

THE SAMPLE FILES HANDBOOK

A 100-page sample document

Fourteen chapters about file formats and testing, with bookmarks, a linked table of contents, tables, charts and lists.

Property	Value
Pages	100
Chapters	14
Navigation	Bookmarks + linked contents
Author	Technology Channel
License	Free for testing and development

Contents

Chapter	Title	Page
01	What makes a good test file	3
02	Inside the Portable Document Format	10
03	Images on the web	17
04	Video containers and codecs	24
05	Sound and audio formats	31
06	Structured data: JSON, CSV and XML	38
07	Office documents are ZIP files	45
08	Testing file uploads	52
09	Character encodings	59
10	Archives and messy folders	66
11	Metadata and privacy	73
12	Large files and performance	80
13	Accessible documents	87
14	Automating tests with sample files	94

Select a title to jump to that chapter, or use the bookmarks panel.

CHAPTER 01

What makes a good test file

A good test file is boring in the best possible way. It opens everywhere, it is exactly what its name says, and it behaves the same way every time you use it. When a bug appears, you want to be sure the problem is in your code and not in a half-broken file you found somewhere online. That is why every file in this collection is generated from scratch, checked after it is written, and labelled with the details that matter: size, dimensions, duration, page count or row count.

Good test files also come in families. A single PDF tells you that your viewer can open a PDF; a family of PDFs tells you how it copes with one page and a hundred pages, with a few kilobytes and ten megabytes, with plain text, scanned images, form fields and passwords. Testing across a family is how you find the edges of a system before your users do.

Finally, a good test file is honest about what it contains. Real documents carry personal data, licensed fonts and copyrighted photos. Synthetic files avoid all of that. The names, numbers and pictures in these samples are invented, so you can attach them to bug reports, commit them to a repository or share them with a vendor without a second thought.

1.1 Key points

- A good test file is boring in the best possible way.
- It opens everywhere, it is exactly what its name says, and it behaves the same way every time you use it.
- When a bug appears, you want to be sure the problem is in your code and not in a half-broken file you found somewhere online.
- That is why every file in this collection is generated from scratch, checked after it is written, and labelled with the details that matter: size, dimensions, duration, page count or row count.
- Good test files also come in families.
- A single PDF tells you that your viewer can open a PDF; a family of PDFs tells you how it copes with one page and a hundred pages, with a few kilobytes and ten megabytes, with plain text, scanned images, form fields and passwords.
- Testing across a family is how you find the edges of a system before your users do.

JPEG remains the workhorse for photographs. It uses lossy compression that throws away detail the eye is unlikely to miss, which keeps files small but means every re-save loses a little quality. PNG is lossless and supports transparency, which makes it the usual choice for screenshots, logos and interface graphics, at the cost of larger files for photographic content.

1.2 Sample data

The table below lists fictional test runs. It is here so that table extraction can be checked page after page.

Case	Sample file	Size	Result
TC-0101	sample-100-rows.csv	7.4 MB	Pass
TC-0102	sample.xml	2.8 MB	Pass
TC-0103	sample-5-slides.pptx	5.6 MB	Pass
TC-0104	sample.xml	1.8 MB	Pass
TC-0105	sample-1920x1080.jpg	7.3 MB	Pass
TC-0106	sample-4032x3024.heic	10.4 MB	Skipped
TC-0107	sample-30s.mp3	3.7 MB	Pass
TC-0108	sample.xml	3.3 MB	Skipped
TC-0109	sample-100-rows.csv	3.0 MB	Fail
TC-0110	sample-30s.mp3	10.0 MB	Pass
TC-0111	sample.txt	3.4 MB	Pass
TC-0112	sample-10-pages.docx	8.9 MB	Pass
TC-0113	sample.xml	6.0 MB	Pass
TC-0114	sample-4032x3024.heic	10.2 MB	Pass

1.3 Discussion

PDF was designed to make a document look the same on every screen and every printer. A PDF file is a collection of numbered objects, such as pages, fonts, images and annotations, tied together by a cross-reference table that lets a reader jump straight to any object without reading the whole file. That structure is why a viewer can show page 80 of a long report almost instantly.

H.264 video with AAC audio in an MP4 container is still the safest combination for playback everywhere. For streaming tests, keep a range of resolutions from 360p to 4K and a range of bitrates, because adaptive players switch between them as network conditions change.

XML is more verbose but brings namespaces, schemas and a long tail of enterprise use: RSS feeds, sitemaps, SOAP services and office document internals. Parsers must also be configured carefully, because some XML features can be abused to read local files or exhaust memory.

1.4 Figure

Figure 1. Sample measurements

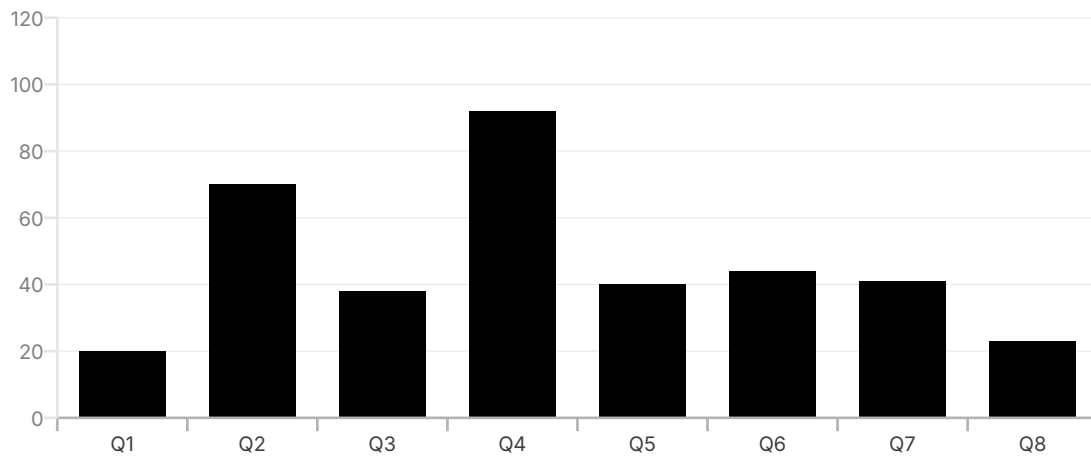


Figure 1: a vector bar chart drawn directly into the PDF. Values are random and seeded, so the chart is identical on every build.

Audio bugs often hide in the details: an unexpected sample rate, a mono file where stereo was assumed, 24-bit samples in a pipeline built for 16-bit, or a long ID3 tag with cover art that pushes the first audio frame further into the file than a naive parser expects.

1.5 Checklist

Work through these steps when you test this area:

- 1 Check audio and video stay in sync to the end.
- 2 Upload the same file twice in a row.
- 3 Compare checksums before and after storage.
- 4 Play a video from the middle using seek.
- 5 Repeat the test on a throttled mobile connection.
- 6 Rename a file so its extension does not match its contents.
- 7 Test a password-protected document.
- 8 Upload a file exactly at the size limit.
- 9 Unpack an archive with deeply nested folders.

1.6 Summary

This chapter looked at what makes a good test file. A good test file is boring in the best possible way. Good test files also come in families. Finally, a good test file is honest about what it contains.

Key takeaway. The names, numbers and pictures in these samples are invented, so you can attach them to bug reports, commit them to a repository or share them with a vendor without a second thought.

When you generate office documents yourself, open them in more than one application. A file that one program tolerates may trigger a repair prompt in another, and that prompt is enough to make users distrust the export feature you worked hard to build.

CHAPTER 02

Inside the Portable Document Format

PDF was designed to make a document look the same on every screen and every printer. A PDF file is a collection of numbered objects, such as pages, fonts, images and annotations, tied together by a cross-reference table that lets a reader jump straight to any object without reading the whole file. That structure is why a viewer can show page 80 of a long report almost instantly.

Text in a PDF is not stored as flowing paragraphs. Each piece of text is placed at exact coordinates on the page, often a few words at a time. Extracting readable text therefore means reassembling lines, columns and reading order, which is harder than it sounds. Multi-column layouts, tables and footnotes are classic sources of extraction bugs, and they are worth including in any test set.

PDFs can also carry much more than text. Interactive forms, bookmarks, links, embedded files, digital signatures and encryption are all part of the format. Each feature is optional, and each one is supported unevenly across viewers and libraries. If your product accepts PDFs from users, assume you will eventually receive every one of them.

2.1 Key points

- PDF was designed to make a document look the same on every screen and every printer.
- A PDF file is a collection of numbered objects, such as pages, fonts, images and annotations, tied together by a cross-reference table that lets a reader jump straight to any object without reading the whole file.
- That structure is why a viewer can show page 80 of a long report almost instantly.
- Text in a PDF is not stored as flowing paragraphs.
- Each piece of text is placed at exact coordinates on the page, often a few words at a time.
- Extracting readable text therefore means reassembling lines, columns and reading order, which is harder than it sounds.
- Multi-column layouts, tables and footnotes are classic sources of extraction bugs, and they are worth including in any test set.

A video file has two layers that are easy to confuse. The container, such as MP4, MOV, WebM or MKV, is the box that holds the streams, timing information and metadata. The codec, such as H.264, HEVC, VP9 or AV1, is the method used to compress the pictures inside the box. Two MP4 files can contain completely different codecs, and a player that handles one may refuse the other.

2.2 Sample data

The table below lists fictional test runs. It is here so that table extraction can be checked page after page.

Case	Sample file	Size	Result
TC-0201	sample-5-slides.pptx	9.9 MB	Pass
TC-0202	sample-transparent-512x512.png	11.4 MB	Pass
TC-0203	sample-1000-rows.xlsx	5.4 MB	Pass
TC-0204	sample-30s.mp3	8.7 MB	Fail
TC-0205	sample-30s.mp3	7.7 MB	Pass
TC-0206	sample.pdf	1.6 MB	Fail
TC-0207	sample-4032x3024.heic	7.0 MB	Pass
TC-0208	sample.txt	11.3 MB	Pass
TC-0209	sample.xml	8.2 MB	Fail
TC-0210	sample-1000-rows.xlsx	4.1 MB	Pass
TC-0211	sample-4032x3024.heic	9.4 MB	Fail
TC-0212	sample-100-rows.csv	896 KB	Pass
TC-0213	sample-5-slides.pptx	11.1 MB	Pass
TC-0214	sample.pdf	9.5 MB	Fail

2.3 Discussion

JPEG remains the workhorse for photographs. It uses lossy compression that throws away detail the eye is unlikely to miss, which keeps files small but means every re-save loses a little quality. PNG is lossless and supports transparency, which makes it the usual choice for screenshots, logos and interface graphics, at the cost of larger files for photographic content.

Audio bugs often hide in the details: an unexpected sample rate, a mono file where stereo was assumed, 24-bit samples in a pipeline built for 16-bit, or a long ID3 tag with cover art that pushes the first audio frame further into the file than a naive parser expects.

When you generate office documents yourself, open them in more than one application. A file that one program tolerates may trigger a repair prompt in another, and that prompt is enough to make users distrust the export feature you worked hard to build.

