Package 'otelsdk'

November 5, 2026

Title R SDK and Exporters for OpenTelemetry

Version 0.3.0

Description OpenTelemetry is a collection of tools,
APIs, and SDKs used to instrument, generate, collect, and export
telemetry data (metrics, logs, and traces) for analysis in order to
understand your software's performance and behavior.
This package contains the OpenTelemetry SDK, and exporters.
Use this package to export traces, metrics, logs from instrumented
R code. Use the otel package to instrument your R code for
OpenTelemetry.

License MIT + file LICENSE

Encoding UTF-8

SystemRequirements cmake, protobuf-compiler, libprotobuf, libcurl,
zlib, GNU make

Imports otel (>= 0.2.0), utils

Suggests callr, cli, curl, gh, jsonlite, processx, ps, rlang,
spelling, testthat (>= 3.0.0), webfakes, withr

Config/testthat/edition 3

Config/testthat/parallel TRUE

URL https://otelsdk.r-lib.org, https://github.com/r-lib/otelsdk

Config/Needs/website tidyverse/tidytemplate

BugReports https://github.com/r-lib/otelsdk/issues

Config/roxygen2/version 8.1.1

NeedsCompilation yes

Author Gábor Csárdi [aut, cre],
Posit Software, PBC [cph, fnd] (ROR: <https://ror.org/03wc8by49>),
opentelemetry-cpp authors [ctb]

Maintainer Gábor Csárdi <csardi.gabor@gmail.com>

Repository CRAN

Date/Publication 2026-11-05 08:30:02 UTC

Contents

add_span_processor . . . 2
Collecting Telemetry Data . . . 5
Environment Variables . . . 7
File Size Options . . . 15
logger_provider_file . . . 15
logger_provider_http . . . 17
logger_provider_memory . . . 22
logger_provider_stdstream . . . 23
meter_provider_file . . . 24
meter_provider_http . . . 26
meter_provider_memory . . . 31
meter_provider_stdstream . . . 32
Time Interval Options . . . 33
tracer_provider_file . . . 34
tracer_provider_http . . . 36
tracer_provider_memory . . . 41
tracer_provider_stdstream . . . 42
with_otel_record . . . 43

Index **45**

`add_span_processor` *Add or remove a span processor*

Description

A span processor is a set of callbacks that otelsdk calls for every span, no matter which package or function created it. You can use a span processor to add attributes to spans that are created by code you do not control, e.g. to add the current user to every span that Shiny creates.

Usage

```
add_span_processor(on_start = NULL, ..., id = NULL)

remove_span_processor(id)
```

Arguments

`on_start`	Function to call when a span starts, or `NULL`. See details below.
`...`	Reserved for more callbacks in the future. Must be empty.
`id`	Id of the span processor, a string. For `add_span_processor()` it may be `NULL`, then a new id is generated. If a span processor with the same id already exists, it is replaced. For `remove_span_processor()` it is the id returned by `add_span_processor()`.

Details

Span processors are part of the OpenTelemetry SDK, not the API, so they are meant to be set up by whoever configures the telemetry output of an app, not by the authors of instrumented packages.

Span processors are global: they apply to the spans of all tracer providers.

Currently span processors only support an `on_start` callback.

`on_start`**:**

The `on_start` callback is called right after a span has started, for every span that is recording. It is called with two arguments:

- `span`: the new span, an `otel_span` object. You can call its methods, e.g. `span$set_attribute()` or `span$set_attributes()`.
- `info`: a named list with more information about the span:
  - `name`: span name.
  - `tracer_name`: name of the tracer that created the span. This is usually the name of the instrumented package.
  - `kind`: span kind, a string.
  - `parent`: the `parent` span option, if it was set explicitly when starting the span, or `NULL`.

The return value of `on_start` is ignored.

If a callback fails, otelsdk emits an `otel_error_message` message, and continues with the next span processor. The span is still created. Set the `OTEL_ENV=dev` environment variable to get errors instead.

If a callback starts a new span, span processors are not called for that span.

Setting up span processors from an environment variable:

Use the `OTEL_R_SPAN_PROCESSORS` environment variable to set up span processors that are defined in packages, without changing the code of the app. It is a comma separated list of `pkg::name` entries. Each entry must refer to a named list of callbacks, e.g.:

```
span_processor <- list(
  on_start = function(span, info) {
    span$set_attribute("deployment.region", "eu-west-1")
  }
)
```

and then

```
OTEL_R_SPAN_PROCESSORS=mypkg::span_processor
```

otelsdk reads `OTEL_R_SPAN_PROCESSORS` when the first span is started. The id of these span processors is `env:` followed by the entry, e.g. `env:mypkg::span_processor`.

If the list has callbacks that this version of otelsdk does not support, otelsdk ignores them with a message, and uses the rest of the span processor. This way a package can define callbacks for newer otelsdk versions and still work with older ones.

Example Shiny app:

This span processor adds the user and the session token to every span that starts inside a Shiny session:

```
# Add the Shiny user and session token to every span that Shiny creates
# for this app. See README.md in this directory for how to run it.

library(shiny)

otelsdk::add_span_processor(
  id = "shiny-session",
  on_start = function(span, info) {
    session <- shiny::getDefaultReactiveDomain()
    if (is.null(session)) {
      return()
    }
    user <- if (is.null(session$user)) "anonymous" else session$user
    span$set_attribute("app.user", user)
    span$set_attribute("app.session.token", session$token)
  }
)

ui <- fluidPage(
  titlePanel("Span processor example"),
...
```

See the full example app, with instructions for running it, in the `examples/shiny-span-processor` directory of the installed otelsdk package:

```
system.file(package = "otelsdk", "examples", "shiny-span-processor")
```

Value

`add_span_processor()` returns the id of the span processor, invisibly. `remove_span_processor()` returns `TRUE` if the span processor was removed and `FALSE` if there was no span processor with this id, invisibly.

Examples

```
id <- add_span_processor(
  on_start = function(span, info) {
    span$set_attribute("tracer", info$tracer_name)
  }
)
spns <- with_otel_record({
  trc <- otel::get_tracer("mytracer")
  spn <- trc$start_local_active_span("myspan")
  spn$end()
})
spns$traces$myspan$attributes
remove_span_processor(id)
```

`Collecting Telemetry Data`
Collecting Telemetry Data

Description

This page is about collecting telemetry data from instrumented R packages. If you need help with instrumenting your R package, see Getting Started in the otel package.

Value

Not applicable.

Usage

- Instrument your R package or project using the otel package. Alternatively, use Zero Code Instrumentation to automatically instrument selected packages.
- Install the otelsdk package.
- Choose an exporter from the otelsdk package. The `http` exporter sends OpenTelemetry output through OTLP/HTTP.
- Set the `OTEL_TRACES_EXPORTER` environment variable to point to the exporter of your choice. E.g. for OTLP/HTTP set `OTEL_TRACES_EXPORTER=http`.
- If you are sending telemetry data through HTTP, then you typically need to configure the URL of your OpenTelemetry collector, and you possibly also need to supply a token in an HTTP header, possibly some resource attributes. Follow the instructions of the provider of your collector. They typically don't have instructions for R, but generic instructions about environment variables will work for the otelsdk R package. E.g. for Grafana you need something like

  ```
  OTEL_EXPORTER_OTLP_PROTOCOL="http"
  OTEL_EXPORTER_OTLP_ENDPOINT="https://otlp-gateway-prod-eu-central-0.grafana.net/otlp" \
  OTEL_RESOURCE_ATTRIBUTES="service.name=<name-of-your-app>,service.namespace=<name-of-your-names
  OTEL_EXPORTER_OTLP_HEADERS="Authorization=Basic%20<base64-encoded-token>"
  ```

  See more examples below.
- See all environment variables and the manual of your chosen provider for details on configuring otelsdk exporters.
- Start R and your app. Telemetry data will be exported to the chosen exporter.

Setup

Setup for remote collectors:

There are a lot of services that offer an OpenTelemetry collector for tracers, logs and metrics, many of them supporting all three of them. There are also local apps that work as a collector. We tried otelsdk with the following ones:

Grafana:
Follow the documentation. Create an API token. You'll need to use the Grafana instrance ID as your username, and the token as the password in HTTP Basic auth. E.g. in R do this to get the base64 encoded token. `instance-id` is a (currently seven digit) number and a string with a `glc_` prefix.

```
openssl::base64_encode("<instance-id>:<api-token>")
```

Then use this encoded token to set the `Authorization` header:

```
OTEL_EXPORTER_OTLP_PROTOCOL="http"
OTEL_EXPORTER_OTLP_ENDPOINT="https://otlp-gateway-prod-eu-central-0.grafana.net/otlp" \
OTEL_EXPORTER_OTLP_HEADERS="Authorization=Basic%20<base64-encoded-token>"
OTEL_RESOURCE_ATTRIBUTES="service.name=<name-of-your-app,service.namespace=<name-of-your-namespa
```

Your endpoint URL is probably different, use the one that you see on your dashboard.
If you want to export logs and/or metrics, set these environment variables, respectively:

```
OTEL_LOGS_EXPORTER=http
OTEL_LOG_LEVEL=debug
OTEL_METRICS_EXPORTER=http
```

It also makes sense to set the desired log level.
Grafana suggests running an OpenTelemetry collector on premise instead of sending telemetry data to them directly. But nevertheless you can start out without running your own collector, they call this "quick start" mode.

Pydantic Logfire:
Create a project and a write token. Note that the URLs you need to use are different if you are within the EU! You probably need to replace `us` with `eu` in the URL if you are in the EU. Set these environment variables:

```
OTEL_TRACES_EXPORTER=http
OTEL_EXPORTER_OTLP_ENDPOINT="https://logfire-us.pydantic.dev"
OTEL_EXPORTER_OTLP_HEADERS="Authorization=<your-write-token>"
```

For logs also set `OTEL_LOGS_EXPORTER` and the desired log level:

```
OTEL_LOGS_EXPORTER=http
OTEL_LOG_LEVEL=debug
```

For exporting metrics also set

```
OTEL_METRICS_EXPORTER=http
```

Setup for local collectors:

otel-tui:
`otel-tui` is a terminal app that supports traces, logs and metrics. It is great for development, as you can keep all your telemetry local while instrumenting your package or app. Follow the installation instructions and then run the app from a terminal:

```
otel-tui
```

It listens on the default port, so the setup is very simple, set these environment variables (or a subset if you don't want metrics or logs):

```
OTEL_TRACES_EXPORTER=http
OTEL_LOGS_EXPORTER=http
OTEL_LOG_LEVEL=debug
OTEL_METRICS_EXPORTER=http
```

otel-desktop-viewer:

`otel-desktop-viewer` is similar to `otel-tui`, but has a web UI. Follow the installation instructions and start the app from a terminal:

```
otel-desktop-viewer
```

It should start a new windows or tab in your local web browser. Set the usual environment variable for your R app:

```
OTEL_TRACES_EXPORTER=http
```

Jaeger:

If you have Docker, you can start a Jaeger container on the default port:

```
docker run --rm --name jaeger \
 -p 16686:16686 \
 -p 4317:4317 \
 -p 4318:4318 \
 -p 5778:5778 \
 -p 9411:9411 \
 jaegertracing/jaeger:2.4.0
```

Go to `http://localhost:16686/` to view the Jaeger UI.

SigNoz:

To run SigNoz locally with Docker, clone the repository at https://github.com/SigNoz/signoz:

```
git clone --depth 1 https://github.com/SigNoz/signoz
```

and then run Docker Compose from the `deploy/docker/` subdirectory:

```
cd deploy/docker
docker compose up
```

Go to `http://localhost:8080` to see the SigNoz UI.

Examples

```
# See above
```

`Environment Variables` *Environment variables to configure otelsdk*

Description

See also the Environment Variables in the otel package, which is charge of selecting the exporters to use.

Value

Not applicable.

The OpenTelemetry Specification

Most of these environment variables are based on the OpenTelemetry Specification, version 1.62.0.

The environment variables with an 'OTEL_R_' prefix are not (yet) in the standard, and are specific for the otel and otelsdk R packages.

General SDK Configuration

- `OTEL_SDK_DISABLED`
  Set to a 'true' value to disable the SDK for all signals.
- `OTEL_RESOURCE_ATTRIBUTES`
  Key-value pairs to be used as resource attributes. See the Resource SDK for more details.
- `OTEL_SERVICE_NAME`
  Sets the value of the `service.name` resource attribute.
- `OTEL_LOG_LEVEL`
  Log level used by the SDK internal logger. In R it is also used for the default log level of the OpenTelemetry loggers.

Selecting Exporters

otel is responsible for selecting the providers to use for traces, logs and metrics. You can use the environment variables below to point the otel functions to the desired providers.

If none of these environment variables are set, then otel will not emit any telemetry data.

