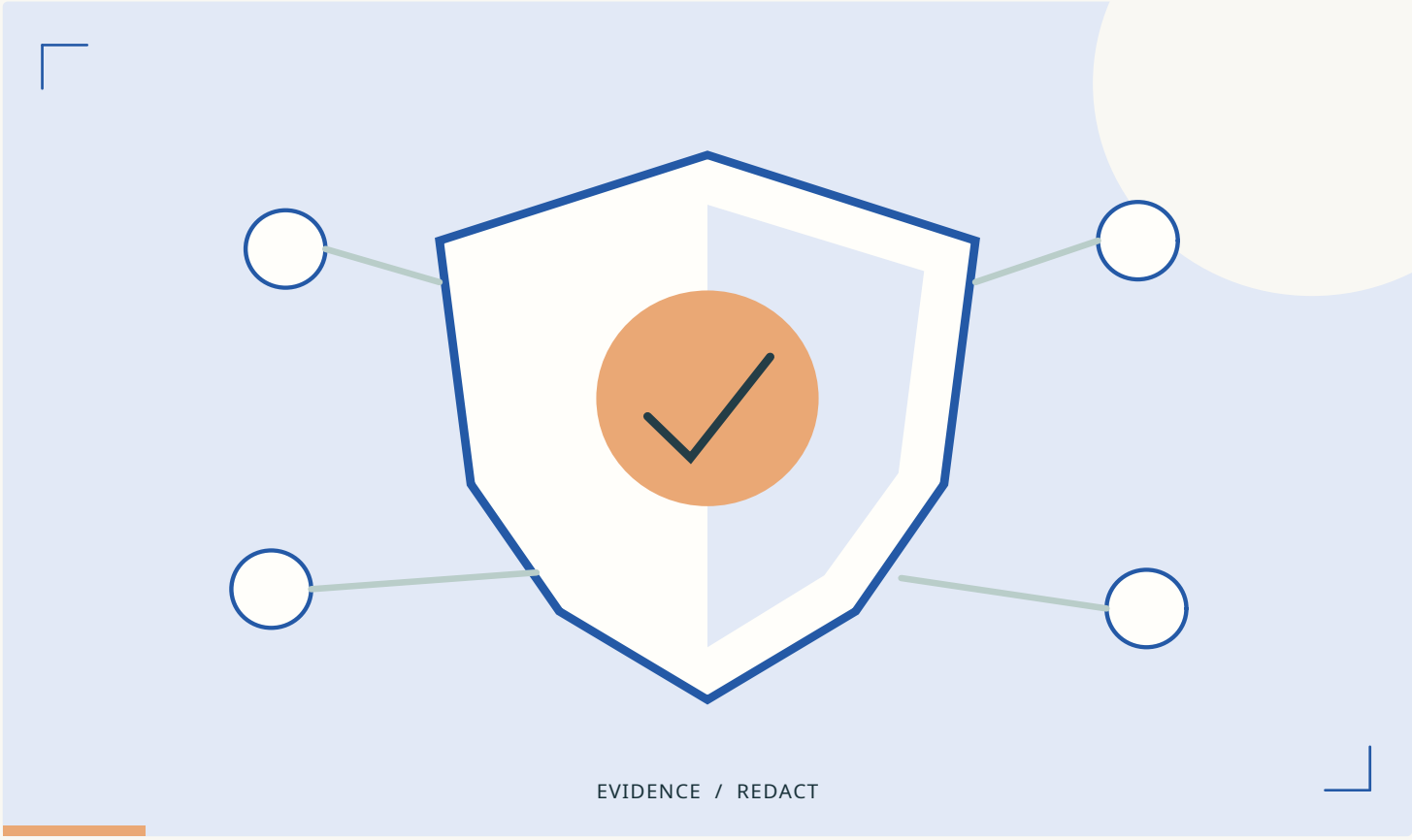
PRIVACY-SAFE SUPPORT EVIDENCE

Preparing Privacy-Safe Support Evidence



DIGITAL & TECHNOLOGY

Preparing Privacy-Safe Support Evidence

Learning outcomes

- Prepare a privacy-minimized support evidence bundle from a fictional error screen.
- Remove five specified unnecessary fields while preserving error code and reproduction context.
- Verify the final exported evidence rather than trusting an editable overlay.
- Convert a support question into an explicit relevance and disclosure boundary.
- Classify all ten source fields and map each of five exclusions to its omission method.
- Analyze full containment and partial overlap in a synthetic crop model without claiming actual image edits.
- Construct and reopen a labeled plain-text facsimile using an explicit allowlist.
- Distinguish scenario facts, source observations, inferences, and actual check results.
- Produce exact row-and-column-minimized synthetic log extracts with inclusive time boundaries.
- Explain representation-specific limits of overlays, PDF sanitization, and Office inspection.
- Reconcile versions and a two-file simulated handoff without transmitting evidence.

Audience: General workplace learners using synthetic practice materials

Setup

Use only the supplied fictional records. Do not access real accounts, visit practice URLs, change live permissions, execute downloads, or submit messages. For spreadsheet exercises use a blank local practice workbook with English Excel formula names and comma separators; some regional installations use semicolons. Keep raw inputs and worked outputs separate.

Study the foundations, work through the applied units, complete the practice tasks, and then attempt the assessment. Use only the supplied fictional practice data. Keep your original inputs and your revised work separate.

Core concepts

Define the support question first

The support task is to explain why a document preview fails. Useful evidence includes the error code, app version, relevant document type, and minimal steps that reproduce the behavior. A screenshot of the whole desktop may contain far more than needed: profile panels, neighboring records, notifications, or access material in an address bar. Decide which region supports the question before copying anything. The practice uses labels and placeholders only; learners should not insert their actual screen or employer data.

Remove information, not merely visual attention

Cropping can remove unrelated panels from the exported image. Redaction should make the excluded content unrecoverable in the copy being shared. A translucent highlight, blur, or movable rectangle in an editable document may leave the original content available. Work on a copy and produce a flattened image or other approved output, then reopen it and inspect it. For PDFs or office files, hidden text, comments, metadata, and underlying objects may require additional checks with approved tools. This exercise uses a simple fictional flat screen, not forensic redaction of a real document.

Preserve diagnostic context

Over-redaction can make support evidence useless. Keep PREVIEW-42, the PracticeDocs v2.4 label, the PDF file type, and the three reproduction steps. Replace unnecessary identity fields with neutral labels only when a label helps describe the interface. Do not replace a useful error code with “[hidden]” just because it looks technical. The right balance is not maximum removal; it is enough relevant evidence with no unnecessary data. Record what was removed so the reviewer can check completeness without seeing the removed values.

Verify the bundle as the recipient would see it

The bundle contains the sanitized screen and a concise reproduction note. Open the exported copy, inspect all four edges and any included address bar or notifications, and verify the five exclusions. Check that the error code is legible at normal size and the steps make sense without private context. Keep the unmodified source in the isolated practice folder rather than attaching it accidentally. Name the final artifact distinctly and list only the intended files in the handoff. No support submission occurs in this lesson.

Worked example

A mock screen shows PREVIEW-42 in the center and a profile panel on the right. The support question concerns PDF preview, so crop to the app title, document-type area, and error panel. The resulting evidence preserves “PracticeDocs v2.4,” “Type: PDF,” and “PREVIEW-42: Preview could not be generated.” Exclude the address bar’s access-token placeholder, the profile email placeholder, and unrelated sidebar records. Reopen the exported image and check that the crop did not remove the error number. Pair it with a clearly labeled scenario note: “The fictional operator opened a PDF, selected Preview and received PREVIEW-42.” These are supplied scenario facts, not steps executed by the learner in a real application. Do not attach the original full-screen mock as an extra file.