2.4 Figure

Figure 2. Sample measurements

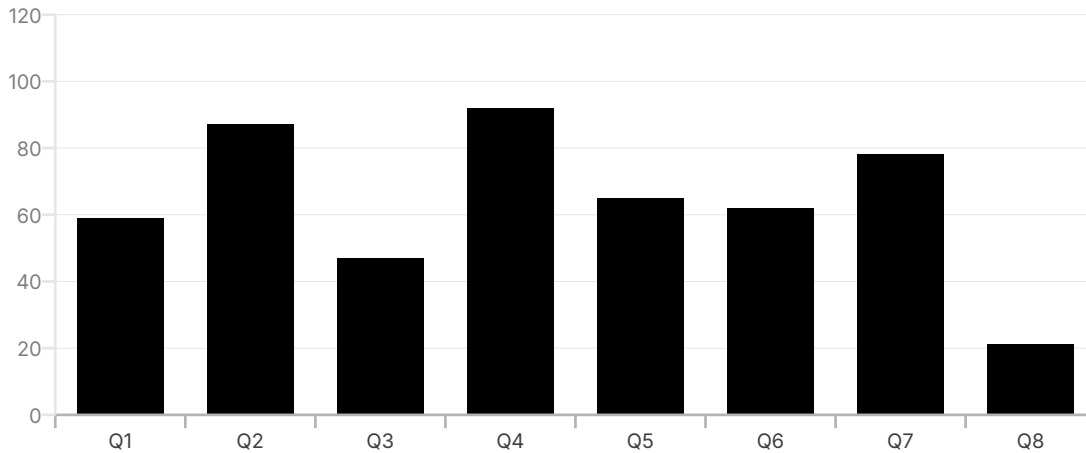


Figure 2: a vector bar chart drawn directly into the PDF. Values are random and seeded, so the chart is identical on every build.

CSV looks even simpler, with values separated by commas and one record per line, but there is no single CSV standard in everyday use. Fields may contain commas, quotes or line breaks; files may use semicolons in European locales; spreadsheet programs add a byte-order mark; and old systems still write legacy code pages instead of UTF-8. A robust importer must handle all of these.

2.5 Checklist

Work through these steps when you test this area:

- 1 Unpack an archive with deeply nested folders.
- 2 Try an empty, zero-byte file.
- 3 Check that text extraction keeps reading order.
- 4 Cancel an upload halfway through and retry.
- 5 Confirm scripts inside SVG files are stripped.
- 6 Make sure extracted paths never leave the target folder.
- 7 Play a video from the middle using seek.
- 8 Upload a file exactly at the size limit.
- 9 Repeat the test on a throttled mobile connection.

2.6 Summary

This chapter looked at inside the Portable Document Format. PDF was designed to make a document look the same on every screen and every printer. Text in a PDF is not stored as flowing paragraphs. PDFs can also carry much more than text.

Key takeaway. If your product accepts PDFs from users, assume you will eventually receive every one of them.

Then look at what happens after the upload succeeds. Are thumbnails generated correctly? Is metadata such as GPS location stripped when it should be? Are SVG files sanitized so they cannot run scripts? Is the original file stored safely and served with the right content type? Each of these questions deserves its own test file.

CHAPTER 03

Images on the web

JPEG remains the workhorse for photographs. It uses lossy compression that throws away detail the eye is unlikely to miss, which keeps files small but means every re-save loses a little quality. PNG is lossless and supports transparency, which makes it the usual choice for screenshots, logos and interface graphics, at the cost of larger files for photographic content.

Newer formats such as WebP and HEIC promise similar quality at a fraction of the size. WebP is now supported by every major browser. HEIC is the default photo format on iPhones, yet many web services still reject it or fail to convert it, which makes it one of the most common causes of upload support tickets.

When testing images, dimensions and file size are only the beginning. Check how your system handles transparency, color profiles, CMYK files from print workflows, very large images that strain memory, and EXIF orientation tags that tell a viewer to rotate a photo taken with a sideways phone.

3.1 Key points

- JPEG remains the workhorse for photographs.
- It uses lossy compression that throws away detail the eye is unlikely to miss, which keeps files small but means every re-save loses a little quality.
- PNG is lossless and supports transparency, which makes it the usual choice for screenshots, logos and interface graphics, at the cost of larger files for photographic content.
- Newer formats such as WebP and HEIC promise similar quality at a fraction of the size.
- WebP is now supported by every major browser.
- HEIC is the default photo format on iPhones, yet many web services still reject it or fail to convert it, which makes it one of the most common causes of upload support tickets.
- When testing images, dimensions and file size are only the beginning.

Digital audio starts as a stream of samples, typically 44,100 or 48,000 per second for each channel. WAV files usually store those samples uncompressed, which makes them large but simple and lossless. MP3 and AAC use perceptual compression to shrink the same audio by a factor of ten or more while keeping it pleasant to listen to.

3.2 Sample data

The table below lists fictional test runs. It is here so that table extraction can be checked page after page.

Case	Sample file	Size	Result
TC-0301	sample-5-slides.pptx	1.7 MB	Pass
TC-0302	sample-5-slides.pptx	2.4 MB	Pass
TC-0303	sample-4032x3024.heic	10.3 MB	Pass
TC-0304	sample-transparent-512x512.png	639 KB	Pass
TC-0305	sample-100-rows.csv	811 KB	Pass
TC-0306	sample.txt	774 KB	Skipped
TC-0307	sample-720p.mp4	2.5 MB	Pass
TC-0308	sample-100-rows.csv	3.4 MB	Pass
TC-0309	sample-1920x1080.jpg	4.0 MB	Pass
TC-0310	sample-5-slides.pptx	6.4 MB	Pass
TC-0311	messy-folders.zip	7.6 MB	Pass
TC-0312	sample-5-slides.pptx	7.3 MB	Pass
TC-0313	sample-1000-rows.xlsx	9.3 MB	Fail
TC-0314	messy-folders.zip	2.3 MB	Fail

3.3 Discussion

A video file has two layers that are easy to confuse. The container, such as MP4, MOV, WebM or MKV, is the box that holds the streams, timing information and metadata. The codec, such as H.264, HEVC, VP9 or AV1, is the method used to compress the pictures inside the box. Two MP4 files can contain completely different codecs, and a player that handles one may refuse the other.

CSV looks even simpler, with values separated by commas and one record per line, but there is no single CSV standard in everyday use. Fields may contain commas, quotes or line breaks; files may use semicolons in European locales; spreadsheet programs add a byte-order mark; and old systems still write legacy code pages instead of UTF-8. A robust importer must handle all of these.

Then look at what happens after the upload succeeds. Are thumbnails generated correctly? Is metadata such as GPS location stripped when it should be? Are SVG files sanitized so they cannot run scripts? Is the original file stored safely and served with the right content type? Each of these questions deserves its own test file.

3.4 Figure

Figure 3. Sample measurements

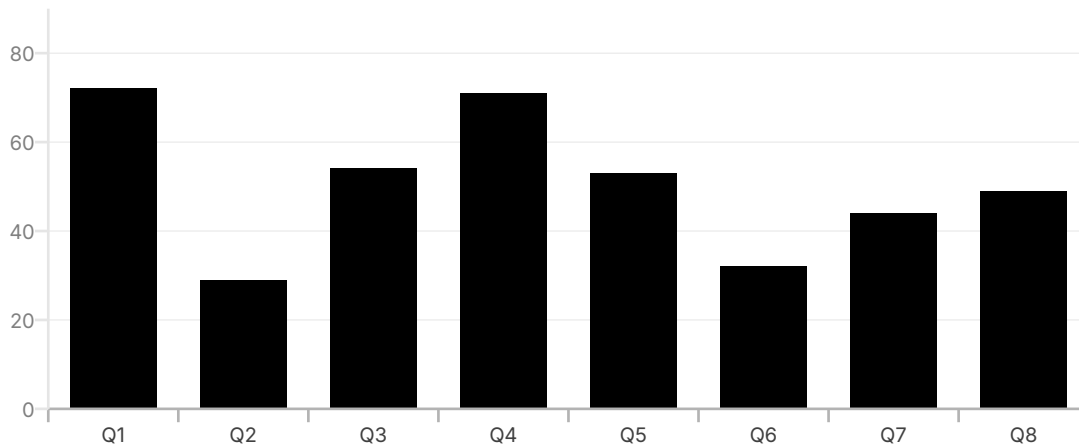


Figure 3: a vector bar chart drawn directly into the PDF. Values are random and seeded, so the chart is identical on every build.

Because they are archives, office files can contain surprises: embedded images that make a short document many megabytes in size, macros in the macro-enabled variants, comments and tracked changes that reveal earlier drafts, and hidden sheets in spreadsheets. Document viewers, search indexers and data-loss-prevention tools should all be tested against these cases.

3.5 Checklist

Work through these steps when you test this area:

- 1 Upload a file exactly at the size limit.
- 2 Repeat the test on a throttled mobile connection.
- 3 Check thumbnails for portrait and landscape images.
- 4 Upload a file just under the size limit.
- 5 Check audio and video stay in sync to the end.
- 6 Confirm GPS metadata is removed from public photos.
- 7 Make sure extracted paths never leave the target folder.
- 8 Play a video from the middle using seek.
- 9 Import a CSV file that contains quoted line breaks.

3.6 Summary

This chapter looked at images on the web. JPEG remains the workhorse for photographs. Newer formats such as WebP and HEIC promise similar quality at a fraction of the size. When testing images, dimensions and file size are only the beginning.

Key takeaway. Check how your system handles transparency, color profiles, CMYK files from print workflows, very large images that strain memory, and EXIF orientation tags that tell a viewer to rotate a photo taken with a sideways phone.

Line endings are a quieter source of trouble. Windows traditionally ends lines with a carriage return and line feed, while macOS and Linux use a line feed alone. Diff tools, parsers and checksums can all behave differently depending on which one they meet.

CHAPTER 04

Video containers and codecs

A video file has two layers that are easy to confuse. The container, such as MP4, MOV, WebM or MKV, is the box that holds the streams, timing information and metadata. The codec, such as H.264, HEVC, VP9 or AV1, is the method used to compress the pictures inside the box. Two MP4 files can contain completely different codecs, and a player that handles one may refuse the other.

H.264 video with AAC audio in an MP4 container is still the safest combination for playback everywhere. For streaming tests, keep a range of resolutions from 360p to 4K and a range of bitrates, because adaptive players switch between them as network conditions change.

Useful test videos carry visible information: a running timecode, a frame counter, the resolution printed on screen and a short beep or flash at regular intervals. With those clues you can spot dropped frames, audio drift and scaling problems by eye, without special tools.

4.1 Key points

- A video file has two layers that are easy to confuse.
- The container, such as MP4, MOV, WebM or MKV, is the box that holds the streams, timing information and metadata.
- The codec, such as H.264, HEVC, VP9 or AV1, is the method used to compress the pictures inside the box.
- Two MP4 files can contain completely different codecs, and a player that handles one may refuse the other.
- H.264 video with AAC audio in an MP4 container is still the safest combination for playback everywhere.
- For streaming tests, keep a range of resolutions from 360p to 4K and a range of bitrates, because adaptive players switch between them as network conditions change.
- Useful test videos carry visible information: a running timecode, a frame counter, the resolution printed on screen and a short beep or flash at regular intervals.

JSON has become the default language of web APIs. It is easy to read, maps naturally onto objects in most programming languages and supports nesting. Its simplicity also hides traps: very large numbers lose precision in JavaScript, duplicate keys are technically allowed, and a single missing comma makes the whole document invalid.

4.2 Sample data

The table below lists fictional test runs. It is here so that table extraction can be checked page after page.

