

# Agilent MassHunter

.D / AcqData, MSTree2.bin index, AgilentMassHunterDataProvider, quant method vs method-from-data.

How Quanto reads Agilent MassHunter .D folders (AcqData layout), builds or uses the MSTree2.bin index, and imports MassHunter quant methods or methods from data. For classic ChemStation .D , see [ChemStation / MSD](#). For OpenLab CDS ( .dx / .pmx ), see [OpenLab CDS](#) — do not treat OpenLab as MassHunter.

## Raw data layout

| Item                | Notes                                                                                                                                                 |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| Sample              | .D folder (MassHunter acquisition)                                                                                                                    |
| Acquisition data    | AcqData\ inside the sample folder                                                                                                                     |
| Index (for extract) | sample.d\AcqData\MSTree2.bin                                                                                                                          |
| Modes               | Detector classifies <b>SingleQuadrupole</b> vs <b>QQQ</b> / <b>MRM</b> / <b>SIM</b> / <b>SCAN</b> from the embedded acquisition method and scan types |

Reader add-in: AgilentMassHunterDataProvider (DLL AgilentMHData.dll ). Vendor native libraries ship under:

C:\Program Files\Quanto\AddIns\VendorSupportFiles\Agilent\

Do not mix these DLLs with another Quanto build.

## Indexing and MSTree2.bin

MassHunter extract needs the index file:

sample.d\AcqData\MSTree2.bin

If it is missing, Quanto’s built-in conversion writes MSTree2.bin **into the sample** .D . The Windows account needs **write** permission on that folder (and parent share). ChemStation .D does **not** use this step — see contrast below.

Typical conversion outcomes:

| Outcome                | Meaning                                                                                                      |
|------------------------|--------------------------------------------------------------------------------------------------------------|
| <b>Success</b>         | Index written; extract can proceed                                                                           |
| <b>Already indexed</b> | MSTree2.bin (or equivalent) was already present                                                              |
| <b>Error strings</b>   | Corrupt MSScan or conversion failure — repair or re-acquire the .D , or re-run conversion on a writable copy |

First-time indexing is usually well under a few seconds with Quanto's native indexer (see table below and [Hardware and performance](#)). Antivirus scanning of \*.d / MSTree2.bin during Analyze is a common cause of slow or failed conversion.

### Native indexer vs TDAConverterConsole (measured)

Bench on laptop L (2026-09-16): fresh copies of each .D, wipe MSTree2.bin, then time Agilent TDAConverterConsole.exe vs Quanto AcqData.IndexConvert (overwrite, parallel). Two runs per mode; table shows the faster run.

| Mode                 | Sample                            | MSScan.bin | TDAConverter | Quanto native | Speed-up |
|----------------------|-----------------------------------|------------|--------------|---------------|----------|
| <b>MRM</b> (LC QQQ)  | 10ppb.d (FastPositive demo)       | 11.6 MB    | 2.60 s       | 0.12 s        | ~21×     |
| <b>SIM</b> (GC-MSD)  | B2092-Pesticide-SIM-fromQuant-1.d | 0.82 MB    | 0.55 s       | 0.019 s       | ~29×     |
| <b>SCAN</b> (GC-MSD) | PesticideScan-B2079-5977-4.D      | 0.58 MB    | 0.34 s       | 0.050 s       | ~7×      |

Quanto's built-in path does not need MassHunter Quant installed. Use vendor TDAConverterConsole.exe only when you prefer pre-indexing on an acquisition PC or when a lab policy requires the Agilent tool.

### When MassHunter Quant / TDAConverter is needed

Quanto does **not** install Agilent MassHunter. Built-in conversion writes MSTree2.bin when the sample folder is writable. Lab checklist guidance: install MassHunter Quant / TDAConverterConsole.exe **only if** instruments do not already write MSTree2.bin and you prefer vendor-side indexing (for example on the acquisition PC) before analysts open read-only copies. For read-only network snapshots, copy to a writable study folder or pre-index on the acquisition PC.

### Method import

| Source                             | Add-in (class)              | What you point at                                                             | What comes in                                                   |
|------------------------------------|-----------------------------|-------------------------------------------------------------------------------|-----------------------------------------------------------------|
| MassHunter quant method            | AgilentMhMethod             | .m method folder ( ... \DaMethod\Quant\quantitative.xml ) or .quantmethod.xml | Compounds, ISTD flags, transitions, expected RT, curve hints    |
| MassHunter method <b>from data</b> | AgilentMassHunterDataMethod | .D data folder                                                                | Method inferred from acquisition / MRM content in the data file |

Import: Method → Load / Import — Replace or Append ([Method design](#)). Vendor methods do **not** set Quanto's Calibration mode; new work defaults to Initial calibration unless you change it.

## SQ vs QQQ / SIM / MRM / SCAN

For MassHunter .D, Quanto classifies single quadrupole vs triple quadrupole (QQQ) and the acquisition modes **SIM**, **MRM**, and **SCAN** from the embedded acquisition method and scan types. Internally, chromatogram extraction uses `ChromatogramAcquisitionMode: Auto, Scan, Sim, MsMs`. **Auto** detects from the file and tries alternates if a channel comes back empty.

If a chromatogram stays empty: confirm the Quant ion / transition exists in that sample; force Scan / Sim / MsMs if Auto picked the wrong mode; confirm `AcqData\MSTree2.bin` exists or conversion returned Success / Already indexed.

## Contrast: ChemStation (no index step)

| Kind              | Provider                                    | Needs                                                                 | Conversion                                  |
|-------------------|---------------------------------------------|-----------------------------------------------------------------------|---------------------------------------------|
| MassHunter        | <code>AgilentMassHunterDataProvider</code>  | <code>AcqData\</code> (+ <code>MSTree2.bin</code> for indexed access) | Writes <code>MSTree2.bin</code> if missing  |
| ChemStation / MSD | <code>AgilentChemStationDataProvider</code> | <code>data.ms</code> and/or <code>dataSim.ms</code>                   | Native MSDChem — <b>no</b> index conversion |

Both use .D folders but different layouts. Waiting for index conversion on ChemStation data, or opening MassHunter data without `AcqData/index`, causes empty or failed extracts. Details: [ChemStation / MSD](#).

## OpenLab CDS is separate

OpenLab CDS uses .dx sample/result data and .pmx processing methods via `AgilentOpenLabDataProvider` / `AgilentOpenLabMethod`. That is not MassHunter. See [OpenLab CDS](#) and [Compatibility](#).

## Related

- [Compatibility](#) — capability checklist and method importers
- [Installation and folders](#) — permissions, Vendor extras, antivirus exclusions
- [Troubleshooting](#) — `AcqData`, `MSTree2`, `ConvertResult`, MH vs CS provider
- [Agilent ChemStation / MSD](#) — no index step
- [Agilent OpenLab CDS](#) — .dx / .pmx
- [Reports, import, and export](#)