- `OTEL_TRACES_EXPORTER`
  The name of the selected tracer provider. See `otel::get_default_tracer_provider()` for the possible values.
- `OTEL_R_TRACES_EXPORTER`
  R specific version of `OTEL_TRACES_EXPORTER`.
- `OTEL_LOGS_EXPORTER`
  The name of the selected logger provider. See `otel::get_default_logger_provider()` for the possible values.
- `OTEL_R_LOGS_EXPORTER`
  R specific version of `OTEL_LOGS_EXPORTER`.
- `OTEL_METRICS_EXPORTER`
  The name of the selected meter provider. See `otel::get_default_meter_provider()` for the possible values.
- `OTEL_R_METRICS_EXPORTER`
  R specific version of `OTEL_METRICS_EXPORTER`.

Suppressing Instrumentation Scopes (R Packages)

otel has two environment variables to fine tune which instrumentation scopes (i.e. R packages, typically) emit telemetry data. By default, i.e. if neither of these are set, all packages emit telemetry data.

- `OTEL_R_EMIT_SCOPES`
  Set this environment variable to a comma separated string of instrumentation scope names or R package names to restrict telemetry to these packages only. The name of the instrumentation scope is the same as the name of the tracer, logger or meter, see `otel::default_tracer_name()`.
  You can mix package names and instrumentation scope names and you can also use wildcards (globbing). For example the value

OTEL_R_EMIT_SCOPES="org.r-lib.*,dplyr"

selects all packages with an instrumentation scope that starts with `org.r-lib.` and also dplyr.

- `OTEL_R_SUPPRESS_SCOPES`

  Set this environment variable to a comma separated string of instrumentation scope names or R package names to suppress telemetry data from these packages. The name of the instrumentation scope is the same as the name of the tracer, logger or meter, see `otel::default_tracer_name()`.

  You can mix package names and instrumentation scope names and you can also use wildcards (globbing). For example the value

  OTEL_R_SUPPRESS_SCOPES="org.r-lib.*,dplyr"

  excludes packages with an instrumentation scope that starts with `org.r-lib.` and also dplyr.

Zero Code Instrumentation

otel can instrument R packages for OpenTelemetry data collection without changing their source code. This relies on changing the code of the R functions manually using `base::trace()` and can be configured using environment variables.

- `OTEL_R_INSTRUMENT_PKGS`

  Set `OTEL_R_INSTRUMENT_PKGS` to a comma separated list of packages to instrument. The automatic instrumentation happens when the otel package is loaded, so in general it is best to set this environment variable before loading R.
- `OTEL_R_INSTRUMENT_PKGS_<pkg>_INCLUDE`

  For an automatically instrumented package, set this environment variable to only instrument a subset of its functions. It is parsed as a comma separated string of function names, which may also include `?` and `*` wildcards (globbing).
- `OTEL_R_INSTRUMENT_PKGS_<pkg>_EXCLUDE`

  For an automatically instrumented package, set this environment variable to exclude some functions from instrumentation. It has the same syntax as its `*_INCLUDE` pair. If both are set, then inclusion is applied and the exclusion.

Attribute Limits

- `OTEL_ATTRIBUTE_COUNT_LIMIT`

  Set this environment variable to limit the number of attributes for a single span, log record, metric measurement, etc. If unset, the default limit is 128 attributes. Note that only attributes specified with `otel::as_attributes()` are subject to this environment variable.
- `OTEL_ATTRIBUTE_VALUE_LENGTH_LIMIT`

  Set this environment variable to limit the length of vectors in attributes for a single span, log record, metric measurement, etc. If unset, there is no limit on the lengths of vectors in attributes. Note that only attributes specified with `otel::as_attributes()` are subject to this environment variable.

OTLP/HTTP Exporters

These environment variables are used by tracer_provider_http, meter_provider_http and logger_provider_http.

For every set of environment variables, the signal specific ones have priority over the generic one.

- `OTEL_EXPORTER_OTLP_ENDPOINT` | `OTEL_EXPORTER_OTLP_TRACES_ENDPOINT` | `OTEL_EXPORTER_OTLP_METRICS_ENDPO` | `OTEL_EXPORTER_OTLP_LOGS_ENDPOINT`

  OTLP URL to send telemetry data to. When the generic environment variable is used, the exporter appends the signal specific endpoint to it. The signal specific environment variables are used as is. More in the OpenTelemetry specification.

- `OTEL_EXPORTER_OTLP_PROTOCOL` | `OTEL_EXPORTER_OTLP_TRACES_PROTOCOL` | `OTEL_EXPORTER_OTLP_METRICS_PROTO` | `OTEL_EXPORTER_OTLP_LOGS_PROTOCOL`

  The transport protocol. Possible values: `http/json`, `http/protobuf`.

- `OTEL_R_EXPORTER_OTLP_JSON_BYTES_MAPPING` | `OTEL_R_EXPORTER_OTLP_TRACES_JSON_BYTES_MAPPING` | `OTEL_R_EXPORTER_OTLP_METRICS_JSON_BYTES_MAPPING` | `OTEL_R_EXPORTER_OTLP_LOGS_JSON_BYTES_MAPPING`

  Encoding used for trace ids and spans id. Possible values: `hexid`, `base64`, `hex`.

- `OTEL_R_EXPORTER_OTLP_USE_JSON_NAME` | `OTEL_R_EXPORTER_OTLP_TRACES_USE_JSON_NAME` | `OTEL_R_EXPORTER_OTLP_METRICS_USE_JSON_NAME` | `OTEL_R_EXPORTER_OTLP_LOGS_USE_JSON_NAME`

  Whether to use json name of protobuf field to set the key of json. A boolean value (flag, `true` or `false`).

- `OTEL_R_EXPORTER_OTLP_CONSOLE_DEBUG` | `OTEL_R_EXPORTER_OTLP_TRACES_CONSOLE_DEBUG` | `OTEL_R_EXPORTER_OTLP_METRICS_CONSOLE_DEBUG` | `OTEL_R_EXPORTER_OTLP_LOGS_CONSOLE_DEBUG`

  Whether to print debug messages to the console. A boolean value (flag, `true` or `false`).

- `OTEL_EXPORTER_OTLP_TIMEOUT` | `OTEL_EXPORTER_OTLP_TRACES_TIMEOUT` | `OTEL_EXPORTER_OTLP_METRICS_TIMEOUT` | `OTEL_EXPORTER_OTLP_LOGS_TIMEOUT`

  HTTP timeout in milliseconds.

- `OTEL_EXPORTER_OTLP_HEADERS` | `OTEL_EXPORTER_OTLP_TRACES_HEADERS` | `OTEL_EXPORTER_OTLP_METRICS_HEADERS` | `OTEL_EXPORTER_OTLP_LOGS_HEADERS`

  Additional HTTP headers to send. E.g. `Authorization` is commonly used. It must be a comma separated list of headers, each in the `header=value` form.

- `OTEL_R_EXPORTER_OTLP_SSL_INSECURE_SKIP_VERIFY` | `OTEL_R_EXPORTER_OTLP_TRACES_SSL_INSECURE_SKIP_VER` | `OTEL_R_EXPORTER_OTLP_METRICS_SSL_INSECURE_SKIP_VERIFY` | `OTEL_R_EXPORTER_OTLP_LOGS_SSL_INSECURE_S`

  Whether to disable SSL. A boolean value (flag, `true` or `false`).

- `OTEL_EXPORTER_OTLP_CERTIFICATE` | `OTEL_EXPORTER_OTLP_TRACES_CERTIFICATE` | `OTEL_EXPORTER_OTLP_METRICS` | `OTEL_EXPORTER_OTLP_LOGS_CERTIFICATE`

  CA certificate, path to a file.

- `OTEL_EXPORTER_OTLP_CERTIFICATE_STRING` | `OTEL_EXPORTER_OTLP_TRACES_CERTIFICATE_STRING` | `OTEL_EXPORTER_OTLP_METRICS_CERTIFICATE_STRING` | `OTEL_EXPORTER_OTLP_LOGS_CERTIFICATE_STRING`

  CA certificate, as a string.

- `OTEL_EXPORTER_OTLP_CLIENT_KEY` | `OTEL_EXPORTER_OTLP_TRACES_CLIENT_KEY` | `OTEL_EXPORTER_OTLP_METRICS_C` | `OTEL_EXPORTER_OTLP_LOGS_CLIENT_KEY`

  SSL client key, path to a file.

- `OTEL_EXPORTER_OTLP_CLIENT_KEY_STRING` | `OTEL_EXPORTER_OTLP_TRACES_CLIENT_KEY_STRING` | `OTEL_EXPORTER_OTLP_METRICS_CLIENT_KEY_STRING` | `OTEL_EXPORTER_OTLP_LOGS_CLIENT_KEY_STRING`

  SSL client key as a string.

- `OTEL_EXPORTER_OTLP_CLIENT_CERTIFICATE` | `OTEL_EXPORTER_OTLP_TRACES_CLIENT_CERTIFICATE` | `OTEL_EXPORTER_OTLP_METRICS_CLIENT_CERTIFICATE` | `OTEL_EXPORTER_OTLP_LOGS_CLIENT_CERTIFICATE`

  SSL client certificate, path to a file.

- `OTEL_EXPORTER_OTLP_CLIENT_CERTIFICATE_STRING` | `OTEL_EXPORTER_OTLP_TRACES_CLIENT_CERTIFICATE_STRIN` | `OTEL_EXPORTER_OTLP_METRICS_CLIENT_CERTIFICATE_STRING` | `OTEL_EXPORTER_OTLP_LOGS_CLIENT_CERTIFICAT`
  SSL client certificate, as a string.
- `OTEL_R_EXPORTER_OTLP_SSL_MIN_TLS` | `OTEL_R_EXPORTER_OTLP_TRACES_SSL_MIN_TLS` | `OTEL_R_EXPORTER_OTLP_N` | `OTEL_R_EXPORTER_OTLP_LOGS_SSL_MIN_TLS`
  Minimum TLS version.
- `OTEL_R_EXPORTER_OTLP_SSL_MAX_TLS` | `OTEL_R_EXPORTER_OTLP_TRACES_SSL_MAX_TLS` | `OTEL_R_EXPORTER_OTLP_N` | `OTEL_R_EXPORTER_OTLP_LOGS_SSL_MAX_TLS`
  Maximum TLS version.
- `OTEL_R_EXPORTER_OTLP_SSL_CIPHER` | `OTEL_R_EXPORTER_OTLP_TRACES_SSL_CIPHER` | `OTEL_R_EXPORTER_OTLP_MET` | `OTEL_R_EXPORTER_OTLP_LOGS_SSL_CIPHER`
  TLS cipher.
- `OTEL_R_EXPORTER_OTLP_SSL_CIPHER_SUITE` | `OTEL_R_EXPORTER_OTLP_TRACES_SSL_CIPHER_SUITE` | `OTEL_R_EXPORTER_OTLP_METRICS_SSL_CIPHER_SUITE` | `OTEL_R_EXPORTER_OTLP_LOGS_SSL_CIPHER_SUITE`
  TLS cipher suite.
- `OTEL_EXPORTER_OTLP_COMPRESSION` | `OTEL_EXPORTER_OTLP_TRACES_COMPRESSION` | `OTEL_EXPORTER_OTLP_METRICS` | `OTEL_EXPORTER_OTLP_LOGS_COMPRESSION`
  Compression to use. `none`, `gzip`.
- `OTEL_R_EXPORTER_OTLP_RETRY_POLICY_MAX_ATTEMPTS` | `OTEL_R_EXPORTER_OTLP_TRACES_RETRY_POLICY_MAX_ATT` | `OTEL_R_EXPORTER_OTLP_METRICS_RETRY_POLICY_MAX_ATTEMPTS` | `OTEL_R_EXPORTER_OTLP_LOGS_RETRY_POLICY_`
  The maximum number of call attempts, including the original attempt.
- `OTEL_R_EXPORTER_OTLP_RETRY_POLICY_INITIAL_BACKOFF` | `OTEL_R_EXPORTER_OTLP_TRACES_RETRY_POLICY_INIT` | `OTEL_R_EXPORTER_OTLP_METRICS_RETRY_POLICY_INITIAL_BACKOFF` | `OTEL_R_EXPORTER_OTLP_LOGS_RETRY_POLI`
  The maximum initial back-off delay between retry attempts. The actual backoff delay is uniform random between zero and this. It is in milliseconds.
- `OTEL_R_EXPORTER_OTLP_RETRY_POLICY_MAX_BACKOFF` | `OTEL_R_EXPORTER_OTLP_TRACES_RETRY_POLICY_MAX_BACK` | `OTEL_R_EXPORTER_OTLP_METRICS_RETRY_POLICY_MAX_BACKOFF` | `OTEL_R_EXPORTER_OTLP_LOGS_RETRY_POLICY_M`
  The maximum backoff places an upper limit on exponential backoff growth.
- `OTEL_R_EXPORTER_OTLP_RETRY_POLICY_BACKOFF_MULTIPLIER` | `OTEL_R_EXPORTER_OTLP_TRACES_RETRY_POLICY_E` | `OTEL_R_EXPORTER_OTLP_METRICS_RETRY_POLICY_BACKOFF_MULTIPLIER` | `OTEL_R_EXPORTER_OTLP_LOGS_RETRY_P`
  The backoff will be multiplied by this value after each retry attempt.

