This overall settling time is determined by the longest of all the individual *active* channel delays; channels which are turned **off** do not contribute.
- **readtime** is the overall reading time in milliseconds; this is the additional time the sensor needs to acquire data from all active channels once the settling time has passed. This overall readtime is determined by the longest value of all the individual *active* channels; any which are turned **off** do not contribute.
  Most channels have a fixed, pre-determined readtime, but for some it may be variable. An example would be a channel which supports, and is configured to use, the auto-ranging feature: the readtime is longer when the channel is in auto-ranging mode than when operated in a fixed-gain mode. The instrument adjusts the reported value of the readtime to reflect the operating mode and status of all active channels.
- **minperiod** is the minimum sampling period in milliseconds with the currently active channels. This takes account of the current overall values for the settling and read times, and adds some overhead and safety margin for both fixed and per-channel activities. The value applies to the 1Hz or slower sampling modes supported by all sensors; sensors configured to support fast sampling modes as well may use selected periods less than this value.

Examples

```
>> channels
<< channels count = 2, on = 2, settlingtime = 160, readtime = 150, minperiod = 1000
```

```
>> channels settlingtime, readtime
<< channels settlingtime = 160, readtime = 150
```

Errors

Error E0108 invalid argument to command: *<invalid-argument>*
The command was given with an argument which is unrecognised or misplaced.

Error E0505 no channels configured
There is a serious fault with the instrument. Please contact RBR for help.

17.5.8 settings

Usage

```
>> settings [ fetchpoweroffdelay | inputtimeout | temperature | pressure | atmosphere |
density | salinity ]
```

Description

Reports the values of miscellaneous settings in the sensor.

The parameters applicable for the RBR*coda*^{3} instruments are:

- **`fetchpoweroffdelay`** `[= <timeoutinmilliseconds>]` is the delay in milliseconds between successful completion of a `fetch` command, and power to the front-end sensors being removed by the instrument. Power is left on for a short time to avoid excessive power cycling when sending repeated **`fetch`** commands; this parameter allows that delay to be adjusted. The default value is 8000. This parameter is only available for the RBR*coda*^{3} T.ODO instruments.
- **`inputtimeout`** `[= <timeoutinmilliseconds>]` specifies the value of a timeout in milliseconds when receiving command input. It is used to temporarily blank other output such as streamed data, and to assist in power saving by turning off the serial communication interface when it is not needed. The default value for all instruments is 10000 (10 seconds).
- **`temperature`**, **`pressure`**, **`atmosphere`**, **`density`**, **`salinity`** `[= <value>]`: are default parameter values, to be used when the instrument does not have a channel which measures the named parameter, but cross-channel calibration equations require it as an input.
  When specifying a value, any simple numeric format compatible with floating point representation may be used; for example 11, 11.000, or 1.10e+1 would all be accepted. The units of these parameter values are implicit, and *must* be as shown below. If these parameter values are never explicitly set, they will have default values based on standard seawater properties:
  - temperature in °C, default value 15.0
  - absolute pressure in dbar, default value 10.132501 (1 standard atmosphere)
  - atmospheric pressure in dbar, default value 10.132501
  - water density in g/cm^3, default value 1.026021
  - salinity in PSU, default value 35

Examples

```
>> settings atmosphere
<< settings atmosphere = 10.132501
```

```
>> settings density
<< settings density = 1.0295
```

Errors

`Error E0108 invalid argument to command: <invalid-argument>`
The command was given with an argument which is unrecognised or misplaced.

17.5.9 serial

Usage

```
>> serial [ baudrate | mode | availablemodes | availablebaudrates]
```

Description

Reports or sets the parameters which apply to the serial link.

Care must be taken if the serial link is used to change its own operating parameters. In this case, new settings are acknowledged while the old parameters are still in force, then the changes are applied. The next command sent must use the new configuration of the link if the sensor is to recognize it. In particular, changing the **mode** is not recommended unless you have discussed it with RBR first. The individual parameters are described below.

- **baudrate** [= *<baudrate>*]**:** corresponds to the baudrate of the serial link. The default baudrate from the factory is 9600.

- **mode** [= *<mode>*]**:** allows the electrical interface standard used for the serial link to be changed, available choices being listed below.

  > ❗ Contact RBR for advice before changing the **mode** parameter.
  >
  > Different modes typically require different hardware. An inappropriate change to the **mode** parameter may cripple the sensor communications.