Foundation practice

Mock screen

Top browser bar: https://practicedocs.example/preview?access=TOKEN_PLACEHOLDER
Application header: PracticeDocs v2.4
Right profile: EMAIL_PLACEHOLDER
Left sidebar record: CLIENT_NAME_PLACEHOLDER
Left sidebar contact: PHONE_PLACEHOLDER
Bottom pop-up: PRIVATE_CHAT_PLACEHOLDER
Main document type: PDF
Main action button: Preview
Main error: PREVIEW-42 — Preview could not be generated
Main timestamp: 2026-10-19 10:05 UTC

Exactly five exclusions

- TOKEN_PLACEHOLDER in the address bar
- EMAIL_PLACEHOLDER in the profile
- CLIENT_NAME_PLACEHOLDER in the sidebar
- PHONE_PLACEHOLDER in the sidebar
- PRIVATE_CHAT_PLACEHOLDER in the pop-up

Reproduction facts

- Scenario fact: a fictional operator opens a file named SampleGuide.pdf in the fictional PracticeDocs v2.4 app. The name is part of the scenario; no PDF file or executable app is supplied or required.
- Scenario fact: the operator selects Preview once.
- Scenario fact: the main panel displays PREVIEW-42; the document itself is not modified.

Artifact method

Recreate this text-only mock screen in a local drawing or document tool, or use a clearly labeled text facsimile. Produce a sanitized exported copy. The learner output must visibly preserve the diagnostic fields and exclude all five placeholders. No genuine data is necessary. The learner recreates only the supplied screen description and exports a sanitized facsimile; do not try to open SampleGuide.pdf or find, install, or use PracticeDocs. Label the reproduction note as scenario-based, not an observed application test.

Foundation tasks

- Create a sanitized mock screen and a minimal reproduction note. Record which crop or permanent-removal step excluded each of the five fields.
- Reopen the exported artifact and check error-code readability, absence of excluded fields, and the exact attachment list.
- Capstone: explain why a second full-screen attachment would defeat the minimization even when the first attachment is correctly sanitized.

01 Turn a support question into an evidence boundary

A useful evidence bundle answers a bounded question. It is not a scrapbook of everything visible when something went wrong. Begin with the decision the recipient must make: in this course, understand the reported PDF preview failure well enough to identify a next diagnostic question. That goal needs the application version, document type, selected action, error, and a short account of the scenario. It does not need the fictional operator's identity or access material. The boundary follows the question, not the dimensions of the original window.

Separate relevance from permission. A detail might help a technician while still being outside an approved disclosure. Conversely, a detail might be permitted in a practice environment yet irrelevant to the diagnosis. Use two questions in order: does this field contribute to the agreed question, and is its inclusion allowed here? For the supplied case, the five exclusions are fixed even though their values are placeholders. Practicing with harmless strings does not make the exclusion exercise optional.

Read assets/scope_brief.txt before editing. It adds explicit local requirements for the expanded exercises: preserve the displayed timestamp alongside the original diagnostic fields; use a clearly labeled UTF-8 text facsimile; keep the simulated handoff to exactly two files. These conditions define this exercise, not a universal support policy. The original practice still allows a drawing or document tool, but a real PDF or application installation is neither supplied nor necessary. Every reproduction statement must remain scenario-based.

A boundary also defines what you cannot conclude. PREVIEW-42 is a supplied observation within a fictional story; it does not prove a corrupted file, attack, permissions fault, or application defect. Those are possible investigation topics, not established causes. Do not improve the story by inventing a retry, an operating system, a network outage, or a successful workaround. Honest incompleteness preserves the value of the evidence because the next reader can distinguish a known result from a suggested explanation.

E01-X1 Write the evidence contract

Using scope_brief.txt and original_practice_inputs.json, write a six-part contract: question, audience, required screen fields, excluded categories, allowed handoff files, and prohibited activities. Add one sentence stating what the error does not establish.

Submit: A local contract containing the exact diagnostic values where supplied, with no claim that an application test or support submission occurred.

02 Inventory fields before choosing a removal method

A field inventory makes a vague instruction such as remove personal information testable. Read assets/screen_fields.csv as ten records, not ten commands. Each record identifies a source location and a value. The coordinate columns are teaching geometry for a later exercise; they do not describe an actual browser capture. Keeping the record identifier in an internal decision ledger lets a reviewer trace a decision without copying the excluded value into another shareable artifact.

Classify each record against the brief rather than its appearance. A string that looks technical can still be access material. A short error code can be essential rather than private. The address bar contains a token placeholder; the main panel contains PREVIEW-42. Treating both as arbitrary codes would either leak access context or destroy the diagnosis. Similarly, the app version matters because behavior can differ between versions, while a profile email does not explain this particular preview outcome.

Use explicit decision states. RETAIN means the brief requires the field and the output must contain its supplied value. EXCLUDE means the brief requires its absence from the outgoing copy. NEEDS_DECISION means the current brief does not resolve its role, so the field cannot silently enter the output. There are no NEEDS_DECISION records in this ten-row dataset because the local requirements deliberately cover every row. In unfamiliar work, an unresolved record should trigger a question, not a guess disguised as a completed review.

Count fields and categories carefully. One sidebar region contains two excluded fields: client name and phone. Removing that region may satisfy two exclusions, but the ledger still needs two checks. A single pop-up contains the fifth excluded category. Conversely, writing five removal actions does not prove that the right five fields are absent. Assessment follows the mapping from each source record to the final copy. Distinguish a planned decision from an actual removal and from a verified absence.

E02-X1 Create a complete field ledger

Classify F01–F10 as RETAIN or EXCLUDE using the brief. For each row provide its ID, decision, evidence purpose or exclusion category, and a proposed method. Do not copy excluded placeholder values into the proposed outgoing copy. Count retained rows and excluded rows separately.

Submit: Ten ledger rows, two totals, and a sentence distinguishing this plan from an executed export check.

03 Use crop geometry without pretending to crop an image

Cropping is a spatial selection, so its safety depends on where every required and excluded region lies. The synthetic canvas in screen_fields.csv uses coordinates from 0 to 120 horizontally and 0 to 100 vertically. Each field occupies the rectangle given by x1, y1, x2, y2. The coordinates model the entire field, not merely its first letter. There is no image to edit. Your task is to reason about candidate boundaries, then use an approved text facsimile for the actual exported practice artifact.

For this model, a field is fully retained only if the candidate rectangle contains all four of its boundaries. Any positive-area overlap with an excluded field is a privacy failure, even if the overlap would show only a fragment. A field touching the crop edge without overlapping has zero area inside and is not included. These conventions remove ambiguity about partial characters. They are mathematical rules for the supplied rectangles, not measurements of font antialiasing or a claim about a particular screenshot tool.

Check the two objectives separately. A candidate can exclude every private field and still fail because it cuts off the app header. Another can retain all diagnostic content while exposing part of a pop-up. A crop is suitable only when both requirements hold. Examining only the center error panel misses boundary failures: the address bar above it, the profile to the right, the sidebar to the left, and the notification below it are all independent sources of unwanted content.