Case	Sample file	Size	Result
TC-0401	sample-720p.mp4	6.4 MB	Pass
TC-0402	sample-1920x1080.jpg	3.5 MB	Pass
TC-0403	sample-4032x3024.heic	10.0 MB	Pass
TC-0404	sample-1920x1080.jpg	9.4 MB	Pass
TC-0405	sample-4032x3024.heic	7.1 MB	Pass
TC-0406	sample.xml	3.0 MB	Skipped
TC-0407	sample-1920x1080.jpg	376 KB	Pass
TC-0408	sample.txt	5.6 MB	Fail
TC-0409	sample-transparent-512x512.png	2.3 MB	Pass
TC-0410	sample-5-slides.pptx	54.7 KB	Pass
TC-0411	sample.pdf	11.3 MB	Pass
TC-0412	messy-folders.zip	2.8 MB	Pass
TC-0413	sample-5-slides.pptx	7.1 MB	Pass
TC-0414	sample-100-rows.csv	9.6 MB	Fail

4.3 Discussion

Digital audio starts as a stream of samples, typically 44,100 or 48,000 per second for each channel. WAV files usually store those samples uncompressed, which makes them large but simple and lossless. MP3 and AAC use perceptual compression to shrink the same audio by a factor of ten or more while keeping it pleasant to listen to.

Because they are archives, office files can contain surprises: embedded images that make a short document many megabytes in size, macros in the macro-enabled variants, comments and tracked changes that reveal earlier drafts, and hidden sheets in spreadsheets. Document viewers, search indexers and data-loss-prevention tools should all be tested against these cases.

Line endings are a quieter source of trouble. Windows traditionally ends lines with a carriage return and line feed, while macOS and Linux use a line feed alone. Diff tools, parsers and checksums can all behave differently depending on which one they meet.

4.4 Figure

Figure 4. Sample measurements

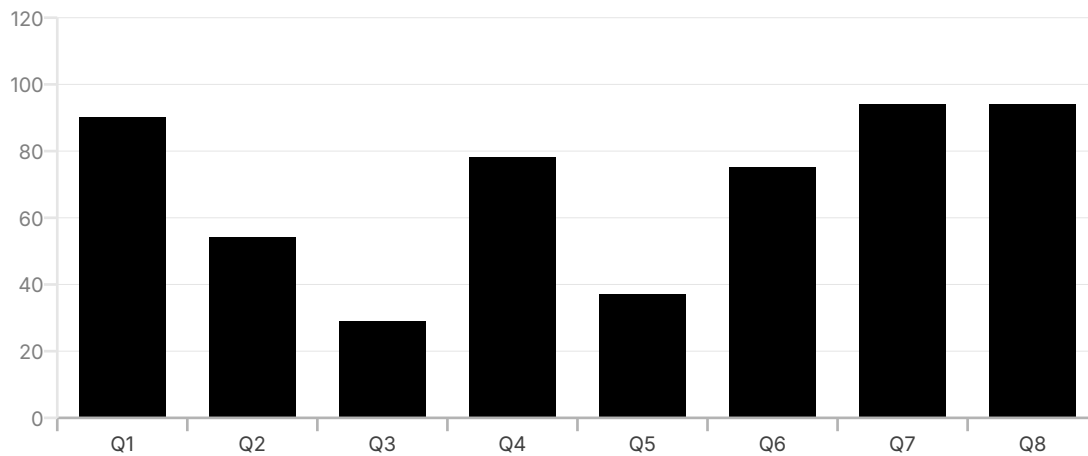


Figure 4: a vector bar chart drawn directly into the PDF. Values are random and seeded, so the chart is identical on every build.

Start with limits. Try a file just under the maximum size, exactly at it and just over it. Try an empty file, a file with no extension, a file whose extension does not match its contents and a file whose name contains spaces, accents or emoji. Check what happens when a user uploads the same file twice, or cancels halfway through.

4.5 Checklist

Work through these steps when you test this area:

- 1 Confirm scripts inside SVG files are stripped.
- 2 Try an empty, zero-byte file.
- 3 Play a video from the middle using seek.
- 4 Confirm GPS metadata is removed from public photos.
- 5 Rename a file so its extension does not match its contents.
- 6 Unpack an archive with deeply nested folders.
- 7 Upload the same file twice in a row.
- 8 Upload a file exactly at the size limit.
- 9 Repeat the test on a throttled mobile connection.

4.6 Summary

This chapter looked at video containers and codecs. A video file has two layers that are easy to confuse. H.264 video with AAC audio in an MP4 container is still the safest combination for playback everywhere. Useful test videos carry visible information: a running timecode, a frame counter, the resolution printed on screen and a short beep or flash at regular intervals.

Key takeaway. With those clues you can spot dropped frames, audio drift and scaling problems by eye, without special tools.

Safety matters too. An unpacker should never write outside its target directory, should cap the total size it will extract and should treat the archive's own claims about file sizes with healthy suspicion.

CHAPTER 05

Sound and audio formats

Digital audio starts as a stream of samples, typically 44,100 or 48,000 per second for each channel. WAV files usually store those samples uncompressed, which makes them large but simple and lossless. MP3 and AAC use perceptual compression to shrink the same audio by a factor of ten or more while keeping it pleasant to listen to.

Audio bugs often hide in the details: an unexpected sample rate, a mono file where stereo was assumed, 24-bit samples in a pipeline built for 16-bit, or a long ID3 tag with cover art that pushes the first audio frame further into the file than a naive parser expects.

Testing a media player or transcription service with several short clips, one long track and a few unusual encodings will surface most of these problems quickly. Keep at least one silent file and one very short file in the set, because edge cases at both ends of the scale are where players tend to fail.

5.1 Key points

- Digital audio starts as a stream of samples, typically 44,100 or 48,000 per second for each channel.
- WAV files usually store those samples uncompressed, which makes them large but simple and lossless.
- MP3 and AAC use perceptual compression to shrink the same audio by a factor of ten or more while keeping it pleasant to listen to.
- Audio bugs often hide in the details: an unexpected sample rate, a mono file where stereo was assumed, 24-bit samples in a pipeline built for 16-bit, or a long ID3 tag with cover art that pushes the first audio frame further into the file than a naive parser expects.
- Testing a media player or transcription service with several short clips, one long track and a few unusual encodings will surface most of these problems quickly.
- Keep at least one silent file and one very short file in the set, because edge cases at both ends of the scale are where players tend to fail.

Modern Word, Excel and PowerPoint files are ZIP archives in disguise. Rename a .docx file to .zip and you will find folders of XML describing the text, styles, images and relationships between them. This design, known as Office Open XML, made the formats far easier for other software to read and write.

5.2 Sample data

The table below lists fictional test runs. It is here so that table extraction can be checked page after page.

Case	Sample file	Size	Result
TC-0501	sample.xml	7.3 MB	Pass
TC-0502	sample-30s.mp3	1.4 MB	Pass
TC-0503	sample-10-pages.docx	6.7 MB	Pass
TC-0504	sample-100-rows.csv	7.7 MB	Pass
TC-0505	sample-1920x1080.jpg	4.7 MB	Pass
TC-0506	sample-1920x1080.jpg	8.9 MB	Pass
TC-0507	sample-100-rows.csv	2.2 MB	Skipped
TC-0508	sample-transparent-512x512.png	9.3 MB	Pass
TC-0509	sample-30s.mp3	9.1 MB	Pass
TC-0510	sample-5-slides.pptx	1.7 MB	Pass
TC-0511	sample-10-pages.docx	5.2 MB	Pass
TC-0512	sample.xml	7.6 MB	Skipped
TC-0513	sample.json	3.6 MB	Pass
TC-0514	sample.xml	5.6 MB	Pass

5.3 Discussion

JSON has become the default language of web APIs. It is easy to read, maps naturally onto objects in most programming languages and supports nesting. Its simplicity also hides traps: very large numbers lose precision in JavaScript, duplicate keys are technically allowed, and a single missing comma makes the whole document invalid.

Start with limits. Try a file just under the maximum size, exactly at it and just over it. Try an empty file, a file with no extension, a file whose extension does not match its contents and a file whose name contains spaces, accents or emoji. Check what happens when a user uploads the same file twice, or cancels halfway through.

Safety matters too. An unpacker should never write outside its target directory, should cap the total size it will extract and should treat the archive's own claims about file sizes with healthy suspicion.

5.4 Figure

Figure 5. Sample measurements

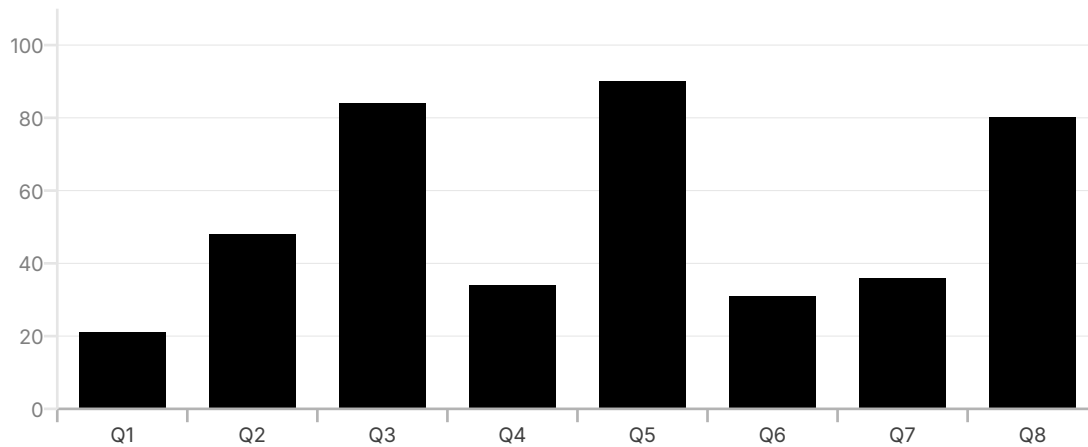


Figure 5: a vector bar chart drawn directly into the PDF. Values are random and seeded, so the chart is identical on every build.

Problems appear when software guesses the encoding wrong. The result is garbled text: accented letters turned into strange pairs of symbols, or Asian scripts reduced to question marks. UTF-16 files, byte-order marks and legacy code pages are frequent culprits, and so are emoji that need four bytes in UTF-8 and two code units in UTF-16.

5.5 Checklist

Work through these steps when you test this area:

- 1 Play a video from the middle using seek.
- 2 Upload a file exactly at the size limit.
- 3 Check that text extraction keeps reading order.
- 4 Measure peak memory while processing the largest sample.
- 5 Confirm scripts inside SVG files are stripped.
- 6 Check audio and video stay in sync to the end.
- 7 Cancel an upload halfway through and retry.
- 8 Import a CSV file that contains quoted line breaks.
- 9 Unpack an archive with deeply nested folders.

5.6 Summary

This chapter looked at sound and audio formats. Digital audio starts as a stream of samples, typically 44,100 or 48,000 per second for each channel. Audio bugs often hide in the details: an unexpected sample rate, a mono file where stereo was assumed, 24-bit samples in a pipeline built for 16-bit, or a long ID3 tag with cover art that pushes the first audio frame further into the file than a naive parser expects. Testing a media player or transcription service with several short clips, one long track and a few unusual encodings will surface most of these problems quickly.

Key takeaway. Keep at least one silent file and one very short file in the set, because edge cases at both ends of the scale are where players tend to fail.

The sample files in this collection include clearly fictional metadata, such as a public landmark instead of a home address, an invented camera and the Technology Channel name as author, so you can test metadata handling without exposing anyone's real information.