OTLP/FILE Exporters

These environment variables are used by tracer_provider_file, meter_provider_file and logger_provider_file.

For every set of environment variables, the signal specific ones have priority over the generic one.

- `OTEL_EXPORTER_OTLP_FILE` | `OTEL_EXPORTER_OTLP_TRACES_FILE` | `OTEL_EXPORTER_OTLP_METRICS_FILE` | `OTEL_EXPORTER_OTLP_LOGS_FILE`
  Output file pattern. May contain placeholders, see the manual pages of the providers, linked at the beginning of the section.
- `OTEL_EXPORTER_OTLP_FILE_ALIAS` | `OTEL_EXPORTER_OTLP_TRACES_FILE_ALIAS` | `OTEL_EXPORTER_OTLP_METRICS_F` | `OTEL_EXPORTER_OTLP_LOGS_FILE_ALIAS`
  The file which always point to the latest file. May contain placeholders, see the manual pages of the providers, linked at the beginning of the section.

- `OTEL_EXPORTER_OTLP_FILE_FLUSH_INTERVAL` | `OTEL_EXPORTER_OTLP_TRACES_FILE_FLUSH_INTERVAL` | `OTEL_EXPORTER_OTLP_METRICS_FILE_FLUSH_INTERVAL` | `OTEL_EXPORTER_OTLP_LOGS_FILE_FLUSH_INTERVAL`

  Interval to force flush output. A time interval specification, see Time Interval Options.

- `OTEL_EXPORTER_OTLP_FILE_FLUSH_COUNT` | `OTEL_EXPORTER_OTLP_TRACES_FILE_FILE_SIZE` | `OTEL_EXPORTER_OTLP_METRICS_FILE_FILE_SIZE` | `OTEL_EXPORTER_OTLP_LOGS_FILE_FILE_SIZE`

  Force flush output after every `flush_count` records.

- `OTEL_EXPORTER_OTLP_FILE_FILE_SIZE` | `OTEL_EXPORTER_OTLP_TRACES_FILE_FILE_SIZE` | `OTEL_EXPORTER_OTLP_METRICS_FILE_FILE_SIZE` | `OTEL_EXPORTER_OTLP_LOGS_FILE_FILE_SIZE`

  File size to rotate output files. A file size specification, see File Size Options.

- `OTEL_EXPORTER_OTLP_FILE_ROTATE_SIZE` | `OTEL_EXPORTER_OTLP_TRACES_FILE_ROTATE_SIZE` | `OTEL_EXPORTER_OTLP_METRICS_FILE_ROTATE_SIZE` | `OTEL_EXPORTER_OTLP_LOGS_FILE_ROTATE_SIZE`

  How many rotated output files to keep.

Batch Processor

These environment variables are used by tracer_provider_http, and logger_provider_http.

- `OTEL_BSP_SCHEDULE_DELAY`

  The maximum buffer/queue size. After the size is reached, spans are dropped. Must be positive.

- `OTEL_BSP_MAX_QUEUE_SIZE`

  The maximum batch size of every export. It must be smaller or equal to max_queue_size. Must be positive.

- `OTEL_BSP_MAX_EXPORT_BATCH_SIZE`

  The time interval between two consecutive exports, in milliseconds.

Metric Reader

These environment variables are used by meter_provider_http, meter_provider_stdstream meter_provider_memory and meter_provider_file.

- `OTEL_METRIC_EXPORT_INTERVAL`

  The time interval between the start of two export attempts, in milliseconds.

- `OTEL_METRIC_EXPORT_TIMEOUT`

  Maximum allowed time to export data, in milliseconds.

Metric Exporters

These environment variables are used by meter_provider_http and meter_provider_memory.

- `OTEL_R_EXPORTER_OTLP_AGGREGATION_TEMPORALITY`

  Aggregation temporality. Possible values: `unspecified`, `delta`, `cumulative`, `lowmemory`. See the OpenTelemetry data model.

Standard Stream Exporters

These environment variables are used by tracer_provider_stdstream, meter_provider_stdstream and logger_provider_stdstream.

The signal specific environment variables have priority over the generic one.

- `OTEL_R_EXPORTER_STDSTREAM_OUTPUT` | `OTEL_R_EXPORTER_STDSTREAM_TRACES_OUTPUT` | `OTEL_R_EXPORTER_STDSTR` | `OTEL_R_EXPORTER_STDSTREAM_LOGS_OUTPUT`

  Where to write the output. Can be

  - `stdout`: write output to the standard output,
  - `stderr`: write output to the standard error,
  - another string: write output to a file. (To write output to a file named `stdout` or `"stderr`, use a `./` prefix.)

In-Memory Exporters

These environment variables are used by tracer_provider_memory, meter_provider_memory and logger_provider_memory.

The signal specific environment variables have priority over the generic one.

- `OTEL_R_EXPORTER_MEMORY_BUFFER_SIZE` | `OTEL_R_EXPORTER_MEMORY_TRACES_BUFFER_SIZE` | `OTEL_R_EXPORTER_MEMORY_METRICS_BUFFER_SIZE` | `OTEL_R_EXPORTER_MEMORY_LOGS_BUFFER_SIZE` | `OTEL_R_EXPORTER_MEMORY_BUFFER_SIZE`

  Buffer size, this is the maximum number of spans, metrics measurements or log records that the provider can record.

Span Processors

- `OTEL_R_SPAN_PROCESSORS`

  Comma separated list of span processors to add, in the `pkg::name` form. Each entry must refer to a named list of callbacks. See `add_span_processor()` for details.

Support Matrix of all OpenTelemetry Environment Variables for R

Name	*Supported*
OTEL_SDK_DISABLED	+
OTEL_RESOURCE_ATTRIBUTES	+
OTEL_SERVICE_NAME	+
OTEL_LOG_LEVEL	+
OTEL_PROPAGATORS	-
OTEL_BSP_SCHEDULE_DELAY	+
OTEL_BSP_EXPORT_TIMEOUT	-
OTEL_BSP_MAX_QUEUE_SIZE	+
OTEL_BSP_MAX_EXPORT_BATCH_SIZE	+
OTEL_EXPORTER_OTLP_ENDPOINT	+
OTEL_EXPORTER_OTLP_*_ENDPOINT	+
OTEL_EXPORTER_OTLP_INSECURE	-
OTEL_EXPORTER_OTLP_*_INSECURE	-

OTEL_EXPORTER_OTLP_CERTIFICATE	+
OTEL_EXPORTER_OTLP_*_CERTIFICATE	+
OTEL_EXPORTER_OTLP_CLIENT_KEY	+
OTEL_EXPORTER_OTLP_*_CLIENT_KEY	+
OTEL_EXPORTER_OTLP_CLIENT_CERTIFICATE	+
OTEL_EXPORTER_OTLP_CLIENT_*_CERTIFICATE	+
OTEL_EXPORTER_OTLP_HEADERS	+
OTEL_EXPORTER_OTLP_*_HEADERS	+
OTEL_EXPORTER_OTLP_COMPRESSION	+
OTEL_EXPORTER_OTLP_*_COMPRESSION	+
OTEL_EXPORTER_OTLP_PROTOCOL	+
OTEL_EXPORTER_OTLP_*_PROTOCOL	+
OTEL_EXPORTER_OTLP_SPAN_INSECURE (obsolete)	-
OTEL_EXPORTER_OTLP_METRIC_INSECURE (obsolete)	-
OTEL_EXPORTER_ZIPKIN_ENDPOINT	-
OTEL_EXPORTER_ZIPKIN_TIMEOUT	-
OTEL_EXPORTER_PROMETHEUS_HOST	-
OTEL_EXPORTER_PROMETHEUS_PORT	-
OTEL_TRACES_EXPORTER	+
OTEL_METRICS_EXPORTER	+
OTEL_LOGS_EXPORTER	+
OTEL_SPAN_ATTRIBUTE_COUNT_LIMIT	-
OTEL_SPAN_ATTRIBUTE_VALUE_LENGTH_LIMIT	-
OTEL_SPAN_EVENT_COUNT_LIMIT	-
OTEL_SPAN_LINK_COUNT_LIMIT	-
OTEL_EVENT_ATTRIBUTE_COUNT_LIMIT	-
OTEL_LINK_ATTRIBUTE_COUNT_LIMIT	-
OTEL_LOGRECORD_ATTRIBUTE_COUNT_LIMIT	-
OTEL_LOGRECORD_ATTRIBUTE_VALUE_LENGTH_LIMIT	-
OTEL_TRACES_SAMPLER	-
OTEL_TRACES_SAMPLER_ARG	-
OTEL_ATTRIBUTE_VALUE_LENGTH_LIMIT (1)	+
OTEL_ATTRIBUTE_COUNT_LIMIT (1)	+
OTEL_METRIC_EXPORT_INTERVAL	+
OTEL_METRIC_EXPORT_TIMEOUT	+
OTEL_METRICS_EXEMPLAR_FILTER	-
OTEL_EXPORTER_OTLP_METRICS_TEMPORALITY_PREFERENCE	-
OTEL_EXPORTER_OTLP_METRICS_DEFAULT_HISTOGRAM_AGGREGATION	-
OTEL_EXPERIMENTAL_CONFIG_FILE	-
OTEL_CONFIG_FILE	-

(1) In `otel::as_attributes()`.

See Also

Environment Variables in otel

Examples

```
# To start an R session using the OTLP exporter:
# OTEL_TRACES_EXPORTER=http R -q -f script.R
```

`File Size Options` *File Size Options*

Description

otel and otelsdk accept file size options in the following format:

- As a positive numeric scalar, interpreted as number of bytes.
- A string scalar with a positive number and a unit suffix. Possible units: B, KB, KiB, MB, MiB, GB, GiB, TB, TiB, PB, PiB. Units are case insensitive.

Value

Not applicable.

Examples

```
# Maximum output file size is 128 MiB:
# OTEL_EXPORTER_OTLP_FILE_FILE_SIZE=128MiB
```

`logger_provider_file` *Logger provider to write log messages into a JSONL file.*

Description

This is the OTLP file exporter. It writes logs to a JSONL file, each log is a line in the file, a valid JSON value. The line separator is `\n`. The preferred file extension is `jsonl`.

Select this tracer provider with `OTEL_LOGS_EXPORTER=otlp/file`.

Select `OTEL_LOGS_EXPORTER=otlp/stdout` to write to the standard output instead of a file. This is the same as setting the `file_pattern` option to `stdout`.

Value

`logger_provider_file$new()` returns an otel::otel_logger_provider object.

`logger_provider_file$options()` returns a named list, the current values of the options.

Usage

Externally:

```
OTEL_LOGS_EXPORTER=otlp/file
```

From R:

```
logger_provider_file$new(opts = NULL)
logger_provider_file$options()
```

Arguments

- `opts`: Named list of options. See below.

Options

File exporter options:

- `file_pattern`: Output file pattern. Value is set from:
  - `opts` argument, or
  - `OTEL_EXPORTER_OTLP_LOGS_FILE` environment variable, or
  - `OTEL_EXPORTER_OTLP_FILE` environment variable, or
  - the default is: `logs-%N.jsonl`.

  May contain placeholders, see below.
  If it is `stdout` or `stderr`, then the output is written to the standard output or standard error, instead of a file. In this case the other options are ignored.
- `alias_pattern`: The file which always point to the latest file. Value is set from:
  - `opts` argument, or
  - `OTEL_EXPORTER_OTLP_LOGS_FILE_ALIAS` environment variable, or
  - `OTEL_EXPORTER_OTLP_FILE_ALIAS environment variable`, or
  - the default is: `logs-latest.jsonl`.

  If `file_pattern` is set, but `alias_pattern` is not, then the default alias is created from `file_pattern`: the last group of placeholders in the file name is replaced by `latest`, and all other placeholders are removed, e.g. `trace-%F-%T-%N.jsonl` becomes `trace-latest.jsonl`.
  May contain placeholders, see below.
- `flush_interval`: Interval to force flush output. A time interval specification, see Time Interval Options. Value is set from
  - `opts` argument, or
  - `OTEL_EXPORTER_OTLP_LOGS_FILE_FLUSH_INTERVAL` environment variable, or
  - `OTEL_EXPORTER_OTLP_FILE_FLUSH_INTERVAL` environment variable, or
  - the default is `30s`, thirty seconds.
- `flush_count`: Force flush output after every `flush_count` records. Value is set from
  - `opts` argument, or
  - `OTEL_EXPORTER_OTLP_LOGS_FILE_FLUSH_COUNT` environment variable, or
  - `OTEL_EXPORTER_OTLP_FILE_FLUSH_COUNT` environment variable, or

  - the default is `256`.
- `file_size`: File size to rotate output files. A file size specification, see File Size Options. Value is set from
  - `opts` argument, or
  - `OTEL_EXPORTER_OTLP_LOGS_FILE_FILE_SIZE` environment variable, or
  - `OTEL_EXPORTER_OTLP_FILE_FILE_SIZE` environment variable, or
  - the default is `20MB`.
- `rotate_size`: How many rotated output files to keep. Value is set from
  - `opts` argument, or
  - `OTEL_EXPORTER_OTLP_LOGS_FILE_ROTATE_SIZE` environment variable, or
  - `OTEL_EXPORTER_OTLP_FILE_ROTATE_SIZE` environment variable, or
  - the default is `10`.

