

Antora Tracer

Requirements

Version 0.7.0, 2026-07-29

This document lists the traceability requirements for the Antora Tracer extension. Each requirement is extracted from the OpenSpec specifications under [openspec/changes/unified-item-architecture/specs/](#).

1. Unified Item Macro

REQ-001 — Single `item` macro replaces all block macros

The system SHALL use a single `[item]` block macro to define all traceable artifacts, replacing the previous role-specific macros.

Related specs: [unified-item-macro/spec.md](#)

Source: OpenSpec specification — Scenario: Define a requirement using `[#REQ-001, item, role=requirement]`

▼ Addressed-by

- [ARC-001 — Project scope and architectural goals](#)
- [ARC-002 — Component-level architecture](#)
- [ARC-003 — Processing flow during Antora build](#)
- [ARC-004 — Key architectural decisions and their rationale](#)

▼ Verified-by

- [TST-001 — Core Extension Tests](#)

REQ-002 — `item` macro accepts role attribute

The `[item]` macro SHALL accept a `role` attribute that specifies the type of traceable artifact. Items without a `role` attribute receive the default role `unknown` with a warning.

Related specs: [unified-item-macro/spec.md](#)

▼ Addressed-by

- [ARC-002 — Component-level architecture](#)
- [ARC-003 — Processing flow during Antora build](#)

▼ Verified-by

- [TST-001 — Core Extension Tests](#)

REQ-003 — `item` macro maintains existing attributes

The `[item]` macro SHALL support the attributes `id`, `title`, `status`, and any custom attributes. The `title` defaults to the item's `id` if omitted. The `status` field is free-form.

Related specs: [unified-item-macro/spec.md](#)

▼ Addressed-by

- [ARC-002 — Component-level architecture](#)

▼ Verified-by

- [TST-001 — Core Extension Tests](#)

REQ-004 — `item` macro supports block content

The `[item]` macro SHALL accept block content between `====` delimiters as the artifact description. Multi-line content and AsciiDoc formatting within the block SHALL be preserved.

Related specs: [unified-item-macro/spec.md](#)

▼ *Addressed-by*

- [ARC-002 — Component-level architecture](#)
- [ARC-003 — Processing flow during Antora build](#)

▼ *Verified-by*

- [TST-001 — Core Extension Tests](#)

REQ-005 — Old macro syntax generates deprecation error

The system SHALL detect and report the old macro syntax (`[req]`, `[imp]`, `[test]`, `[doc]`) as a deprecation error, suggesting the equivalent `[item, role=...]` syntax.

Related specs: [unified-item-macro/spec.md](#)

2. Configuration System

REQ-006 — Configuration defines roles

The system SHALL read roles from a YAML configuration file. Items with roles not in this list receive warnings. An empty roles list generates an error.

Related specs: [configuration-system/spec.md](#)

▼ Addressed-by

- [ARC-001 — Project scope and architectural goals](#)
- [ARC-002 — Component-level architecture](#)
- [ARC-004 — Key architectural decisions and their rationale](#)

▼ Verified-by

- [TST-004 — ConfigLoader Tests](#)

REQ-007 — Configuration defines relations

The system SHALL read relation definitions from configuration as a nested structure: [source_role](#) → [target_role](#) → [[allowed_types](#)]. Each entry defines which relation types are valid from one role to another.

Related specs: [configuration-system/spec.md](#)

▼ Addressed-by

- [ARC-002 — Component-level architecture](#)
- [ARC-004 — Key architectural decisions and their rationale](#)

▼ Verified-by

- [TST-004 — ConfigLoader Tests](#)

REQ-008 — Configuration defines matrices

The system SHALL read matrix definitions from configuration. Each matrix specifies a [name](#), [rows](#) role, [columns](#) roles, and [coverageRelations](#) mapping column roles to relation types that count as coverage.

Related specs: [configuration-system/spec.md](#)

▼ Addressed-by

- [ARC-002 — Component-level architecture](#)
- [ARC-004 — Key architectural decisions and their rationale](#)

▼ Verified-by

- [TST-004 — ConfigLoader Tests](#)
- [TST-006 — MatrixGenerator Tests](#)

REQ-009 — Configuration is required for validation

The system SHALL require a valid configuration to perform role-based relation validation. Without a ConfigLoader, all relations are allowed (unguarded mode).