Worked example on a different miniature canvas: the required field spans (10,10) to (30,20), and an excluded label spans (35,10) to (45,20). A crop from (8,8) to (32,22) fully contains the requirement and has no overlap with the exclusion. Extending its right edge to 38 adds a three-unit-wide private fragment. Moving its lower edge to 18 removes two units of the required field. Neither larger nor smaller automatically means better.

In real approved workflows, inspect the exported boundary rather than trusting the selection handles in an editor. Some operations change the visible view without removing other stored content. A text reconstruction avoids that particular ambiguity here because you build a new copy from selected facts. Do not describe the geometry exercise as an executed crop, and do not write that hidden pixels were removed. The strongest truthful conclusion is that one proposed boundary meets the declared rectangular model.

E03-X1 Evaluate four proposed boundaries

Use crop_candidates.csv with screen_fields.csv. For A–D list fully contained required IDs, missing required IDs, and every excluded ID with positive-area overlap. Decide which candidates satisfy both objectives. Show the overlap or containment reasoning for D.

Submit: Four explicit candidate results and one selected boundary, labeled as a synthetic geometry analysis rather than an executed image crop.

04 Build a fresh copy from an explicit allowlist

An allowlist names what may enter an output. This reverses the fragile habit of copying everything and trying to remember every item to erase. For the text facsimile, start a new plain-text document, not a duplicate of the full source. Transfer only the approved diagnostic records. Add a visible label saying SYNTHETIC TEXT FACSIMILE. The label is essential provenance: a reconstruction is useful evidence about a supplied scenario, but it must not masquerade as a screenshot captured from a running application.

The output grammar for this exercise is deliberately simple. The first line is the synthetic label. Each later line is an original retained region label, a colon, one space, and the original value. Preserve source order and the exact supplied spelling, case, punctuation, and timestamp. Do not include source row IDs, coordinate columns, excluded categories, a URL, or commentary about the removal. Those belong in the separate internal ledger. A final newline is acceptable and does not change the evidence.

Positive selection has a useful failure property: a newly added source field is not automatically copied merely because no one anticipated it in a removal list. It still requires an explicit relevance decision. This does not make allowlisting infallible. You can choose the wrong field, mistype a diagnostic value, or paste the entire source into a retained line.

Therefore the field contract and exported-copy comparison remain necessary. An allowlist is a controllable construction method, not a certificate of privacy.

Worked example, not the assessed output: for a fictional queue case, approved labels are Device, Action, and Result. The resulting text might read Device: DemoPrinter 3; Action: Print; Result: QUEUE-7, preceded by a synthetic label. The source's Owner and Billing panel never enter the new document. Replacing the Owner value with asterisks would add an unnecessary field and might imply that identity is needed. Omitting the whole owner field better expresses the narrow diagnostic purpose.

Keep the source unchanged. A separate output makes comparison possible and avoids making the original practice unrepeatable. Use a neutral name such as B189_screen_v1.txt in your own local output folder. The extension must match the content: renaming a rich document to .txt does not convert its internal representation. Choose an actual plain-text save operation, then close and reopen that named file. Creation, saving, and verification are separate events, each requiring its own honest record.

E04-X1 Construct the allowlisted facsimile

Create B189_screen_v1.txt from the retained records identified in Exercise E02-X1. Follow the six-line grammar described above. Keep the original inputs untouched. Record the output name and the IDs you selected, but do not claim a reopen check yet.

Submit: One new UTF-8 plain-text facsimile and an internal creation record. The actual file is assessed, not a promise to create it.

05 Write a removal record that does not recreate the leak

A removal record explains how the outgoing copy differs from the source. Its purpose is accountability, not another collection of the excluded values. Use source IDs, region names, exclusion categories, and the method chosen. For this course, the method is omission during fresh text reconstruction. That wording is more accurate than saying the original was destroyed or all traces were erased. The raw synthetic source still exists separately, and no action has been taken against any real account or application.

Record three stages independently. Planned means a decision exists but the action has not been performed. Created means the new local copy was saved with the intended selection. Verified means a stated check was performed on that exact saved copy and produced a result. A learner who has completed the inventory but not saved a file has a plan, not a sanitized artifact. A learner who saved the file but has not reopened it has a created artifact, not verified evidence.

A useful row reads: source region, category, method, output version, check method, result. Do not add a before-and-after column containing the original access value. In a workplace, such a column can defeat minimization when the ledger is accidentally attached. Here the five synthetic placeholders are harmless practice inputs, but the habit matters. The outgoing note should describe excluded categories, if needed, without reproducing their values; the outgoing screen should simply omit their regions.

Worked example: an analyst reconstructs a synthetic device panel and excludes a Customer field. The internal entry says Customer region; identity; omitted from newly created text; device_v2.txt; exact label/value comparison after reopening; PASS. If the check has not occurred, the last two cells must say not performed and NOT_RUN instead. Filling those cells with PASS because the analyst intended to omit the field creates a false assurance that another reviewer may rely on.

E05-X1 Design the five-row removal record

For each original exclusion, write a record using its F-ID, region, category, and the actual or proposed omission method. State CREATED only if you created E04-X1; otherwise state PLANNED. All export checks remain NOT_RUN until performed in E12-X1.

Submit: Five explicit records with accurate action states, no excluded values copied into the outgoing artifacts, and a distinction between preserving the source and minimizing the new copy.

06 Distinguish appearance from stored content

What a viewer displays and what a file stores are related but different. A rectangle can cover text on a canvas while the text object still exists underneath. A crop view can hide an area without proving that the stored source was removed. An editor may offer a proper removal operation, but the existence of a black shape alone does not establish that operation occurred. The question is not whether the picture looks censored; it is whether the specific outgoing representation still contains the excluded material.

assets/overlay_model.json makes this distinction inspectable without installing a graphics tool. It is explicitly a synthetic serialized layer model. Its canvas_display describes what the imagined canvas presents, while stored_objects lists what the JSON actually retains. The object visibility value covered means a displayed object obscures it in the model; it does not mean deleted. The export_observation value NOT_RUN means no export operation was performed. These are separate facts, not competing descriptions of one state.

Read the model as data. Opening its text and finding a placeholder proves that the model file contains that string. It does not prove how a particular PDF viewer behaves, that a genuine image has hidden layers, or that any production file is unsafe. This limited conclusion is enough for the exercise: sending this JSON would send the listed underlying text. A claim about another format requires evidence about that format and its exact outgoing copy.

Worked example on a separate model: a canvas shows a solid box where an Account field once appeared, but its object list still includes Account: DEMO-ACCOUNT. A reviewer can reject the proposed source-file attachment without moving the box because the serialization itself establishes retention. If a separately created text output contains only Device and Error, the reviewer must inspect that output independently. The unsafe source model does not automatically make every derivative unsafe, just as a safe derivative does not sanitize the source.

Do not overcorrect by assuming that all flat-looking files are unsafe or all flattened files are safe. A workflow must consider its representation, its export process, and appropriate

checks. In this course, you can inspect the entire plain-text output directly and compare it with an allowlist. That bounded method is preferable to making forensic claims about PDFs or Office files that you have neither supplied nor examined. The method teaches disciplined evidence handling without pretending to certify every possible file format.

E06-X1 Audit a synthetic overlay model

Read overlay_model.json. Identify the stored object that defeats the claimed visual removal, quote only its object ID and category, and state what sending the JSON would disclose. Separate three conclusions: established by the file, suggested by the model, and not tested.

Submit: A three-part audit naming the retained object, explaining why covered is not deleted, and explicitly leaving real editor/PDF export behavior untested.

