

Uploading Code and Running Parsing Workflows

August 25, 2026

Opening the code parsing workspace and understanding what it accepts

In Atloria, the code parsing workspace is the place where you prepare source code before turning it into technical documentation. If you already know how parsed results support writing work, use this page alongside [Using Code Parsing Results to Improve Technical Documentation](#). This guide stays focused on the workspace itself: what you can load, what the screen is waiting for, and how to tell when you are ready to run parsing.

When you open the code parsing workspace, look for four main areas:

- an upload area for adding source files
- a code snippet area for pasting a smaller block of code
- a parse action that starts the workflow
- a results area where Atloria shows the parsed output after the run finishes

[SCREENSHOT: code parsing workspace showing file upload area, snippet editor, parse action, and results panel]

Atloria accepts two main input types in this workspace:

- **Uploaded files** when you want Atloria to read code from saved source files
- **Pasted snippets** when you only need to analyze a focused section of code

Before you start, prepare code that gives Atloria enough context to recognize structure clearly. Full source files are usually best when you want broader coverage. Focused excerpts work well when you only need one part of the code, but those excerpts should still be complete enough to stand on their own.

You will usually move through a few simple workspace states before parsing begins:

- **Empty workspace:** no file or snippet has been added yet
- **Loaded input:** a file appears in the upload area, or the snippet box contains code

- **Ready to parse:** the workspace has valid input and is ready for the parse action

If the workspace is still empty, Atloria has nothing to process. Once your file or snippet is visible on the screen, you can move on to preparing it for a clean parsing run.

Preparing files and snippets for successful parsing

Good parsing starts before you click the parse action. In Atloria, the quality of the file or snippet you load has a direct effect on how useful the results will be in later documentation work. If you need help reviewing parsed coverage after a run, see [Reviewing Parsed Code Results and Reference Coverage](#).

Choose your input method based on what you want to document:

1. Use **file upload** when you want broader structure from a complete source file, such as all major code elements in one screen.
2. Use the **snippet area** when you want to isolate one specific section without bringing in unrelated code.
3. Keep each input focused on a single documentation target whenever possible.

For the best results, make sure the code you provide is complete enough to be understood as a whole. In practice, that means:

- include the full declaration rather than only a few lines from the middle
- keep the original indentation and formatting
- avoid cutting off the beginning or end of the code block
- include nearby supporting lines when they are necessary to understand the main section

Partial fragments often lead to output that is too thin or confusing to reuse. For example, a short excerpt taken from the middle of a larger file may leave Atloria without enough context to identify the structure you actually want documented.

It also helps to separate unrelated code before loading it into the workspace. If one file or snippet mixes several different topics, the results area may be harder to review and reuse. Smaller, clearly grouped inputs are easier to trace back to their source and easier to apply in documentation drafts.

Before uploading, quickly review file names and snippet content. Clear naming and clean organization make it much easier to recognize which parsed result belongs to which source, especially when you are comparing multiple runs.

Uploading files or pasting code into the workspace

Once your code is ready, add it to the workspace using the input method that matches your goal. Atloria supports both uploaded files and pasted snippets, so you can work from saved source code or from a smaller excerpt you copied for review.

1. In the code parsing workspace, use the **file upload** control to add one or more source files.
2. If you only need a focused section, click into the **snippet input** area and paste the code directly.
3. Check the workspace before parsing to confirm that the correct input is visible.
4. If you notice a mistake, replace or remove the input before starting the run.

When you upload files, Atloria should reflect that change in the workspace by showing the loaded file. When you paste a snippet, the code should remain visible in the snippet area so you can review it before continuing.

[SCREENSHOT: workspace with an uploaded file listed and a populated snippet area]

Use file upload when your documentation work depends on the full structure of a source file. Use the snippet area when you want to test or document a narrower section without extra surrounding material. In either case, pause and confirm what Atloria is showing on screen. This is the easiest point to catch a wrong file, an incomplete paste, or a code block that includes too much unrelated content.

If the wrong content was added, correct it immediately:

- remove the current file and upload the right one
- clear the snippet area and paste the full code again
- replace a partial excerpt with a more complete block
- reload the input if the visible content does not match what you intended to parse

Do not move straight to parsing just because something is loaded. The workspace should show the exact file or snippet you want Atloria to interpret. A quick review here saves time and reduces unnecessary reruns later.