Related specs: [configuration-system/spec.md](#)

▼ Addressed-by

- [ARC-001 — Project scope and architectural goals](#)

▼ Verified-by

- [TST-003 — CLI Tests](#)

REQ-010 — Configuration file location

The system SHALL search for `traceability.yml` or `traceability.yaml` in common locations. A custom path SHALL be specifiable via `--config` CLI flag or `configPath` playbook option.

Related specs: [configuration-system/spec.md](#)

▼ Addressed-by

- [ARC-004 — Key architectural decisions and their rationale](#)

▼ Verified-by

- [TST-003 — CLI Tests](#)
- [TST-004 — ConfigLoader Tests](#)
- [TST-008 — Relationship Link Substitution Tests](#)

REQ-011 — Configuration validation

The system SHALL validate the configuration structure: roles must be non-empty strings, relation source/target roles must be in the roles list, matrix row/column roles must be in the roles list, and relation types must be arrays. Invalid configurations produce clear error messages.

Related specs: [configuration-system/spec.md](#)

▼ Addressed-by

- [ARC-002 — Component-level architecture](#)

▼ Verified-by

- [TST-004 — ConfigLoader Tests](#)

REQ-012 — Configuration can extend presets

The system SHALL support the `extends` field in configuration to merge with a built-in preset. User configuration overrides preset values. Referencing a non-existent preset generates an error listing available presets.

Related specs: [configuration-system/spec.md](#)

3. Role-Based Validation

REQ-013 — Role-based relation validation

The system SHALL validate that relations between items are allowed based on their roles as defined in configuration. Valid relations are accepted; invalid relations generate errors listing the allowed types.

Related specs: [role-based-validation/spec.md](#)

▼ Addressed-by

- [ARC-001 — Project scope and architectural goals](#)
- [ARC-003 — Processing flow during Antora build](#)
- [ARC-004 — Key architectural decisions and their rationale](#)

▼ Verified-by

- [TST-005 — Graph and API Tests](#)

REQ-014 — Directional relation checking

Validation SHALL be directional: the system checks the source item's role against the target item's role. The same relation type may be valid in one direction but not the reverse.

Related specs: [role-based-validation/spec.md](#)

▼ Addressed-by

- [ARC-003 — Processing flow during Antora build](#)

▼ Verified-by

- [TST-005 — Graph and API Tests](#)

REQ-015 — Error messages list allowed relations

Error messages for invalid relations SHALL include the relation type, source role, target role, and the list of allowed relation types between those roles.

Related specs: [role-based-validation/spec.md](#)

▼ Addressed-by

- [ARC-004 — Key architectural decisions and their rationale](#)

▼ Verified-by

- [TST-005 — Graph and API Tests](#)

REQ-016 — Warnings for unknown roles

The system SHALL generate warnings (not errors) when relations involve items with unknown roles. This enables graceful degradation and incremental role adoption.

Related specs: [role-based-validation/spec.md](#)

▼ *Addressed-by*

- [ARC-003 — Processing flow during Antora build](#)

▼ *Verified-by*

- [TST-005 — Graph and API Tests](#)

REQ-017 — Graceful degradation for unknown roles

Items with unknown roles SHALL still be stored in the graph and appear in matrices. Relations involving unknown roles SHALL be stored but marked with warnings. Processing continues despite unknown roles.

Related specs: [role-based-validation/spec.md](#)

4. Preset System

REQ-018 — Built-in preset configurations

The system SHALL ship with built-in preset configurations: `requirements-engineering`, `agile`, `medical-iec62304`, and `minimal`. Users SHALL be able to list and inspect presets via CLI.

Related specs: `preset-system/spec.md`

▼ *Addressed-by*

- [ARC-001 — Project scope and architectural goals](#)
- [ARC-004 — Key architectural decisions and their rationale](#)

▼ *Verified-by*

- [TST-004 — ConfigLoader Tests](#)

REQ-019 — Each preset contains complete configuration

Each preset SHALL contain `roles`, `relations`, and `matrices` definitions that form a valid, loadable configuration. Presets with invalid configuration are rejected at load time.

