SynchroAgent User Guide

Version 1.1.5

25 September 2026

Contents

1. About SynchroAgent

SynchroAgent measures how a machine is performing – processor, memory, disks, file systems, network and running processes – and reports those measurements to SynchroServer, where they are stored and graphed.

It installs as a **single file**. There is no runtime to install first, no interpreter, no shared libraries and no system-wide changes. Installation does not require administrator or root privileges.

This guide describes **SynchroAgent 1.1.5**.

1.1 One version for every operating system

SynchroAgent carries a single version number across all supported operating systems. Version 1.1.5 on Linux, on macOS and on Windows is the same release, built from the same source, documented by this guide, and delivered in one package containing all of them.

There is no separate Linux version number or Windows version number to track. When you are asked which version you are running, the answer is one number:

```
~/SynchroAgent/bin/synchroagent -version
# synchroagent 1.1.5 built ...
```

The agent also reports its own version to SynchroServer, so the version running on every monitored machine can be seen centrally. See section 9.8.

2. What you need

2.1 Supported operating systems

SynchroAgent is compiled native code, and the versions below are hard floors: on anything older the agent will not start at all. They come from the Go toolchain the agent is built with, not from a choice made by the agent.

Operating system	Minimum version	Processor
Linux	kernel 3.2	x86-64, arm64
Windows	Windows Server 2016, or Windows 10	x86-64
macOS	macOS 13 Ventura	Apple silicon

Linux has no distribution requirement and no library requirement. The binary is statically linked, so it does not depend on the C library and runs on any distribution meeting the kernel version, regardless of age or how the machine is built.

Windows older than Server 2016 or Windows 10 cannot run this agent. There is no configuration or compatibility mode that changes this. Windows Server 2012 R2, 2012, 2008 R2 and earlier are not supported, and neither are Windows 8.1 and earlier. If you must monitor those machines, they need a different agent. Note also that Windows deployment is not yet supported at all; see section 8.2.

macOS builds are for Apple silicon. Intel Macs are not covered by the supplied binary.

If you are unsure whether a machine qualifies, the quickest check is to run the agent with `-version`: it either prints a version or refuses to start.

2.2 Everything else

Network access: the monitored machine must be able to open an outbound TCP connection to SynchroServer, on port 9123 unless your installation uses a different one. No inbound connection to the monitored machine is required.

Disk space: about 10 MB, plus log files.

Privileges: on Linux and macOS, none. The agent runs as an ordinary user account and reports on the whole machine; it does not need root.

On Windows, Administrator is needed to *install* the service, and the service runs as `LocalSystem`. That is not a convenience: measuring another account's processes requires it, and an agent running as an ordinary user would report only its own processes while appearing to work normally.

Clock: measurements are timestamped by the monitored machine in its own local time. A machine whose clock or time zone differs from SynchroServer's will have its measurements filed under the wrong time. Make sure the clock is synchronised and the time zone matches the server's.

> The connection to SynchroServer is not encrypted and not authenticated. It should run over a trusted network or through a tunnel.

3. Installing

You will be supplied with a release archive named `SynchroAgent-1.1.5.tar.gz`, together with a `.sha256` checksum file.

3.1 Check the download

```
sha256sum -c SynchroAgent-1.1.5.tar.gz.sha256
# SynchroAgent-1.1.5.tar.gz: OK
```

On macOS, use `shasum -a 256 -c` instead.

3.2 Unpack and install, Linux and macOS

```
tar xzf SynchroAgent-1.1.5.tar.gz
cd SynchroAgent-1.1.5
./install.sh
```

The installer selects the agent matching the operating system it is run on, and reports what it did:

```
Installing SynchroAgent-Linux into /home/you/SynchroAgent
Created conf/synchroagent-local.ini for site settings

Installed: synchroagent 1.1.5 built 20260923
```

By default everything goes into `SynchroAgent` in your home directory. To install elsewhere, pass the location:

```
./install.sh /opt/monitoring/SynchroAgent
```

3.3 Unpack and install, Windows

Windows uses the `.zip` and a PowerShell installer. Both are in the same release, so there is nothing extra to fetch.