Special placeholders are available for `file_pattern` and `alias_pattern`:

- `%Y`: Writes year as a 4 digit decimal number.
- `%y`: Writes last 2 digits of year as a decimal number (range `[00,99]`).
- `%m`: Writes month as a decimal number (range `[01,12]`).
- `%j`: Writes day of the year as a decimal number (range `[001,366]`).
- `%d`: Writes day of the month as a decimal number (range `[01,31]`).
- `%w`: Writes weekday as a decimal number, where Sunday is 0 (range `[0-6]`).
- `%H`: Writes hour as a decimal number, 24 hour clock (range `[00-23]`).
- `%I`: Writes hour as a decimal number, 12 hour clock (range `[01,12]`).
- `%M`: Writes minute as a decimal number (range `[00,59]`).
- `%S`: Writes second as a decimal number (range `[00,60]`).
- `%F`: Equivalent to `%Y-%m-%d` (the ISO 8601 date format).
- `%T`: Equivalent to `%H:%M:%S` (the ISO 8601 time format).
- `%R`: Equivalent to `%H:%M`.
- `%N`: Rotate index, start from 0.
- `%n`: Rotate index, start from 1.

Examples

```
logger_provider_file$options()
```

---

logger_provider_http *Logger provider to log over HTTP*

Description

This is the OTLP HTTP exporter.

Value

`logger_provider_http$new()` returns an otel::otel_logger_provider object.

`logger_provider_http$options()` returns a named list, the current values for all options.

Usage

Externally:

```
OTEL_LOGS_EXPORTER=otlp
```

From R:

```
logger_provider_http$new(opts = NULL)
logger_provider_http$options()
```

Arguments

- `opts`: Named list of options. See below.

Options

HTTP exporter options:

- `url`: OTLP URL to send telemetry data to. Value is set from
  - the `opts` argument, needs to point to the logs endpoint, or
  - `OTEL_EXPORTER_OTLP_LOGS_ENDPOINT` environment variable, needs to point to the logs endpoint, or
  - `OTEL_EXPORTER_OTLP_ENDPOINT` environment variable + `/v1/logs`, or
  - the default is `http://localhost:4318/v1/logs`.
- `content_type`: data format used, JSON or binary. Possible values: `"http/json"`, `"http/protobuf"`. Value if set from
  - the `opts` argument, or
  - the `OTEL_EXPORTER_OTLP_LOGS_PROTOCOL` environment variable, or
  - the `OTEL_EXPORTER_OTLP_PROTOCOL` environment variable, or
  - the default is `"http/protobuf"`.
- `json_bytes_mapping`: encoding used for trace id and span id. Possible values: `"hexid"`, `"base64"`, `"hex"`. Value is set from
  - the `opts` argument, or
  - the `OTEL_R_EXPORTER_OTLP_LOGS_JSON_BYTES_MAPPING` environment variable, or
  - the `OTEL_R_EXPORTER_OTLP_JSON_BYTES_MAPPING` environment variable, or
  - the default is `"hexid"`.
- `use_json_name`: whether to use json name of protobuf field to set the key of json. A flag. Value is set from
  - the `opts` argument, or
  - the `OTEL_R_EXPORTER_OTLP_LOGS_USE_JSON_NAME` environment variable, or
  - the `OTEL_R_EXPORTER_OTLP_USE_JSON_NAME` environment variable, or
  - the default is `FALSE`.
- `console_debug`: whether to print debug messages to the console. Value is set from
  - the `opts` argument, or
  - the `OTEL_R_EXPORTER_OTLP_LOGS_CONSOLE_DEBUG` environment variable, or

  - the OTEL_R_EXPORTER_OTLP_CONSOLE_DEBUG environment variable, or
  - the default is FALSE.
- timeout: HTTP timeout in milliseconds. Value is set from
  - the opts argument, or
  - the OTEL_EXPORTER_OTLP_LOGS_TIMEOUT} environment variable, or
  - the OTEL_EXPORTER_OTLP_TIMEOUT environment variable, or
  - the default is 10000.
- http_headers: additional HTTP headers to send, e.g. Authorization is commonly used. A named character vector without NA values. Value is set from
  - the opts argument, or
  - the OTEL_EXPORTER_OTLP_LOGS_HEADERS environment variable, or
  - the OTEL_EXPORTER_OTLP_HEADERS environment variable, or
  - the default is an empty named character vector. When specified in environment variables, it must be a comma separated list of headers, each in the header=value form.
- ssl_insecure_skip_verify: whether to disable SSL. Value is set from
  - the opts argument, or
  - the OTEL_R_EXPORTER_OTLP_LOGS_SSL_INSECURE_SKIP_VERIFY environment variable, or
  - the OTEL_R_EXPORTER_OTLP_SSL_INSECURE_SKIP_VERIFY environment variable, or
  - the default is FALSE.
- ssl_ca_cert_path: CA certificate, path to a file. Empty string uses the system default. Value is set from
  - the opts argument, or
  - the OTEL_EXPORTER_OTLP_LOGS_CERTIFICATE environment variable, or
  - the OTEL_EXPORTER_OTLP_CERTIFICATE environment variable, or
  - the default is ‘’.
- ssl_ca_cert_string: CA certificate, as a string. Empty string uses the system default. Value is set from
  - the opts argument, or
  - the OTEL_EXPORTER_OTLP_LOGS_CERTIFICATE_STRING environment variable, or
  - the OTEL_EXPORTER_OTLP_CERTIFICATE_STRING environment variable, or
  - the default is ‘’.
- ssl_client_key_path: SSL client key, path to a file. Empty string uses the system default. Value is set from
  - the opts argument, or
  - the OTEL_EXPORTER_OTLP_LOGS_CLIENT_KEY environment variable, or
  - the OTEL_EXPORTER_OTLP_CLIENT_KEY environment variable, or
  - the default is ‘’.
- ssl_client_key_string: SSL client key as a string. Empty string uses the system default. Value is set from
  - the opts argument, or
  - the OTEL_EXPORTER_OTLP_LOGS_CLIENT_KEY_STRING environment variable, or
  - the OTEL_EXPORTER_OTLP_CLIENT_KEY_STRING environment variable, or

  - the default is ‘’.
- `ssl_client_cert_path`: SSL client certificate, path to a file. Empty string uses the system default. Value is set from
  - the `opts` argument, or
  - the `OTEL_EXPORTER_OTLP_LOGS_CLIENT_CERTIFICATE` environment variable, or
  - the `OTEL_EXPORTER_OTLP_CLIENT_CERTIFICATE` environment variable, or
  - the default is ‘’.
- `ssl_client_cert_string`: SSL client certificate, as a string. Empty string uses the system default. Value is set from
  - the `opts` argument, or
  - the `OTEL_EXPORTER_OTLP_LOGS_CLIENT_CERTIFICATE_STRING` environment variable, or
  - the `OTEL_EXPORTER_OTLP_CLIENT_CERTIFICATE_STRING` environment variable, or
  - the default is ‘’.
- `ssl_min_tls`: minimum TLS version. Empty string uses the system default. Value is set from
  - the `opts` argument, or
  - the `OTEL_R_EXPORTER_OTLP_LOGS_SSL_MIN_TLS` environment variable, or
  - the `OTEL_R_EXPORTER_OTLP_SSL_MIN_TLS` environment variable, or
  - the default is ‘’.
- `ssl_max_tls`: maximum TLS version. Empty string uses the system default. Value is set from
  - the `opts` argument, or
  - the `OTEL_R_EXPORTER_OTLP_LOGS_SSL_MAX_TLS` environment variable, or
  - the `OTEL_R_EXPORTER_OTLP_SSL_MAX_TLS` environment variable, or
  - the default is ‘’.
- `ssl_cipher`: TLS cipher. Empty string uses the system default. Value is set from
  - the `opts` argument, or
  - the `OTEL_R_EXPORTER_OTLP_LOGS_SSL_CIPHER` environment variable, or
  - the `OTEL_R_EXPORTER_OTLP_SSL_CIPHER` environment variable, or
  - the default is ‘’.
- `ssl_cipher_suite`: TLS cipher suite. Empty string uses the system default. Value is set from
  - the `opts` argument, or
  - the `OTEL_R_EXPORTER_OTLP_LOGS_SSL_CIPHER_SUITE` environment variable, or
  - the `OTEL_R_EXPORTER_OTLP_SSL_CIPHER_SUITE` environment variable, or
  - the default is ‘’.
- `compression`: compression to use. Possible values are `"none"`, `"gzip"`. Value is the set from
  - the `opts` argument, or
  - the `OTEL_EXPORTER_OTLP_LOGS_COMPRESSION` environment variable, or
  - the `OTEL_EXPORTER_OTLP_COMPRESSION` environment variable, or
  - the default is `none`.

- `retry_policy_max_attempts`: the maximum number of call attempts, including the original attempt. Value is set from
  - the `opts` argument, or
  - the `OTEL_R_EXPORTER_OTLP_LOGS_RETRY_POLICY_MAX_ATTEMPTS` environment variable, or
  - the `OTEL_R_EXPORTER_OTLP_RETRY_POLICY_MAX_ATTEMPTS` environment variable, or
  - the default is `5`.
- `retry_policy_initial_backoff`: the maximum initial back-off delay between retry attempts. The actual backoff delay is uniform random between zero and this. Value is set from
  - the `opts` argument, or
  - the `OTEL_R_EXPORTER_OTLP_LOGS_RETRY_POLICY_INITIAL_BACKOFF` environment variable, or
  - the `OTEL_R_EXPORTER_OTLP_RETRY_POLICY_INITIAL_BACKOFF` environment variable, or
  - the default is `1000`.
- `retry_policy_max_backoff`: the maximum backoff places an upper limit on exponential backoff growth. Value is set from
  - the `opts` argument, or
  - the `OTEL_R_EXPORTER_OTLP_LOGS_RETRY_POLICY_MAX_BACKOFF` environment variable, or
  - the `OTEL_R_EXPORTER_OTLP_RETRY_POLICY_MAX_BACKOFF` environment variable, or
  - the default is `5000`.
- `retry_policy_backoff_multiplier`: the backoff will be multiplied by this value after each retry attempt. Value is set from
  - the `opts` argument, or
  - the `OTEL_R_EXPORTER_OTLP_LOGS_RETRY_POLICY_BACKOFF_MULTIPLIER` environment variable, or
  - the `OTEL_R_EXPORTER_OTLP_RETRY_POLICY_BACKOFF_MULTIPLIER` environment variable, or
  - the default is `1.5`.

Batch processor options:

- `max_queue_size`: The maximum buffer/queue size. After the size is reached, spans are dropped. Must be positive. Value is set from
  - the `opts` argument, or
  - the `OTEL_BSP_MAX_QUEUE_SIZE` environment variable, or
  - the default is `2048`.
- `max_export_batch_size`: the maximum batch size of every export. It must be smaller or equal to max_queue_size. Must be positive. Value is set from
  - the `opts` argument, or
  - the `OTEL_BSP_MAX_EXPORT_BATCH_SIZE` environment variable, or
  - the default is `512`.
- `schedule_delay`: the time interval between two consecutive exports, in milliseconds. Value is set from

  - the `opts` argument‘ or
  - the `OTEL_BSP_SCHEDULE_DELAY` environment variable, or
  - the default is `1000`.

Examples

```
logger_provider_http$options()
```

`logger_provider_memory`

In-memory logger provider for testing

Description

Collects log records in memory. This is useful for testing your instrumented R package or application.

Value

`logger_provider_memory$new()` returns an otel::otel_logger_provider object. `lp$get_logs()` returns an unnamed list of recorded log records, in the order they were emitted. If the buffer is full, then the oldest log records are dropped.

`logger_provider_memory$options()` returns a named list, the current values for all options.

Usage

```
lp <- logger_provider_memory$new(opts = NULL)
lp$get_logs()
logger_provider_memory$options()
```

`lp$get_logs()` erases the internal buffer of the logger provider.

Arguments

- `opts`: Named list of options. See below.

Options

Memory exporter options:

- `buffer_size`: buffer size, this is the maximum number of spans or metrics measurements that the provider can record. Must be positive. Value is set from
  - the `opts` argument, or
  - the `OTEL_R_EXPORTER_MEMORY_LOGS_BUFFER_SIZE` environment variable, or
  - the `OTEL_R_EXPORTER_MEMORY_BUFFER_SIZE` environment variable, or
  - the default is `100`.

Examples

```
logger_provider_memory$options()
```

`logger_provider_stdstream` *Logger provider to write to the standard output or standard error or to a file*

Description

Writes logs to the standard output or error, or to a file. Useful for debugging.

Value

`logger_provider_stdstream$new()` returns an otel::otel_logger_provider object.

