

— MEASUREMENT SYSTEM SPECIFICATION

MDX Technical Brief

MDF4 measurement indexing, high-resolution signal inspection, and local Model Context Protocol integration.

ASAM MDF v3 / v4

Vector BLF / ASC

12 Native MCP Tools

Local-First & Air-Gapped

Automotive test logging exceeds legacy desktop capacity.

Modern vehicle validation records hundreds of synchronous and event-driven signals across multiple vehicle buses, creating gigabyte-scale datasets per drive cycle.

DATA VOLUME

Memory Exhaustion

Traditional calibration viewers attempt to load raw measurement arrays fully into desktop RAM, causing interface lockups and crashes on multi-hour high-bandwidth bus logs.

SIGNAL ALIGNMENT

Desynchronized Channels

Comparing multi-run test sweeps requires manual offset adjustment when ECU timestamps, GPS coordinates, and dynamometer bench clocks drift or sample at disparate rates.

AUTOMATION CEILING

Proprietary Script Walls

Automating routine trace reviews requires proprietary COM scripts or closed desktop macros, preventing direct inspection by modern AI agents and command-line pipelines.

Local-first measurement analysis with agent connectivity.

MDX is an offline-first vehicle measurement and telemetry platform that couples an oscilloscope-grade signal viewer with a Model Context Protocol interface.

HIGH-FREQUENCY WORKBENCH

Oscilloscope Precision for Engineers

Operates as a standalone engineering station for interactive time-series review. Provides independent signal lane scaling, live dual A/B cursors with interval deltas, windowed FFT analysis, calculated channel expressions, and synchronized GPS path tracing.

STANDARDIZED AGENT PROTOCOL

MCP Interface for AI Agents

Hosts a local loopback server compliant with the Model Context Protocol. AI assistants inspect channel schemas, query windowed statistics, identify transient threshold anomalies, and generate plots without moving proprietary measurement binaries off the host.

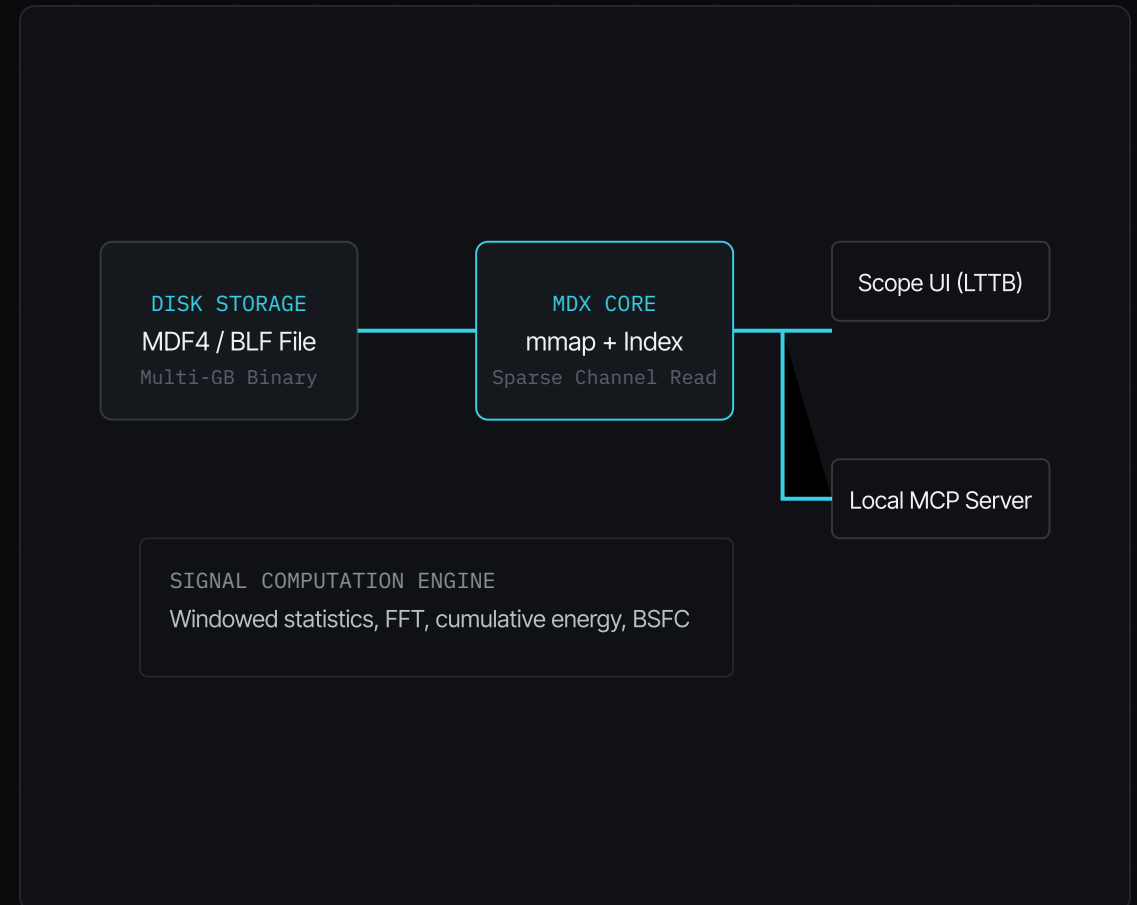
Architectural Principle: Both human calibration engineers and automated AI agents execute queries against the exact same memory-mapped analytics core. Automation is an integrated interface, not a cloud wrapper.