Open an **elevated** PowerShell – right-click Start, "Windows PowerShell (Admin)" – and run:

```
Expand-Archive SynchroAgent-1.1.5.zip -DestinationPath C:\Temp
cd C:\Temp\SynchroAgent-1.1.5
.\install.ps1 -Server synchro.example.com -Component web-01 -InstallService
```

That installs to `C:\SynchroAgent`, writes your settings into `conf\synchroagent-local.ini`, and registers the service to start automatically as `LocalSystem`. Pass `-Target` to install elsewhere.

The installer prompts for nothing, so it works unattended. To deploy across an estate:

```
Invoke-Command -ComputerName SRV1,SRV2 -FilePath .\install.ps1 `
    -ArgumentList '-Server','synchro.example.com','-InstallService'
```

Omit `-InstallService` to lay the files down without registering the service, which then needs no Administrator rights. Add `-Force` to upgrade over a running service; the installer otherwise refuses rather than replace a binary underneath it.

A `.zip` is supplied because Windows Server 2016 has no built-in `tar`.

3.4 What gets installed

```
SynchroAgent/
    bin/     the agent, and the scripts to start and stop it
    conf/    configuration
    doc/     this guide
    log/     one log file per start
    run/     the process id of the running agent
```

3.5 What is in the package

One archive contains the agents for every supported operating system, so the same download serves a mixed estate:

File	For
`SynchroAgent-Linux`	Linux, 64-bit Intel or AMD
`SynchroAgent-Darwin`	macOS, Apple silicon
`SynchroAgent-Windows.exe`	Windows, 64-bit

`install.sh` handles Linux and macOS, `install.ps1` handles Windows. The guide you are reading is in the package too, under `doc`.

4. Configuration

4.1 Which file to edit

There are two configuration files in `conf`:

File	Purpose
`synchroagent.ini`	supplied defaults – **do not edit**, an upgrade replaces it
`synchroagent-local.ini`	your settings – an upgrade never touches it

Always edit `synchroagent-local.ini`. Anything set there overrides the supplied default of the same name.

At minimum, set where the server is:

```
Server=synchro.example.com
Port=9123
```

4.2 Settings

Setting	Default	Meaning
`Server`	*(none)*	SynchroServer host name or address
`Port`	`9123`	port SynchroServer listens on
`Component`	*(host name)*	the name this machine reports under
`Verbose`	`false`	write detailed diagnostics to the log
`HOST`	`yes`	whole-machine processor and memory measurements

Setting	Default	Meaning
CORE	yes	per-processor-core measurements
DISK	yes	disk activity measurements
NET	yes	network measurements
FS	yes	file system space measurements
PROC	yes	per-process measurements
PROC-EXTENDED	yes	extra per-process memory detail, Windows only
THRESHOLD	0.01	measurements at or below this are not sent
PROC-CPU-THRESHOLD	0.01	as above, for process processor use
PROC-RSS-THRESHOLD	0.01	as above, for process memory use

Settings accept yes/no or true/false.

There is one setting per measurement type, matching the types described in section 9: set DISK=no and no DISK_* measurement is reported. HOST covers both processor and memory, because both are reported as whole-machine HOST measurements.

Turning a type off reduces both network traffic and the load the agent puts on the machine. PROC=no makes by far the largest difference; see section 9.10.

4.3 How the machine is identified

Measurements are filed on the server under a **component name**. By default this is the machine's host name, which is usually what you want.

You will need a different name when the host name is already used by another reporter on the same SynchroServer, or when a host name is not meaningful – several machines built from one image, for example.

Where the name comes from

The agent takes the first of these that is set:

Order	Source	Set it with
1	command-line flag	`-component NAME`
2	environment variable	`SYNCHRO_COMPONENT=NAME`
3	configuration file	`Component=NAME` in `synchroagent-local.ini`
4	the machine's host name	*(automatic)*

Blank or whitespace-only values are ignored and the next source is used.

Use the configuration file for a permanent setting. It is the only one of the four that an upgrade preserves: the flag and the environment variable are set wherever the agent is started from, and that start script is replaced when you upgrade.

```
# in conf/synchroagent-local.ini
Component=web-server-01
```

Use the environment variable for a machine whose identity is decided when it is deployed rather than when it is configured:

```
SYNCHRO_COMPONENT=web-server-01 ~/SynchroAgent/bin/start-client.sh
```

Use the flag for a one-off run, such as reporting under a temporary name while testing.

Changing the component in the configuration file takes effect at the next configuration check, and the agent reconnects so that it announces itself under the new name. Measurements recorded under the old name stay on the server; they do not move.