Related specs: `preset-system/spec.md`

▼ *Verified-by*

- [TST-004 — ConfigLoader Tests](#)

REQ-020 — Each preset includes documentation

Each preset SHALL include a human-readable description, domain explanation, and example AsciiDoc snippets demonstrating typical usage.

Related specs: `preset-system/spec.md`

▼ *Verified-by*

- [TST-004 — ConfigLoader Tests](#)

REQ-021 — Each preset includes Neo4j Cypher queries

Each preset SHALL include a `neo4j.queries` section with named Cypher queries relevant to the domain, such as uncovered requirements, coverage summaries, and traceability chains.

Related specs: `preset-system/spec.md`

▼ *Verified-by*

- [TST-004 — ConfigLoader Tests](#)

REQ-022 — Presets have independent versioning

Presets SHALL have version numbers independent of the extension version, following semantic versioning. A `minExtensionVersion` compatibility field SHALL indicate the minimum extension version required.

Related specs: `preset-system/spec.md`

▼ *Verified-by*

- [TST-004 — ConfigLoader Tests](#)

REQ-023 — Users can define custom presets

The system SHALL allow users to create their own preset YAML files. Custom presets are loaded and validated the same way as built-in presets. Invalid custom presets generate validation errors.

Related specs: [preset-system/spec.md](#)

5. Neo4j Export

REQ-024 — Neo4j CSV export

The system SHALL export the traceability graph as `nodes.csv` and `relationships.csv` files with proper headers, comma delimiters, and escaped special characters. Files SHALL be importable via Neo4j's `LOAD CSV` command.

Related specs: [neo4j-export/spec.md](#)

▼ Addressed-by

- [ARC-001 — Project scope and architectural goals](#)
- [ARC-002 — Component-level architecture](#)
- [ARC-004 — Key architectural decisions and their rationale](#)

▼ Verified-by

- [TST-007 — Neo4jExporter Tests](#)

REQ-025 — Neo4j Cypher export

The system SHALL export the traceability graph as `import.cypher` containing CREATE statements for all nodes (with `:Item` and role-specific labels) and relationships (with type-specific labels). Special characters SHALL be properly escaped.

Related specs: [neo4j-export/spec.md](#)

▼ Addressed-by

- [ARC-002 — Component-level architecture](#)

▼ Verified-by

- [TST-007 — Neo4jExporter Tests](#)

REQ-026 — Export command in CLI

The system SHALL provide an `export neo4j` CLI command supporting `--format csv` and `--format cypher` options, with configurable input and output directories.

Related specs: [neo4j-export/spec.md](#)

▼ Verified-by

- [TST-003 — CLI Tests](#)

REQ-027 — Export includes all item attributes

Neo4j exports SHALL include all item attributes (id, role, title, content, status, custom attributes) as node properties. Special characters in attribute values SHALL be properly escaped.

Related specs: [neo4j-export/spec.md](#)

▼ Verified-by

- [TST-007 — Neo4jExporter Tests](#)

REQ-028 — Export is optional

The Neo4j export functionality SHALL be completely optional. Normal extension operation (parsing, validation, matrix generation) SHALL not require any Neo4j runtime dependencies.

Related specs: [neo4j-export/spec.md](#)

6. Broader Capabilities

REQ-029 — Configurable matrix generation with coverage calculations

The system SHALL generate matrices from configuration definitions, computing per-item coverage based on `coverageRelations`. Each cell SHALL show related item IDs or `-` for no relationships. Coverage SHALL be expressed as overall, complete, partial, and missing counts.

Source: Design Decision 2 — Configuration-Based Roles and Relations

▼ *Addressed-by*

- [ARC-002 — Component-level architecture](#)
- [ARC-003 — Processing flow during Antora build](#)
- [ARC-004 — Key architectural decisions and their rationale](#)

▼ *Verified-by*

- [TST-001 — Core Extension Tests](#)
- [TST-006 — MatrixGenerator Tests](#)

REQ-030 — Role statistics reporting

The system SHALL provide `getRoleStatistics()` returning item counts per role and total relationship counts, enabling quick inspection of the traceability graph.

Source: RequirementsTraceabilityExtension API

▼ *Verified-by*

- [TST-001 — Core Extension Tests](#)

REQ-031 — Graph path finding

The system SHALL provide BFS-based path finding between any two items in the graph via `findPath(fromId, toId, maxDepth?)`. Connected items return the path; disconnected items return null. Depth is capped by `maxDepth` (default: 5).