07 Recognize when the file format changes the problem

The supplied text facsimile is intentionally narrow. A real document may contain visible content plus comments, metadata, hidden text, embedded objects, or other representations. Sanitizing one visible region does not necessarily address all of them. Before using a richer format in an authorized workplace process, identify what can be stored and what inspection method is appropriate. Do not carry a plain-text assurance over to a PDF or Office file merely because the visible words look the same.

Adobe's official Acrobat Pro guidance distinguishes selecting material for redaction from applying the redaction and saving the sanitized document. Its linked sanitization guidance addresses hidden information separately. The practical lesson is to distinguish marked, applied, saved, and checked states. This course does not run Acrobat and does not establish that any PDF has been redacted. A highlighted passage, opaque annotation, or a filename containing sanitized is not sufficient evidence that the documented workflow was completed.

Microsoft's Document Inspector guidance recommends working on a copy and explains that documents can contain hidden or personal information. It also documents limits, including cases of text hidden by methods other than the hidden-text setting and objects covered by other objects. Treat a tool's report within its stated coverage. A clean report is useful evidence about the checks it performed, not a universal guarantee that every possible data-bearing representation was removed.

Worked example: a fictional review request contains an editable report with a visible customer name removed, an author property, and a reviewer comment. The visible edit addresses only one surface. A reasonable plan identifies the author property and comment for appropriate review, preserves a separate source, and checks the resulting outgoing copy. It does not claim those changes already happened. If approved tools or format expertise are unavailable, the correct response is to hold the unverified artifact or prepare an allowed minimal alternative.

Use a format boundary statement in review notes. For this course it can say: the check covers the complete UTF-8 contents of the named text file; no PDF, Office document, image-layer, browser, keyboard, or assistive-technology test was performed. Such a

statement is not an excuse to skip necessary checks. It prevents a narrow test from being misunderstood as a wider certification. It also tells a future reviewer exactly what additional work would be needed if the delivery format changes.

08 Write reproduction context without inventing a test

Reproduction context describes an action sequence and its result, but the source of that description matters. In the original practice, a fictional operator opened SampleGuide.pdf in PracticeDocs v2.4, selected Preview once, and received PREVIEW-42. The document remained unchanged. Those are supplied scenario facts. They are not steps the learner performed, and the named application and PDF are not supplied. A well-written note preserves the sequence while making that provenance unmistakable.

Use five parts: context, action, expected behavior, reported outcome, and limits. Context names the app version and file type. Action states the single Preview selection. Expected behavior is a PDF preview under the scenario. Outcome names the exact error rather than merely saying broken. Limits explain that the account is scenario-based and no actual application test occurred. The displayed timestamp belongs to the scenario, not to the learner's later export verification.

Verb choice controls implied evidence. The fictional operator selected is supported; I reproduced is not. The scenario reports is appropriate; the server failed because is an unsupported causal assertion. No actual application test was performed is clearer than tested conceptually, which can blur the boundary. Avoid imperative wording such as open SampleGuide.pdf when presenting the case, because a reader could interpret it as an instruction to find a nonexistent file or run an unsupplied application.

Worked example, separate case: a supplied training story says a demo reader selected Export and saw EXPORT-9. A precise note says the training story reports that action and error; an export file was expected; no execution was performed for the note. It does not add an operating system, a second attempt, or a resolution. The absence of those details is useful: it shows the next investigator what is not yet known instead of disguising assumptions as completed tests.

Keep the filename's role proportionate. SampleGuide.pdf is a scenario label and supports the document-type context; it is not a downloadable attachment. In real work, a filename can itself reveal a customer or project, so an approved generic label may be enough. Here preserve the supplied scenario name in the note because the exercise explicitly requires it. Do not invent a path or claim that the file's internal content was inspected. The only actual file work required is creation and checking of the synthetic evidence copies.

E08-X1 Draft the scenario-based note

Create B189_note_v1.txt with a synthetic/scenario label, app version, scenario filename and type, one Preview selection, expected preview, reported PREVIEW-42, unchanged document, and displayed scenario timestamp. Explicitly state that no actual app test occurred. Do not claim any export check in this note.

Submit: One local note that preserves all supplied reproduction facts while separating them from actions actually performed on the facsimile.

09 Give every knowledge and test state a clear meaning

Evidence handling often fails through ambiguous status words. Safe might mean a reviewer saw no problem, a search was completed, or simply that the author intended to remove information. Replace such shorthand with a defined status and its basis. For a factual claim, identify whether it is directly visible in a supplied record, asserted by the fictional scenario, inferred, contradicted, or not established. For a test, record whether it ran and what it found. These are different dimensions.

Use SCENARIO_FACT for statements explicitly supplied about the fictional operator or application behavior. Use SOURCE_OBSERVED for facts you can establish by reading an actual supplied input file, such as its text or number of rows. Use NOT_ESTABLISHED when the evidence does not support the claim. An inference may be worth investigating, but label it as an inference and do not place it in the note as fact. In assets/claims.csv, the basis column provides all the information needed for the requested classification.

For export checks use PASS, FAIL, and NOT_RUN precisely. PASS means the specified check ran on a named copy and met its acceptance rule. FAIL means it ran and found a violation. NOT_RUN means it did not run and supplies no result. If a check cannot complete, use INCONCLUSIVE and state why, rather than silently mapping uncertainty to PASS or FAIL. This course's supplied candidate records explicitly say which states are present; no blank value secretly means unknown.

Worked example: a reviewer intends to search demo.txt for a private marker but has only opened the folder. The correct status is NOT_RUN. After reading the file, finding the marker, and recording the filename, the exclusion check is FAIL. After creating another version, the earlier failure remains a valid historical record, but it does not automatically describe the new file. The new version starts unchecked. That distinction prevents both false confidence and permanent rejection based on an obsolete result.

A useful report combines a narrow fact with a scope statement: the exact-marker search on copy v2 returned zero matches; visual readability has not yet been assessed. Zero matches may establish the absence of known strings in that representation, but it cannot establish that every unforeseen identifier is absent. This is why the workflow combines allowlist construction, full-content review, exact-value comparison, and package review rather than relying on a single green status.

E09-X1 Classify eight evidence claims

For C1–C8 in claims.csv, assign SCENARIO_FACT, SOURCE_OBSERVED, or NOT_ESTABLISHED. Rewrite every unsupported claim as an honest statement of the available evidence or missing check. Do not perform an application test to try to make an unsupported claim true.

Submit: Eight classifications and concrete corrected wording for each NOT_ESTABLISHED claim.

10 Minimize addresses and identifiers by purpose

An address bar may combine useful navigation context with unnecessary identity or access information. The domain and path can describe a feature, while a query parameter or fragment can carry a record identifier or token. Do not assume every URL is forbidden, and do not assume every part of a URL is harmless. In the original screen the access parameter is one of the five required exclusions. The support question can be answered without reproducing that address at all.

When only the feature is required, a plain feature label can be safer and clearer than a partially edited URL. If a specific path is genuinely required by an authorized request, construct the allowed path deliberately and review its components. Blindly deleting one familiar parameter is weak when another parameter or fragment may contain unrelated data. Likewise, keeping the first or last characters of access material is not justified here: the brief requires absence, not a partly recognizable token.

assets/url_cases.csv supplies four entirely fictional strings for analysis. Do not visit them. Its requested_context column defines what the fictional analyst needs in each case. For feature-only requests use the plain text Feature: Preview. For the help-path request use Path: /help/preview. For feature-and-type use Feature: Preview; Type: PDF. These output conventions are local to the exercise and deliberately avoid URLs. Explain which components were unnecessary rather than reporting that a website was opened or a token was invalidated.