4.4 Changing settings later

The agent checks its configuration files every ten minutes and re-reads them when they have changed. A restart is not required.

Most changes take effect on the next measurement with no interruption: the measurement-type settings, the thresholds and the logging level are all read afresh each time the agent reports. Switching a type off and on again does not produce a spike covering the period it was off.

Changing `Server` or `Port` moves the agent to a different SynchroServer, so it does reconnect, leaving a gap of a few seconds.

Changes are picked up within ten minutes. To apply one immediately, restart the agent.

If an edited file cannot be read, the agent logs the problem and carries on with the settings it already had, so a mistake in an edit does not stop it reporting.

5. Starting and stopping

```
~/SynchroAgent/bin/start-client.sh
~/SynchroAgent/bin/stop.sh
```

Starting prints the process id and the component name. Stopping closes the connection to the server cleanly.

The agent does not start automatically when the machine reboots. If it should, add `start-client.sh` to whatever your site uses to start user services.

On Windows the agent is a service, so use the service controls instead:

```
Start-Service SynchroAgent
Stop-Service SynchroAgent
Get-Service SynchroAgent
```

It starts automatically at boot, and restarts itself after 30, 60 and then 120 seconds if it ever exits unexpectedly. Stopping is graceful: the agent tells the server it is going before closing the connection.

To remove the service entirely:

```
C:\SynchroAgent\bin\synchroagent.exe -uninstall-service
```

6. Checking that it works

6.1 On the monitored machine

Look at the newest log file in `log` – `~/SynchroAgent/log` on Linux and macOS, `C:\SynchroAgent\log` on Windows:

```
tail ~/SynchroAgent/log/SynchroAgent.*.stdout
```

```
Get-Content (Get-ChildItem C:\SynchroAgent\log\*.log |
    Sort-Object LastWriteTime | Select-Object -Last 1).FullName -Tail 10
```

A Windows service has no console, so the agent writes its own log file there. Elsewhere the start script redirects its output.

Three lines within a second of starting mean everything is working:

```
msg=starting version=1.1.5 built=20260923 component=HOSTNAME server=...
port=9123
msg=connecting address=...:9123
msg=configured interval=1s dumpconf=true
```

The third line is the one that matters. `configured` means the server accepted the agent and told it how often to report. If it is missing, the agent reached the server but got no reply.

Check that it is using a sensible amount of the machine:

```
ps -o pid,pcpu,rss,etime -p "$(cat ~/SynchroAgent/run/SynchroAgent.pid)"
```

Expect a few percent of one processor and under 15 MB of memory.

6.2 On SynchroServer

Measurements for the machine appear under its component name. They may take up to ten minutes to be written to disk for the first time; this is normal and does not mean anything is wrong.

6.3 Seeing the measurements locally

To see exactly what a machine will report, without involving the server:

```
~/SynchroAgent/bin/synchroagent -once -interval 5s
```

This takes two readings the given interval apart and prints the result, one measurement per line. Two readings are needed because most measurements describe change over time.

7. If SynchroServer is unavailable

The agent is built to outlive the server. It never gives up and never needs to be restarted when the server comes back.

7.1 What happens

If the server is down, the network fails, or the connection drops, the agent closes the connection, waits **60 seconds**, and starts again: connect, identify itself, ask for its configuration, resume reporting.

This repeats for as long as necessary. A server down for a minute and a server down for a week are handled identically; only the number of attempts differs.

Each attempt writes one line to the log, roughly 1,440 lines per day during an outage. The agent's memory use does not grow while it is disconnected.

Four situations are detected at different speeds:

Situation	Noticed after
Server not running	immediately
Server unreachable	30 seconds
Server connects but does not respond	10 seconds
Connection silently dropped, e.g. by a firewall	9 minutes

The last case matters on long-running connections. If a firewall discards the connection without telling either end, it still looks open but nothing gets through. The agent gives up on a silent connection after nine minutes rather than waiting indefinitely.

7.2 What happens to the measurements

Measurements are not stored while the server is unavailable. The agent reports only while connected, so an outage leaves a gap in the graphs covering exactly the period the server could not be reached.

This is intentional. Storing them would consume steadily more memory during a long outage, and the server rejects measurements that arrive too late anyway.

When the connection returns, the agent takes a fresh reading before reporting again. The first measurement after an outage therefore describes the interval just measured, not the whole outage, so graphs show a clean gap rather than a false spike.