`logger_provider_stdstream$options()` returns a named list, the current values of the options.

Usage

Externally:

```
OTEL_LOGS_EXPORTER=console
OTEL_LOGS_EXPORTER=stderr
```

From R:

```
logger_provider_stdstream$new(opts = NULL)
logger_provider_stdstream$options()
```

Arguments

`opts`: Named list of options. See below.

Options

Standard stream exporter options:

- `output`: where to write the output. Can be
  - `"stdout"`: write output to the standard output,
  - `"stderr"`: write output to the standard error,
  - another string: write output to a file. (To write output to a file named `"stdout"` or `"stderr"`, use a `./` prefix.)

  Value is set from
  - the `opts` argument, or
  - the `OTEL_R_EXPORTER_STDSTREAM_LOGS_OUTPUT` environment variable, or
  - the `OTEL_R_EXPORTER_STDSTREAM_OUTPUT` environment variable, or
  - the default is `"stdout"`.

Examples

```
logger_provider_stdstream$options()
```

`meter_provider_file` *Meter provider to collect metrics in JSONL files*

Description

This is the OTLP file exporter. It writes measurements to a JSONL file, each measurement is a line in the file, a valid JSON value. The line separator is `\n`. The preferred file extension is `jsonl`.

Select this tracer provider with `OTEL_METRICS_EXPORTER=otlp/file`.

Select `OTEL_METRICS_EXPORTER=otlp/stdout` to write to the standard output instead of a file. This is the same as setting the `file_pattern` option to `stdout`.

Value

`meter_provider_file$new()` returns an otel::otel_meter_provider object.

`meter_provider_file$options()` returns a named list, the current values of the options.

Usage

Externally:

```
OTEL_METRICS_EXPORTER=otlp/file
```

From R:

```
meter_provider_file$new(opts = NULL)
meter_provider_file$options()
```

Arguments

- `opts`: Named list of options. See below.

Options

File exporter options:

- `file_pattern`: Output file pattern. Value is set from:
  - `opts` argument, or
  - `OTEL_EXPORTER_OTLP_METRICS_FILE` environment variable, or
  - `OTEL_EXPORTER_OTLP_FILE` environment variable, or
  - the default is: `metrics-%N.jsonl`.

  May contain placeholders, see below.
  If it is `stdout` or `stderr`, then the output is written to the standard output or standard error, instead of a file. In this case the other options are ignored.
- `alias_pattern`: The file which always point to the latest file. Value is set from:
  - `opts` argument, or
  - `OTEL_EXPORTER_OTLP_METRICS_FILE_ALIAS` environment variable, or

  - `OTEL_EXPORTER_OTLP_FILE_ALIAS` environment variable, or
  - the default is: `metrics-latest.jsonl`.

  If `file_pattern` is set, but `alias_pattern` is not, then the default alias is created from `file_pattern`: the last group of placeholders in the file name is replaced by `latest`, and all other placeholders are removed, e.g. `trace-%F-%T-%N.jsonl` becomes `trace-latest.jsonl`. May contain placeholders, see below.
- `flush_interval`: Interval to force flush output. A time interval specification, see Time Interval Options. Value is set from
  - `opts` argument, or
  - `OTEL_EXPORTER_OTLP_METRICS_FILE_FLUSH_INTERVAL` environment variable, or
  - `OTEL_EXPORTER_OTLP_FILE_FLUSH_INTERVAL` environment variable, or
  - the default is `30s`, thirty seconds.
- `flush_count`: Force flush output after every `flush_count` records. Value is set from
  - `opts` argument, or
  - `OTEL_EXPORTER_OTLP_METRICS_FILE_FLUSH_COUNT` environment variable, or
  - `OTEL_EXPORTER_OTLP_FILE_FLUSH_COUNT` environment variable, or
  - the default is `256`.
- `file_size`: File size to rotate output files. A file size specification, see File Size Options. Value is set from
  - `opts` argument, or
  - `OTEL_EXPORTER_OTLP_METRICS_FILE_FILE_SIZE` environment variable, or
  - `OTEL_EXPORTER_OTLP_FILE_FILE_SIZE` environment variable, or
  - the default is `20MB`.
- `rotate_size`: How many rotated output files to keep. Value is set from
  - `opts` argument, or
  - `OTEL_EXPORTER_OTLP_METRICS_FILE_ROTATE_SIZE` environment variable, or
  - `OTEL_EXPORTER_OTLP_FILE_ROTATE_SIZE` environment variable, or
  - the default is `10`.

Special placeholders are available for `file_pattern` and `alias_pattern`:

- `%Y`: Writes year as a 4 digit decimal number.
- `%y`: Writes last 2 digits of year as a decimal number (range `[00,99]`).
- `%m`: Writes month as a decimal number (range `[01,12]`).
- `%j`: Writes day of the year as a decimal number (range `[001,366]`).
- `%d`: Writes day of the month as a decimal number (range `[01,31]`).
- `%w`: Writes weekday as a decimal number, where Sunday is 0 (range `[0-6]`).
- `%H`: Writes hour as a decimal number, 24 hour clock (range `[00-23]`).
- `%I`: Writes hour as a decimal number, 12 hour clock (range `[01,12]`).
- `%M`: Writes minute as a decimal number (range `[00,59]`).
- `%S`: Writes second as a decimal number (range `[00,60]`).
- `%F`: Equivalent to `%Y-%m-%d` (the ISO 8601 date format).
- `%T`: Equivalent to `%H:%M:%S` (the ISO 8601 time format).
- `%R`: Equivalent to `%H:%M`.

- `%N`: Rotate index, start from 0.
- `%n`: Rotate index, start from 1.

Metric reader options:

- `export_interval`: the time interval between the start of two export attempts, in milliseconds. Value is set from
  - the `opts` argument, or
  - the `OTEL_METRIC_EXPORT_INTERVAL` environment variable, or
  - the default is `60000`.
- `export_timeout`: Maximum allowed time to export data, in milliseconds. Value is set from
  - the `opts` argument, or
  - the `OTEL_METRIC_EXPORT_TIMEOUT` environment variable, or
  - the default is `30000`.

Examples

```
meter_provider_file$options()
```

`meter_provider_http` *Meter provider to send collected metrics over HTTP*

Description

This is the OTLP HTTP exporter.

Select this tracer provider with `OTEL_METRICS_EXPORTER=otlp`.

Value

`meter_provider_http$new()` returns an otel::otel_meter_provider object.

`meter_provider_http$options()` returns a named list, the current values for all options.

Usage

Externally:

```
OTEL_METRICS_EXPORTER=otlp
```

From R:

```
meter_provider_http$new(opts = NULL)
meter_provider_http$options()
```

Arguments

- `opts`: Named list of options. See below.

Options

HTTP exporter options:

- `url`: OTLP URL to send telemetry data to. Value is set from
  - the `opts` argument, needs to point to the metrics endpoint, or
  - `OTEL_EXPORTER_OTLP_METRICS_ENDPOINT` environment variable, needs to point to the metrics endpoint, or
  - `OTEL_EXPORTER_OTLP_ENDPOINT` environment variable + `/v1/metrics`, or
  - the default is `http://localhost:4318/v1/metrics`.
- `content_type`: data format used, JSON or binary. Possible values: `"http/json"`, `"http/protobuf"`. Value if set from
  - the `opts` argument, or
  - the `OTEL_EXPORTER_OTLP_METRICS_PROTOCOL` environment variable, or
  - the `OTEL_EXPORTER_OTLP_PROTOCOL` environment variable, or
  - the default is `"http/protobuf"`.
- `json_bytes_mapping`: encoding used for trace id and span id. Possible values: `"hexid"`, `"base64"`, `"hex"`. Value is set from
  - the `opts` argument, or
  - the `OTEL_R_EXPORTER_OTLP_METRICS_JSON_BYTES_MAPPING` environment variable, or
  - the `OTEL_R_EXPORTER_OTLP_JSON_BYTES_MAPPING` environment variable, or
  - the default is `"hexid"`.
- `use_json_name`: whether to use json name of protobuf field to set the key of json. A flag. Value is set from
  - the `opts` argument, or
  - the `OTEL_R_EXPORTER_OTLP_METRICS_USE_JSON_NAME` environment variable, or
  - the `OTEL_R_EXPORTER_OTLP_USE_JSON_NAME` environment variable, or
  - the default is `FALSE`.
- `console_debug`: whether to print debug messages to the console. Value is set from
  - the `opts` argument, or
  - the `OTEL_R_EXPORTER_OTLP_METRICS_CONSOLE_DEBUG` environment variable, or
  - the `OTEL_R_EXPORTER_OTLP_CONSOLE_DEBUG` environment variable, or
  - the default is `FALSE`.
- `timeout`: HTTP timeout in milliseconds. Value is set from
  - the `opts` argument, or
  - the `OTEL_EXPORTER_OTLP_METRICS_TIMEOUT}` environment variable, or
  - the `OTEL_EXPORTER_OTLP_TIMEOUT` environment variable, or
  - the default is `10000`.
- `http_headers`: additional HTTP headers to send, e.g. `Authorization` is commonly used. A named character vector without `NA` values. Value is set from
  - the `opts` argument, or
  - the `OTEL_EXPORTER_OTLP_METRICS_HEADERS` environment variable, or
  - the `OTEL_EXPORTER_OTLP_HEADERS` environment variable, or
  - the default is an empty named character vector. When specified in environment variables, it must be a comma separated list of headers, each in the `header=value` form.

- `ssl_insecure_skip_verify`: whether to disable SSL. Value is set from
  - the `opts` argument, or
  - the `OTEL_R_EXPORTER_OTLP_METRICS_SSL_INSECURE_SKIP_VERIFY` environment variable, or
  - the `OTEL_R_EXPORTER_OTLP_SSL_INSECURE_SKIP_VERIFY` environment variable, or
  - the default is `FALSE`.
- `ssl_ca_cert_path`: CA certificate, path to a file. Empty string uses the system default. Value is set from
  - the `opts` argument, or
  - the `OTEL_EXPORTER_OTLP_METRICS_CERTIFICATE` environment variable, or
  - the `OTEL_EXPORTER_OTLP_CERTIFICATE` environment variable, or
  - the default is ‘’.
- `ssl_ca_cert_string`: CA certificate, as a string. Empty string uses the system default. Value is set from
  - the `opts` argument, or
  - the `OTEL_EXPORTER_OTLP_METRICS_CERTIFICATE_STRING` environment variable, or
  - the `OTEL_EXPORTER_OTLP_CERTIFICATE_STRING` environment variable, or
  - the default is ‘’.
- `ssl_client_key_path`: SSL client key, path to a file. Empty string uses the system default. Value is set from
  - the `opts` argument, or
  - the `OTEL_EXPORTER_OTLP_METRICS_CLIENT_KEY` environment variable, or
  - the `OTEL_EXPORTER_OTLP_CLIENT_KEY` environment variable, or
  - the default is ‘’.
- `ssl_client_key_string`: SSL client key as a string. Empty string uses the system default. Value is set from
  - the `opts` argument, or
  - the `OTEL_EXPORTER_OTLP_METRICS_CLIENT_KEY_STRING` environment variable, or
  - the `OTEL_EXPORTER_OTLP_CLIENT_KEY_STRING` environment variable, or
  - the default is ‘’.
- `ssl_client_cert_path`: SSL client certificate, path to a file. Empty string uses the system default. Value is set from
  - the `opts` argument, or
  - the `OTEL_EXPORTER_OTLP_METRICS_CLIENT_CERTIFICATE` environment variable, or
  - the `OTEL_EXPORTER_OTLP_CLIENT_CERTIFICATE` environment variable, or
  - the default is ‘’.
- `ssl_client_cert_string`: SSL client certificate, as a string. Empty string uses the system default. Value is set from
  - the `opts` argument, or
  - the `OTEL_EXPORTER_OTLP_METRICS_CLIENT_CERTIFICATE_STRING` environment variable, or
  - the `OTEL_EXPORTER_OTLP_CLIENT_CERTIFICATE_STRING` environment variable, or
  - the default is ‘’.