CHAPTER 06

Structured data: JSON, CSV and XML

JSON has become the default language of web APIs. It is easy to read, maps naturally onto objects in most programming languages and supports nesting. Its simplicity also hides traps: very large numbers lose precision in JavaScript, duplicate keys are technically allowed, and a single missing comma makes the whole document invalid.

CSV looks even simpler, with values separated by commas and one record per line, but there is no single CSV standard in everyday use. Fields may contain commas, quotes or line breaks; files may use semicolons in European locales; spreadsheet programs add a byte-order mark; and old systems still write legacy code pages instead of UTF-8. A robust importer must handle all of these.

XML is more verbose but brings namespaces, schemas and a long tail of enterprise use: RSS feeds, sitemaps, SOAP services and office document internals. Parsers must also be configured carefully, because some XML features can be abused to read local files or exhaust memory.

6.1 Key points

- JSON has become the default language of web APIs.
- It is easy to read, maps naturally onto objects in most programming languages and supports nesting.
- Its simplicity also hides traps: very large numbers lose precision in JavaScript, duplicate keys are technically allowed, and a single missing comma makes the whole document invalid.
- CSV looks even simpler, with values separated by commas and one record per line, but there is no single CSV standard in everyday use.
- Fields may contain commas, quotes or line breaks; files may use semicolons in European locales; spreadsheet programs add a byte-order mark; and old systems still write legacy code pages instead of UTF-8.
- A robust importer must handle all of these.
- XML is more verbose but brings namespaces, schemas and a long tail of enterprise use: RSS feeds, sitemaps, SOAP services and office document internals.

File upload is one of the riskiest features in any web application. It accepts arbitrary bytes from strangers and often passes them to image processors, document converters and storage services. A good test plan covers the happy path and the many ways it can go wrong.

6.2 Sample data

The table below lists fictional test runs. It is here so that table extraction can be checked page after page.

Case	Sample file	Size	Result
TC-0601	sample-4032x3024.heic	8.1 MB	Pass
TC-0602	sample.pdf	9.7 MB	Pass
TC-0603	sample.txt	1.7 MB	Pass
TC-0604	sample.pdf	9.3 MB	Pass
TC-0605	sample-10-pages.docx	11.1 MB	Pass
TC-0606	sample-720p.mp4	5.9 MB	Fail
TC-0607	sample-1920x1080.jpg	10.0 MB	Pass
TC-0608	sample.json	4.9 MB	Fail
TC-0609	sample-5-slides.pptx	7.0 MB	Skipped
TC-0610	sample-720p.mp4	5.2 MB	Pass
TC-0611	sample-30s.mp3	9.6 MB	Pass
TC-0612	sample-1000-rows.xlsx	2.5 MB	Skipped
TC-0613	sample-5-slides.pptx	6.8 MB	Pass
TC-0614	sample-30s.mp3	9.7 MB	Pass

6.3 Discussion

Modern Word, Excel and PowerPoint files are ZIP archives in disguise. Rename a .docx file to .zip and you will find folders of XML describing the text, styles, images and relationships between them. This design, known as Office Open XML, made the formats far easier for other software to read and write.

Problems appear when software guesses the encoding wrong. The result is garbled text: accented letters turned into strange pairs of symbols, or Asian scripts reduced to question marks. UTF-16 files, byte-order marks and legacy code pages are frequent culprits, and so are emoji that need four bytes in UTF-8 and two code units in UTF-16.

The sample files in this collection include clearly fictional metadata, such as a public landmark instead of a home address, an invented camera and the Technology Channel name as author, so you can test metadata handling without exposing anyone's real information.

6.4 Figure

Figure 6. Sample measurements

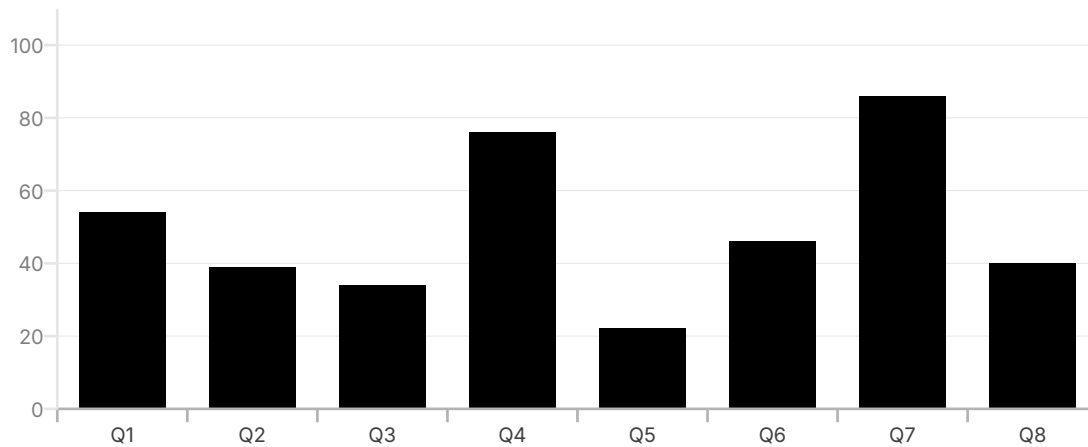


Figure 6: a vector bar chart drawn directly into the PDF. Values are random and seeded, so the chart is identical on every build.

Deeply nested paths can exceed operating-system limits. Very many small files can be slower to process than one large file. Archives inside archives raise the question of how deep to go. Files with identical names in different folders test whether your code keeps paths intact when it flattens a structure.

6.5 Checklist

Work through these steps when you test this area:

- 1 Check that text extraction keeps reading order.
- 2 Parse a JSON file with deeply nested objects.
- 3 Check audio and video stay in sync to the end.
- 4 Upload a file exactly at the size limit.
- 5 Test a password-protected document.
- 6 Compare checksums before and after storage.
- 7 Measure peak memory while processing the largest sample.
- 8 Upload a file just under the size limit.
- 9 Upload a file one byte over the size limit.

6.6 Summary

This chapter looked at structured data: JSON, CSV and XML. JSON has become the default language of web APIs. CSV looks even simpler, with values separated by commas and one record per line, but there is no single CSV standard in everyday use. XML is more verbose but brings namespaces, schemas and a long tail of enterprise use: RSS feeds, sitemaps, SOAP services and office document internals.

Key takeaway. Parsers must also be configured carefully, because some XML features can be abused to read local files or exhaust memory.

Streaming techniques, which read and write data in chunks rather than all at once, are the usual cure. They keep memory use flat no matter how large the file grows, and they allow work to begin before the transfer has finished.

CHAPTER 07

Office documents are ZIP files

Modern Word, Excel and PowerPoint files are ZIP archives in disguise. Rename a .docx file to .zip and you will find folders of XML describing the text, styles, images and relationships between them. This design, known as Office Open XML, made the formats far easier for other software to read and write.

Because they are archives, office files can contain surprises: embedded images that make a short document many megabytes in size, macros in the macro-enabled variants, comments and tracked changes that reveal earlier drafts, and hidden sheets in spreadsheets. Document viewers, search indexers and data-loss-prevention tools should all be tested against these cases.

When you generate office documents yourself, open them in more than one application. A file that one program tolerates may trigger a repair prompt in another, and that prompt is enough to make users distrust the export feature you worked hard to build.

7.1 Key points

- Modern Word, Excel and PowerPoint files are ZIP archives in disguise.
- Rename a .docx file to .zip and you will find folders of XML describing the text, styles, images and relationships between them.
- This design, known as Office Open XML, made the formats far easier for other software to read and write.
- Because they are archives, office files can contain surprises: embedded images that make a short document many megabytes in size, macros in the macro-enabled variants, comments and tracked changes that reveal earlier drafts, and hidden sheets in spreadsheets.
- Document viewers, search indexers and data-loss-prevention tools should all be tested against these cases.
- When you generate office documents yourself, open them in more than one application.
- A file that one program tolerates may trigger a repair prompt in another, and that prompt is enough to make users distrust the export feature you worked hard to build.

Every text file is a sequence of bytes, and an encoding is the rule that turns those bytes into characters. UTF-8 is now the dominant encoding on the web because it can represent every character in Unicode while remaining compatible with plain ASCII.

7.2 Sample data

The table below lists fictional test runs. It is here so that table extraction can be checked page after page.

Case	Sample file	Size	Result
TC-0701	sample-10-pages.docx	997 KB	Pass
TC-0702	messy-folders.zip	2.5 MB	Pass
TC-0703	sample-100-rows.csv	3.3 MB	Pass
TC-0704	sample-720p.mp4	4.7 MB	Pass
TC-0705	messy-folders.zip	4.2 MB	Pass
TC-0706	sample-5-slides.pptx	4.3 MB	Pass
TC-0707	sample-transparent-512x512.png	2.8 MB	Pass
TC-0708	messy-folders.zip	1.5 MB	Pass
TC-0709	sample-1920x1080.jpg	2.7 MB	Pass
TC-0710	sample-10-pages.docx	5.9 MB	Pass
TC-0711	sample-4032x3024.heic	8.3 MB	Pass
TC-0712	sample.pdf	7.5 MB	Fail
TC-0713	sample.pdf	4.2 MB	Pass
TC-0714	sample.pdf	1.2 MB	Pass

7.3 Discussion

File upload is one of the riskiest features in any web application. It accepts arbitrary bytes from strangers and often passes them to image processors, document converters and storage services. A good test plan covers the happy path and the many ways it can go wrong.

Deeply nested paths can exceed operating-system limits. Very many small files can be slower to process than one large file. Archives inside archives raise the question of how deep to go. Files with identical names in different folders test whether your code keeps paths intact when it flattens a structure.

Streaming techniques, which read and write data in chunks rather than all at once, are the usual cure. They keep memory use flat no matter how large the file grows, and they allow work to begin before the transfer has finished.

7.4 Figure

Figure 7. Sample measurements

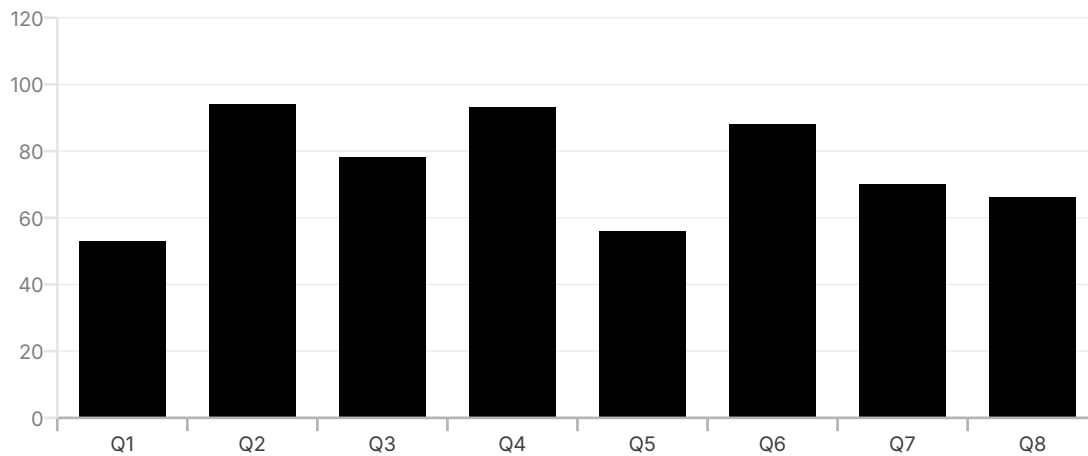


Figure 7: a vector bar chart drawn directly into the PDF. Values are random and seeded, so the chart is identical on every build.

