Guidedog Discussions GDS 0014

Embedded Typst diagrams

STATE
committed
TYPE
Standards Track
AUTHORS
Vikrant Rathore
CREATED
2026-10-31
UPDATED
2026-10-31

DISCUSSION
Local review; no external discussion URL assigned

LABELS
architecture, renderer, typst, build

Record status

The catalog defines the review process and the meaning of each state. This numbered record forms part of the catalog.

Contents

1. Abstract 1
2. Motivation and scope 1
3. Specification 1
4. Rationale and alternatives 3
5. Security and operational considerations 3
6. Backwards compatibility 3
7. Acceptance criteria and implementation status 3
8. Open questions 4
9. Review history 4
10. References 4

1. Abstract

We add a `typst-diagram` directive to RST and MyST projects. The embedded Typst 0.15.1 compiler draws its content as one SVG or PNG image for HTML and PDF. We keep ordinary Typst code blocks as source examples.

2. Motivation and scope

The sibling jlavici project keeps diagrams in `docs/jcd/figures/src`. Its `cargo xtask figures` compiles each Typst file to HTML, extracts one SVG, and commits the result for RST image directives. Its shared `_jcd.typ` imports Fletcher 0.5.8 and CeTZ 0.3.4 and defines reusable drawing helpers. We remove the separate compilation and extraction step for new diagrams. Existing image directives and external figure tasks continue to work. We do not modify jlavici as part of this proposal.

3. Specification

We write a diagram as follows.

```
.. typst-diagram::
   :caption: A small drawing
   :alt: A circle
   :name: circle

   #circle(radius: 12pt)
```

The corresponding MyST fence begins with ```{typst-diagram} and carries the same colon options and Typst body. An optional file argument supplies the entire source instead of the body. Exactly one source is required. File arguments use the reader's existing confined include mechanism. Inline imports resolve beside the document containing the directive, including an included RST document. File-source imports resolve beside the named Typst file. An absolute Typst import resolves from the project source directory.
Options are `format` (`svg` by default, or `png`), `alt`, `caption`, `name`, `class`, and `align` (`left`, `center`, or `right`). PNG uses a fixed 144 pixels per inch. The compiler prepends `#set page(width: auto, height: auto, margin: 0pt)`. Authors may override page settings, but image export requires exactly one finite page. We reject multiple pages rather than silently discard content. CeTZ and Fletcher are ordinary version-pinned imports, with no implicit package injection. We reuse `pdf_packages`, `pdf_fonts` and `pdf_font_paths` for diagram compilation in both builders. This keeps one package and font policy per project.
The AST mapping is deliberately small.

Table 26 Source and output mapping

Source	Semantic representation	HTML and PDF
RST or MyST `typst-diagram`	`CodeBlock` with language `typst` and custom `typst-diagram=1`	Generated image
`caption`	Parsed caption child and caption attribute	Figure caption and figure numbering
`name` and `class`	Existing identifier and class attributes	Preserved on the image or figure
`alt` and `align`	Custom directive attributes	Image alternative text and alignment
`format`	Custom format attribute	SVG or PNG asset
Ordinary `code-block:: typst` or ```typst	Unmarked code block	Source example

Text output retains source and caption without invoking Typst. RST writing reconstructs the directive and its options from the AST, retaining file arguments so relative helper imports keep their base directory. We preserve drawing source when converting markup; we do not reconstruct it from rendered images or PDF. This directive is a Guidedog extension, not a claim of compatibility with a Sphinx Typst extension.
The internal `gd-typst` adapter owns image export, generated-source locations and file-read observation. `guidedoc-sphinx` owns directive parsing, also reached by MyST's existing directive bridge. `gd-link` owns compilation policy, diagnostics, assets and image substitution. `gd-read` classifies captioned diagrams as figures. We reuse the generated-figure rewrite shared with Graphviz rather than duplicate caption and identifier handling. Public options remain plain structs with defaults; functions borrow their inputs and return structured errors.
Every rewritten diagram is compiled again. We record imported project files, data files and package files as incremental output dependencies, including failed file lookups. Output asset names include the compiled bytes, so a changed drawing receives a new URL. An existing asset with matching bytes keeps its modification time, so redrawing an unchanged diagram does not invalidate a cached PDF. We do not reuse an image merely because its source text has not changed. The existing staged output publication owns the final HTML and PDF assets. Scratch artifacts live under doctrees and are removed after use. The PDF typesetter can read the generated `graphs` and `diagrams` folders even when a mounted build keeps its cache outside the source tree; it gains no access to other cache folders.

4. Rationale and alternatives

Direct paged export avoids extracting an image from experimental Typst HTML. SVG preserves vector geometry in both builders; PNG is useful when an image consumer requires raster output. Separate `cetz` and `fletcher` directives would tie markup to package APIs and require implicit version choices. A single Typst directive also supports local helpers and other drawing libraries. Automatic execution of every Typst code block would change the meaning of documentation examples. Caching only the source hash would miss imported helper and data changes. We prefer incremental dependency tracking and fresh compilation when a page is rewritten.