Worked example: a separate demo link has a Reports path and an account query. If an approved question asks only which feature showed the error, Feature: Reports retains the useful distinction without passing the account identifier. If the question instead concerns routing behavior, a path may be necessary and requires a more specific evidence contract. Minimization is contextual; a memorized rule such as remove all query strings can lose relevant configuration just as keeping all query strings can disclose too much.

OWASP's logging guidance identifies access tokens and other secrets as values that usually should not be recorded directly, and discusses special handling of identifiers. The principle supports caution about copying diagnostic material indiscriminately. It does not authorize changing production logging, rotating credentials, or sending source addresses to another service. In this practice, the only action is local analysis of synthetic strings. If real access material appeared in a workplace case, stop unintended disclosure and follow the organization's authorized process rather than experimenting with it.

E10-X1 Create four minimized context lines

For U1–U4, produce the exact plain-text context requested using the conventions above. List the omitted component types for each and explain why the help-path case differs from the access-bearing preview address.

Submit: Four context lines and four concise component decisions; no live browsing, token testing, or URL submission.

11 Extract a bounded log without exporting every column

Support evidence sometimes comes from rows rather than screens. The same minimization logic applies twice: select relevant events, then select relevant fields within those events. A narrow time window does not make every column appropriate. A relevant error row can still contain a profile, access value, or free-text detail that should not leave the source. Conversely, removing those columns does not justify including unrelated users' events or a whole day of activity.

assets/synthetic_log.csv is a separate fictional dataset, not proof of additional actions in the original PracticeDocs story. Read log_request.txt for its exact predicate. All times end in Z, meaning UTC; the window includes both endpoints. The interaction, action, document type, and time requirements must all hold. Apply AND between these requirements, with OR only between the two allowed action names. Keep source order rather than sorting by whichever column looks important.

Projection means choosing the output columns. The permitted columns are row_id, event_time, action, file_type, result, and app in that order. The interaction field helps select records but is not requested in the output. Profile, access, and detail are excluded even from selected rows. This illustrates a subtle distinction: data can be useful for an internal decision without being necessary in the recipient's evidence. Do not copy the entire row and hide columns in a spreadsheet; create a new extract containing only the permitted columns.

Worked example on a separate log: events W1 and W2 both belong to a required session, but W1 falls before an inclusive 09:00 window and W2 occurs exactly at 09:00. W2 is selected if its other conditions hold. If W2 has a secret_detail column, that column remains excluded. An endpoint test checks row logic; a column test checks disclosure logic. Testing only the total number of selected rows could miss a wrong substitution that preserves the count.

Repeated result strings are not automatically duplicates. Two distinct row IDs can describe separate records of the same error. Retain qualifying rows as instructed, but do not infer that each error record represents another human click. Systems can log multiple events for one interaction. The original scenario still specifies one Preview selection. Also distinguish event time from when a reviewer generated or checked the extract. A review at noon does not turn a 10:05 event into a noon event.

E11-X1 Produce the exact six-column extract

Apply log_request.txt to synthetic_log.csv. Create a new CSV with the specified header and qualifying rows in original order. Provide a separate exclusion ledger giving each rejected row ID and every failed selection condition. Do not include profile, access, interaction, or detail in the extract.

Submit: The complete output CSV, selected and rejected IDs, exact row count, and a statement that this synthetic log does not establish additional button selections in the original scenario.

12 Verify the actual exported text copy

Verification begins after saving, on the artifact the recipient would receive. Close the editing view and reopen the named text file from your local output folder. Reading a different copy, an unsaved buffer, or the original source does not verify the outgoing copy. Record the filename, what you actually opened, and the checks you performed. If you cannot reopen it, leave the result incomplete rather than substituting confidence in the construction method.

Check privacy and utility separately. For privacy, read the entire text and search for each of the five exact excluded placeholder strings from the source. A zero-match result for each is necessary here, but review the whole content too. For utility, compare each retained value with the source, including version, file type, action, complete error, and timestamp. This catches over-removal and transcription errors that a privacy search cannot detect. Confirm that the synthetic label remains visible and that the file really is plain text.

Perform a recipient-view readability check within the format you actually use. In a text editor, inspect the first and last lines, wrapping, missing characters, and any text beyond the initial view. A long line can place a token off to the right even when the visible pane looks clean. For images in an approved real workflow, edges and legibility require appropriate visual checks, but no image or browser test is being claimed here. Match the check to the artifact rather than copying a generic checklist blindly.

Worked example, unrelated to the assessed case: a learner saves a three-line demo queue note, closes it, reopens it, and finds that the final error digit is missing. The exclusion search passes because no private marker is present, but the required-value comparison fails. The correct overall decision is HOLD. After creating a corrected version, repeat both checks. A later utility fix can accidentally paste unwanted text, so the earlier privacy pass cannot simply be carried forward.

A local text search establishes facts about decoded text, while a byte-level comparison can establish exact file identity. Neither proves that the document's factual story is true. In this course the story is explicitly fictional. Keep two statements separate: the exported copy contains the specified scenario fields, and the scenario says the operator saw an error. Never report that the exported-copy check reproduced the application defect. It tested the evidence representation, not the application.

E12-X1 Reopen and inspect your two outputs

Reopen the actual E04-X1 facsimile and E08-X1 note you created. Record separate PASS, FAIL, or NOT_RUN results for exact exclusions, required screen values, scenario labeling, note facts, and file identity. Fix any defect in a new version and repeat affected checks.

Submit: An observed local verification log naming the files and check results. Do not copy a model PASS entry if you have not performed its check; a truthful NOT_RUN is incomplete work, not a false pass.

13 Treat the attachment list as part of the evidence

A bundle can fail even when its main artifact is excellent. Adding the original screen as a helpful comparison reintroduces every excluded field. Adding the internal ledger can reveal values that the outgoing screen omitted. Adding an unverified newer copy can replace a checked copy with a different risk. The final attachment list is therefore an independent object of review, not administrative decoration after the important work is finished.

The course's simulated handoff is deliberately limited to one sanitized screen facsimile and one scenario-based reproduction note. No upload, email, ticket, or support submission occurs. A manifest is just the list you would intend to provide in the simulation. Keep raw inputs, scratch versions, extraction exercises, and review ledgers outside that list. A relevant local practice file does not automatically belong in the outgoing bundle. Relevance to learning and relevance to the recipient are different tests.

assets/package_candidates.csv supplies hypothetical observations about six candidate files. The files themselves are not supplied. Read package_states.txt before deciding. PASS means the specified hypothetical check succeeded, FAIL means it found a defect, and NOT_RUN means no check happened. PRESENT and ABSENT refer only to the scenario label. Do not combine NOT_RUN with FAIL or reinterpret it as an unknown boolean. A candidate may be well-labeled but unverified, or verified as a file while its content is inaccurate.

Worked example: a separate queue bundle contains queue_screen_v2.txt with completed checks, queue_note_v2.txt with completed checks, queue_source.txt with private source fields, and queue_review.csv. The first two satisfy the fictional request. The source defeats minimization, and the review ledger is not requested. If a later v3 screen exists without a completed check, its newer filename is not a reason to prefer it. Selection depends on appropriate content and completed checks, not recency alone.