This metadata is useful for search and organization, but it can also leak private details when files are shared publicly. Platforms that publish user uploads usually strip location data and personal fields, and that stripping step deserves dedicated tests with files that are known to contain metadata.

7.5 Checklist

Work through these steps when you test this area:

- 1 Open the file in at least two different applications.
- 2 Use a file name with spaces, accents and emoji.
- 3 Rename a file so its extension does not match its contents.
- 4 Unpack an archive with deeply nested folders.
- 5 Make sure extracted paths never leave the target folder.
- 6 Check that text extraction keeps reading order.
- 7 Repeat the test on a throttled mobile connection.
- 8 Verify the served Content-Type header.
- 9 Upload the same file twice in a row.

7.6 Summary

This chapter looked at office documents are ZIP files. Modern Word, Excel and PowerPoint files are ZIP archives in disguise. Because they are archives, office files can contain surprises: embedded images that make a short document many megabytes in size, macros in the macro-enabled variants, comments and tracked changes that reveal earlier drafts, and hidden sheets in spreadsheets. When you generate office documents yourself, open them in more than one application.

Key takeaway. A file that one program tolerates may trigger a repair prompt in another, and that prompt is enough to make users distrust the export feature you worked hard to build.

When you test a document viewer or converter, include files with headings, lists, tables and images, and check that the structure survives the round trip. A conversion that keeps the words but loses the headings has quietly made the document harder to use.

CHAPTER 08

Testing file uploads

File upload is one of the riskiest features in any web application. It accepts arbitrary bytes from strangers and often passes them to image processors, document converters and storage services. A good test plan covers the happy path and the many ways it can go wrong.

Start with limits. Try a file just under the maximum size, exactly at it and just over it. Try an empty file, a file with no extension, a file whose extension does not match its contents and a file whose name contains spaces, accents or emoji. Check what happens when a user uploads the same file twice, or cancels halfway through.

Then look at what happens after the upload succeeds. Are thumbnails generated correctly? Is metadata such as GPS location stripped when it should be? Are SVG files sanitized so they cannot run scripts? Is the original file stored safely and served with the right content type? Each of these questions deserves its own test file.

8.1 Key points

- File upload is one of the riskiest features in any web application.
- It accepts arbitrary bytes from strangers and often passes them to image processors, document converters and storage services.
- A good test plan covers the happy path and the many ways it can go wrong.
- Start with limits.
- Try a file just under the maximum size, exactly at it and just over it.
- Try an empty file, a file with no extension, a file whose extension does not match its contents and a file whose name contains spaces, accents or emoji.
- Check what happens when a user uploads the same file twice, or cancels halfway through.

Real users rarely send tidy files. They send ZIP archives full of nested folders, duplicate names, temporary files left behind by office applications, hidden system files from macOS and Windows, and file names written in several languages at once. Any feature that unpacks archives should be tested against that kind of mess.

8.2 Sample data

The table below lists fictional test runs. It is here so that table extraction can be checked page after page.

Case	Sample file	Size	Result
TC-0801	sample.xml	8.0 MB	Pass
TC-0802	sample-transparent-512x512.png	3.4 MB	Pass
TC-0803	sample-720p.mp4	1.3 MB	Pass
TC-0804	sample.xml	4.5 MB	Fail
TC-0805	sample-720p.mp4	5.4 MB	Pass
TC-0806	sample.pdf	9.5 MB	Pass
TC-0807	sample-30s.mp3	11.0 MB	Pass
TC-0808	sample-5-slides.pptx	11.1 MB	Pass
TC-0809	sample.json	9.8 MB	Pass
TC-0810	sample-30s.mp3	4.9 MB	Skipped
TC-0811	messy-folders.zip	7.0 MB	Pass
TC-0812	sample.json	1.3 MB	Pass
TC-0813	sample.xml	11.1 MB	Fail
TC-0814	sample.txt	4.4 MB	Pass

8.3 Discussion

Every text file is a sequence of bytes, and an encoding is the rule that turns those bytes into characters. UTF-8 is now the dominant encoding on the web because it can represent every character in Unicode while remaining compatible with plain ASCII.

This metadata is useful for search and organization, but it can also leak private details when files are shared publicly. Platforms that publish user uploads usually strip location data and personal fields, and that stripping step deserves dedicated tests with files that are known to contain metadata.

When you test a document viewer or converter, include files with headings, lists, tables and images, and check that the structure survives the round trip. A conversion that keeps the words but loses the headings has quietly made the document harder to use.

8.4 Figure

Figure 8. Sample measurements

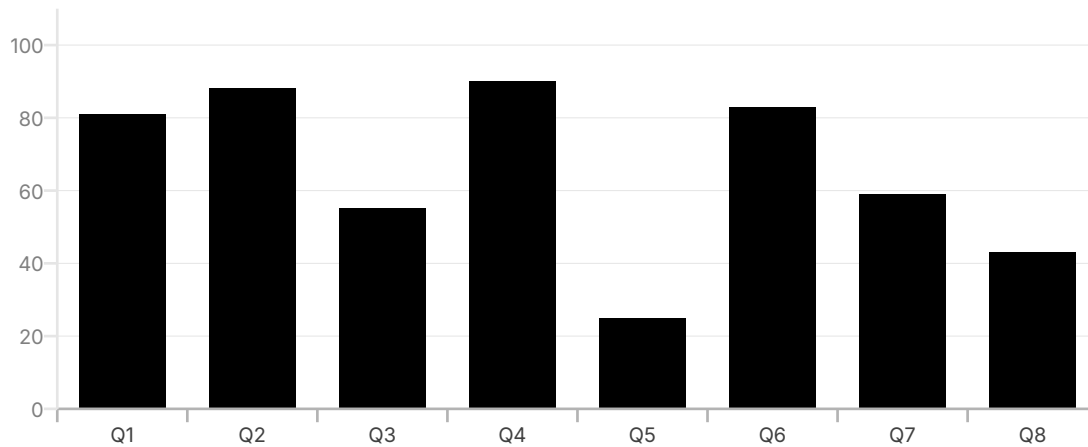


Figure 8: a vector bar chart drawn directly into the PDF. Values are random and seeded, so the chart is identical on every build.

Test with a range of sizes that matches your real limits: something tiny, something typical and something close to the maximum you allow. Measure memory as well as time, and repeat the test on a slow network connection, because mobile users experience your upload flow very differently from developers on office Wi-Fi.

8.5 Checklist

Work through these steps when you test this area:

- 1 Upload a file one byte over the size limit.
- 2 Make sure extracted paths never leave the target folder.
- 3 Check that text extraction keeps reading order.
- 4 Parse a JSON file with deeply nested objects.
- 5 Cancel an upload halfway through and retry.
- 6 Confirm scripts inside SVG files are stripped.
- 7 Unpack an archive with deeply nested folders.
- 8 Open the file in at least two different applications.
- 9 Rename a file so its extension does not match its contents.

8.6 Summary

This chapter looked at testing file uploads. File upload is one of the riskiest features in any web application. Start with limits. Then look at what happens after the upload succeeds.

Key takeaway. Each of these questions deserves its own test file.

Over time, every bug you fix should leave behind a test file that reproduces it. That growing collection becomes one of the most valuable assets a team owns: a precise, executable memory of everything that has gone wrong before.

CHAPTER 09

Character encodings

Every text file is a sequence of bytes, and an encoding is the rule that turns those bytes into characters. UTF-8 is now the dominant encoding on the web because it can represent every character in Unicode while remaining compatible with plain ASCII.

Problems appear when software guesses the encoding wrong. The result is garbled text: accented letters turned into strange pairs of symbols, or Asian scripts reduced to question marks. UTF-16 files, byte-order marks and legacy code pages are frequent culprits, and so are emoji that need four bytes in UTF-8 and two code units in UTF-16.

Line endings are a quieter source of trouble. Windows traditionally ends lines with a carriage return and line feed, while macOS and Linux use a line feed alone. Diff tools, parsers and checksums can all behave differently depending on which one they meet.

9.1 Key points

- Every text file is a sequence of bytes, and an encoding is the rule that turns those bytes into characters.
- UTF-8 is now the dominant encoding on the web because it can represent every character in Unicode while remaining compatible with plain ASCII.
- Problems appear when software guesses the encoding wrong.
- The result is garbled text: accented letters turned into strange pairs of symbols, or Asian scripts reduced to question marks.
- UTF-16 files, byte-order marks and legacy code pages are frequent culprits, and so are emoji that need four bytes in UTF-8 and two code units in UTF-16.
- Line endings are a quieter source of trouble.
- Windows traditionally ends lines with a carriage return and line feed, while macOS and Linux use a line feed alone.

Many files carry information that is invisible at first glance. Photos store the camera model, the time they were taken and often the exact GPS location. Office documents record authors, editing time and sometimes the names of everyone who commented on a draft. PDFs list the software that produced them.

9.2 Sample data

The table below lists fictional test runs. It is here so that table extraction can be checked page after page.

Case	Sample file	Size	Result
TC-0901	messy-folders.zip	8.1 MB	Pass
TC-0902	sample-30s.mp3	6.5 MB	Pass
TC-0903	sample-1000-rows.xlsx	6.9 MB	Fail
TC-0904	sample.xml	11.0 MB	Pass
TC-0905	sample-4032x3024.heic	1.3 MB	Pass
TC-0906	sample.pdf	4.6 MB	Pass
TC-0907	sample-10-pages.docx	9.6 MB	Pass
TC-0908	sample-720p.mp4	7.2 MB	Pass
TC-0909	sample-5-slides.pptx	8.5 MB	Fail
TC-0910	sample-100-rows.csv	7.6 MB	Pass
TC-0911	sample.txt	10.3 MB	Pass
TC-0912	sample.pdf	412 KB	Skipped
TC-0913	sample-1920x1080.jpg	10.2 MB	Skipped
TC-0914	sample.xml	9.2 MB	Skipped

9.3 Discussion

Real users rarely send tidy files. They send ZIP archives full of nested folders, duplicate names, temporary files left behind by office applications, hidden system files from macOS and Windows, and file names written in several languages at once. Any feature that unpacks archives should be tested against that kind of mess.

Test with a range of sizes that matches your real limits: something tiny, something typical and something close to the maximum you allow. Measure memory as well as time, and repeat the test on a slow network connection, because mobile users experience your upload flow very differently from developers on office Wi-Fi.

Over time, every bug you fix should leave behind a test file that reproduces it. That growing collection becomes one of the most valuable assets a team owns: a precise, executable memory of everything that has gone wrong before.

