

# Support Features

Available in Nexus Repository OSS and Nexus Repository Pro

- [Analytics](#)
- [Logging and Log Viewer](#)
- [Metrics](#)
- [Support ZIP](#)
  - [Creating a Support ZIP](#)
- [System Information](#)

Nexus Repository Manager provides a number of features that allow you to ensure your server is configured correctly and provides you with tools to investigate details about the configuration. This information can be useful for troubleshooting and support activities.

All support features are available in the *Support* group of the *Administration* menu in the main menu section.

## Analytics

The analytics integration allows Sonatype to gather data about of your repository manager usage, since it enables the collection of event data. It collects non-sensitive information about how you are using the repository manager and allows Sonatype to achieve a better understanding of usage overall and therefore drive product innovation following your needs.

The collected information is limited the primary interaction points between your environment and the repository manager. None of the request specific data (e.g., credentials or otherwise sensitive information) is ever captured.

The data is can be useful to you from a compatibility perspective, since it gathers answers to questions such as what features are most important, where are users having difficulties, and what integrations/APIs are actively in use.

You can enable the event logging in the *Analytics* feature view available via *Analytics* menu item in the *Support* section of the *Administration* menu. Select the checkbox beside *Collect analytics* events and press the *Save* button.

You can choose to provide this data automatically to Sonatype by selecting the checkbox beside *Enable anonymized analytics submission to Sonatype*. It enables Sonatype to tailor the ongoing development of the product. Alternatively, you can submit the data manually or just use the gathered data for your own analysis only.

Once enabled, all events logged can be inspected in the *Events* feature view available via the *Analytics* section of the *Administration* menu displayed in *Figure 4.10, "List of Analytics Events"*.

### Figure 4.10. List of Analytics Events

The list of events shows the *Event type*, the *Timestamp*, the *Sequence* number and the *Duration* of the event as well as the *User* that triggered it and any *Attributes*. Each row has a + symbol in the first column that allows you to expand the row vertically. Each attribute will be expanded into a separate line allowing you to inspect all the information that is potentially submitted to Sonatype.

The *User* value is replaced by a salted hash so that no username information is transmitted. The *Anonymization Salt* is automatically randomly generated and can optionally be configured in the *Analytics: Collection* capability manually. This administration area can additionally be used to change the random identifier for the repository manager instance.

More information about capabilities can be found in [Accessing and Configuring Capabilities](#).

If you desire to further inspect the data that is potentially submitted, you can select to download the file containing the JSON files in a zip archive by clicking the *Export* button above the events list and downloading the file. The *Submit* button can be used to manually submit the events to Sonatype.

Sonatype values your input greatly and hopes you will activate the analytics feature and the automatic submission to allow us to ensure ongoing development is well aligned with your needs. In addition, Sonatype appreciates any further direct contact and feedback in person and looks forward to hearing from you.

## Logging and Log Viewer

You can configure the level of logging for the repository manager and all plugins as well as inspect the current log using the user interface with the *Logging* and the *Log Viewer* feature views. Access the *Logging* feature view displayed in *Figure 4.11, "The Logging Feature View for Configuring Loggers"* with the *Logging* menu item in the *Support* section in the *Administration* main menu.

#### Figure 4.11. The Logging Feature View for Configuring Loggers

The *Logging* feature view allows you to configure the preconfigured loggers as well as add and remove loggers. You can modify the log level for a configured logger by clicking on the *Level* value e.g., *INFO*. It will change into a drop-down of the valid levels including *OFF*, *DEFAULT*, *INFO* and others. Press the *Update* button to apply the change.

The *Create logger* button can be used to create new loggers. You will need to know the *Logger* name you want to configure. Typically this corresponds to the Java package name used in the source code. Depending on your needs you can inspect the source of Nexus Repository Manager OSS and the plugins as well as the source of your own plugins to determine the related loggers or contact Sonatype support for detailed help.