7.3 What the log looks like

```
msg=session ended err="dial tcp 10.0.0.5:9123: connect: connection refused"
msg=connecting address=10.0.0.5:9123
...
msg=configured interval=1s dumpconf=true
```

One pair of lines per minute while the server is away, then `configured` the moment it returns. Nothing needs to be done by hand.

8. Platform notes

8.1 macOS

macOS is fully supported and installs exactly as described in section 3, on macOS 13 Ventura or newer running on Apple silicon.

macOS reports fewer measurements than Linux, because the operating system makes less information available: there is no interrupt, I/O-wait or processor-steal accounting, no swap or commit detail, and no per-process virtual size worth reporting. Section 9 shows exactly what each platform provides.

To have the agent start when the machine boots, use a `launchd` agent; this is not supplied.

8.2 Windows

Windows is supported from Windows Server 2016 or Windows 10 onward, as described in section 2.1. Install it with `install.ps1` (section 3.3).

The agent runs as a **Windows service** named `SynchroAgent`, registered to start automatically as `LocalSystem`. It is a single executable: there is no service wrapper, no runtime and no other file to deploy.

It reports everything the other platforms do, plus the memory, paging and kernel-pool detail in section 9.9 that has no equivalent elsewhere.

Two Windows behaviours are worth knowing:

The machine name. Windows machines built from an image often have a generated host name such as `WIN-55EP3L28376`. That would become the component, so set `Component=` explicitly (section 4.3). The real host name is still recorded on the server in the agent's `WindowsInfo` report, so nothing is lost.

Disk measurements are read directly from each volume rather than through a general-purpose library, because the usual route truncates service times to whole seconds and so reports nothing at a normal sampling interval. A volume that does not answer, such as a virtual drive, is skipped and the rest are still reported.

9. Metric dictionary

9.1 Key structure

Every metric is a three-part key plus a value:

```
TYPE_METRIC_TARGET=value
```

The component is not part of the key; it comes from the connection. Values are decimal with two places. A metric at or below its threshold is not sent, so a series can be absent simply because the quantity was zero.

TYPE	Covers	TARGET is
`HOST`	whole machine	always `HOST`
`CORE`	one logical CPU	`cpu0`, `cpu1`, …
`PROC`	processes sharing a name	the process name
`DISK`	block devices	device name
`FS`	file systems	mount point
`NET`	network interfaces	interface name
`SYNCHROAGENT`	the agent itself	always `internal`

`FS` and `DISK` answer different questions. `FS` is about file systems: a mount point, its size and how full it is, including tmpfs mounts such as `/dev/shm` that live in RAM. `DISK` is about block devices: transfer rates and service times. A tmpfs never appears under `DISK`; a RAID member never appears under `FS`.

A metric absent from a platform below is not collected there at all, rather than reported as zero.

9.2 HOST — processor

All values are percentages of the interval.

Metric	Meaning	Linux	macOS	Windows
`CPU`	busy time, i.e. 100 minus idle	yes	yes	yes
`USER`	user mode	yes	yes	yes
`SYSTEM`	kernel mode	yes	yes	yes
`IDLE`	idle	yes	yes	yes
`NICE`	low-priority user mode	yes	yes	–
`WAIT`	blocked on I/O	yes	–	–
`STEAL`	time taken by the hypervisor	yes	–	–
`HINTR`	hardware interrupts	yes	–	–
`SINTR`	software interrupts	yes	–	–

Metric	Meaning	Linux	macOS	Windows
INTR	interrupts, not split	–	–	yes

These are reported even when zero, so a flat WAIT line means no I/O wait rather than a gap in collection.

9.3 HOST — memory

Metric	Meaning	Linux	macOS	Windows
MEM, MEM–PCT	memory in use, percent	yes	MEM only	yes
MEM–MB	memory in use, MB	yes	–	yes
USED–PCT, USED–MB	everything not free, cache included	yes	–	yes
CACHED–PCT, CACHED–MB	reclaimable page cache	yes	–	yes
COMMITTED–PCT	commit charge against the commit limit	yes	–	yes
COMMITTED–MB	commit charge	yes	–	yes
SWAP, SWAP–PCT, SWAP–MB	swap or pagefile in use	yes	–	yes

> **MEM is not the same quantity on every platform, deliberately.** On Linux it is physical memory in use excluding reclaimable cache. On macOS it is physical memory in use. On Windows it is commit charge against the commit limit, which can exceed physical memory and is the figure that governs whether allocations succeed. Each follows its own platform's convention.