9.4 Figure

Figure 9. Sample measurements

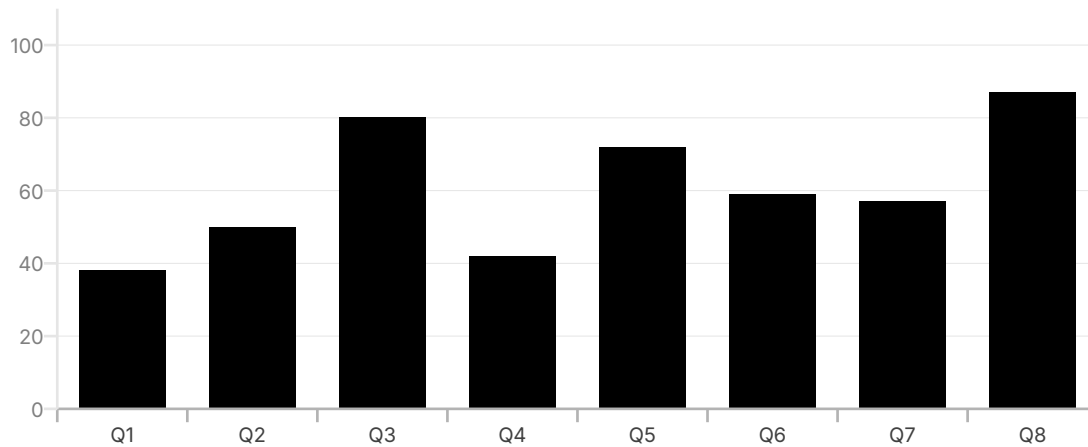


Figure 9: a vector bar chart drawn directly into the PDF. Values are random and seeded, so the chart is identical on every build.

Accessibility also benefits automated systems. Search engines, summarization tools and data-extraction pipelines all work better when a document's structure is explicit. A well-structured file is easier for both people and programs to understand.

9.5 Checklist

Work through these steps when you test this area:

- 1 Repeat the test on a throttled mobile connection.
- 2 Make sure extracted paths never leave the target folder.
- 3 Unpack an archive with deeply nested folders.
- 4 Import a CSV file that contains quoted line breaks.
- 5 Check that text extraction keeps reading order.
- 6 Try an empty, zero-byte file.
- 7 Test a password-protected document.
- 8 Rename a file so its extension does not match its contents.
- 9 Upload a file exactly at the size limit.

9.6 Summary

This chapter looked at character encodings. Every text file is a sequence of bytes, and an encoding is the rule that turns those bytes into characters. Problems appear when software guesses the encoding wrong. Line endings are a quieter source of trouble.

Key takeaway. Diff tools, parsers and checksums can all behave differently depending on which one they meet.

Finally, a good test file is honest about what it contains. Real documents carry personal data, licensed fonts and copyrighted photos. Synthetic files avoid all of that. The names, numbers and pictures in these samples are invented, so you can attach them to bug reports, commit them to a repository or share them with a vendor without a second thought.

CHAPTER 10

Archives and messy folders

Real users rarely send tidy files. They send ZIP archives full of nested folders, duplicate names, temporary files left behind by office applications, hidden system files from macOS and Windows, and file names written in several languages at once. Any feature that unpacks archives should be tested against that kind of mess.

Deeply nested paths can exceed operating-system limits. Very many small files can be slower to process than one large file. Archives inside archives raise the question of how deep to go. Files with identical names in different folders test whether your code keeps paths intact when it flattens a structure.

Safety matters too. An unpacker should never write outside its target directory, should cap the total size it will extract and should treat the archive's own claims about file sizes with healthy suspicion.

10.1 Key points

- Real users rarely send tidy files.
- They send ZIP archives full of nested folders, duplicate names, temporary files left behind by office applications, hidden system files from macOS and Windows, and file names written in several languages at once.
- Any feature that unpacks archives should be tested against that kind of mess.
- Deeply nested paths can exceed operating-system limits.
- Very many small files can be slower to process than one large file.
- Archives inside archives raise the question of how deep to go.
- Files with identical names in different folders test whether your code keeps paths intact when it flattens a structure.

Performance problems often stay hidden until a large file arrives. A parser that reads a whole document into memory works perfectly with a few kilobytes and fails with a few hundred megabytes. A progress bar that never moves makes a slow upload look like a broken one.

10.2 Sample data

The table below lists fictional test runs. It is here so that table extraction can be checked page after page.

Case	Sample file	Size	Result
TC-1001	sample-transparent-512x512.png	957 KB	Pass
TC-1002	messy-folders.zip	6.2 MB	Pass
TC-1003	sample.pdf	1.8 MB	Pass
TC-1004	sample-4032x3024.heic	6.1 MB	Pass
TC-1005	sample-transparent-512x512.png	2.6 MB	Fail
TC-1006	sample-10-pages.docx	9.8 MB	Pass
TC-1007	sample-30s.mp3	10.7 MB	Skipped
TC-1008	sample-30s.mp3	6.5 MB	Pass
TC-1009	messy-folders.zip	3.7 MB	Skipped
TC-1010	messy-folders.zip	8.7 KB	Pass
TC-1011	sample-transparent-512x512.png	558 KB	Pass
TC-1012	sample.xml	11.4 MB	Pass
TC-1013	sample-10-pages.docx	6.6 MB	Pass
TC-1014	sample-100-rows.csv	11.3 MB	Pass

10.3 Discussion

Many files carry information that is invisible at first glance. Photos store the camera model, the time they were taken and often the exact GPS location. Office documents record authors, editing time and sometimes the names of everyone who commented on a draft. PDFs list the software that produced them.

Accessibility also benefits automated systems. Search engines, summarization tools and data-extraction pipelines all work better when a document's structure is explicit. A well-structured file is easier for both people and programs to understand.

Finally, a good test file is honest about what it contains. Real documents carry personal data, licensed fonts and copyrighted photos. Synthetic files avoid all of that. The names, numbers and pictures in these samples are invented, so you can attach them to bug reports, commit them to a repository or share them with a vendor without a second thought.

10.4 Figure

Figure 10. Sample measurements

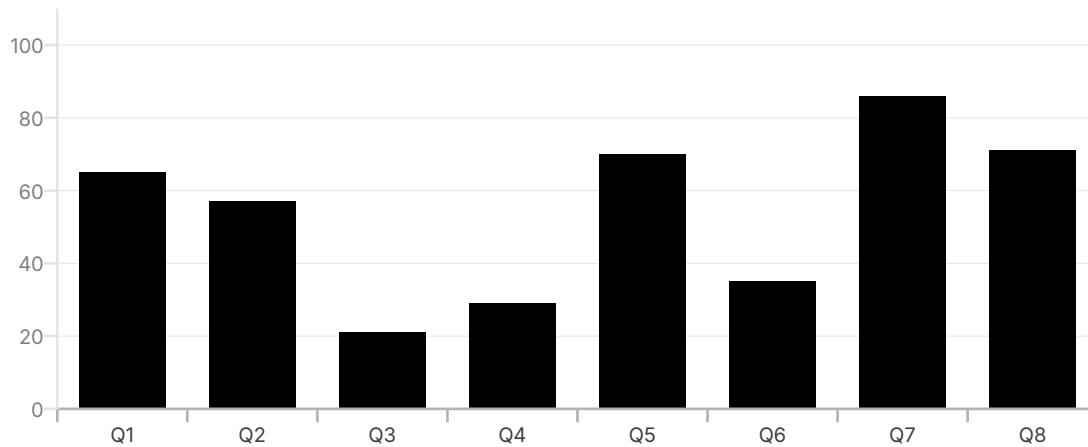


Figure 10: a vector bar chart drawn directly into the PDF. Values are random and seeded, so the chart is identical on every build.

Stable URLs and checksums make this reliable. If a file changes unexpectedly, the checksum comparison fails loudly instead of letting a test pass for the wrong reason. Keep the files small where possible so the suite stays fast, and reserve the largest samples for scheduled performance runs.

10.5 Checklist

Work through these steps when you test this area:

- 1 Cancel an upload halfway through and retry.
- 2 Check audio and video stay in sync to the end.
- 3 Use a file name with spaces, accents and emoji.
- 4 Check thumbnails for portrait and landscape images.
- 5 Upload a file exactly at the size limit.
- 6 Parse a JSON file with deeply nested objects.
- 7 Upload a file one byte over the size limit.
- 8 Confirm scripts inside SVG files are stripped.
- 9 Compare checksums before and after storage.

10.6 Summary

This chapter looked at archives and messy folders. Real users rarely send tidy files. Deeply nested paths can exceed operating-system limits. Safety matters too.

Key takeaway. An unpacker should never write outside its target directory, should cap the total size it will extract and should treat the archive's own claims about file sizes with healthy suspicion.

PDFs can also carry much more than text. Interactive forms, bookmarks, links, embedded files, digital signatures and encryption are all part of the format. Each feature is optional, and each one is supported unevenly across viewers and libraries. If your product accepts PDFs from users, assume you will eventually receive every one of them.

CHAPTER 11

Metadata and privacy

Many files carry information that is invisible at first glance. Photos store the camera model, the time they were taken and often the exact GPS location. Office documents record authors, editing time and sometimes the names of everyone who commented on a draft. PDFs list the software that produced them.

This metadata is useful for search and organization, but it can also leak private details when files are shared publicly. Platforms that publish user uploads usually strip location data and personal fields, and that stripping step deserves dedicated tests with files that are known to contain metadata.

The sample files in this collection include clearly fictional metadata, such as a public landmark instead of a home address, an invented camera and the Technology Channel name as author, so you can test metadata handling without exposing anyone's real information.

11.1 Key points

- Many files carry information that is invisible at first glance.
- Photos store the camera model, the time they were taken and often the exact GPS location.
- Office documents record authors, editing time and sometimes the names of everyone who commented on a draft.
- PDFs list the software that produced them.
- This metadata is useful for search and organization, but it can also leak private details when files are shared publicly.
- Platforms that publish user uploads usually strip location data and personal fields, and that stripping step deserves dedicated tests with files that are known to contain metadata.
- The sample files in this collection include clearly fictional metadata, such as a public landmark instead of a home address, an invented camera and the Technology Channel name as author, so you can test metadata handling without exposing anyone's real information.

An accessible document can be read by people who use screen readers, magnifiers or keyboard navigation. In practice this means real headings rather than large bold text, meaningful alternative text for images, tables with marked header rows and a logical reading order.

11.2 Sample data

The table below lists fictional test runs. It is here so that table extraction can be checked page after page.

Case	Sample file	Size	Result
TC-1101	sample.xml	5.5 MB	Skipped
TC-1102	sample.txt	5.6 MB	Pass
TC-1103	sample-4032x3024.heic	3.2 MB	Fail
TC-1104	sample.txt	5.3 MB	Pass
TC-1105	sample-1000-rows.xlsx	4.7 MB	Pass
TC-1106	sample.txt	4.6 MB	Pass
TC-1107	sample-30s.mp3	5.0 MB	Pass
TC-1108	sample-100-rows.csv	10.7 MB	Pass
TC-1109	sample-10-pages.docx	8.1 MB	Pass
TC-1110	sample-720p.mp4	3.9 MB	Pass
TC-1111	messy-folders.zip	7.5 MB	Pass
TC-1112	messy-folders.zip	5.6 MB	Pass
TC-1113	sample-5-slides.pptx	10.1 MB	Pass
TC-1114	sample-720p.mp4	4.5 MB	Pass

11.3 Discussion

Performance problems often stay hidden until a large file arrives. A parser that reads a whole document into memory works perfectly with a few kilobytes and fails with a few hundred megabytes. A progress bar that never moves makes a slow upload look like a broken one.

Stable URLs and checksums make this reliable. If a file changes unexpectedly, the checksum comparison fails loudly instead of letting a test pass for the wrong reason. Keep the files small where possible so the suite stays fast, and reserve the largest samples for scheduled performance runs.