Source: TraceabilityGraph API

▼ *Verified-by*

- [TST-005 — Graph and API Tests](#)

REQ-032 — Impact analysis

The system SHALL provide `getImpactAnalysis(itemId)` returning all reachable items in both directions (forward and reverse relationships), excluding the source item from results.

Source: TraceabilityGraph API

▼ *Verified-by*

- [TST-005 — Graph and API Tests](#)

REQ-033 — Graph merge

The system SHALL support merging another TraceabilityGraph into the current graph via `merge()`. Items not already present are added; existing items are not duplicated. Relationships from the source graph are copied if both endpoints exist in the target graph.

Source: TraceabilityGraph API

▼ Verified-by

- [TST-005 — Graph and API Tests](#)

REQ-034 — Mustache template rendering

The system SHALL use Mustache templates for HTML matrix output via `TemplateRenderer`. Built-in templates SHALL be shipped; custom template directories SHALL be supported with fallback to built-ins for missing files.

Source: Design Decision 4 — Mustache Templates

▼ Addressed-by

- [ARC-002 — Component-level architecture](#)
- [ARC-004 — Key architectural decisions and their rationale](#)

▼ Verified-by

- [TST-001 — Core Extension Tests](#)

REQ-035 — Antora extension integration

The system SHALL integrate with Antora via `AntoraTraceabilityExtension`, registering for `contentClassified` (parse items) and `sitePublished` (generate matrices and coverage). Configuration SHALL be read from the Antora playbook's extension config. Disabled extensions SHALL skip initialization.

Source: Antora Extension implementation

▼ Addressed-by

- [ARC-001 — Project scope and architectural goals](#)
- [ARC-003 — Processing flow during Antora build](#)

▼ Verified-by

- [TST-002 — Antora Extension Tests](#)

REQ-036 — CLI commands

The system SHALL provide a Commander-based CLI with six subcommands: **process** (parse items), **matrix** (generate matrices), **validate** (check graph integrity), **export neo4j** (CSV/Cypher export), **stats** (role statistics), and **preset** (list/show/init). Global **--config** and **--preset** flags SHALL apply to all commands.

Source: CLI implementation

▼ Addressed-by

- [ARC-001 — Project scope and architectural goals](#)

▼ Verified-by

- [TST-003 — CLI Tests](#)

7. Matrix Item Linking

REQ-037 — Matrix row items are clickable links

The system SHALL render row item IDs as hyperlinks that navigate to the item's definition in the rendered HTML documentation.

Source: openspec/specs/matrix-item-linking/spec.md

▼ *Verified-by*

- [TST-006 — MatrixGenerator Tests](#)

REQ-038 — Matrix cell items are clickable links

The system SHALL render cell item IDs as hyperlinks that navigate to the item's definition in the rendered HTML documentation. Multiple items in a cell SHALL each render as separate clickable links.

Source: openspec/specs/matrix-item-linking/spec.md

▼ *Verified-by*

- [TST-006 — MatrixGenerator Tests](#)

REQ-039 — Source file tooltips on matrix links

The system SHALL display the source file name as a tooltip when users hover over item links in the matrix.

Source: openspec/specs/matrix-item-linking/spec.md

▼ *Verified-by*

- [TST-006 — MatrixGenerator Tests](#)

REQ-040 — Context-aware link generation

The system SHALL generate consistent relative link paths (`../../\``) from matrix output to component root, regardless of execution context. The LinkResolver SHALL accept a configurable ``relativePathPrefix`\`` option.

Source: openspec/specs/matrix-item-linking/spec.md

▼ *Verified-by*

- [TST-002 — Antora Extension Tests](#)

REQ-041 — Source file path normalization

The system SHALL normalize item ``sourceFile`\`` paths by stripping any ``pages/\`` prefix and ``.adoc\`` extension at link resolution time, producing clean HTML paths.

Source: openspec/specs/matrix-item-linking/spec.md

REQ-042 — Backward compatibility without LinkResolver

The system SHALL maintain existing matrix rendering behavior when no ``LinkResolver`\`` is

provided. Items SHALL render as plain text without links, and existing matrix HTML structure SHALL be preserved.