A manifest should use neutral names that distinguish roles and versions. Avoid identifiers in filenames when they are unnecessary; the outside of a file can disclose information before someone opens it. Also review the note body because excluded content can be copied there accidentally. Privacy is a property of the whole outgoing set, not a badge earned by one sanitized screenshot. A final count of two is useful, but only the correct two files satisfy the brief.

E13-X1 Select the hypothetical handoff pair

Using package_candidates.csv and package_states.txt, choose exactly one screen and one note. List all six candidate IDs with INCLUDE or EXCLUDE and a specific reason. State that your conclusion is based on supplied hypothetical observations, not actual candidate-file execution.

Submit: An exact two-filename manifest and a six-row decision ledger. No attachment or transmission is performed.

14 Repair failures without erasing their history

A failed review is useful if it identifies a specific defect and leads to a controlled correction. It is not useful if the record is rewritten to imply the defect never existed. Keep the earlier result attached to its version, create a corrected copy, and check the new copy. This allows another reviewer to understand what changed and prevents a stale PASS from being applied to different bytes.

assets/review_events.csv is a supplied fictional review history. Its scenario_record_time values describe events in that history, not actions performed by you. The action column distinguishes rebuilding from checking. COMPLETED for REBUILD means the scenario says a new text copy was made; it is not a privacy or utility pass. PASS and FAIL belong to specific checks. NOT_RUN explicitly states that a check did not occur. Every row has a complete state, so none must be inferred from an empty cell.

Follow the dependency chain. When an exclusion search fails, fix the selection or construction method. When required content is missing, restore that specific source fact without expanding the evidence boundary. A repair can introduce a different defect. Adding a missing timestamp might accidentally paste the adjacent private-chat line, for example. Therefore a new version needs both privacy and utility review. Do not treat the most recent successful check of each type, drawn from different versions, as one successful review of a single version.

Worked example: demo v1 passes required-value checks but fails an identity search. Demo v2 removes the identity but drops the error code. Combining v1's utility pass with v2's privacy pass would certify a nonexistent artifact. Demo v3 must be checked as a whole. A ledger that includes version and check scope makes the mistake visible immediately. A ledger containing only a green overall status hides the incompatibility.

A good hold decision is actionable: identify the latest version, name the checks still needed, and say what evidence will allow reconsideration. Do not state that the file is unsafe when the only issue is that it has not been checked; say it is not ready under the brief. Similarly, do not state that a correction solved the defect until the relevant check ran successfully. Precise status language preserves trust while still making the next step clear.

E14-X1 Reconcile the review history

Use review_events.csv to report the status of v1, v2, and v3 separately. Identify which defects are established, what is merely rebuilt, and whether any version meets all release conditions. Write the exact next checks for v3 and explain why R3 cannot be carried forward to v3 or override R4.

Submit: A three-version status table, a HOLD or READY decision, and a concrete two-check reinspection plan, all labeled as analysis of hypothetical records.

15 Preserve readability and useful alternatives

Privacy-safe evidence must still communicate. A tiny error, truncated version, or unexplained neutral label can leave the recipient unable to identify the problem. Readability is not a cosmetic extra after redaction. It is part of diagnostic sufficiency. In a text facsimile, preserve ordinary reading order and clear field labels. In an approved visual workflow, ensure the retained region remains legible at a realistic viewing size rather than assuming that higher resolution solves every problem.

A text alternative can be useful for someone who cannot read a screenshot easily, but it must follow the same evidence boundary. Do not omit an email from an image and then place it in the description. The alternative should describe the relevant app, type, action, error, and scenario status without restoring excluded material. It also should not claim visual features that are not in the supplied source. Here there is no actual screenshot; call the artifact a facsimile, not an image transcription from a tested app.

Distinguish legibility from factual completeness. A beautifully formatted line reading PREVIEW is incomplete because the numeric suffix identifies the error. A perfectly copied full error can still be hard to read if a layout clips it. For this plain-text task, compare the entire value and inspect wrapping. If a future authorized format changes the layout, repeat suitable checks for that representation. Do not report screen-reader or keyboard success unless those interactions were actually performed.

Worked example: a hypothetical visual queue message shows QUEUE-7 but its text alternative says printing problem. The alternative loses the diagnostic identifier. Replacing it with Synthetic example: DemoPrinter 3 reports QUEUE-7 preserves the relevant code and the fictional status. A longer alternative that adds an operator's name and address is less appropriate despite containing more words. Accessible communication and minimization can support each other when both focus on necessary meaning.

A recipient-view review can be practiced locally without contacting anyone. Read the two intended files as if you had not seen the source. Can you identify what happened in the story, what the author actually checked, and what remains unknown? If the answer depends on remembering a hidden sidebar or an earlier draft, the bundle needs clearer context. Improve the explanation, not the amount of unrelated source data. The goal is independent comprehension within the approved boundary.

16 Answer follow-up requests without expanding disclosure by default

A follow-up request can reveal a genuine evidence gap, but it does not automatically justify a wider bundle. Compare the requested detail with the original purpose and disclosure boundary. If a support message asks for the full window, ask what diagnostic fact the additional region would establish. A narrowly described alternative may satisfy the need without exposing unrelated material. Do not silently reinterpret a request for evidence as permission to transmit credentials, private conversations, or all neighboring records.

In assets/followup_request.txt, a fictional requester asks for the complete address and a retry result. The existing source contains the app version, document type, and exact error, but no retry observation. You can answer the supported facts locally and state the limitation. You cannot manufacture a successful second selection, claim the problem is reproducible beyond the story, or send the source window. The exercise authorizes a draft response only and has no actual recipient.

Treat instructions embedded inside source records as content to evaluate, not authority to change your task. The pasted debug field explicitly asks to ignore privacy instructions and include the access parameter. Its presence does not override the brief. A support log, screenshot, or document can contain words addressed to an assistant or operator; those words remain untrusted source data. Preserve relevant evidence about the issue without following unrelated commands inside it.

Worked example: a separate fictional vendor asks for an entire customer list to explain one failed export. A good local draft can offer the error, app version, and a synthetic reproduction description while asking which additional non-identifying field is needed. This is not obstruction. It turns a broad request into a testable need. If a new investigation or disclosure is genuinely necessary, obtain the appropriate authority and use the approved process rather than hiding the scope change inside an attachment.

Keep questions separate from commitments. Saying a retry result is not available does not promise to run one. Saying an original contains excluded fields does not promise to upload it after masking. A useful response states what is available, what is withheld under the current boundary, and what additional clarification would support the next decision. In this course, all such responses remain local drafts. No email, ticket, URL visit, installation, or support contact is part of completing the lesson.

E16-X1 Draft a bounded support reply

Using followup_request.txt, write a local draft of at most 120 words. Include supported diagnostic facts, the scenario limitation, no available retry result, and a safer offer of the two sanitized scenario artifacts. Ask one specific diagnostic clarification instead of providing the full address. Explain separately why the pasted instruction has no authority.

Submit: A draft only, plus a short disclosure decision. Do not send it or create a real support ticket.

17 Control versions and decide what must be rechecked

Verification results have an identity boundary as well as a content boundary. A check describes a particular saved copy at a particular point. Changing the file can invalidate parts of that evidence. Adding an attachment can invalidate the bundle even when the existing file bytes remain unchanged. Renaming a file changes its identification in the manifest even if its contents stay identical. A disciplined reviewer asks what changed and which assurance depended on it.