PDFs can also carry much more than text. Interactive forms, bookmarks, links, embedded files, digital signatures and encryption are all part of the format. Each feature is optional, and each one is supported unevenly across viewers and libraries. If your product accepts PDFs from users, assume you will eventually receive every one of them.

11.4 Figure

Figure 11. Sample measurements

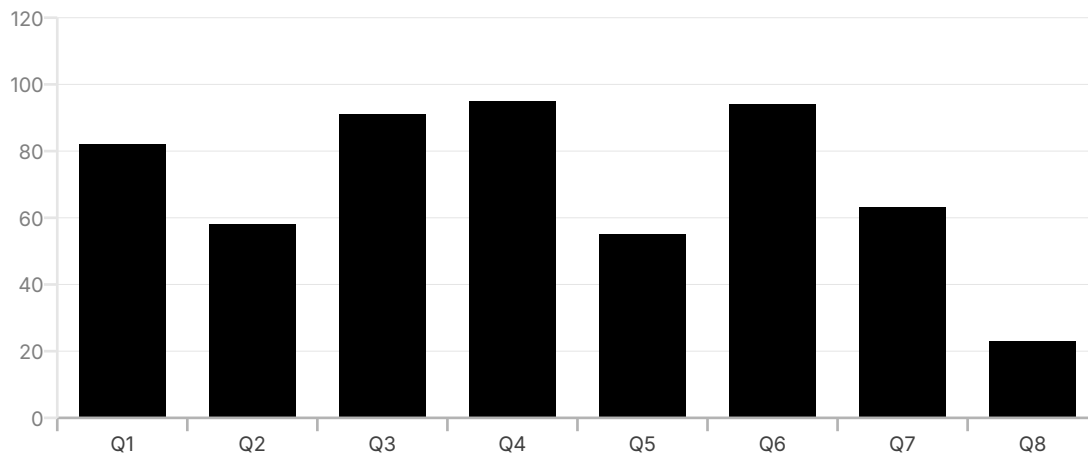


Figure 11: a vector bar chart drawn directly into the PDF. Values are random and seeded, so the chart is identical on every build.

Good test files also come in families. A single PDF tells you that your viewer can open a PDF; a family of PDFs tells you how it copes with one page and a hundred pages, with a few kilobytes and ten megabytes, with plain text, scanned images, form fields and passwords. Testing across a family is how you find the edges of a system before your users do.

11.5 Checklist

Work through these steps when you test this area:

- 1 Repeat the test on a throttled mobile connection.
- 2 Parse a JSON file with deeply nested objects.
- 3 Make sure extracted paths never leave the target folder.
- 4 Test a password-protected document.
- 5 Verify the served Content-Type header.
- 6 Check that text extraction keeps reading order.
- 7 Confirm GPS metadata is removed from public photos.
- 8 Confirm scripts inside SVG files are stripped.
- 9 Unpack an archive with deeply nested folders.

11.6 Summary

This chapter looked at metadata and privacy. Many files carry information that is invisible at first glance. This metadata is useful for search and organization, but it can also leak private details when files are shared publicly. The sample files in this collection include clearly fictional metadata, such as a public landmark instead of a home address, an invented camera and the Technology Channel name as author, so you can test metadata handling without exposing anyone's real information.

Key takeaway. The sample files in this collection include clearly fictional metadata, such as a public landmark instead of a home address, an invented camera and the Technology Channel name as author, so you can test metadata handling without exposing anyone's real information.

When testing images, dimensions and file size are only the beginning. Check how your system handles transparency, color profiles, CMYK files from print workflows, very large images that strain memory, and EXIF orientation tags that tell a viewer to rotate a photo taken with a sideways phone.

CHAPTER 12

Large files and performance

Performance problems often stay hidden until a large file arrives. A parser that reads a whole document into memory works perfectly with a few kilobytes and fails with a few hundred megabytes. A progress bar that never moves makes a slow upload look like a broken one.

Test with a range of sizes that matches your real limits: something tiny, something typical and something close to the maximum you allow. Measure memory as well as time, and repeat the test on a slow network connection, because mobile users experience your upload flow very differently from developers on office Wi-Fi.

Streaming techniques, which read and write data in chunks rather than all at once, are the usual cure. They keep memory use flat no matter how large the file grows, and they allow work to begin before the transfer has finished.

12.1 Key points

- Performance problems often stay hidden until a large file arrives.
- A parser that reads a whole document into memory works perfectly with a few kilobytes and fails with a few hundred megabytes.
- A progress bar that never moves makes a slow upload look like a broken one.
- Test with a range of sizes that matches your real limits: something tiny, something typical and something close to the maximum you allow.
- Measure memory as well as time, and repeat the test on a slow network connection, because mobile users experience your upload flow very differently from developers on office Wi-Fi.
- Streaming techniques, which read and write data in chunks rather than all at once, are the usual cure.
- They keep memory use flat no matter how large the file grows, and they allow work to begin before the transfer has finished.

Manual testing with sample files is a good start, but the real value comes from automation. Store a small set of representative files alongside your test suite, or download them by URL during a continuous-integration run, and assert on the results: the number of pages extracted, the dimensions of a generated thumbnail, the rows imported from a spreadsheet.

12.2 Sample data

The table below lists fictional test runs. It is here so that table extraction can be checked page after page.

Case	Sample file	Size	Result
TC-1201	sample-5-slides.pptx	458 KB	Pass
TC-1202	messy-folders.zip	2.3 MB	Skipped
TC-1203	sample-30s.mp3	9.5 MB	Pass
TC-1204	sample-30s.mp3	6.8 MB	Pass
TC-1205	sample-100-rows.csv	2.0 MB	Pass
TC-1206	sample-1920x1080.jpg	7.5 MB	Pass
TC-1207	sample-100-rows.csv	1012 KB	Pass
TC-1208	sample.xml	8.8 MB	Pass
TC-1209	sample-5-slides.pptx	10.2 MB	Pass
TC-1210	sample-1920x1080.jpg	6.3 MB	Pass
TC-1211	sample-10-pages.docx	10.3 MB	Pass
TC-1212	sample.xml	9.3 MB	Pass
TC-1213	sample-5-slides.pptx	9.1 MB	Fail
TC-1214	sample-1000-rows.xlsx	2.4 MB	Pass

12.3 Discussion

An accessible document can be read by people who use screen readers, magnifiers or keyboard navigation. In practice this means real headings rather than large bold text, meaningful alternative text for images, tables with marked header rows and a logical reading order.

Good test files also come in families. A single PDF tells you that your viewer can open a PDF; a family of PDFs tells you how it copes with one page and a hundred pages, with a few kilobytes and ten megabytes, with plain text, scanned images, form fields and passwords. Testing across a family is how you find the edges of a system before your users do.

When testing images, dimensions and file size are only the beginning. Check how your system handles transparency, color profiles, CMYK files from print workflows, very large images that strain memory, and EXIF orientation tags that tell a viewer to rotate a photo taken with a sideways phone.

12.4 Figure

Figure 12. Sample measurements

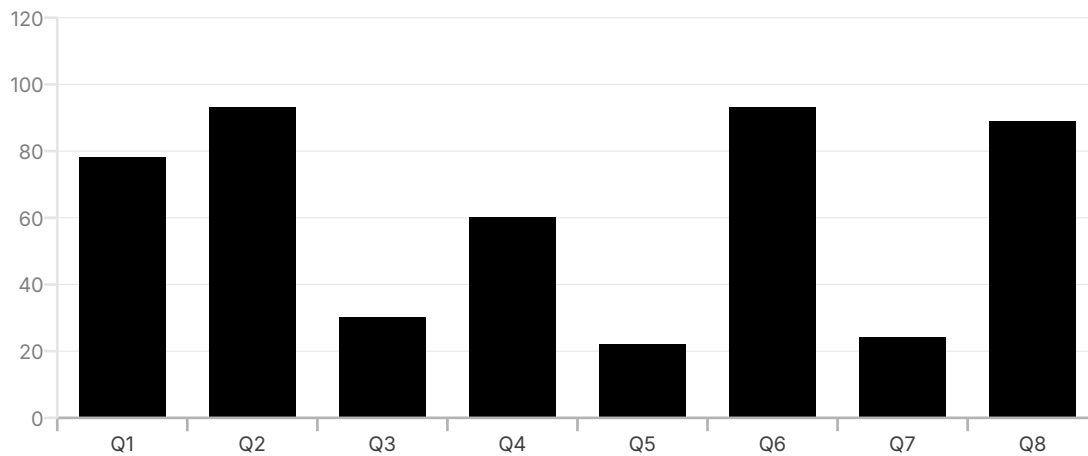


Figure 12: a vector bar chart drawn directly into the PDF. Values are random and seeded, so the chart is identical on every build.

Text in a PDF is not stored as flowing paragraphs. Each piece of text is placed at exact coordinates on the page, often a few words at a time. Extracting readable text therefore means reassembling lines, columns and reading order, which is harder than it sounds. Multi-column layouts, tables and footnotes are classic sources of extraction bugs, and they are worth including in any test set.

12.5 Checklist

Work through these steps when you test this area:

- 1 Test a password-protected document.
- 2 Try an empty, zero-byte file.
- 3 Confirm scripts inside SVG files are stripped.
- 4 Unpack an archive with deeply nested folders.
- 5 Upload a file one byte over the size limit.
- 6 Measure peak memory while processing the largest sample.
- 7 Verify the served Content-Type header.
- 8 Play a video from the middle using seek.
- 9 Confirm GPS metadata is removed from public photos.

12.6 Summary

This chapter looked at large files and performance. Performance problems often stay hidden until a large file arrives. Test with a range of sizes that matches your real limits: something tiny, something typical and something close to the maximum you allow. Streaming techniques, which read and write data in chunks rather than all at once, are the usual cure.

Key takeaway. They keep memory use flat no matter how large the file grows, and they allow work to begin before the transfer has finished.

Useful test videos carry visible information: a running timecode, a frame counter, the resolution printed on screen and a short beep or flash at regular intervals. With those clues you can spot dropped frames, audio drift and scaling problems by eye, without special tools.

CHAPTER 13

Accessible documents

An accessible document can be read by people who use screen readers, magnifiers or keyboard navigation. In practice this means real headings rather than large bold text, meaningful alternative text for images, tables with marked header rows and a logical reading order.

Accessibility also benefits automated systems. Search engines, summarization tools and data-extraction pipelines all work better when a document's structure is explicit. A well-structured file is easier for both people and programs to understand.

When you test a document viewer or converter, include files with headings, lists, tables and images, and check that the structure survives the round trip. A conversion that keeps the words but loses the headings has quietly made the document harder to use.

13.1 Key points

- An accessible document can be read by people who use screen readers, magnifiers or keyboard navigation.
- In practice this means real headings rather than large bold text, meaningful alternative text for images, tables with marked header rows and a logical reading order.
- Accessibility also benefits automated systems.
- Search engines, summarization tools and data-extraction pipelines all work better when a document's structure is explicit.
- A well-structured file is easier for both people and programs to understand.
- When you test a document viewer or converter, include files with headings, lists, tables and images, and check that the structure survives the round trip.
- A conversion that keeps the words but loses the headings has quietly made the document harder to use.

A good test file is boring in the best possible way. It opens everywhere, it is exactly what its name says, and it behaves the same way every time you use it. When a bug appears, you want to be sure the problem is in your code and not in a half-broken file you found somewhere online. That is why every file in this collection is generated from scratch, checked after it is written, and labelled with the details that matter: size, dimensions, duration, page count or row count.

