

1. Lines, Java and artifacts			
Line	Latest	Java	Status (2026-10-13)
4.x	4.1.0	17+	Active; no 4.2 release yet
3.x	3.3.2	11+	Maintenance; ends Jan 2027 (wiki) or Mar 2027 (roadmap)
2.x / 1.x	2.9.4 / 1.28.5	8	End of life; affected by the 2025 XXE CVEs
Artifact (org.apache.tika)		For	
tika-bom 4.1.0		One version for all Tika artifacts (Boot does not manage Tika)	
tika-core		Detection, Parser/Detector, Metadata, TikaInputStream, facade	
tika-parsers-standard-package (<type>pom</type>)		All standard parsers and charset detectors	
tika-parser-pdf-module, -microsoft-, -html- ...		One family each, for a smaller class path	
tika-serialization		tika-config.json and TikaLoader	
tika-pipes-fork-parser		PipesForkParser (parse in a separate JVM)	
tika-langdetect-charsoup		Language detection (recommended)	
tika-vlm, tika-inference		VLM recognizers, embeddings (experimental)	

Keep Tika-core and every parser module at one version (mixing them caused CVE-2025-66516).

2. Which way to run Tika?

Option	Isolation	Use for
Tika facade / AutoDetectParser	None: OOM, hang, crash hit your JVM	Trusted files, tests, tools
PipesForkParser	Forked JVM, restarted after a crash	Java app receiving untrusted files
tika-app	Optional --fork, pipes mode	CLI, scripts, batch folders
tika-server (REST, :9998)	Parses in forked JVMs	Any language, Docker, scaling
tika-grpc (:9090 image)	Forked pipes workers	gRPC clients, streaming
Tika Pipes	Forks + limits, fetcher to emitter	Millions of files, indexing, RAG

Official advice: consider a service first. In-process parsing is not a sandbox.

3. Facade one-liners

```
Tika tika = new Tika(); // one shared instance
tika.setMaxStringLength(-1); // default 100,000 chars
String t1 = tika.detect(Path.of("catalogue.pdf"));
String t2 = tika.detect("notes.txt"); // by name only
String txt = tika.parseToString(Path.of("manuscript.docx"));
try (Reader r = tika.parse(path, metadata)) { /* stream */ }
```

detect(InputStream) consumes the stream in 4.x: wrap it in TikaInputStream and rewind().

4. AutoDetectParser skeleton

```
static final AutoDetectParser PARSER = new AutoDetectParser();
Metadata md = new Metadata(); // per document
BodyContentHandler h = new BodyContentHandler(-1); // no limit
ParseContext ctx = new ParseContext(); // per parse
try (TikaInputStream tis =
    TikaInputStream.get(Path.of("manuscript.docx"), md)) {
    PARSER.parse(tis, h, md, ctx); // 4.x: TikaInputStream
}
String text = h.toString();
String type = md.get(Headers.CONTENT_TYPE);
```

Catch ZeroByteFileNotFoundException, EncryptedDocumentException (supply a PasswordProvider), WriteLimitReachedException and TikaException per file. tk:parsed-by= EmptyParser means nothing parsed it.

5. Detection and media types

DefaultDetector: (1) tk:content-type-override wins; (2) MimeTypes: magic on the first 64 KiB, refined by the name glob (tk:resource-name) and the Content-Type hint; (3) container detectors (ZIP: OOXML, ODF, EPUB; OLE2: doc, xls, .msg); (4) TextDetector fallback; most specific wins.
MediaTypeRegistry: docx -> application/x-tika-ooxml -> application/zip; use isInstanceOf, normalize (aliases such as application/x-pdf). Custom types: custom-mimetypes.xml with sub-class-of. Validate uploads on detected bytes, never on the client's header; application/octet-stream = unknown.

6. Charset detection (4.x chain)

bom-detector -> metadata-charset-detector (HTTP charset) -> html-encoding-detector (<meta charset>) -> mojibuster-encoding-detector (UTF gates, Naive Bayes) -> junk-filter-encoding-detector arbitrates by decode quality (must stay last). Results: tk:detected-encoding.
tk:encoding-detector, tk:encoding-detection-trace.
The 3.x universal/ICU4J detectors need extra jars and explicit config. AutoDetectorReader decodes for you.

7. Content handlers

HANDLER_TYPE	Handler	Names
MARKDOWN (4.x default)	ToMarkdownContentHandler	markdown, md
TEXT	ToTextContentHandler	text, txt
BODY	BodyContentHandler over text	body
HTML / XML	ToHTMLContentHandler / ToXMLContentHandler	html / xml
IGNORE	no-op (metadata only)	ignore

new BodyContentHandler() silently stops at 100,000 chars: pass a limit or -1. Build handlers with BasicContentHandlerFactory(type, writeLimit) or newInstance(type, ctx); combine with TeeContentHandler. Parsers emit XHTML SAX events underneath.