  The most common mode is **rs232**, used on most equipment with serial ports and referred to as RS-232, EIA-232, TIA-232, or variations on one of these depending on the revision. This is the default setting typically shipped from the factory. The implementation of RS-232 on the instrument is always full duplex, with no hardware flow control lines required: **transmit**, **receive**, and **ground** are the three connections needed.
  If an instrument has been built to use one of the other interfaces, the mode will be correctly set when the instrument is shipped.

- **availablemodes:** reports the list of available modes.

- **availablebaudrates:** reports the list of available baudrates

Examples

```
>> serial
<< serial baudrate = 19200

>> serial baudrate = 115200
<< serial baudrate = 115200
```

```
>> serial mode
<< serial mode = rs232
```

```
>> serial availablebaudrates

<< serial availablebaudrates = 115200|19200|9600|4800|2400|1200
```

```
>> serial availablemodes
<< serial availablemodes = rs232
```

Errors

`Error E0108 invalid argument to command: <invalid-argument>`
The supplied argument was not a recognised parameter name, baudrate value, or mode setting.

17.5.10 getall

Usage

`>> getall`

Description

Reports all the parameters and settings in use.

It is a read-only command, which outputs the result over several lines, unlike standard commands.

The output of this command will vary from one instrument to another. The example below is just an illustration.

Example

```
>> getall
<< prompt state = off
confirmation state = on
link type = serial
E0102 invalid command 'powerinternal'
memformat type = rawbin00, newtype = rawbin00, availabletypes = rawbin00|none
settings fetchpoweroffdelay = 8000, sensorpoweralwayson = off, temperature =
15.0000, atmosphere = 10.1325010, pressure = 10.1325, density = 1.0260207,
density = 1.0260207, castdetection = off, inputtimeout = 10000
clock datetime = 20000101010815, offsetfromutc = unknown
sampling mode = continuous, period = 63, gate = none, userperiodlimit = 31,
availablefastperiods = 500|250|125|63|42|31
deployment starttime = 20000101000000, endtime = 20991231235959, status =
streaming
calibration 1 label = temperature_00, datetime = 20000401000000, c0 =
3.5000000e-003, c1 = -250.00002e-006, c2 = 2.7000000e-006, c3 = 23.000000e-009
|| calibration 2 label = pressure_00, datetime = 20000401000000, c0 =
0.0000000e+000, c1 = 1.0000000e+000, c2 = 0.0000000e+000, c3 = 0.0000000e+000,
x0 = 0.0000000e+000, x1 = 0.0000000e+000, x2 = 0.0000000e+000, x3 =
0.0000000e+000, x4 = 0.0000000e+000, x5 = 0.0000000e+000, n0 = 3
outputformat type = caltext06, availabletypes = caltext06|caltext08, labelslist
= temperature_00|pressure_00
id model = RBRcoda3, version = 1.000, serial = 092087, fwtype = 105, flavour =
rt
info pn = SEN3-M32-F35-SEC33-INT32-ST32-SP31
hwrev pcb = G, cpu = 5528H, sbsl = n/a
channels count = 2, on = 2, settlingtime = 50, readtime = 335, minperiod = 465
channel 1 type = temp12, module = 1, status = on, settlingtime = 50, readtime =
300, equation = tmp, userunits = C, gain = none, availablegains = none, derived
= off, label = temperature_00 || channel 2 type = pres26, module = 2, status =
on, settlingtime = 50, readtime = 335, equation = corr_pres2, userunits = dbar,
gain = none, availablegains = none, derived = off, label = pressure_00
sensor 1 || sensor 2 serial = H163989
```

Errors

`E0102 invalid command` ***`<unknown-command-name>`***

The response to **`getall`** may contain error messages about invalid commands or arguments and, in this case, such errors can be safely ignored. They mean that the instrument is configured with certain options disabled.

17.5.11 id

Usage

`>> id [model | version | serial | fwtype ]`

Description

Identifies the sensor.

It is a read-only command which reports the model name, the version of firmware in the CPU, the unit serial number, firmware type, and may contain the flavour of instrument (rt = realtime). The serial number is always reported using six digits, padded with leading zeroes if necessary.

Examples

```
>> id serial
<< id serial = 092087
```

```
>> id
<< id model = RBRcoda3, version = 1.000, serial = 092087, fwtype = 105, flavour = rt
```

Errors
None.

17.5.12 info

Usage
>> info [pn]

Description
Reports the RBR part number of the instrument. This is a read-only command.

