ETW logging driver

- Usage

The Event Tracing for Windows (ETW) logging driver forwards container logs as ETW events. ETW stands for Event Tracing in Windows, and is the common framework for tracing applications in Windows. Each ETW event contains a message with both the log and its context information. A client can then create an ETW listener to listen to these events.

The ETW provider that this logging driver registers with Windows, has the GUID identifier of: `{a3693192-9ed6-46d2-a981-f8226c8363bd}`. A client creates an ETW listener and registers to listen to events from the logging driver's provider. It doesn't matter the order in which the provider and listener are created. A client can create their ETW listener and start listening for events from the provider, before the provider has been registered with the system.

Usage

Here is an example of how to listen to these events using the logman utility program included in most installations of Windows:

1. `logman start -ets DockerContainerLogs -p "{a3693192-9ed6-46d2-a981-f8226c8363bd}" 0x0 -o trace.etl`
2. Run your container(s) with the etwlogs driver, by adding `--log-driver=etwlogs` to the Docker run command, and generate log messages.
3. `logman stop -ets DockerContainerLogs`
4. This generates an etl file that contains the events. One way to convert this file into human-readable form is to run: `tracerpt -y trace.etl`.

Each ETW event contains a structured message string in this format:

```
container_name: %s, image_name: %s, container_id: %s, image_id: %s, source: [stdout | stderr], log: %s
```

Details on each item in the message can be found below:

Field	Description

`container_name`	The container name at the time it was started.
`image_name`	The name of the container's image.
`container_id`	The full 64-character container ID.
`image_id`	The full ID of the container's image.
`source`	`stdout` or `stderr`.
`log`	The container log message.

Here is an example event message (output formatted for readability):

```
container_name: backstabbing_spence,
image_name: windowsservercore,
container_id: f14bb55aa862d7596b03a33251c1be7dbbec8056bbdead1da8ec5ecebbe29731,
image_id: sha256:2f9e19bd998d3565b4f345ac9aaf6e3fc555406239a4fb1b1ba879673713824b,
source: stdout,
log: Hello world!
```

A client can parse this message string to get both the log message, as well as its context information. The timestamp is also available within the ETW event.

> This ETW provider only emits a message string, and not a specially structured ETW event. Therefore, you don't have to register a manifest file with the system to read and interpret its ETW events.

□□□ #2
□ □□□ □□□ 2026-10-31 19:24:27 UTC
□ □□□ □□□ 2026-10-31 19:24:28 UTC