- `ssl_min_tls`: minimum TLS version. Empty string uses the system default. Value is set from
  - the `opts` argument, or
  - the `OTEL_R_EXPORTER_OTLP_METRICS_SSL_MIN_TLS` environment variable, or
  - the `OTEL_R_EXPORTER_OTLP_SSL_MIN_TLS` environment variable, or
  - the default is ‘’.
- `ssl_max_tls`: maximum TLS version. Empty string uses the system default. Value is set from
  - the `opts` argument, or
  - the `OTEL_R_EXPORTER_OTLP_METRICS_SSL_MAX_TLS` environment variable, or
  - the `OTEL_R_EXPORTER_OTLP_SSL_MAX_TLS` environment variable, or
  - the default is ‘’.
- `ssl_cipher`: TLS cipher. Empty string uses the system default. Value is set from
  - the `opts` argument, or
  - the `OTEL_R_EXPORTER_OTLP_METRICS_SSL_CIPHER` environment variable, or
  - the `OTEL_R_EXPORTER_OTLP_SSL_CIPHER` environment variable, or
  - the default is ‘’.
- `ssl_cipher_suite`: TLS cipher suite. Empty string uses the system default. Value is set from
  - the `opts` argument, or
  - the `OTEL_R_EXPORTER_OTLP_METRICS_SSL_CIPHER_SUITE` environment variable, or
  - the `OTEL_R_EXPORTER_OTLP_SSL_CIPHER_SUITE` environment variable, or
  - the default is ‘’.
- `compression`: compression to use. Possible values are `"none"`, `"gzip"`. Value is the set from
  - the `opts` argument, or
  - the `OTEL_EXPORTER_OTLP_METRICS_COMPRESSION` environment variable, or
  - the `OTEL_EXPORTER_OTLP_COMPRESSION` environment variable, or
  - the default is `none`.
- `retry_policy_max_attempts`: the maximum number of call attempts, including the original attempt. Value is set from
  - the `opts` argument, or
  - the `OTEL_R_EXPORTER_OTLP_METRICS_RETRY_POLICY_MAX_ATTEMPTS` environment variable, or
  - the `OTEL_R_EXPORTER_OTLP_RETRY_POLICY_MAX_ATTEMPTS` environment variable, or
  - the default is 5.
- `retry_policy_initial_backoff`: the maximum initial back-off delay between retry attempts. The actual backoff delay is uniform random between zero and this. Value is set from
  - the `opts` argument, or
  - the `OTEL_R_EXPORTER_OTLP_METRICS_RETRY_POLICY_INITIAL_BACKOFF` environment variable, or
  - the `OTEL_R_EXPORTER_OTLP_RETRY_POLICY_INITIAL_BACKOFF` environment variable, or

    - the default is `1000`.
- `retry_policy_max_backoff`: the maximum backoff places an upper limit on exponential backoff growth. Value is set from
    - the `opts` argument, or
    - the `OTEL_R_EXPORTER_OTLP_METRICS_RETRY_POLICY_MAX_BACKOFF` environment variable, or
    - the `OTEL_R_EXPORTER_OTLP_RETRY_POLICY_MAX_BACKOFF` environment variable, or
    - the default is `5000`.
- `retry_policy_backoff_multiplier`: the backoff will be multiplied by this value after each retry attempt. Value is set from
    - the `opts` argument, or
    - the `OTEL_R_EXPORTER_OTLP_METRICS_RETRY_POLICY_BACKOFF_MULTIPLIER` environment variable, or
    - the `OTEL_R_EXPORTER_OTLP_RETRY_POLICY_BACKOFF_MULTIPLIER` environment variable, or
    - the default is `1.5`.

Metric reader options:

- `export_interval`: the time interval between the start of two export attempts, in milliseconds. Value is set from
    - the `opts` argument, or
    - the `OTEL_METRIC_EXPORT_INTERVAL` environment variable, or
    - the default is `60000`.
- `export_timeout`: Maximum allowed time to export data, in milliseconds. Value is set from
    - the `opts` argument, or
    - the `OTEL_METRIC_EXPORT_TIMEOUT` environment variable, or
    - the default is `30000`.

Metric exporter options:

- `aggregation_temporality`: possible values: `"unspecified"`, `"delta"`, `"cumulative"`, `"lowmemory"`. See the OpenTelemetry data model. Value is set from
    - the `opts` argument, or
    - the `OTEL_R_EXPORTER_OTLP_AGGREGATION_TEMPORALITY` environment variable, or
    - the default is `"cumulative"`.

Examples

```
meter_provider_http$options()
```

`meter_provider_memory` *In-memory meter provider for testing*

Description

Collects metrics measurements in memory. This is useful for testing your instrumented R package or application.

`with_otel_record()` uses this meter provider. Use `with_otel_record()` in your tests to record telemetry and check that it is correct.

Value

`meter_provider_memory$new()` returns an otel::otel_meter_provider object. `mp$get_metrics()` returns a named list of recorded metrics.

`meter_provider_memory$options()` returns a named list, the current values for all options.

Usage

```
mp <- meter_provider_memory$new(opts = NULL)
mp$get_metrics()
meter_provider_memory$options()
```

`mp$get_metrics()` erases the internal buffer of the meter provider.

Arguments

- `opts`: Named list of options. See below.

Options

Memory exporter options:

- `buffer_size`: buffer size, this is the maximum number of spans or metrics measurements that the provider can record. Must be positive. Value is set from
  - the `opts` argument, or
  - the `OTEL_R_EXPORTER_MEMORY_METRICS_BUFFER_SIZE` environment variable, or
  - the `OTEL_R_EXPORTER_MEMORY_BUFFER_SIZE` environment variable, or
  - the default is `100`.

Metric reader options:

- `export_interval`: the time interval between the start of two export attempts, in milliseconds. Value is set from
  - the `opts` argument, or
  - the `OTEL_METRIC_EXPORT_INTERVAL` environment variable, or
  - the default is `60000`.
- `export_timeout`: Maximum allowed time to export data, in milliseconds. Value is set from
  - the `opts` argument, or

  - the `OTEL_METRIC_EXPORT_TIMEOUT` environment variable, or
  - the default is `30000`.

Metric exporter options:

- `aggregation_temporality`: possible values: `"unspecified"`, `"delta"`, `"cumulative"`, `"lowmemory"`. See the OpenTelemetry data model. Value is set from
  - the `opts` argument, or
  - the `OTEL_R_EXPORTER_OTLP_AGGREGATION_TEMPORALITY` environment variable, or
  - the default is `"cumulative"`.

Examples

```
meter_provider_memory$options()
```

`meter_provider_stdstream`

Meter provider to write to the standard output or standard error or to a file

---

Description

Writes metrics measurements to the standard output or error, or to a file. Useful for debugging.

Value

`meter_provider_stdstream$new()` returns an otel::otel_meter_provider object.

`meter_provider_stdstream$options()` returns a named list, the current values of the options.

Usage

Externally:

```
OTEL_METRICS_EXPORTER=console
OTEL_METRICS_EXPORTER=stderr
```

From R:

```
meter_provider_stdstream$new(opts = NULL)
meter_provider_stdstream$options()
```

Arguments

`opts`: Named list of options. See below.

Options

Standard stream exporter options:

- `output`: where to write the output. Can be
  - `"stdout"`: write output to the standard output,
  - `"stderr"`: write output to the standard error,
  - another string: write output to a file. (To write output to a file named `"stdout"` or `"stderr"`, use a `./` prefix.)

  Value is set from
  - the `opts` argument, or
  - the `OTEL_R_EXPORTER_STDSTREAM_METRICS_OUTPUT` environment variable, or
  - the `OTEL_R_EXPORTER_STDSTREAM_OUTPUT` environment variable, or
  - the default is `"stdout"`.

Metric reader options:

- `export_interval`: the time interval between the start of two export attempts, in milliseconds. Value is set from
  - the `opts` argument, or
  - the `OTEL_METRIC_EXPORT_INTERVAL` environment variable, or
  - the default is `60000`.
- `export_timeout`: Maximum allowed time to export data, in milliseconds. Value is set from
  - the `opts` argument, or
  - the `OTEL_METRIC_EXPORT_TIMEOUT` environment variable, or
  - the default is `30000`.

Examples

```
meter_provider_stdstream$options()
```

`Time Interval Options` *Time Interval Options*

Description

otel and otelsdk accept time interval options in the following format:

- A base::difftime object.
- A positive numeric scalar, interpreted as number of milliseconds. It may be fractional.
- A string scalar with a positive number and a time unit suffix. Possible time units: us (microseconds), ms (milliseconds), s (seconds), m (minutes), h (hours), d (days).

When the time interval is specified in an environment variable, it may be:

- A positive number, interpreted as number of milliseconds. It may be fractional.
- A positive number with a time unit suffix. Possible time units: us (microseconds), ms (milliseconds), s (seconds), m (minutes), h (hours), d (days).

Value

Not applicable.

Examples

```
# Write pending telemetry data to the output file every 5 seconds:
# OTEL_EXPORTER_OTLP_FILE_FLUSH_INTERVAL=5s
```

---

tracer_provider_file *Tracer provider to write traces into a JSONL file*

Description

This is the OTLP file exporter. It writes spans to a JSONL file, each span is a line in the file, a valid JSON value. The line separator is `\n`. The preferred file extension is `jsonl`.

Select this tracer provider with `OTEL_TRACES_EXPORTER=otlp/file`.

Select `OTEL_TRACES_EXPORTER=otlp/stdout` to write to the standard output instead of a file. This is the same as setting the `file_pattern` option to `stdout`.

Value

`tracer_provider_file$new()` returns an otel::otel_tracer_provider object.

`tracer_provider_file$options()` returns a named list, the current values of the options.

Usage

Externally:

```
OTEL_TRACES_EXPORTER=otlp/file
```

From R:

```
tracer_provider_file$new(opts = NULL)
tracer_provider_file$options()
```

Arguments

- `opts`: Named list of options. See below.

Options

File exporter options:

- `file_pattern`: Output file pattern. Value is set from:
  - `opts` argument, or
  - `OTEL_EXPORTER_OTLP_TRACES_FILE` environment variable, or
  - `OTEL_EXPORTER_OTLP_FILE` environment variable, or
  - the default is: `trace-%N.jsonl`.

  May contain placeholders, see below.
  If it is `stdout` or `stderr`, then the output is written to the standard output or standard error, instead of a file. In this case the other options are ignored.
- `alias_pattern`: The file which always point to the latest file. Value is set from:
  - `opts` argument, or
  - `OTEL_EXPORTER_OTLP_TRACES_FILE_ALIAS` environment variable, or
  - `OTEL_EXPORTER_OTLP_FILE_ALIAS environment variable`, or
  - the default is: `trace-latest.jsonl`.

  If `file_pattern` is set, but `alias_pattern` is not, then the default alias is created from `file_pattern`: the last group of placeholders in the file name is replaced by `latest`, and all other placeholders are removed, e.g. `trace-%F-%T-%N.jsonl` becomes `trace-latest.jsonl`.
  May contain placeholders, see below.
- `flush_interval`: Interval to force flush output. A time interval specification, see Time Interval Options. Value is set from
  - `opts` argument, or
  - `OTEL_EXPORTER_OTLP_TRACES_FILE_FLUSH_INTERVAL` environment variable, or
  - `OTEL_EXPORTER_OTLP_FILE_FLUSH_INTERVAL` environment variable, or
  - the default is `30s`, thirty seconds.
- `flush_count`: Force flush output after every `flush_count` records. Value is set from
  - `opts` argument, or
  - `OTEL_EXPORTER_OTLP_TRACES_FILE_FLUSH_COUNT` environment variable, or
  - `OTEL_EXPORTER_OTLP_FILE_FLUSH_COUNT` environment variable, or
  - the default is `256`.
- `file_size`: File size to rotate output files. A file size specification, see File Size Options. Value is set from
  - `opts` argument, or
  - `OTEL_EXPORTER_OTLP_TRACES_FILE_FILE_SIZE` environment variable, or
  - `OTEL_EXPORTER_OTLP_FILE_FILE_SIZE` environment variable, or
  - the default is `20MB`.
- `rotate_size`: How many rotated output files to keep. Value is set from
  - `opts` argument, or
  - `OTEL_EXPORTER_OTLP_TRACES_FILE_ROTATE_SIZE` environment variable, or
  - `OTEL_EXPORTER_OTLP_FILE_ROTATE_SIZE` environment variable, or
  - the default is `10`.

Special placeholders are available for `file_pattern` and `alias_pattern`:

- %Y: Writes year as a 4 digit decimal number.
- %y: Writes last 2 digits of year as a decimal number (range [00,99]).
- %m: Writes month as a decimal number (range [01,12]).
- %j: Writes day of the year as a decimal number (range [001,366]).
- %d: Writes day of the month as a decimal number (range [01,31]).
- %w: Writes weekday as a decimal number, where Sunday is 0 (range [0-6]).
- %H: Writes hour as a decimal number, 24 hour clock (range [00-23]).
- %I: Writes hour as a decimal number, 12 hour clock (range [01,12]).
- %M: Writes minute as a decimal number (range [00,59]).
- %S: Writes second as a decimal number (range [00,60]).
- %F: Equivalent to %Y-%m-%d (the ISO 8601 date format).
- %T: Equivalent to %H:%M:%S (the ISO 8601 time format).
- %R: Equivalent to %H:%M.
- %N: Rotate index, start from 0.
- %n: Rotate index, start from 1.

Examples

```
tracer_provider_file$options()
```

---

tracer_provider_http *Tracer provider to export traces over HTTP*

Description

This is the OTLP HTTP exporter.

Select this tracer provider with OTEL_TRACES_EXPORTER=otlp.

Value

tracer_provider_http$new() returns an otel::otel_tracer_provider object.

tracer_provider_http$options() returns a named list, the current values for all options.

Usage

Externally:

```
OTEL_TRACES_EXPORTER=otlp
```

From R:

```
tracer_provider_http$new(opts = NULL)
tracer_provider_http$options()
```

Arguments

- opts: Named list of options. See below.