Examples

```
>> info pn
<< info pn = SEN3-M32-F35-SEC33-INT32-ST32-SP31
```

Errors
None.

17.5.13 confirmation

Usage

```
>> confirmation [ state ]
```

Description

Returns the state of the sensor confirmation responses.

It is normally sent after a parameter has been modified if the state is **on**. If the state is **`off`**, successful parameter modifications occur without confirmation messages.

- **`state`** `[=` *`<state>`*`]`
  - **`on`**, the confirmation is sent
  - **`off`**, the confirmation is suppressed.

A change of the on/off state takes place immediately, so for example there will be no confirmation of the command which turns it **off**.

> ⚠ A request for information or a failed attempt to set a parameter always provoke a response, even when the state is **`off`**. For example:
> - Requests to simply report a parameter always generate output.
> - Error messages resulting from a *failed* attempt to set a parameter are always sent.
> - Some action commands always generate a confirmation message.
> - The **`Ready:`** prompt is controlled separately by the **`prompt`** command.

Turning confirmation off is not normally recommended, unless there is a very good reason for doing so. For example, if it is interfering with the parsing of responses by an automated system, it may be necessary to suppress it.

Examples

```
>> confirmation
<< confirmation state = on
>> confirmation state = off
<<
```

Confirmation of a parameter change is immediately suppressed.

```
>> confirmation
<< confirmation state = off

>> serial baudrate
<< serial baudrate = 19200
```

While the `confirmation state = off`, a request for information always provokes a response.

Errors

`Error E0108 invalid argument to command:` ***`<invalid-argument>`***

The command was given with an argument which is unrecognised or misplaced.

17.5.14 prompt

Usage

>> prompt [state]

Description

Returns the state of the **`Ready:`** prompt.

It is normally sent by the sensor in response to almost any command after any other output generated by the command is complete.

- **`state`** `[=` *`<state>`*`]`
    a. **`on`**, the prompt is sent
    b. **`off`**, the prompt is suppressed.

A change of the on/off state takes place immediately, so for example there will be no prompt following the command which turns it off. Turning the prompt off is not normally recommended, unless there is a very good reason for doing so. For example, if it is interfering with the parsing of responses by an automated system, it may be necessary to suppress it.

Examples

```
>> prompt
<< prompt state = on
```

```
>> prompt state = off
<< prompt state = off
```

Errors

`Error E0108 invalid argument to command:` ***`<invalid-argument>`***
The command was given with an argument which is unrecognised or misplaced.

17.5.15 config

Usage

`>> config restore`

Description

Restores factory configuration for the RBR*tridente.* This is a protected command.

This command requires permission. Please contact RBR before attempting to use it.

The instruments may ship with different sets of default coefficients. Use **`restore`** to recover this information.

- **`restore`** returns the sensor to factory configuration (calibration coefficients will be untouched). The sensor configuration is stored as a binary table in non-volatile memory. When a restore request is received (`config restore`), the backup configuration is read from memory and used to reconfigure the sensor as follows:
  - All sections are copied one-to-one (except for the ID)
  - The channel status (on / off) is reverted back to default
  - The channel gain (for relevant channels) is reverted back to default (automatic for the RBR*tridente*)

  The ID section is not updated. The channel calibration coefficients are not reverted.

Examples

```
>> permit command = config
<< permit command = config
>> config restore
```

Factory defaults will be restored from the backup copy.

Errors

`Error E0107 expected argument missing`
An argument expected by the instrument was not given with the command; for example, instead of **`config restore`** there was only **`config`**.

`Error E0108 invalid argument to command:` ***`<invalid-argument>`***
The command was given with an argument which is unrecognised or misplaced.

`Error E0111 command failed`
The factory configuration was not stored before restoring or the factory configuration became corrupted.

18 Revision history

Revision No.	Release Date	Notes
1.0 - H	2011-2020	Various
J	31-August-2022	Restructured the user guide to reflect Ruskin interface, added a dedicated page for RSK files, edited existing instructions for clarity, removed outdated sections and screenshots. Added RBR*tridente* to Realtime output format and OEM command reference (channel types).
K	31-December-2022	Added a section on the Search option under Support (for Mac users). Updated the RSK files and File info sections to reflect new file structure labels. Enhanced Plot window instructions. Updated the Power section. Added RBR*quadrante*, RBR*coda* Tu to Realtime output format and OEM command reference (channel types). Added the config command (restoring factory configuration) for the RBR*tridente.* Corrected typos in the User calibration section.