13.2 Sample data

The table below lists fictional test runs. It is here so that table extraction can be checked page after page.

Case	Sample file	Size	Result
TC-1301	sample.pdf	9.8 MB	Pass
TC-1302	sample-10-pages.docx	2.9 MB	Pass
TC-1303	sample-5-slides.pptx	6.7 MB	Pass
TC-1304	sample-4032x3024.heic	10.4 MB	Skipped
TC-1305	sample-transparent-512x512.png	10.1 MB	Pass
TC-1306	sample.xml	4.2 MB	Pass
TC-1307	sample-10-pages.docx	1.5 MB	Pass
TC-1308	sample.txt	7.3 MB	Pass
TC-1309	sample-4032x3024.heic	4.0 MB	Pass
TC-1310	messy-folders.zip	639 KB	Pass
TC-1311	sample.xml	9.1 MB	Pass
TC-1312	sample.json	3.0 MB	Pass
TC-1313	sample.txt	6.9 KB	Pass
TC-1314	messy-folders.zip	2.6 MB	Fail

13.3 Discussion

Manual testing with sample files is a good start, but the real value comes from automation. Store a small set of representative files alongside your test suite, or download them by URL during a continuous-integration run, and assert on the results: the number of pages extracted, the dimensions of a generated thumbnail, the rows imported from a spreadsheet.

Text in a PDF is not stored as flowing paragraphs. Each piece of text is placed at exact coordinates on the page, often a few words at a time. Extracting readable text therefore means reassembling lines, columns and reading order, which is harder than it sounds. Multi-column layouts, tables and footnotes are classic sources of extraction bugs, and they are worth including in any test set.

Useful test videos carry visible information: a running timecode, a frame counter, the resolution printed on screen and a short beep or flash at regular intervals. With those clues you can spot dropped frames, audio drift and scaling problems by eye, without special tools.

13.4 Figure

Figure 13. Sample measurements

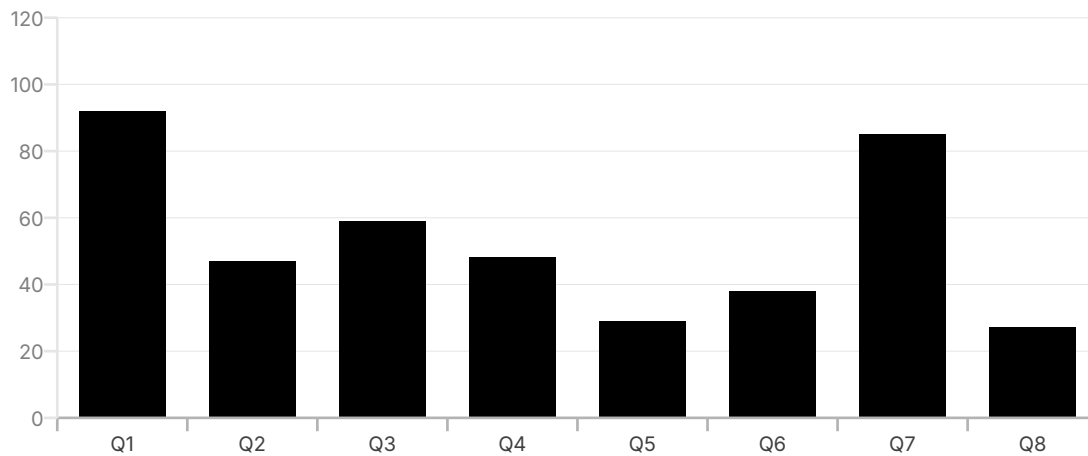


Figure 13: a vector bar chart drawn directly into the PDF. Values are random and seeded, so the chart is identical on every build.

Newer formats such as WebP and HEIC promise similar quality at a fraction of the size. WebP is now supported by every major browser. HEIC is the default photo format on iPhones, yet many web services still reject it or fail to convert it, which makes it one of the most common causes of upload support tickets.

13.5 Checklist

Work through these steps when you test this area:

- 1 Test a password-protected document.
- 2 Open the file in at least two different applications.
- 3 Try an empty, zero-byte file.
- 4 Import a CSV file that contains quoted line breaks.
- 5 Play a video from the middle using seek.
- 6 Confirm GPS metadata is removed from public photos.
- 7 Cancel an upload halfway through and retry.
- 8 Check audio and video stay in sync to the end.
- 9 Repeat the test on a throttled mobile connection.

13.6 Summary

This chapter looked at accessible documents. An accessible document can be read by people who use screen readers, magnifiers or keyboard navigation. Accessibility also benefits automated systems. When you test a document viewer or converter, include files with headings, lists, tables and images, and check that the structure survives the round trip.

Key takeaway. A conversion that keeps the words but loses the headings has quietly made the document harder to use.

Testing a media player or transcription service with several short clips, one long track and a few unusual encodings will surface most of these problems quickly. Keep at least one silent file and one very short file in the set, because edge cases at both ends of the scale are where players tend to fail.

CHAPTER 14

Automating tests with sample files

Manual testing with sample files is a good start, but the real value comes from automation. Store a small set of representative files alongside your test suite, or download them by URL during a continuous-integration run, and assert on the results: the number of pages extracted, the dimensions of a generated thumbnail, the rows imported from a spreadsheet.

Stable URLs and checksums make this reliable. If a file changes unexpectedly, the checksum comparison fails loudly instead of letting a test pass for the wrong reason. Keep the files small where possible so the suite stays fast, and reserve the largest samples for scheduled performance runs.

Over time, every bug you fix should leave behind a test file that reproduces it. That growing collection becomes one of the most valuable assets a team owns: a precise, executable memory of everything that has gone wrong before.

14.1 Key points

- Manual testing with sample files is a good start, but the real value comes from automation.
- Store a small set of representative files alongside your test suite, or download them by URL during a continuous-integration run, and assert on the results: the number of pages extracted, the dimensions of a generated thumbnail, the rows imported from a spreadsheet.
- Stable URLs and checksums make this reliable.
- If a file changes unexpectedly, the checksum comparison fails loudly instead of letting a test pass for the wrong reason.
- Keep the files small where possible so the suite stays fast, and reserve the largest samples for scheduled performance runs.
- Over time, every bug you fix should leave behind a test file that reproduces it.
- That growing collection becomes one of the most valuable assets a team owns: a precise, executable memory of everything that has gone wrong before.

PDF was designed to make a document look the same on every screen and every printer. A PDF file is a collection of numbered objects, such as pages, fonts, images and annotations, tied together by a cross-reference table that lets a reader jump straight to any object without reading the whole file. That structure is why a viewer can show page 80 of a long report almost instantly.

14.2 Sample data

The table below lists fictional test runs. It is here so that table extraction can be checked page after page.

Case	Sample file	Size	Result
TC-1401	sample-1000-rows.xlsx	5.3 MB	Pass
TC-1402	sample-30s.mp3	6.4 MB	Pass
TC-1403	sample-4032x3024.heic	8.0 MB	Pass
TC-1404	sample-5-slides.pptx	11.2 MB	Pass
TC-1405	sample-5-slides.pptx	5.4 MB	Pass
TC-1406	sample-transparent-512x512.png	2.7 MB	Pass
TC-1407	sample.txt	8.8 MB	Pass
TC-1408	sample.txt	6.0 MB	Pass
TC-1409	sample.pdf	5.3 MB	Pass
TC-1410	sample-5-slides.pptx	8.5 MB	Pass
TC-1411	sample.xml	10.4 MB	Pass
TC-1412	sample-4032x3024.heic	8.1 MB	Pass
TC-1413	sample.xml	7.3 MB	Pass
TC-1414	sample-100-rows.csv	1.5 MB	Pass

14.3 Discussion

A good test file is boring in the best possible way. It opens everywhere, it is exactly what its name says, and it behaves the same way every time you use it. When a bug appears, you want to be sure the problem is in your code and not in a half-broken file you found somewhere online. That is why every file in this collection is generated from scratch, checked after it is written, and labelled with the details that matter: size, dimensions, duration, page count or row count.

Newer formats such as WebP and HEIC promise similar quality at a fraction of the size. WebP is now supported by every major browser. HEIC is the default photo format on iPhones, yet many web services still reject it or fail to convert it, which makes it one of the most common causes of upload support tickets.

Testing a media player or transcription service with several short clips, one long track and a few unusual encodings will surface most of these problems quickly. Keep at least one silent file and one very short file in the set, because edge cases at both ends of the scale are where players tend to fail.

14.4 Figure

Figure 14. Sample measurements

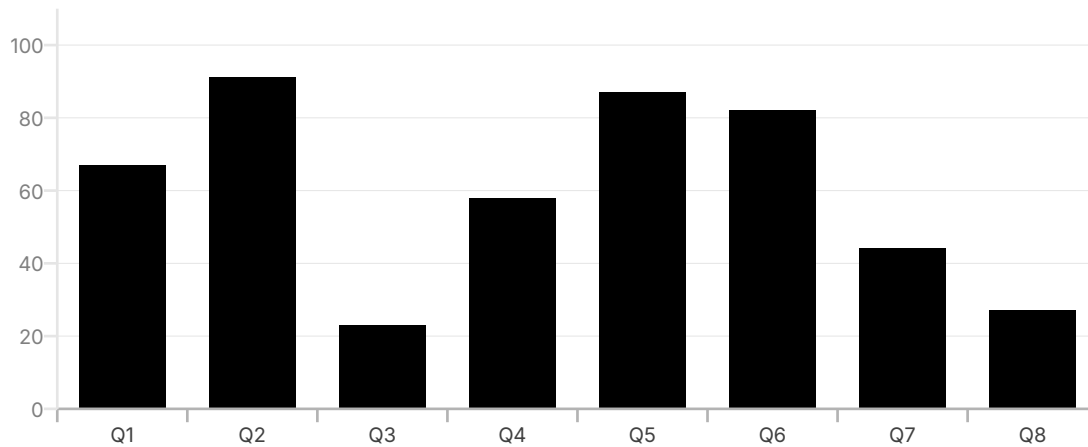


Figure 14: a vector bar chart drawn directly into the PDF. Values are random and seeded, so the chart is identical on every build.

H.264 video with AAC audio in an MP4 container is still the safest combination for playback everywhere. For streaming tests, keep a range of resolutions from 360p to 4K and a range of bitrates, because adaptive players switch between them as network conditions change.

14.5 Checklist

Work through these steps when you test this area:

- 1 Measure peak memory while processing the largest sample.
- 2 Upload the same file twice in a row.
- 3 Cancel an upload halfway through and retry.
- 4 Compare checksums before and after storage.
- 5 Try an empty, zero-byte file.
- 6 Parse a JSON file with deeply nested objects.
- 7 Play a video from the middle using seek.
- 8 Upload a file exactly at the size limit.
- 9 Check that text extraction keeps reading order.

14.6 Summary

This chapter looked at automating tests with sample files. Manual testing with sample files is a good start, but the real value comes from automation. Stable URLs and checksums make this reliable. Over time, every bug you fix should leave behind a test file that reproduces it.

Key takeaway. That growing collection becomes one of the most valuable assets a team owns: a precise, executable memory of everything that has gone wrong before.

XML is more verbose but brings namespaces, schemas and a long tail of enterprise use: RSS feeds, sitemaps, SOAP services and office document internals. Parsers must also be configured carefully, because some XML features can be abused to read local files or exhaust memory.