Memory-mapped indexing and LTTB downsampling.

Rather than loading multi-gigabyte log files into memory, MDX memory-maps raw binary blocks directly from disk storage and builds an on-demand signal index.

PIPELINE CHARACTERISTICS

- **Zero-Copy Block Parsing:** Reads MDF4 block headers (ID, HD, DG, CG, CN) without bulk data transposition.
- **LTTB Decimation:** Largest-Triangle-Three-Buckets decimation preserves waveform extrema across millions of data points on fixed-width displays.
- **Non-Linear Interpolation:** Supports Kriging, radial basis functions (RBF), and B-spline surfaces for empirical map construction.



The built-in Model Context Protocol server.

MDX implements the open Model Context Protocol specification over stdio and local Server-Sent Events (SSE), standardizing telemetry access across diverse AI clients.

SUPPORTED CLIENT ECOSYSTEM

Universal Client Interoperability

Any client supporting standard MCP connects locally with a single configuration block. Standardized client endpoints include:

- Claude Code CLI & Claude Desktop
- Google Antigravity & Gemini CLI
- Cursor & VS Code (GitHub Copilot)
- OpenAI Codex CLI & Custom SSE Connectors
- Windsurf & Zed Editor

DATA CONTAINMENT

Zero Raw Telemetry Leakage

Large binary logs remain on the host filesystem. Connected agents receive structured responses: channel schemas, numerical summaries, event timestamp arrays, and downsampled plot vectors.

```
TRANSPORT: stdio / 127.0.0.1 SSE  
EGRESS: 0 bytes outside loopback
```

The twelve native MCP tool definitions.

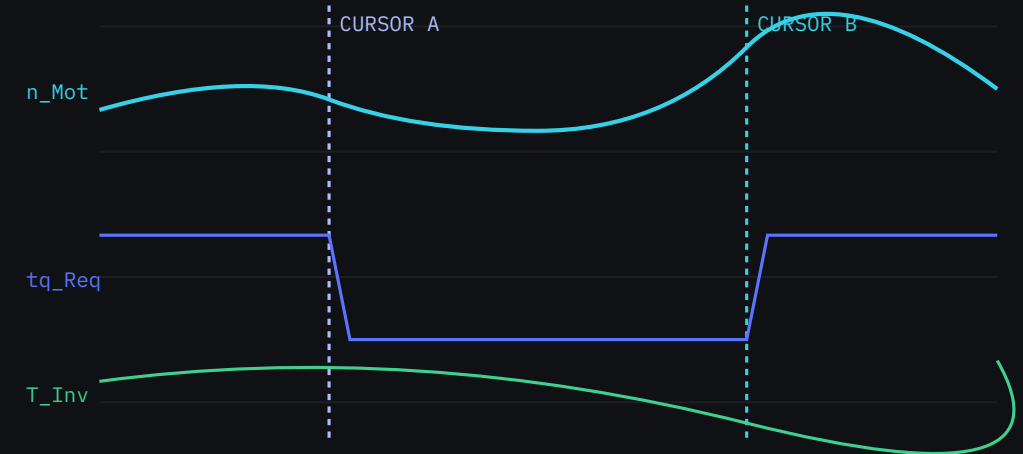
TOOL IDENTIFIER	DESCRIPTION & OPERATIONAL CAPABILITY	RETURN DATA STRUCTURE
mdx_health	Checks server responsiveness, backend status, and active session configuration.	JSON { status, version, uptime }
mdx_list_logs	Enumerates indexed measurement files in the active session workspace.	Array<LogSummary>
mdx_load_log	Initializes memory-mapping for a specific MDF/BLF/ASC file and mounts signal hierarchy.	JSON { log_id, channel_count }
mdx_query_channels	Searches signal channels by exact string match, wildcard, or decoded DBC message.	Array<ChannelMeta>
mdx_signal_stats	Calculates mean, variance, min, max, percentiles, and sample counts over time windows.	JSON { min, max, mean, std, n }
mdx_plot_signal	Returns decimated vector pairs (time/value) suitable for agent visualization.	JSON { t: [], v: [] }
mdx_detect_events	Identifies threshold violations, state transitions, and derivative spikes.	Array<TimestampInterval>
mdx_find_similar_events	Performs cross-correlation matching against a reference waveform pattern.	Array<EventMatch>
mdx_import_config	Imports legacy desktop workspace layouts, signal groups, and display templates.	JSON { imported_groups }

Multi-lane oscilloscope and cursor inspection.

The primary inspection view provides an oscilloscope interface configured for transient signal dynamics and bus validation.