If you select a row in the list of loggers, you can delete the highlighted logger by pressing the *Delete logger* button above the list. This only applies to previously created custom loggers. To disable a default configured logger, set it to *OFF*.

When upgrading the repository manager, keep in mind that some loggers change between versions, so if you rely on specific loggers, you might have to reconfigure them.

The *Reset to default levels* button allows you to remove all your custom loggers and get back to the setup shipped with a fresh install of the repository manager.

The loggers configured in the user interface are persisted into `$data-dir/etc/logback/logback-overrides.xml` and override any logging levels configured in the main `$install-dir/etc/logback/logback.xml` file as well as other `logback-*` files. If you need to edit a logging level in those files, edit the overrides file. This will give you access to edit the configuration in the user interface at a later stage and also ensure that the values you configure take precedence.

The *ROOT* logger level controls how verbose the logging is in general. If set to *DEBUG*, logging will be very verbose, printing all log messages including debugging statements. If set to *ERROR*, logging will be far less verbose, only printing out a log statement if the repository manager encounters an error. *INFO* represents an intermediate amount of logging.

When configuring logging, keep in mind that heavy logging can have a significant performance impact on an application and any changes trigger the change to the logging immediately.

Once logging is configured as desired, you can inspect the impact of your configuration in the *Log Viewer* feature view. It allows you to copy the log from the server to your machine by pressing the *Download* button. The *Create mark* button allows you to add a custom text string into the log, so that you can create a reference point in the log file for an analysis of the file. It will insert the text you entered surrounded by *\** symbols as visible in *Figure 4.12, "Viewing the Log with an Inserted Mark"*.

#### Figure 4.12. Viewing the Log with an Inserted Mark

The *Refresh interval* configuration on the right on the top of the view allows you to configure the timing for the refresh as well as the size of the log displayed. A manual refresh can be triggered with the general refresh button in the main toolbar.

## Metrics

The *Metrics* feature view is available in the *Support* section of the *Administration* main menu. It provides insight to characteristics of the Java virtual machine JVM running the repository manager and is displayed in *Figure 4.13, "JVM Metrics"*.

#### Figure 4.13. JVM Metrics

The *Memory usage*, *Memory distribution* and *Thread states* charts provide some simple visualizations. The *Download* button allows you to retrieve a large number of properties from the JVM and download them in a JSON-formatted text file. Pressing the *Thread dump* button triggers the creation of a thread dump of the JVM and a download of the resulting text file.

## Support ZIP

The *Support ZIP* feature view allows you to create a ZIP archive file that you can submit to Sonatype support via email or a support ticket. The checkboxes in *Contents* and *Options* allow you to control the content and size of the archive.

The repository manager implements security measures when a support ZIP is generated. Sensitive password related information is removed from generated files. When a support ZIP download is attempted, you may be prompted to verify your repository manager account credentials.

Support ZIP archive files are stored on the server under the `$data-dir/downloads` directory using a name which includes the time the file was

generated.

## Creating a Support ZIP

1. Sign in to the user interface using the default admin account or any account with the *nx-admin* role.
2. Click the cog icon



in the top toolbar to open the administration interface.

3. Select *Support* and then *Support ZIP* sidebar menu items.
4. Review the options and click the *Create Support ZIP* button. A popup dialog will be shown when the support zip generation is complete.
5. Either download the support ZIP using the *Download* button or use the file path shown to retrieve the file from the `$data-dir/downloads` directory.

## System Information

The *System Information* feature view displays a large number of configuration details related to

### Nexus

details about the versions of the repository manager and the installed plugins, install and work directory location, application host and port and a number of other properties.

### Java Virtual Machine

all system properties like `java.runtime.name`, `os.name` and many more as known by the JVM running the repository manager

### Operating System

including environment variables like `JAVA_HOME` or `PATH` as well as details about the runtime in terms of processor, memory and threads, network connectors and storage file stores.

You can copy a subsection of the text from the panel or use the *Download* button to retrieve a JSON-formatted text file.