Options

HTTP exporter options:

- url: OTLP URL to send telemetry data to. Value is set from
  - the opts argument, needs to point to the traces endpoint, or
  - OTEL_EXPORTER_OTLP_TRACES_ENDPOINT environment variable, needs to point to the traces endpoint, or
  - OTEL_EXPORTER_OTLP_ENDPOINT environment variable + /v1/traces, or
  - the default is http://localhost:4318/v1/traces.
- content_type: data format used, JSON or binary. Possible values: "http/json", "http/protobuf". Value if set from
  - the opts argument, or
  - the OTEL_EXPORTER_OTLP_TRACES_PROTOCOL environment variable, or
  - the OTEL_EXPORTER_OTLP_PROTOCOL environment variable, or
  - the default is "http/protobuf".
- json_bytes_mapping: encoding used for trace id and span id. Possible values: "hexid", "base64", "hex". Value is set from
  - the opts argument, or
  - the OTEL_R_EXPORTER_OTLP_TRACES_JSON_BYTES_MAPPING environment variable, or
  - the OTEL_R_EXPORTER_OTLP_JSON_BYTES_MAPPING environment variable, or
  - the default is "hexid".
- use_json_name: whether to use json name of protobuf field to set the key of json. A flag. Value is set from
  - the opts argument, or
  - the OTEL_R_EXPORTER_OTLP_TRACES_USE_JSON_NAME environment variable, or
  - the OTEL_R_EXPORTER_OTLP_USE_JSON_NAME environment variable, or
  - the default is FALSE.
- console_debug: whether to print debug messages to the console. Value is set from
  - the opts argument, or
  - the OTEL_R_EXPORTER_OTLP_TRACES_CONSOLE_DEBUG environment variable, or
  - the OTEL_R_EXPORTER_OTLP_CONSOLE_DEBUG environment variable, or
  - the default is FALSE.
- timeout: HTTP timeout in milliseconds. Value is set from
  - the opts argument, or
  - the OTEL_EXPORTER_OTLP_TRACES_TIMEOUT} environment variable, or
  - the OTEL_EXPORTER_OTLP_TIMEOUT environment variable, or
  - the default is 10000.
- http_headers: additional HTTP headers to send, e.g. Authorization is commonly used. A named character vector without NA values. Value is set from
  - the opts argument, or

  - the OTEL_EXPORTER_OTLP_TRACES_HEADERS environment variable, or
  - the OTEL_EXPORTER_OTLP_HEADERS environment variable, or
  - the default is an empty named character vector. When specified in environment variables, it must be a comma separated list of headers, each in the header=value form.
- ssl_insecure_skip_verify: whether to disable SSL. Value is set from
  - the opts argument, or
  - the OTEL_R_EXPORTER_OTLP_TRACES_SSL_INSECURE_SKIP_VERIFY environment variable, or
  - the OTEL_R_EXPORTER_OTLP_SSL_INSECURE_SKIP_VERIFY environment variable, or
  - the default is FALSE.
- ssl_ca_cert_path: CA certificate, path to a file. Empty string uses the system default. Value is set from
  - the opts argument, or
  - the OTEL_EXPORTER_OTLP_TRACES_CERTIFICATE environment variable, or
  - the OTEL_EXPORTER_OTLP_CERTIFICATE environment variable, or
  - the default is ".
- ssl_ca_cert_string: CA certificate, as a string. Empty string uses the system default. Value is set from
  - the opts argument, or
  - the OTEL_EXPORTER_OTLP_TRACES_CERTIFICATE_STRING environment variable, or
  - the OTEL_EXPORTER_OTLP_CERTIFICATE_STRING environment variable, or
  - the default is ".
- ssl_client_key_path: SSL client key, path to a file. Empty string uses the system default. Value is set from
  - the opts argument, or
  - the OTEL_EXPORTER_OTLP_TRACES_CLIENT_KEY environment variable, or
  - the OTEL_EXPORTER_OTLP_CLIENT_KEY environment variable, or
  - the default is ".
- ssl_client_key_string: SSL client key as a string. Empty string uses the system default. Value is set from
  - the opts argument, or
  - the OTEL_EXPORTER_OTLP_TRACES_CLIENT_KEY_STRING environment variable, or
  - the OTEL_EXPORTER_OTLP_CLIENT_KEY_STRING environment variable, or
  - the default is ".
- ssl_client_cert_path: SSL client certificate, path to a file. Empty string uses the system default. Value is set from
  - the opts argument, or
  - the OTEL_EXPORTER_OTLP_TRACES_CLIENT_CERTIFICATE environment variable, or
  - the OTEL_EXPORTER_OTLP_CLIENT_CERTIFICATE environment variable, or
  - the default is ".
- ssl_client_cert_string: SSL client certificate, as a string. Empty string uses the system default. Value is set from
  - the opts argument, or

  - the OTEL_EXPORTER_OTLP_TRACES_CLIENT_CERTIFICATE_STRING environment variable, or
  - the OTEL_EXPORTER_OTLP_CLIENT_CERTIFICATE_STRING environment variable, or
  - the default is ‘’.
- ssl_min_tls: minimum TLS version. Empty string uses the system default. Value is set from
  - the opts argument, or
  - the OTEL_R_EXPORTER_OTLP_TRACES_SSL_MIN_TLS environment variable, or
  - the OTEL_R_EXPORTER_OTLP_SSL_MIN_TLS environment variable, or
  - the default is ‘’.
- ssl_max_tls: maximum TLS version. Empty string uses the system default. Value is set from
  - the opts argument, or
  - the OTEL_R_EXPORTER_OTLP_TRACES_SSL_MAX_TLS environment variable, or
  - the OTEL_R_EXPORTER_OTLP_SSL_MAX_TLS environment variable, or
  - the default is ‘’.
- ssl_cipher: TLS cipher. Empty string uses the system default. Value is set from
  - the opts argument, or
  - the OTEL_R_EXPORTER_OTLP_TRACES_SSL_CIPHER environment variable, or
  - the OTEL_R_EXPORTER_OTLP_SSL_CIPHER environment variable, or
  - the default is ‘’.
- ssl_cipher_suite: TLS cipher suite. Empty string uses the system default. Value is set from
  - the opts argument, or
  - the OTEL_R_EXPORTER_OTLP_TRACES_SSL_CIPHER_SUITE environment variable, or
  - the OTEL_R_EXPORTER_OTLP_SSL_CIPHER_SUITE environment variable, or
  - the default is ‘’.
- compression: compression to use. Possible values are "none", "gzip". Value is the set from
  - the opts argument, or
  - the OTEL_EXPORTER_OTLP_TRACES_COMPRESSION environment variable, or
  - the OTEL_EXPORTER_OTLP_COMPRESSION environment variable, or
  - the default is none.
- retry_policy_max_attempts: the maximum number of call attempts, including the original attempt. Value is set from
  - the opts argument, or
  - the OTEL_R_EXPORTER_OTLP_TRACES_RETRY_POLICY_MAX_ATTEMPTS environment variable, or
  - the OTEL_R_EXPORTER_OTLP_RETRY_POLICY_MAX_ATTEMPTS environment variable, or
  - the default is 5.
- retry_policy_initial_backoff: the maximum initial back-off delay between retry attempts. The actual backoff delay is uniform random between zero and this. Value is set from

  - the opts argument, or
  - the OTEL_R_EXPORTER_OTLP_TRACES_RETRY_POLICY_INITIAL_BACKOFF environment variable, or
  - the OTEL_R_EXPORTER_OTLP_RETRY_POLICY_INITIAL_BACKOFF environment variable, or
  - the default is 1000.
- retry_policy_max_backoff: the maximum backoff places an upper limit on exponential backoff growth. Value is set from
  - the opts argument, or
  - the OTEL_R_EXPORTER_OTLP_TRACES_RETRY_POLICY_MAX_BACKOFF environment variable, or
  - the OTEL_R_EXPORTER_OTLP_RETRY_POLICY_MAX_BACKOFF environment variable, or
  - the default is 5000.
- retry_policy_backoff_multiplier: the backoff will be multiplied by this value after each retry attempt. Value is set from
  - the opts argument, or
  - the OTEL_R_EXPORTER_OTLP_TRACES_RETRY_POLICY_BACKOFF_MULTIPLIER environment variable, or
  - the OTEL_R_EXPORTER_OTLP_RETRY_POLICY_BACKOFF_MULTIPLIER environment variable, or
  - the default is 1.5.

Batch processor options:

- max_queue_size: The maximum buffer/queue size. After the size is reached, spans are dropped. Must be positive. Value is set from
  - the opts argument, or
  - the OTEL_BSP_MAX_QUEUE_SIZE environment variable, or
  - the default is 2048.
- max_export_batch_size: the maximum batch size of every export. It must be smaller or equal to max_queue_size. Must be positive. Value is set from
  - the opts argument, or
  - the OTEL_BSP_MAX_EXPORT_BATCH_SIZE environment variable, or
  - the default is 512.
- schedule_delay: the time interval between two consecutive exports, in milliseconds. Value is set from
  - the opts argument‘ or
  - the OTEL_BSP_SCHEDULE_DELAY environment variable, or
  - the default is 5000.

Examples

```
tracer_provider_http$options()
```

`tracer_provider_memory` *In-memory tracer provider for testing*

Description

Collects spans in memory. This is useful for testing your instrumented R package or application.

`with_otel_record()` uses this tracer provider. Use `with_otel_record()` in your tests to record telemetry and check that it is correct.

Value

`tracer_provider_memory$new()` returns an otel::otel_tracer_provider object. `tp$get_spans()` returns a named list of recorded spans, with the span names as names.

`tracer_provider_memory$options()` returns a named list, the current values for all options.

Usage

```
tp <- tracer_provider_memory$new(opts = NULL)
tp$get_spans()
tracer_provider_memory$options()
```

`tp$get_spans()` erases the internal buffer of the tracer provider.

Arguments

- `opts`: Named list of options. See below.

Options

Memory exporter options:

- `buffer_size`: buffer size, this is the maximum number of spans or metrics measurements that the provider can record. Must be positive. Value is set from
  - the `opts` argument, or
  - the `OTEL_R_EXPORTER_MEMORY_TRACES_BUFFER_SIZE` environment variable, or
  - the `OTEL_R_EXPORTER_MEMORY_BUFFER_SIZE` environment variable, or
  - the default is `100`.

Examples

```
tracer_provider_memory$options()
```

`tracer_provider_stdstream` *Tracer provider to write to the standard output or standard error or to a file*

Description

Writes spans to the standard output or error, or to a file. Useful for debugging.

Value

`tracer_provider_stdstream$new()` returns an otel::otel_tracer_provider object.

`tracer_provider_stdstream$options()` returns a named list, the current values of the options.

Usage

Externally:

```
OTEL_TRACES_EXPORTER=console
OTEL_TRACES_EXPORTER=stderr
```

From R:

```
tracer_provider_stdstream$new(opts = NULL)
tracer_provider_stdstream$options()
```

Arguments

`opts`: Named list of options. See below.

Options

Standard stream exporter options:

- `output`: where to write the output. Can be
  - `"stdout"`: write output to the standard output,
  - `"stderr"`: write output to the standard error,
  - another string: write output to a file. (To write output to a file named `"stdout"` or `"stderr"`, use a `./` prefix.)

  Value is set from
  - the `opts` argument, or
  - the `OTEL_R_EXPORTER_STDSTREAM_TRACES_OUTPUT` environment variable, or
  - the `OTEL_R_EXPORTER_STDSTREAM_OUTPUT` environment variable, or
  - the default is `"stdout"`.

Examples

```
tracer_provider_stdstream$options()
```

`with_otel_record` *Record OpenTelemetry output, for testing purposes*

Description

You can use this function to test that OpenTelemetry output is correctly generated for your package or application.

Usage

```
with_otel_record(
  expr,
  what = c("traces", "metrics", "logs"),
  tracer_opts = list(),
  meter_opts = list(),
  logger_opts = list()
)
```

Arguments

`expr`	Expression to evaluate.
`what`	Character vector, type(s) of OpenTelemetry output to collect.
`tracer_opts`	Named list of options to pass to the tracer provider.
`meter_opts`	Named list of options to pass to the meter provider.
`logger_opts`	Named list of options to pass to the logger provider.

Details

It evaluates the supplied expression, collects OpenTelemetry output from it and returns it.

`with_otel_record()` uses tracer_provider_memory, meter_provider_memory and logger_provider_memory internally.

Value

A list with the output for each output type. Entries:

- `value`: value of `expr`.
- `traces`: the recorded spans, if requested in `what`.
- `metrics`: the recorded metrics measurements, if requested in `what`.
- `logs`: the recorded log records, if requested in `what`.

