

Headless / post-acquisition reporting

Quanto.Console CLI: import, Analyze, save method, report/export without the WPF UI.

Use **Quanto.Console** when acquisition is finished and you want import → Analyze → report/export **without opening the interactive WPF app**. Typical cases: a scheduled task after the instrument writes a sample folder, a lab automation job, or a review PC that only needs batch reporting.

Quanto.Console is a **Windows console CLI** (`Quanto.Console.exe`). It is **not** a Quanto application server, and it is **not** a replacement for your acquisition software. The WPF UI remains the interactive product for method design, review, and report workbook editing.

Prerequisites

Item	Notes
Install	Same Quanto setup as the desktop app (Program Files). Console ships with the product from 1.0.0.49 / <code>quanto-setup-alpha-49-*</code> onward
OS	Windows 10/11 64-bit (same add-in and vendor stack as WPF)
License	Instrument licenses follow the same rules as WPF; missing licenses still block batch save
Reports	Create report / create export need Spread next to the exe (included in a normal install). Method/batch must define report or export workbooks
Method	A <code>.qtm</code> / <code>.qtb</code> (or vendor method) that already has the compounds, calibration rules, and optional ReportTemplates you want

Help on any machine:

```
"C:\Program Files\Quanto\Quanto.Console.exe" --help
```

What headless is for

1. Acquisition writes raw data to a sample folder (or sequence folder).
2. A scheduled or scripted step runs Quanto.Console with a fixed method and that data path.
3. Console imports method + data, runs Analyze, optionally saves `.qtm` / `.qtb` , and creates defined reports/exports.
4. Analysts open the `.qtb` or the report folder in WPF when they need interactive review.

Typical commands

Full default pipeline (import → analyze → save-method → report → export):

```
"C:\Program Files\Quanto\Quanto.Console.exe" --method "D:\Methods\Assay.qtm" --data "D:\Data\Seq2026-09-20"
```

Post-acquisition report only (method and data are still loaded first):

```
"C:\Program Files\Quanto\Quanto.Console.exe" --method "D:\Methods\Assay.qtm" --data "D:\Data\Seq2026-09-20" --report
```

Named actions in one flag:

```
"C:\Program Files\Quanto\Quanto.Console.exe" -m "D:\Methods\Assay.qtm" -d "D:\Data\Seq2026-09-20" -a import,analyze,report
```

Positional form:

```
"C:\Program Files\Quanto\Quanto.Console.exe" "D:\Methods\Assay.qtm" "D:\Data\Unk1.D" "D:\Data\Unk2.D" --analyze --report
```

Listfile (fixed method, varying data paths — do **not** combine with `--method` / `--data`):

```
"C:\Program Files\Quanto\Quanto.Console.exe" --listfile "D:\Jobs\nightly.listfile" --report
```

Example listfile (`Examples\example.listfile` next to the exe when shipped):

```
# UTF-8; blank lines and # / ; comments ignored
method D:\Methods\Assay.qtm
data D:\Data\2026-09-20\Cal1.D
data D:\Data\2026-09-20\Unk1.D
data D:\Data\2026-09-20
```

A `data` path that is a folder but **not** itself a single sample imports every sample discovered in that folder. Relative paths in a listfile are resolved against the listfile directory.

Actions

Action	Flag	What it does
Import	--import	Load/import the method and add data files
Analyze	--analyze	Get signals → Integrate → Identify → Calibrate → Quantitate (same pipeline as WPF Analyze)
Save method	--save-method	Write the processed method as .qtm
Report	--report	Create defined reports only
Export	--export	Create defined exports only

Or --actions / -a with a comma-separated list: import,analyze,save-method,report,export .

Default when no action flags are given: all five, in that order.

Notes:

- Method and data are **always** loaded from the input mode. --import is the explicit name for that step; later actions still import first.
- report / --export run only workbooks that define Q.Report* / Q.Export* flags. With no definitions they are a quiet skip in the default pipeline.
- If you request --report or --export **explicitly** and they cannot run (no Spread, or nothing defined that can execute), the process exits **non-zero**.

Input modes (pick exactly one)

- method / -m plus one or more --data / -d , or positional <method> <data> [...]
- listfile / -l — do not combine with method/data

Method may be .qtm , .qtb , or a vendor method file/folder. Data may be a sample file/folder.