OPERATIONAL CAPABILITIES

- **Independent Signal Strips:** Separate vertical scales prevent signal masking between high-voltage bus lines and microsecond pulses.
- **Dual Time Cursors:** Dedicated A/B cursors compute point-to-point temporal delta, absolute value differences, and slope (dV/dt).
- **Hold-Last-Sample Reads:** Readout tables track precise scalar states across non-uniform sampling rates.
- **Frequency Domain Analysis:** Windowed FFT displays harmonic distribution for structural vibration and motor ripple.



Multi-run alignment and calculated channels.

MULTI-LOG SYNCHRONIZATION

Serial & Parallel Overlay

Evaluates variation across test runs without pre-merging datasets:

- **Serial Concatenation:** Chains sequential drive cycle logs into a single contiguous timeline.
- **Parallel Comparison:** Overlays identical channels from distinct calibration runs.
- **Sync-Signal Alignment:** Automatically calculates time offsets based on trigger signals (such as ignition key-on, accelerator step, or CAN broadcast sync).

SERVER-SIDE FORMULA ENGINE

Calculated Channel Pipeline

Derives dynamic virtual signals on the fly without altering raw binaries:

- **Accumulators & Integrals:** Energy consumption, cumulative fuel mass, and distance.
- **Windowed Statistics:** Rolling RMS, rolling standard deviation, and moving averages.
- **Domain Math:** Brake-Specific Fuel Consumption (BSFC) and electrical inverter efficiency maps.

Exhaustive standard formats and protocol support.

CATEGORY	STANDARD / FORMAT	ACCESS MODE	TECHNICAL SCOPE
Measurement Files	ASAM MDF v3 (.mdf), MDF v4 (.mf4)	Read (Zero-copy mmap)	Full block hierarchy, compressed data blocks, bus logging channels.
Bus Trace Logs	Vector BLF (.blf), ASCII (.asc)	Read & Extract	CAN, CAN FD, LIN message decoding via associated network databases.
Tabular Data	CSV (.csv), Microsoft Excel (.xlsx)	Read / Export	Time-indexed series import, decimated scope vector export.
Network Descriptions	CAN DBC (.dbc), AUTOSAR ARXML (.arxml)	Read (Schema only)	Frame identifier decoding, signal scaling, offset, unit definitions.
Agent Protocols	Model Context Protocol (MCP)	Bidirectional (stdio / SSE)	12 tools covering session, schema, stats, events, and plotting.
Developer SDK	REST API & Python SDK (mdx_client)	Local loopback (127.0.0.1)	Programmatic batch execution, headless anomaly filters, map jobs.

Air-gapped operation and data containment.

PACKAGING

Tauri Native Desktop

Bundles the UI shell and Rust/C++ backend into a single desktop application for macOS, Windows, and Linux. No external web server runtime required.

ISOLATION

No Licence Server

Does not phone home, does not require an active internet connection, and does not depend on a centralized corporate floating license daemon.

MAP TILES

Embedded Offline Maps

GPS route mapping renders directly from local vector map packages embedded within the client, ensuring location telemetry is never leaked to external tile APIs.

Data Residency Guarantee: All measurement analysis, channel computations, and MCP tool executions execute strictly on the local machine (127.0.0.1). Zero telemetry leaves your engineering environment unless explicitly redirected by custom configuration.

What we do not claim.

Accurate engineering evaluation requires explicit boundaries. MDX is built for telemetry analysis and agent-driven inspection, with the following defined limitations:

NOT A HARDWARE LOGGER

No Direct ECU In-Vehicle Flashing

MDX is an analysis environment. It does not replace physical CAN interface hardware (e.g. Vector VN boxes, Kvaser interfaces) or hardware-in-the-loop flash drivers.

NO AUTONOMOUS SIGN-OFF

Human Verification Required

While AI agents can detect anomalies and extract waveform features over MCP, MDX does not certify calibration compliance without engineering review.

LOCAL CONTEXT CEILING

No Cloud Data Indexing by Default

MDX is designed for local workspaces. It does not provide global multi-fleet cloud indexing unless paired with a centralized database pipeline.

NO LIVE XCP RECORDING

Post-Acquisition & Stream Playback

MDX processes recorded measurement files and incoming network telemetry streams; it is not an active online calibration ECU parameter tuner.

— PROOF OF CAPABILITY

How engineering teams evaluate MDX.

Evaluation is straightforward and operates entirely within your security boundary using representative vehicle data.

STEP 1

Local Build Evaluation

Receive a standalone desktop build (Windows, macOS, or Linux). Verify local execution, memory footprint, and air-gapped performance.

STEP 2

Telemetry Ingestion

Load your proprietary MDF4, BLF, or CSV logs along with network DBCs. Confirm signal decoding accuracy and waveform rendering speed.

STEP 3

Agent Tool Testing

Connect your preferred local agent (Claude Code, Gemini CLI, Cursor) over stdio or SSE. Execute diagnostic queries against real telemetry.

TECHNICAL PILOT & ACCESS

Contact Vehicle Lab Engineering Team

contact@vehiclelab.in · <https://vehiclelab.in/docs>