9.4 CORE

The same processor metrics as section 9.2, one set per logical CPU, targeted `cpu0` through `cpuN`. Useful for spotting a single saturated core that a host average hides. Available on all three platforms, with the same per-platform differences as the host set.

9.5 PROC

Processes sharing a name are summed: twelve `svchost.exe` are reported once. A name is emitted only when its CPU or memory clears the threshold, and then the whole group is sent together.

Metric	Meaning	Linux	macOS	Windows
`CPU`	percent of one CPU-interval	yes	yes	yes
`RSS`	resident set as percent of total	yes	yes	yes
`RSSMBYT`	resident set, MB	yes	yes	yes
`MEM`	virtual size (Linux) / private bytes (Windows), percent	yes	–	yes
`MEMMBYT`	the same quantity in MB	yes	–	yes

On Windows `MEM` is private bytes, the process's own commit charge. It is the figure that tracks a leak: working set shrinks under system memory pressure while private bytes does not, so a leaking process can show a falling `RSS` while `MEM` climbs.

> **Windows: `PROC_RSS` reads lower than it did under the previous agent.** The agent this replaces added the first instance of each process name to its own total, counting it twice. A process with a single instance was therefore reported at roughly double its

> real memory use; one with many instances was closer to correct, since the error is one extra instance divided across all of them.
>
> The values reported now are the correct ones, so there is a step down in the series at the point of upgrade. Any alarm threshold carried over from the previous agent will fire more readily than it used to and should be reviewed. The size of the change depends on the process: close to half for a single-instance service, negligible for something like `svchost` with twenty.

9.6 DISK

Metric	Meaning	Linux	macOS	Windows
`TPS`	transfers, read plus write	yes	yes	yes
`RSEC` / `WSEC`	reads / writes completed	yes	yes	yes
`RKBYTSEC` / `WKBYTSEC`	KB read / written	yes	yes	yes
`RMILLISEC` / `WMILLISEC`	time spent reading / writing	yes	yes	yes
`IOMSEC`	time with I/O in flight	yes	yes	yes
`IOQUEUELEN`	requests in flight	yes	–	yes
`STIME`	average service time	yes	yes	yes
`AWTIME`	average wait per transfer	yes	yes	yes
`RMERGESEC` / `WMERGESEC`	merged requests	yes	–	–
`WEIGHTED_IO_MSEC`	queue-weighted I/O time	yes	–	–

Request merging is a Linux concept and has no Windows or macOS equivalent.

> **Units differ by platform.** On Linux and macOS these are per-second rates. On Windows they are per-interval deltas, despite the `SEC` in the names. At a one-second interval the two are identical; at any other interval they are not.

9.7 FS and NET

Metric	Meaning
`FS_SIZE-GB`	file system size
`FS_USED-GB`	space used
`FS_AVAIL-GB`	space available
`FS_USAGE-PCT`	percent used
`NET_RXMBYTSEC` / `NET_TXMBYTSEC`	MB received / sent
`NET_RXKPCKSEC` / `NET_TXKPCKSEC`	thousands of packets received / sent

All are available on every platform.

File system targets are mount points: `/` is reported as `root`, and deeper paths are flattened, so `/boot/efi` becomes `boot-efi`. On Windows they are drive letters, reported as `Logical-C:`.

Network targets are the interface name on Linux and macOS – `eth0`, `en0`. On Windows they are the adapter's hardware description rather than its friendly name, so an interface appears as `HPE-Ethernet-10-25Gb-2-port-64` rather than `Ethernet`. Two reasons: the friendly name defaults to "Ethernet" on every machine and tells you nothing about the hardware, and the description is what Windows' own performance counters are keyed by. Targets are truncated to 30 characters, which is why a long adapter name appears clipped.

9.8 SYNCHROAGENT — the agent reporting on itself

Target is always `internal`. Available on every platform.

Metric	Meaning
`METRICRATE`	number of metrics in the sample just sent, counting itself
`LATENCY-MS`	milliseconds to collect and transmit one sample
`VERSION`	build date as `YYYYMMDD`

`METRICRATE` and `LATENCY-MS` describe the previous cycle: a sample cannot carry the cost of transmitting itself.