Where outputs go

Output	Flag / default
Method .qtm	--method-out — default: overwrite a .qtm input, or <method-name>.qtm next to the batch
Batch .qtb	--batch-out — default: {data}\QuantoResults\{yyyy-MM-dd-NN}.qtb
Report / export root	--output / -o — default: the batch data folder

Reports and exports from one create-run share one timestamp folder (Excel and PDF together), under a structure like QuantoReports\{batch}_{Batch|Sample|...}_{yyyy-MM-dd_HH-mm-ss} — same idea as WPF create report.

Report workbook templates live in the method/batch package under method/ReportTemplates/ (inside both .qtm and .qtb). Design and define those workbooks in the interactive WPF Report Designer first; Console only runs what is already defined.

Logging

Console writes path and timing lines to stdout (no dialogs). Typical lines include import paths, Analyze stage text (ASCII arrows), saved .qtb / .qtm paths, and report/export folder paths with elapsed times (milliseconds under 1 s, otherwise 1.5 s / 2 min 3 s).

Indexed Agilent conversion prompts, Continue dialogs, and license nags are auto-continued or skipped. Missing instrument licenses still block batch save, same as WPF.

Exit codes

Code	Meaning
0	Success
Non-zero	Parse error, missing files, import/analyze/save failure, or an explicit --report / --export that cannot run

Use the exit code in Task Scheduler or your orchestration script.

Console vs interactive WPF

	Quanto.Console	Quanto WPF
Purpose	Headless post-acq pipeline and reporting	Interactive method design, review, designer
UI	None (stdout logs)	Full desktop UI
Server role	None — still a local Windows process	None — still a local Windows process
Acquisition	Does not acquire; reads finished data	Does not acquire; reads finished data

Do **not** treat the WPF app as a headless batch server. For unattended Analyze / report after acquisition, schedule **Quanto.Console** on a Windows PC that can see the method and data (local disk preferred; see [Hardware and performance](#)).

Performance check (single-sample post-acq)

Use one finished sample as a post-acquisition smoke test with the **default** pipeline (import → analyze → save-method → report → export — no action flags). Stdout already prints stage timings (`Import:` , `Analyze:` , `Report:` , `Export:` , `Job:`). For peak process memory, sample Working Set / Private Bytes while the process runs (Task Manager, Performance Monitor, or a short PowerShell sampler).

Agilent MassHunter VOA example (ships with QuantExamples):

```
"C:\Program Files\Quanto\Quanto.Console.exe" -m "D:\MassHunter\Data\QuantExamples\MS\VOA\QuantoVoaMethod.qtm" -d "D:\MassHunter\
```

Measured reference (Quanto 1.0.0.50, 2026-09-20)

Local disk, Release `Quanto.Console` with AddIns, default actions, one sample:

Metric	Value
Import	242 ms
Analyze	468 ms
Save batch	53 ms
Save method	65 ms
Report	2.9 s
Export	201 ms
Job (stdout)	5.4 s
Wall clock (process start → exit)	~5.6 s
Peak Working Set	~253 MB
Peak Private Bytes	~400 MB
Workload	1 sample · 74 analytes · 209 signals · 224 peaks · 84 results

Cold start includes add-in load. Report/export are no-ops when the method has no defined workbooks; this VOA method creates Excel+PDF report and a TSV export. Re-run on the same PC after method or hardware changes. For larger multi-sample Analyze numbers see [Hardware and performance](#).

Troubleshooting

Symptom	What to check
<code>--help</code> / unknown exe	Confirm install path; Console is next to the desktop product from builds that ship it
Analyze fails / missing integrator	Same add-ins as WPF under the install folder; repair/reinstall if <code>AddIns</code> is incomplete
Report skipped in default run	Method/batch has no defined <code>Q.Report*</code> workbook — open in WPF Report Designer and define report
Explicit <code>--report</code> fails	Spread assemblies must be present (normal install); Windows Desktop runtime; defined report must be runnable
Empty sample set	Data folder may not contain recognizable samples; point at a sample <code>.D / .dx / .wiff</code> / etc., or a folder of samples
Listfile parse error	UTF-8; one <code>method / data</code> directive per line; do not mix <code>--listfile</code> with <code>--method / --data</code>
Batch not saved	License / instrument serial rules — same as WPF
Slow on network share	Copy study data local for Analyze; size the PC per Hardware and performance

Related

- [Silent deploy](#) — roll out the install that includes Console
- [Installation and folders](#) — Program Files layout, `.qtm` / `.qtb` , ReportTemplates
- [Hardware and performance](#) — local disk vs file server for Analyze
- [Reports, import, and export](#) — designing report workbooks in WPF
- [Licensing](#)