5. Security and operational considerations

`--untrusted` refuses to compile diagrams and reports a warning while retaining source. Trusted diagrams use only the embedded compiler, with canonical read roots restricted to the project source directory. Symlinks leaving that directory are refused, including in trusted builds. Packages use Typst's existing package resolver; `pdf_packages = "offline"` prohibits downloads. Fonts use the existing configured font policy. Font stores are cached per process, and missing package probes do not identify a file dependency; after changing fonts or installing a previously missing package, we use a fresh build process with `-a`. An explicit source limit and a raster pixel limit bound adapter work before export; the output reservation charges generated bytes to the build ledger. Typst compilation and layout retain their existing process memory and time limitations; this feature does not claim a compiler sandbox or timeout. Compiler diagnostics identify the containing directive and the position in the diagram or imported file. Errors keep source visible and warn, matching Graphviz; `-W` prevents publication on such warnings. Compilation writes a unique scratch file, then publishes an immutable asset by content digest. No failed compile may be mistaken for an earlier successful artifact.

6. Backwards compatibility

We add a directive. Graphviz, ordinary code blocks and existing images retain their behavior. Projects using the new directive need no installed Typst program. Internal `gd-typst` compiler options gain defaulted fields, and the target enum gains SVG and PNG variants. These are workspace-internal additions during the beta (`gd-typst` sets `publish = false`). The environment schema changes so existing cached documents cannot preserve an unknown directive parse.

7. Acceptance criteria and implementation status

Required evidence covers RST and MyST parsing; SVG and PNG export; one-page rejection; captions, labels and numbering; nested imports; helper-file invalidation; offline packages; untrusted refusal; file confinement; compiler failures; HTML and real PDF builds. We run the workspace build, tests, Clippy, formatting and source limit checker before assessing beta readiness. The implementation lives in `guidedoc-sphinx/src/typst.rs`, `gd-typst/src/embedded/image.rs` and `gd-link/src/typst.rs`, with the shared generated-figure rewrite in `gd-link/src/figure.rs`. Parser, compiler, RST writing and project integration fixtures cover both source languages and both image formats. We also compiled a temporary copy of jlavici's `0002-lease.typ` with its CeTZ 0.3.4 helpers and a Fletcher 0.5.8 diagram offline into HTML and PDF. The copy replaced the extraction-only `html.frame` wrapper with its drawing content; the jlavici checkout was not changed. The manual's combined Graphviz and Typst chapter builds into the 533-page English PDF, and the release smoke test exercises SVG and PNG with an empty PATH. On 31 October 2026, the Linux x86-64 workspace build and all 1,984 tests passed (198 suites; five intentionally ignored tests). `cargo clippy --locked --workspace --all-targets -- -D warnings`, `cargo fmt --all --check`, the nine data-table checks and the source limit checker passed. The source checker retains six pre-existing soft-width warnings and reports no limit violations. The compiler also passes its tests without the embedded feature. All six localized HTML editions build with `-W`. The beta.8 preparation completes the related translations, including previously fuzzy entries. `v1.0.0-beta.8` subsequently passed Linux and macOS CI, website deployment verification, and five-platform release-archive qualification (see `docs/manual/evidence/release-1.0.0-beta.8.rst`). The maintainer requested promotion and implementation-status reconciliation on 31 October 2026; the lifecycle events below record assignment of 0014 and commitment.

8. Open questions

No design questions remain. `v1.0.0-beta.8` shipped with full five-platform archive qualification.

9. Review history

2026-10-31, initial design review: the jlavici workaround established the need for local helper imports and shared font settings. Critique: a source-hash cache would leave stale SVGs after a helper changes, a root-only generated source would resolve imports incorrectly, and first-page-only export could silently lose content. Revision: observe file reads, place generated source beside its origin, require exactly one page, and name assets by output content. Critique: enabling arbitrary Typst for untrusted builds would introduce package downloads and compiler work into a path that currently refuses active input. Revision: refuse diagram compilation under `--untrusted`, enforce canonical source roots for trusted diagrams, reuse offline policy, and bound raster dimensions before allocation. Critique: duplicating Graphviz's figure rewrite would create two implementations of labels, captions and numbering. Revision: share that rewrite and keep compiler-specific drawing in separate modules.

2026-10-31, implementation review: preserve file-source arguments in RST writing; inlining a file's text would otherwise change the base directory of its imports. Reuse the existing streaming file digester for content dependencies. Keep the draft unnumbered and retain maintainer control of acceptance and release tagging.

2026-10-31, mounted-site qualification: the website build exposed a missing PDF read-root grant for the generated diagram folder outside the source tree. Add only the generated-image folder to the existing allowlist, keep other cache folders private, and cover real mounted PDF builds with SVG and PNG.

10. References

- The Guidedog Discussion Process
- Guidedoc: the conversion engine
- Memory budgets and what a build may read
- Typst SVG export
- CeTZ
- Fletcher
- `../jlavici/tools/xtask/src/figures.rs` and `../jlavici/docs/jcd/figures/src/_jcd.typ`

Lifecycle event

```
2026-10-31: prediscussion -> discussion. Maintainer requested promotion after
implementation review and local qualification
```

Lifecycle event

```
2026-10-31: discussion -> accepted. Maintainer requested release and status
reconciliation; diagram design reviewed and implemented
```

Lifecycle event

```
2026-10-31: accepted -> committed. RST and MyST diagrams, generated assets,
dependency tracking and documentation implemented
crates/gd-typst/src/tests/images.rs;
tests/integration/tests/projects/typst_diagrams.rs; 1982 passing workspace
tests; empty-PATH release smoke; manual HTML/PDF builds
```