8. Metadata keys (3.x to 4.x)	
3.x	4.x
resourceName	tk:resource-name
X-TIKA:Parsed-By	tk:parsed-by
X-TIKA:content	tk:content
X-TIKA:embedded_resource_path	tk:embedded-resource-path
X-TIKA:EXCEPTION:write_limit_reached	tk:exception:write-limit-reached
X-TIKA:digest:SHA256	tk:digest:SHA-256
pdf:PDFVersion, Message-From	pdf:pdf-version, message:from

Unchanged: Content-Type (constant now
HttpHeaders.CONTENT_TYPE, dc:title, dc:creator, dcterms:created, xmpTPg:NPages. Typed Property keys:
tikaCoreProperties, Office, PDF, XMPDM.
Legacy-key-migration-filter restores 3.x names during a migration; string writes to tk: keys throw.

9. Embedded documents

```
RecursiveParserWrapper w =
    new RecursiveParserWrapper(new AutoDetectParser());
var rh = new RecursiveParserWrapperHandler(
    new BasicContentHandlerFactory(HANDLER_TYPE.TEXT, -1));
w.parse(tis, rh, new Metadata(), new ParseContext());
List<Metadata> docs = rh.getMetadataList(); // container first
```

Same list: tika-server /rmeta, tika-app -J, Pipes RMETA. Each entry has tk:content, tk:embedded-resource-path, tk:embedded-depth, embedded-limits: maxDepth, maxCount (-1 = unlimited), throwOnMaxDepth/throwOnMaxCount (false: flag key instead of throwing).

10. Language detection

```
LanguageDetector d = LanguageDetector
    .getDefaultLanguageDetector().loadModels();
LanguageResult r = d.detect(text); // resets first
r.getLanguage(); // ISO 639-3: "eng", "spa", "fra"
r.getConfidence(); r.isReasonablyCertain();
CharSoup (tika-langdetect-charsoup, recommended) or
OpenNLP. Stateful: one per thread, reset() between documents.
charsoup-metadata-filter writes tk:detected-language.
1.x LanguageIdentifier was removed in 2.0.
```

11. tika-config.json and TikaLoader

```
{ "parsers": [ { "pdf-parser": { "sortByPosition": true },
  { "default-parser": {} } ],
  "parse-context": {
    "output-limits": { "writeLimit": 1000000,
      "embedded-limits": { "maxDepth": 5, "maxCount": 1000,
        "timeout-limits": { "totalTaskTimeoutMillis": 600000,
          "exception-reporting": { "level": "MESSAGE_REDACTED" } },
        "server": { "host": "127.0.0.1", "port": 9998 } }
    }
```

Top-level keys (strict; unknown key fails the load): parsers, detectors, encoding-detectors, metadata-filters, renderers, text-recognizers, engines, inference, translator, content-handler-factory, auto-detect-parser, parse-context, presets, server, grpc, pipes, fetchers, emitters, pipes-iterator, pipes-reporters, plugin-roots, metadata-list, service-loader, xml-reader-utils.

```
TikaLoader loader =
    TikaLoader.load(Path.of("tika-config.json")); // startup
Parser parser = loader.loadAutoDetectParser(); // shared
ParseContext ctx = loader.loadParseContext(); // per parse
```

Secrets only as \${env:NAME} (4.1.0). Keep default-parser or only the named parsers load. 3.x XML: tika-app --convert-config-xml-to-json= (parsers only).

12. PDF and OCR

```
"engines": { "tesseract": { "tesseract-ocr-parser":
  { "language": "eng+spa", "timeoutMillis": 120000 } },
"text-recognizers": [ { "engine": "tesseract" } ],
"parse-context": { "pages": { "text": "AUTO",
  "renderer": { "dpi": 300, "ocr": { "maxPages": 50 } }
}
```

pages. text: EXTRACT, AUTO (OCR pages with too little text), EXTRACT_AND_OCR, OCR, NONE. "text-recognizers": [] = OCR off. pdf-parser: sortByPosition, extractInlineImages, extractAnnotationText, maxPages, accessCheckMode. Tesseract must be installed, or use the apache/tika:4.1.0-1-full image (Tesseract, GDAL, ImageMagick, ffmpeg).

VLM: openai-vlm-parser, claude-vlm-parser, gemini-vlm-parser as engines in text-recognizers; key via \${env:NAME}; cost in vlm: prompt-tokens. Inference: openai-embedding-engine + inference bindings (TEXT, IMAGES, PAGES, MEDIA, task embed) write vectors to tk:chunks. Both experimental.

13. Extending: custom Parser / Detector

Implement Parser (not AbstractParser) or Detector; annotate @TikaComponent (name = "bookshelf-order-parser") with tika-annotation-processor (writes META-INF/services and META-INF/tika/parsers.idx), or hand-write META-INF/services/org.apache.tika.parser.Parser (no JSON name). A later parser for the same type wins. ExternalParser wraps a CLI tool.

14. Isolated parse: PipesForkParser

```
var cfg = new PipesForkParserConfig() // one, shared
    .setTimeoutLimits(new TimeoutLimits(600_000, 5_000))
    .addJmArg("-Xmx256m");
PipesForkParser fork = new PipesForkParser(cfg) {
    try (TikaInputStream tis = TikaInputStream.get(path)) {
        PipesForkResult r = fork.parse(tis);
        r.getStatus(); // PARSE_SUCCESS, TIMEOUT, OOM, ...
    }
}
```

Crash, hang (progressTimeoutMillis) or OOM kills only the fork, which restarts.