Source: openspec/specs/matrix-item-linking/spec.md

▼ *Verified-by*

- [TST-006 — MatrixGenerator Tests](#)

REQ-043 — Non-HTML matrix outputs unchanged

The system SHALL NOT modify CSV and JSON matrix outputs to include links. CSV and JSON output SHALL contain item data as before.

Source: openspec/specs/matrix-item-linking/spec.md

▼ *Verified-by*

- [TST-006 — MatrixGenerator Tests](#)

8. traceability:outgoing[] and traceability:incoming[] Rendering Macros

REQ-044 — traceability:outgoing[] renders outgoing links

The system SHALL provide a `` macro that expands to a formatted list of all outgoing relationships for the enclosing item. Relationships SHALL be grouped by type and rendered using standard AsciiDoc constructs (lists, tables, inline xrefs).

Source: openspec/specs/traceability-links-macro/spec.md

REQ-044b — traceability:incoming[] renders incoming links

The system SHALL provide a `` macro that expands to a formatted list of all incoming relationships pointing to the enclosing item. Relationships SHALL be grouped by inverse relation type (e.g., ``addresses`` displays as `Addressed by``) and rendered using standard AsciiDoc constructs.

Source: openspec/specs/incoming-links-macro/spec.md

REQ-045 — Opt-in via AsciiDoc attribute

The system SHALL only expand `` and `` when the `:traceability-links:` document attribute is set to a truthy value. Without the attribute, the macro text SHALL be stripped from output.

Source: openspec/specs/traceability-links-macro/spec.md

REQ-046 — Inline macros are pure data markers

Inline relationship macros (``addresses:REQ-001[]``) SHALL always be invisible in rendered output. They are pure data markers stored in the traceability graph. `` and `` are the sole mechanisms for rendering relationships.

Source: openspec/specs/traceability-links-macro/spec.md

REQ-047 — Configurable display style

The system SHALL support configurable display styles via the `:traceability-style:` document attribute: `list` (default, bulleted list), `table` (AsciiDoc table), or `inline` (comma-separated).

Source: openspec/specs/traceability-links-macro/spec.md

REQ-048 — Configurable sort order

The system SHALL support configurable sort order via `:traceability-order:`: `target-id` (default), `target-title`, or `relation-type`.

Source: openspec/specs/traceability-links-macro/spec.md

REQ-049 — PDF compatibility

Generated output from `` and `` SHALL be standard AsciiDoc constructs — not raw HTML — ensuring compatibility with both HTML and PDF backends.

Source: openspec/specs/traceability-links-macro/spec.md

REQ-050 — Source files never modified

The system SHALL NOT modify `.adoc\`` source files on disk. All macro expansion and content substitution SHALL occur in the in-memory content buffer during Antora's ``contentClassified\`` event.

Source: `openspec/specs/traceability-links-macro/spec.md`

9. Traceability Visualization

REQ-051 — Relationship graph via `traceability:graph[]` macro

The system SHALL provide a ``\`` inline macro that renders a GraphViz relationship graph for the enclosing item. The graph SHALL show the item and its direct relationships as labeled nodes and edges, colored by role. An optional depth parameter SHALL control the number of hops.

Source: `openspec/specs/graph-visualization/spec.md`

REQ-052 — Coverage chart via `macro`

The system SHALL provide a ``\`traceability:graph-coverage[]` a Vega-Lite bar chart showing coverage. When used inside an item block, it SHALL show per-relationship-type coverage for that item. When used outside an item block, it SHALL show global coverage by role.

Source: `openspec/specs/graph-visualization/spec.md`

REQ-053 — Graph visualization methods on `TraceabilityGraph`

The ``TraceabilityGraph\`` SHALL provide ``toDot(fromId, depth?)\`` returning a GraphViz DOT source string, and ``toVegaLite(itemId?)\`` returning a Vega-Lite JSON specification.

Source: `openspec/specs/graph-visualization/spec.md`

REQ-054 — Dashboard page in example site

The example site SHALL include a dashboard page with a global coverage chart and select per-item relationship graphs, gated by the ``:traceability-graph:\`` attribute.

Source: `openspec/specs/graph-visualization/spec.md`

10. Changelog

Version	Date	Author	Change
0.7.0	2026-07-28	Team	Initial PDF release