`LATENCY_MS` compared against the sampling interval is the cheapest measure of what the agent costs. At 34 ms against a one-second interval the agent uses about 3.4 % of one CPU; at a ten-second interval the same work would cost a tenth of that.

The server separately derives `HOST_METRICRATE_HOST` from the fields it actually received. A persistent gap between that and `METRICRATE` means loss or truncation in transit.

9.9 Windows-only metrics

Windows can exhaust memory in four unrelated ways, and a single “memory used” figure sees only the first.

Failure	Symptom	Metric
commit exhaustion	allocations fail, RAM looks free	`HOST_COMMITTED_PCT`
kernel pool exhaustion	system hangs, RAM plentiful	`HOST_KERNELNONPAGED_MB`, `PROC_POOLNONPAGED_KB`
page-table exhaustion	allocation failures, no obvious cause	`HOST_FREEPTE`
thrashing	commit fine, disk saturated by paging	`HOST_PAGESSEC`

Host:

Metric	Meaning
`COMMITPEAK_MB`	worst commit charge since boot
`KERNELPAGED_MB`, `KERNELNONPAGED_MB`	kernel pools
`POOLPAGEDRES_MB`	resident part of the paged pool
`PAGESSEC`	hard page faults per second
`PAGEINSEC`, `PAGEOUTSEC`	pages read from / written to disk
`PAGEFAULTSEC`	all page faults, soft included
`FREEPTE`	free system page table entries
`SYSCACHERES_MB`	resident system cache
`MODIFIEDPAGES_MB`	dirty pages awaiting write
`STANDBY_MB`	cached pages, reclaimable
`FREEZERO_MB`	genuinely free pages
`SWAPPEAK_PCT`	peak pagefile usage
`PLIST`, `TCNT`, `HANDLES`	process, thread and handle counts

Metric	Meaning
RUNQ	processor queue length
CNTXTSWT	context switches per second

PAGESSEC is the single most useful addition. “Memory is full” is normal on Windows, which caches aggressively. “Memory is full **and** the machine is thrashing” is not, and only the hard-fault rate tells them apart.

MODIFIEDPAGES_MB, STANDBY_MB and FREEZERO_MB decompose what Windows counts as available memory into dirty, cached and free.

Per process, when PROC_EXTENDED is on:

Metric	Meaning
RSSPEAK_MB	peak working set
MEMPEAK_MB	peak private bytes
POOLPAGED_KB, POOLNONPAGED_KB	kernel pool charged to this process
PAGEFAULTSEC	page faults per second

A climbing MEMPEAK_MB against a flat MEM is a process that has already leaked and been trimmed back. The pool figures are the only per-process attribution Windows offers for pool exhaustion: host totals say the pool is growing, these say which process is growing it.

9.10 How many series to expect

A sixteen-core Linux host with two disks, one interface and 120 distinct process names produces roughly:

	Series
HOST	21
CORE	144
DISK	28
NET	4
FS	32
PROC	600
SYNCHROAGENT	3
Total	**~830**

Process metrics dominate. `PROC=no` removes most of the volume; `CORE=no` removes most of the rest. macOS produces fewer, Windows more.

10. Troubleshooting

No `configured` line in the log. The agent reached the server but got no configuration message within ten seconds. Check that the port is the stat port and not the probe-server port.

`dial tcp: connection refused`. Nothing is listening. The agent retries every 60 seconds indefinitely; no restart is needed once the server returns. See section 7.

A gap in the graphs. The agent does not buffer during an outage, so a period when the server was unreachable produces a gap rather than a backfill. The log will show one reconnect attempt per minute covering exactly that period.

Component directory never appears on the server. The agent is connecting under a different name than expected. The component appears in the first log line; confirm it against `SYNCHRO_COMPONENT` in the start script.

Metrics stop after a schema change. The server creates a directory per metric path and never removes one. After changing agent versions, stale directories remain with old timestamps. Compare modification times to tell live paths from abandoned ones.

CPU higher than expected. Check `LATENCY_MS` against the sampling interval. The dominant cost is per-process collection, which scales with the process count; `PROC=no` or a longer interval both reduce it.

A metric is missing. Values at or below the threshold are not sent, so an idle device or a zero-valued counter produces no series at all. Metrics listed as unavailable for a platform in section 9 are not collected there and will never appear.