15. tika-app flags (4.1.0)	
Flag	Effect
--md (default), -t, -x, -h, -T, -A	Markdown, text, XHTML, HTML, main text, all text
-m, -j, -J, -y, -r	Metadata, JSON, recursive JSON (/rmeta), XMP, pretty
-d, -l, --digest=sha256	Detect type, language (ISO 639-3), digest
-z / -Z, --extract-dir=	Extract direct / all attachments (no --unpack flag)
--config=FILE.json	Load the JSON config with TikaLoader
-f, --task-timeout, --progress-timeout	Parse in a forked JVM, timeouts in ms
-i dir -o dir, -n N, --handler=m	Pipes mode: folders, forks, handler
--list-parser-names, --list-detectors	Component names for JSON

Removed: --server, --client, -ex, -pX, --fork-timeout, --dump-* -config.

16. tika-server

```
java -jar tika-server-standard-4.1.0.jar -c tika-config.json
docker run -d -p 127.0.0.1:9998:9998 apache/tika:4.1.0-1
curl -T file.pdf http://localhost:9998/rmeta/md
```

/tika[/text|html|xml|md], /rmeta[/{handler}], /meta, /detect, /unpack[/all], /language, /parsers, /version, /status (opt-in), /pipes, /async. 429 = fork pool busy: honour Retry-After and retry. 503 = fork died (TIMEOUT, OOM, UNSPECIFIED_CRASH): retry once, then quarantine. A -c file needs a "server" block ({} is enough). allowPipes and allowPerRequestConfig default to false: keep them off.
tika-grpc: apache/tika-grpc:4.1.0-1, port 9090.

17. Tika Pipes

Roles: iterator (lists work, parent) -> fetcher (bytes, fork) -> emitter (writes result) + reporter (status). Unit: FetchEmitTuple.
parseMode: RMETA (default), CONCATENATE, CONTENT_ONLY, NO_PARSE, UNPACK. emitStrategy: DYNAMIC (default), EMIT_ALL, PASSBACK_ALL. PF4J plugins: file system, S3, GCS, Azure Blob, HTTP, JDBC, CSV, JSON, Kafka, Solr, Elasticsearch, OpenSearch.

18. Limits and timeouts (parse-context)

Key	Default
output-limits.writeLimit	-1 (unlimited)
output-limits.maxXmlDepth / maxPackageEntryDepth	100 / 10
output-limits.zipBombThreshold / zipBombRatio	1,000,000 chars / 100
embedded-limits.maxDepth / maxCount	-1 / -1
timeout-limits. totalTaskTimeoutMillis	3,600,000 (1 h; 3.x: 5 min)
timeout-limits. progressTimeoutMillis	120,000 (stall, forks only)
exception-reporting. level	FULL (use MESSAGE_REDACTED)
unpack-config.maxUnpackBytes	10 GiB
server.maxRequestSizeBytes	1 GiB (413 above)

19. Security rules

- Parse untrusted files out of process (PipesForkParser, server, Pipes).
- tika-server and tika-grpc are not security boundaries: trusted network only, never public; bind 127.0.0.1, auth at a gateway, NetworkPolicy.
- External commands (ExternalParser, FileCommandDetector): admins only, absolute paths.
- Secrets via \${env:NAME}; upgrade past CVE-2025-54988 / CVE-2025-66516 (XFA XXE).

20. Spring Boot upload to tika-server

```
// RestClient tika = builder.baseUrl("http://tika:9998")
tika.put().uri("/rmeta/markdown") // no Tika starter
    .contentType(MediaType.APPLICATION_OCTET_STREAM)
    .body(out -> Files.copy(file, out)) // stream from disk
    .exchange((req, res) ->
        switch (res.getStatusCode().value()) {
            case 200 -> MetadataMapper.toExtraction(res.bodyTo(LIST));
            case 429 -> retryAfter(res.getHeaders()); // retry
            case 503 -> throw new DocumentFailedException(...);
            default -> throw new IllegalStateException(...); });
```

Spring AI TikaDocumentReader and LangChain4j
ApacheTikaDocumentParser need Tika 3.x (NoSuchMethodError with 4.x).

21. Search and RAG

Solr: Solr Cell /update/extract with tikaserver backend, or solr-emitter. Elasticsearch / OpenSearch: es-emitter / opensearch-emitter, or the attachment processor. Lucene: index tk:content yourself. RAG: Markdown + /rmeta, one source per embedded document, MarkdownChunker or your splitter.

22. 1.x to 4.x

2.0: parser modules, Tika Pipes, typed metadata, no LanguageIdentifier. 3.0: Java 11, jakarta, PDFBox 3, JSoup HTML. 4.0: Java 17, zip distributions, tika-config.json + TikaLoader (no TikaConfig), TikaInputStream SPI, Markdown default, tk: keys, forked server, PipesForkParser (no ForkParser). 4.1: text-recognizers, presets, metrics, \${env:}.