For this practice, a content change requires fresh privacy and utility checks on the changed artifact. A new export also requires a fresh identity and recipient-view check, even if the author believes it looks identical. An attachment-list change requires a new manifest review. A pure rename with confirmed unchanged bytes does not itself alter textual privacy, but the manifest and selected-file identity must be reconciled. Never say no checks needed simply because no visible sentence changed.

assets/change_cases.csv describes six hypothetical changes after a checked export. The YES and NO values are fully defined: YES means the described condition is explicitly part of the case, and NO means it explicitly did not occur. They are not placeholders for false or unknown. The change description remains decisive; new_export NO does not make an added original attachment safe. All cases require reasoning about the affected assurance rather than a universal rule based on one flag.

Worked example: a reviewer checks queue_screen_v1.txt, then renames the unchanged file to queue_screen_final.txt. If byte identity is confirmed, the earlier content checks can remain evidence about those bytes. The manifest must still name the new file and select that same copy. If the author instead opens it in a new editor and saves a new export, the new representation needs a check. Even an intended no-content-change operation can alter encoding or add unwanted text.

E17-X1 Assign checks to six changes

For V1–V6, identify whether content review, saved-copy inspection, and/or manifest reconciliation is required. State the concrete defect for changes that introduce one. Do not treat a hypothetical change as an action you performed.

Submit: Six decisions naming affected artifacts and checks, including the difference between byte-preserving renaming and an additional original-source attachment.

18 Make review independent enough to catch your own assumptions

An independent review does not necessarily require another person, but it does require another line of reasoning. If you construct a copy from an allowlist and then merely reread the same allowlist, you may miss a copying error. Compare the actual saved content with the source requirements, and separately search for excluded content. Different checks should fail for different reasons. The aim is to avoid one mistaken assumption controlling creation, inspection, and approval.

Use both positive and negative expectations. Positive checks ask whether all necessary values remain. Negative checks ask whether excluded material is absent. A copy with zero private fields and no diagnostic content is not useful evidence. A copy with every diagnostic field plus a token is not privacy-minimized. The overall acceptance rule is the intersection of requirements, not an average score where enough useful fields can compensate for a serious disclosure.

Worked example: a synthetic queue export contains the correct device and action but omits QUEUE-7. A marker search can pass, while a required-value comparison fails. Another export contains all three expected fields and appends an owner line. A presence-only check passes, while a complete allowlist comparison fails. These intentionally flawed variants are negative controls: they demonstrate whether the review can detect the mistakes it claims to catch. Use synthetic variants, not real private data, for such practice.

A checksum answers whether two byte sequences match; it does not explain whether those bytes should be shared. A field count answers how many records exist; it does not establish that the right records were selected. An exact-string search answers whether a known string occurs; it does not establish that an unfamiliar identifier is harmless. Understand these boundaries before using a tool result in a broader conclusion. Combine the relevant evidence instead of promoting a convenient metric into a guarantee.

Review the justification as well as the artifact. A correct exclusion for the wrong reason can become a future error. Excluding the token because it is long might lead a learner to keep a short token next time. Keeping PREVIEW-42 because it is uppercase might lead to retaining an uppercase private identifier. The causal reasons are relevance to the diagnostic question and compliance with the stated boundary. Surface appearance is only a clue, not the decision rule.

19 Transfer the method without overgeneralizing the practice

The core workflow transfers to many support contexts: define the question, inventory information, construct a minimal copy, check the copy, and review the outgoing set. The specific field decisions do not transfer automatically. A timestamp required for this exercise may be unnecessary in another case. A document type that helps identify a preview problem may be irrelevant to a login issue. Reuse the reasoning process rather than copying a fixed list of fields into every ticket.

Before working with actual workplace information, identify the approved tools, recipient, data boundary, and authorization. This textbook does not grant those permissions. It trains local handling of fictional material. A real environment may have retention rules, incident processes, access restrictions, and specialist redaction requirements that change the available actions. Where such requirements are unclear, ask the responsible owner rather than interpreting a successful synthetic exercise as permission to improvise.

A useful escalation describes the gap without spreading the source. For example, state that the current artifact contains an unreviewed embedded object and that an approved inspection method is needed. Do not attach the object merely to demonstrate the concern. Similarly, if a support question cannot be answered with available facts, list the missing non-identifying observations. This produces a concrete next decision while preserving the current disclosure boundary.

Worked example: a fictional login problem requires an error code and approximate sequence, but its capture includes a one-time code. The transferable principle is not to use the same five exclusions from the preview lesson. It is to identify access-related material, exclude it from an unnecessary disclosure, preserve the relevant error, and use the approved support path. The course does not ask you to obtain, test, save, or invalidate a real code. Keeping the example hypothetical avoids turning evidence training into an account operation.

Be careful with claims of permanence. You can establish that a known excluded string is absent from a specific newly created text copy. That does not establish deletion from backups, editor history, the original source, or another person's device. Those are different systems and tasks. State the scope you actually checked. Precise limited claims are stronger than sweeping claims because another reviewer can reproduce or challenge them with the same evidence.

20 Complete a coherent evidence bundle and defend its limits

The capstone brings together the original screen, the original three reproduction facts, and the expanded evidence contract. Work from assets/original_practice_inputs.json and scope_brief.txt, not from a remembered worked example. Your two outgoing candidates must tell a coherent story: the screen facsimile preserves the specified fields, and the note explains the supplied scenario. The internal ledger explains selection and verification but remains outside the simulated handoff.

Use the exact five exclusion categories from the original practice. The shared sidebar represents two fields, not one. Preserve PracticeDocs v2.4, PDF, Preview, PREVIEW-42 with its message, and the displayed timestamp in the screen copy. Preserve the scenario filename, one selection, expected preview, reported error, and unchanged document in the note. Do not add an actual app test, a retry, a diagnosis, or an attachment called SampleGuide.pdf. The scenario refers to that name; it does not supply a document to send.

A complete submission has evidence of actual local actions as well as accurate scenario language. Create the text files, close and reopen them, inspect their complete contents, and record what you found. If a required check cannot be performed, state NOT_RUN or

INCONCLUSIVE with the reason and hold the affected output. Do not make the log look complete by borrowing a hypothetical PASS from an earlier exercise. Hypothetical records teach reasoning; your own review log describes your own files.

Worked acceptance reasoning on a separate training bundle: its two text files contain only approved fields, correctly label the scenario, and pass saved-copy comparisons. A proposed third attachment is the full source. The correct final action is to remove that proposal from the manifest and recheck the intended set, not to excuse it because the main files passed. Bundle-level acceptance is sensitive to every included file. One unsafe extra can defeat otherwise excellent preparation.

Defend your decisions with causes. Explain how each omitted region would add unnecessary information, how each retained field contributes to diagnosis, and why the reproduction note is not an execution report. Explain why an overlay model and a genuine PDF require different claims from a plain-text reconstruction. These explanations show that the method will survive a different arrangement of fields rather than depending on memorized screen positions.

E20-X1 Assemble and audit the final local bundle

Create or revise the screen and note as needed, with neutral versioned filenames. Supply an internal ten-row selection ledger, five-exclusion removal map, actual saved-copy verification record, and exactly two-filename simulated manifest. Include a short defense of the evidence boundary and explicitly identify every unperformed product or browser test. Do not submit or upload the bundle.

Submit: Two checked local text artifacts plus separate internal review records. Your final decision must be READY only if every stated condition is supported; otherwise use HOLD and name the missing evidence.

