FIBER OPTIC FIELD GUIDE

Read a .sor file without writing a parser

OTDR files are proprietary binaries. If your software needs the length, loss, and event table out of them, here's an API that does the parsing in one HTTP call, the same engine behind the free Fiber Analyzer.

Johnstone Technical Services · https://johnstonetechs.com/guides/parse-otdr-sor-files-api

If you were handed a folder of .sor files and told to extract the span length, total loss, and the event table into your own system, you already know the problem. There's no clean, open spec that gets you all the way there. Bellcore SR-4731 covers the base structure. The real files in the real world add vendor extensions, proprietary blocks, bit-packing quirks, and the occasional format that isn't Bellcore at all. EXFO uses .trc. Multi-wavelength files add their own wrapper as .msor. Every brand of OTDR has done something a little different.

The result is predictable. The first week looks tractable. The second week turns into reverse-engineering someone else's hex dumps. By the third week you understand why the vendors sell their own report software.

There's an easier path.

The API

The parsing engine behind the free Fiber Analyzer and Fiber Analysis tools is available as a REST API at johnstonetechs.com/developers. The same code that powers the browser tools runs behind the endpoint. If you've ever opened a trace in the Fiber Analyzer and seen the length, loss, and event table come back correctly, that's the output you get through the API, as JSON.

Two endpoints are live:

POST /api/v1/parse takes one trace file and returns one structured result. Supported formats are .sor and .msor (Bellcore SR-4731) and EXFO .trc. The response includes span length, total loss, worst reflectance, the full event table (every connector, splice, and reflection with distance, loss, and reflectance), and instrument metadata (make, model, wavelength, pulse width, averaging time).

POST /api/v1/analyze takes a whole job (many files, or a zip) and isn't limited to OTDR traces. A real job is a mix, and this endpoint reads that mix: OTDR traces, OLTS and iOLM link-loss results, endface inspection files, and optical power-meter files. It groups the files into fibers, runs the same pass/fail logic as the Fiber Analysis tool, and returns a per-fiber verdict: passing, fail, no-continuity, or not-tested. Each fiber comes back with a plain-language summary and the measured numbers. This is the endpoint that replaces a human opening hundreds of files by hand.

Full reference docs, request and response shapes, and request-a-key instructions are at /developers. What follows is enough to see what it feels like to call.

What you get back

A parse response gives you the measured results and the instrument context together, in one JSON payload. The structure matches what you'd see in the browser Fiber Analyzer: a fiber-level summary (length, total loss, worst reflectance, overall pass/fail against standard thresholds), the event list, and the metadata block that identifies the test equipment and settings. See the reference for the full schema.

The analyze response adds a layer on top: per-fiber verdicts, reasoning across the whole job (resolving each file's identity, flagging conflicts where the filename and the file's internal ID disagree, handling bidirectional shots, and listing any files it couldn't place), and the actionable summary that the Fiber Analysis tool produces for its users.

One minimal example

Base64-encode a file and POST it:

That's the shape of the request. One file in, structured JSON out. See /developers for the full request schema, the response reference, and the error codes.