Examples

```
spns <- with_otel_record({
  trc <- otel::get_tracer("mytracer")
  spn1 <- trc$start_local_active_span()
  spn2 <- trc$start_local_active_span("my")
  spn2$end()
  spn1$end()
  NULL
})
spns
```

Index

add_span_processor, 2
add_span_processor(), *13*

base::difftime, *33*

collecting *(*Collecting Telemetry Data*)*, *5*
Collecting Telemetry Data, *5*

Environment Variables, *7*, 7, *14*
environment variables, *5*

File Size Options, *12*, 15, *17*, *25*, *35*

Getting Started, *5*

logger_provider_file, *11*, 15
logger_provider_http, *9*, *12*, 17
logger_provider_memory, *13*, 22, *43*
logger_provider_stdstream, *13*, 23

meter_provider_file, *11, 12*, 24
meter_provider_http, *9*, *12*, 26
meter_provider_memory, *12, 13*, 31, *43*
meter_provider_stdstream, *12, 13*, 32

otel::as_attributes(), *9*, *14*
otel::default_tracer_name(), *8, 9*
otel::get_default_logger_provider(), *8*
otel::get_default_meter_provider(), *8*
otel::get_default_tracer_provider(), *8*
otel::otel_logger_provider, *15*, *17*, *22, 23*
otel::otel_meter_provider, *24*, *26*, *31, 32*
otel::otel_tracer_provider, *34*, *36*, *41, 42*
OTEL_BSP_MAX_EXPORT_BATCH_SIZE *(*Environment Variables*)*, 7
OTEL_BSP_MAX_QUEUE_SIZE *(*Environment Variables*)*, 7
OTEL_BSP_SCHEDULE_DELAY *(*Environment Variables*)*, 7
OTEL_ENV *(*Environment Variables*)*, 7
OTEL_EXPORTER_OTLP_CERTIFICATE *(*Environment Variables*)*, 7
OTEL_EXPORTER_OTLP_CERTIFICATE_STRING *(*Environment Variables*)*, 7
OTEL_EXPORTER_OTLP_CLIENT_CERTIFICATE *(*Environment Variables*)*, 7
OTEL_EXPORTER_OTLP_CLIENT_CERTIFICATE_STRING *(*Environment Variables*)*, 7
OTEL_EXPORTER_OTLP_CLIENT_KEY *(*Environment Variables*)*, 7
OTEL_EXPORTER_OTLP_CLIENT_KEY_STRING *(*Environment Variables*)*, 7
OTEL_EXPORTER_OTLP_COMPRESSION *(*Environment Variables*)*, 7
OTEL_EXPORTER_OTLP_FILE *(*Environment Variables*)*, 7
OTEL_EXPORTER_OTLP_FILE_ALIAS *(*Environment Variables*)*, 7
OTEL_EXPORTER_OTLP_FILE_FILE_SIZE *(*Environment Variables*)*, 7
OTEL_EXPORTER_OTLP_FILE_FLUSH_COUNT *(*Environment Variables*)*, 7
OTEL_EXPORTER_OTLP_FILE_FLUSH_INTERVAL *(*Environment Variables*)*, 7
OTEL_EXPORTER_OTLP_FILE_ROTATE_SIZE *(*Environment Variables*)*, 7
OTEL_EXPORTER_OTLP_HEADERS *(*Environment Variables*)*, 7
OTEL_EXPORTER_OTLP_LOGS_CERTIFICATE *(*Environment Variables*)*, 7
OTEL_EXPORTER_OTLP_LOGS_CERTIFICATE_STRING *(*Environment Variables*)*, 7
OTEL_EXPORTER_OTLP_LOGS_CLIENT_CERTIFICATE *(*Environment Variables*)*, 7
OTEL_EXPORTER_OTLP_LOGS_CLIENT_CERTIFICATE_STRING *(*Environment Variables*)*, 7

OTEL_EXPORTER_OTLP_LOGS_CLIENT_KEY (Environment Variables), 7
OTEL_EXPORTER_OTLP_LOGS_CLIENT_KEY_STRING (Environment Variables), 7
OTEL_EXPORTER_OTLP_LOGS_COMPRESSION (Environment Variables), 7
OTEL_EXPORTER_OTLP_LOGS_FILE (Environment Variables), 7
OTEL_EXPORTER_OTLP_LOGS_FILE_ALIAS (Environment Variables), 7
OTEL_EXPORTER_OTLP_LOGS_FILE_FILE_SIZE (Environment Variables), 7
OTEL_EXPORTER_OTLP_LOGS_FILE_FLUSH_INTERVAL (Environment Variables), 7
OTEL_EXPORTER_OTLP_LOGS_FILE_ROTATE_SIZE (Environment Variables), 7
OTEL_EXPORTER_OTLP_LOGS_HEADERS (Environment Variables), 7
OTEL_EXPORTER_OTLP_LOGS_PROTOCOL (Environment Variables), 7
OTEL_EXPORTER_OTLP_LOGS_TIMEOUT (Environment Variables), 7
OTEL_EXPORTER_OTLP_METRICS_CERTIFICATE (Environment Variables), 7
OTEL_EXPORTER_OTLP_METRICS_CERTIFICATE_STRING (Environment Variables), 7
OTEL_EXPORTER_OTLP_METRICS_CLIENT_CERTIFICATE (Environment Variables), 7
OTEL_EXPORTER_OTLP_METRICS_CLIENT_CERTIFICATE_STRING (Environment Variables), 7
OTEL_EXPORTER_OTLP_METRICS_CLIENT_KEY (Environment Variables), 7
OTEL_EXPORTER_OTLP_METRICS_CLIENT_KEY_STRING (Environment Variables), 7
OTEL_EXPORTER_OTLP_METRICS_COMPRESSION (Environment Variables), 7
OTEL_EXPORTER_OTLP_METRICS_FILE (Environment Variables), 7
OTEL_EXPORTER_OTLP_METRICS_FILE_ALIAS (Environment Variables), 7
OTEL_EXPORTER_OTLP_METRICS_FILE_FILE_SIZE (Environment Variables), 7
OTEL_EXPORTER_OTLP_METRICS_FILE_FLUSH_INTERVAL (Environment Variables), 7
OTEL_EXPORTER_OTLP_METRICS_FILE_ROTATE_SIZE (Environment Variables), 7
OTEL_EXPORTER_OTLP_METRICS_HEADERS (Environment Variables), 7
OTEL_EXPORTER_OTLP_METRICS_PROTOCOL (Environment Variables), 7
OTEL_EXPORTER_OTLP_METRICS_TIMEOUT (Environment Variables), 7
OTEL_EXPORTER_OTLP_PROTOCOL (Environment Variables), 7
OTEL_EXPORTER_OTLP_TIMEOUT (Environment Variables), 7
OTEL_EXPORTER_OTLP_TRACES_CERTIFICATE (Environment Variables), 7
OTEL_EXPORTER_OTLP_TRACES_CERTIFICATE_STRING (Environment Variables), 7
OTEL_EXPORTER_OTLP_TRACES_CLIENT_CERTIFICATE (Environment Variables), 7
OTEL_EXPORTER_OTLP_TRACES_CLIENT_CERTIFICATE_STRING (Environment Variables), 7
OTEL_EXPORTER_OTLP_TRACES_CLIENT_KEY (Environment Variables), 7
OTEL_EXPORTER_OTLP_TRACES_CLIENT_KEY_STRING (Environment Variables), 7
OTEL_EXPORTER_OTLP_TRACES_COMPRESSION (Environment Variables), 7
OTEL_EXPORTER_OTLP_TRACES_FILE (Environment Variables), 7
OTEL_EXPORTER_OTLP_TRACES_FILE_ALIAS (Environment Variables), 7
OTEL_EXPORTER_OTLP_TRACES_FILE_FILE_SIZE (Environment Variables), 7
OTEL_EXPORTER_OTLP_TRACES_FILE_FLUSH_INTERVAL (Environment Variables), 7
OTEL_EXPORTER_OTLP_TRACES_FILE_ROTATE_SIZE (Environment Variables), 7
OTEL_EXPORTER_OTLP_TRACES_HEADERS (Environment Variables), 7
OTEL_EXPORTER_OTLP_TRACES_PROTOCOL (Environment Variables), 7
OTEL_EXPORTER_OTLP_TRACES_TIMEOUT (Environment Variables), 7
OTEL_LOG_LEVEL (Environment Variables), 7
OTEL_LOGS_EXPORTER (Environment Variables), 7
OTEL_METRIC_EXPORT_INTERVAL (Environment Variables), 7
OTEL_METRIC_EXPORT_TIMEOUT (Environment Variables), 7
OTEL_METRICS_EXPORTER (Environment Variables), 7

OTEL_R_EMIT_SCOPES (Environment Variables), 7
OTEL_R_EXPORTER_MEMORY_BUFFER_SIZE (Environment Variables), 7
OTEL_R_EXPORTER_MEMORY_LOGS_BUFFER_SIZE (Environment Variables), 7
OTEL_R_EXPORTER_MEMORY_METRICS_BUFFER_SIZE (Environment Variables), 7
OTEL_R_EXPORTER_MEMORY_TRACES_BUFFER_SIZE (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_AGGREGATION_TEMPORALITY (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_CONSOLE_DEBUG (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_LOGS_CONSOLE_DEBUG (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_LOGS_RETRY_POLICY_BACKOFF_MULTIPLIER (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_LOGS_RETRY_POLICY_INITIAL_BACKOFF (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_LOGS_RETRY_POLICY_MAX_ATTEMPTS (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_LOGS_RETRY_POLICY_MAX_BACKOFF (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_LOGS_SSL_CIPHER (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_LOGS_SSL_CIPHER_SUITE (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_LOGS_SSL_INSECURE_SKIP_VERIFY (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_LOGS_SSL_MAX_TLS (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_LOGS_SSL_MIN_TLS (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_LOGS_USE_JSON_NAME (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_METRICS_CONSOLE_DEBUG (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_METRICS_RETRY_POLICY_BACKOFF_MULTIPLIER (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_METRICS_RETRY_POLICY_INITIAL_BACKOFF (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_METRICS_RETRY_POLICY_MAX_ATTEMPTS (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_METRICS_RETRY_POLICY_MAX_BACKOFF (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_METRICS_SSL_CIPHER (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_METRICS_SSL_CIPHER_SUITE (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_METRICS_SSL_INSECURE_SKIP_VERIFY (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_METRICS_SSL_MAX_TLS (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_METRICS_SSL_MIN_TLS (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_METRICS_USE_JSON_NAME (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_RETRY_POLICY_BACKOFF_MULTIPLIER (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_RETRY_POLICY_INITIAL_BACKOFF (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_RETRY_POLICY_MAX_ATTEMPTS (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_RETRY_POLICY_MAX_BACKOFF (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_SSL_CIPHER (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_SSL_CIPHER_SUITE (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_SSL_INSECURE_SKIP_VERIFY (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_SSL_MAX_TLS (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_SSL_MIN_TLS (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_TRACES_CONSOLE_DEBUG (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_TRACES_RETRY_POLICY_BACKOFF_MULTIPLIER (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_TRACES_RETRY_POLICY_INITIAL_BACKOFF (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_TRACES_RETRY_POLICY_MAX_ATTEMPTS (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_TRACES_RETRY_POLICY_MAX_BACKOFF (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_TRACES_SSL_CIPHER (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_TRACES_SSL_CIPHER_SUITE (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_TRACES_SSL_INSECURE_SKIP_VERIFY (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_TRACES_SSL_MAX_TLS (Environment Variables), 7
OTEL_R_EXPORTER_OTLP_TRACES_SSL_MIN_TLS (Environment Variables), 7

OTEL_R_EXPORTER_OTLP_TRACES_USE_JSON_NAME
(Environment Variables), 7
OTEL_R_EXPORTER_OTLP_USE_JSON_NAME
(Environment Variables), 7
OTEL_R_EXPORTER_STDSTREAM_LOGS_OUTPUT
(Environment Variables), 7
OTEL_R_EXPORTER_STDSTREAM_METRICS_OUTPUT
(Environment Variables), 7
OTEL_R_EXPORTER_STDSTREAM_OUTPUT
(Environment Variables), 7
OTEL_R_EXPORTER_STDSTREAM_TRACES_OUTPUT
(Environment Variables), 7
OTEL_R_LOGS_EXPORTER (Environment
Variables), 7
OTEL_R_METRICS_EXPORTER (Environment
Variables), 7
OTEL_R_SUPPRESS_SCOPES (Environment
Variables), 7
OTEL_R_TRACES_EXPORTER (Environment
Variables), 7
OTEL_RESOURCE_ATTRIBUTES (Environment
Variables), 7
OTEL_SDK_DISABLED (Environment
Variables), 7
OTEL_SERVICE_NAME (Environment
Variables), 7
OTEL_TRACES_EXPORTER (Environment
Variables), 7

remove_span_processor
(add_span_processor), 2

Time Interval Options, *12*, *16*, *25*, 33, *35*
tracer_provider_file, *11*, 34
tracer_provider_http, *9*, *12*, 36
tracer_provider_memory, *13*, 41, *43*
tracer_provider_stdstream, *13*, 42

with_otel_record, 43
with_otel_record(), *31*, *41*

Zero Code Instrumentation, *5*, *9*