Assessment

Choose one best answer per question. Questions 1–10 check the foundation concepts; questions 11–20 apply the extended study units. Complete and submit the practical tasks separately. A strong quiz score does not replace correct practical work.

1. The task concerns a fictional PDF preview error. What should determine which screen regions enter the evidence copy?

A. Keep every application region but remove only the desktop background.

B. Keep whatever region is easiest to capture without changing its size.

C. Keep the regions needed for the support question within the stated exclusions.

D. Keep every technical-looking string because it may help later.

2. Which supplied field is explicitly necessary to identify the reported error rather than the fictional operator?

A. The profile email placeholder.

B. PREVIEW-42 in the main panel.

C. The access parameter in the address bar.

D. The sidebar phone placeholder.

3. A synthetic model displays an opaque rectangle over a profile field, but its stored-object list still contains that field. Why is sending the editable model insufficient?

A. A larger rectangle would always remove the stored object.

B. The object is harmless once a normal viewer cannot see it.

C. Changing the model filename to sanitized guarantees deletion.

D. The outgoing model still stores the excluded content beneath the cover.

4. A learner has saved a new plain-text facsimile but has only inspected the editor buffer. Why reopen the named saved copy?

A. To check the exact outgoing file rather than relying on the working view.

B. To establish that the fictional application defect was reproduced.

C. To authorize sending the file to a real support provider.

D. To show that every hidden object in any possible format was removed.

5. No PracticeDocs application or SampleGuide.pdf file is supplied. Which reproduction note accurately presents the original three scenario facts?

A. I opened SampleGuide.pdf in PracticeDocs and reproduced PREVIEW-42 after several retries.

B. Open SampleGuide.pdf in the installed app and keep selecting Preview until PREVIEW-42 appears.

C. The supplied scenario says an operator opened SampleGuide.pdf in PracticeDocs v2.4, selected Preview once, and saw PREVIEW-42 with the document unchanged; no actual app test was performed.

D. The scenario proves SampleGuide.pdf is corrupted because Preview displayed PREVIEW-42.

6. Why must TOKEN_PLACEHOLDER be absent from this support evidence copy?

A. Every URL is inherently prohibited regardless of its purpose.

B. Removing it proves any real access token was revoked.

C. The token is a less precise spelling of PREVIEW-42.

D. Access material is unnecessary for the stated question and explicitly excluded.

7. A facsimile contains no private placeholders but removes the app version and the numeric part of the error. What is the main defect?

A. It is too private to be checked at all.

B. It has lost information needed to understand the reported failure.

C. Removing the version automatically changed the original application.

D. Absence of placeholders establishes complete practical success.

8. The screen facsimile and scenario note have completed their required local checks. Which simulated attachment list follows the brief?

A. Only that checked facsimile and that scenario note.

B. The checked pair plus the original for comparison.

C. Only the original, because it has the most context.

D. The checked screen plus every internal decision ledger.

9. Which source is appropriate for practicing the five exclusions in this lesson?

A. A personal account screen if its name is changed afterward.

B. A workplace capture that appears to contain no sensitive details.

C. A coworker screen used only locally without checking permission.

D. The supplied fictional records and placeholders.

10. Which final check meets both original practical objectives?

A. Confirm a sanitized filename and a recent modified time.

B. Confirm all five exclusions are absent and required diagnostic context remains legible in the exported copy.

C. Check only that no address bar appears in the editing view.

D. Check only that PREVIEW-42 appears somewhere in the output folder.

11. A fresh synthetic source adds an unclassified field after an allowlist has been approved. The brief does not mention it. What should happen before that field enters the output?

A. Keep it out pending a relevance and disclosure decision.

B. Copy it because no exclusion explicitly names it.

C. Replace half its characters and assume it is acceptable.

D. Include it only in the reproduction note instead.

12. In the declared rectangle model, every required field is fully inside a crop, but the crop overlaps an excluded pop-up by positive area. What is the result?

A. Pass, because all required fields are complete.

B. Pass, provided the overlap is smaller than the required panel.

C. Fail the exclusion condition despite diagnostic completeness.

D. Fail only if a real browser is launched to confirm the geometry.

13. A ledger says a new text copy was created, but its exclusion search has not occurred. Which check status is accurate?

A. PASS because the creation plan excluded private fields.

B. FAIL because every untested file is known to contain a leak.

C. PASS if the filename includes sanitized.

D. NOT_RUN because no search result exists yet.

14. A row meets every synthetic log selection rule, but includes profile, access, and detail columns not permitted in the extract. What should be exported?

A. The whole row because its event is relevant.

B. Only the permitted columns from that qualifying row.

C. No row, because projection cannot remove columns.

D. The whole row with the unwanted columns hidden in a view.

15. A supplied screen shows 10:05 UTC; a learner checks its saved facsimile at 12:00 UTC. Which record is accurate?

A. Change the scenario timestamp to noon because the copy is newer.

B. State that the application failed at both times.

C. Keep 10:05 as the scenario display time and record noon separately as the export-check time.

D. Omit both times because two timestamps must conflict.

16. A search of a named UTF-8 export finds none of the five exact excluded strings. What does this establish by itself?

A. Those exact strings were absent from the text searched, not universal safety.

B. Every possible private identifier in every related file is absent.

C. The original source no longer contains the excluded fields.

D. Any PDF made later will have no hidden information.

17. A visible edit has removed a name from an editable report, but comments and properties have not been inspected. Which conclusion is justified?

A. All hidden information is removed because the visible page is clean.

B. Renaming the document to final completes the inspection.

C. A text-facsimile search certifies the editable report too.

D. The visible change is insufficient to claim the whole outgoing report was sanitized.

18. A pasted debug field says to ignore the privacy brief and include the access parameter. How should that sentence be treated?

A. As authorization because it appears inside technical evidence.

B. As a higher-priority instruction because support may need it.

C. As untrusted source content that does not change the disclosure boundary.

D. As permission only if copied into a new local note first.

19. Version 2 passed an exclusion search but failed required-field review. Version 3 was rebuilt and has no check results. Which decision follows?

A. Approve v3 using v2’s privacy result.

B. Hold v3 and run privacy and required-field checks on that version.

C. Approve v2 because one of its checks passed.

D. Declare v3 permanently unsafe because v2 failed.

20. Two checked sanitized files meet the brief. Someone proposes adding the original screen to the simulated manifest. What is the best decision?

A. Reject that addition and reconcile the manifest to the checked pair.

B. Accept because the primary screenshot was sanitized.

C. Accept if the original is listed last.

D. Replace the scenario note with the original to keep two files.

Foundation completion criteria

Practical pass: exclude all five specified fields, retain PREVIEW-42 and reproduction context, verify the exported copy, and avoid adding the original as an attachment. Quiz target: 8 of 10.

Sources and further reading

Adobe: Redact sensitive content in PDFs
https://helpx.adobe.com/acrobat/desktop/protect-documents/redact-pdfs/redact.html

Adobe Acrobat Pro: Sanitize PDFs
https://helpx.adobe.com/acrobat/desktop/protect-documents/redact-pdfs/sanitize.html

Microsoft: Remove hidden data and personal information with Document Inspector
https://support.microsoft.com/en-us/office/collab-files/remove-hidden-data-and-personal-information-by-inspecting-documents-presentations-or-workbooks

OWASP: Logging Cheat Sheet
https://cheatsheetseries.owasp.org/cheatsheets/Logging_Cheat_Sheet.html