Starting a parsing run and following workspace state changes

After at least one valid file or snippet is loaded, you can start the parsing workflow. Atloria uses visible workspace states to show where the run stands, so the main thing you need to do is start the run and then watch for the state to change.

1. Confirm that your file or snippet is loaded and that the workspace is ready.
2. Click the parse action to begin the run.
3. Wait while Atloria processes the current input.
4. Review the final state to see whether the run completed successfully or failed.

Most runs move through a simple sequence of states:

Workspace state	What it means
Ready	Your file or snippet is loaded and Atloria can start parsing
Parsing in progress	Atloria is currently processing the input
Parse completed	The results area is ready for review
Parse failed	Atloria could not finish the run with the current input

[SCREENSHOT: parsing workspace showing an in-progress state indicator]

While parsing is running, avoid changing the current input. Do not replace the file, clear the snippet, or paste a different code block until the current run finishes. If you realize you loaded the wrong content, wait for the current state to resolve first, then update the input and run parsing again.

The completed state is your signal that the results area is ready to use. At that point, you can review how Atloria interpreted the code and decide whether the output is suitable for documentation. If the run fails, stay in the same workspace and correct the input before trying again.

This state-based flow is especially useful when you are running several focused parsing attempts in a row. By waiting for each run to finish before making changes, you keep each result tied to a clear source input.

Reviewing parsed output for documentation use

When parsing finishes, move to the results area and compare the output with the file or snippet you loaded. Your goal is not just to confirm that Atloria produced a result, but to check whether the result matches the documentation target you had in mind.

1. Open the results area after the workspace shows **Parse completed**.
2. Compare the parsed output with the original file or snippet.
3. Check whether the output reflects the code elements you expected to document.
4. Decide whether to keep the result or rerun parsing with a better input.

[SCREENSHOT: parsed results area showing structured output beside the original input]

As you review the output, look for whether Atloria captured the intended structure from your source. Depending on what you loaded, that may include major code sections, named elements, grouped logic, or other recognizable parts of the input. The key question is simple: does this result support the documentation page you want to create or improve?

Keep the current result when:

- the parsed structure clearly matches the source you loaded
- the output is focused enough to reuse in technical documentation
- the source is easy to identify and trace back later

Rerun parsing when:

- the output is too broad because the file included unrelated material
- the output is too partial because the snippet was cut too tightly
- important surrounding context is missing
- the result appears tied to the wrong source input

If you need more guidance on turning accepted results into documentation work, continue with [Using Code Parsing Results to Support Technical Docs](#). At this stage, the most important habit is keeping each accepted result connected to a clear source file or snippet. That makes later writing, review, and revision much easier.

Fixing common parsing problems in the workspace

Most parsing problems in Atloria come from the input itself rather than from the workspace screen. When a run does not start, produces weak results, or fails completely, the fastest fix is usually to review what you loaded and clean it up before trying again.

If parsing does not start, check the workspace first:

- make sure a file is actually listed in the upload area, or that code is visible in the snippet input
- confirm the workspace is no longer empty
- verify that the screen appears ready for parsing before you click the parse action

If results are incomplete or inaccurate, the input is often too short or missing surrounding context. Replace a narrow fragment with a more complete code block. Include the full section you want Atloria to interpret, not just a few lines from the middle. Preserve indentation and avoid pasting code that starts or ends abruptly.

If the wrong source was parsed, remove the current input and reload the correct one. This is especially important when you are switching between several files or testing multiple snippets in sequence. Always confirm the visible file or pasted content before starting a new run.

If a parse run fails, review the source for common issues:

- truncated code
- malformed formatting
- pasted content missing its beginning or ending
- unrelated pieces combined into one confusing snippet

[SCREENSHOT: workspace showing failed parse state with input still visible for correction]

After you clean the input, run parsing again rather than trying to work from a failed or misleading result. If you repeatedly get weak output from the same material, split the source into smaller, more focused files or snippets and test them one at a time. That usually makes the results easier to review and more useful in documentation workflows.

Overview

This page covers the practical side of working in Atloria's code parsing workspace: loading source material, starting a parsing run, watching the workspace state change, and checking whether the output is ready to support documentation. It is the hands-on companion to the earlier code parsing articles, especially [Uploading and Parsing Code in the Workspace](#), [Managing Code Parsing Workspace Sessions](#), and [Using Code Parsing Results to Improve Technical Documentation](#).

The main ideas to carry forward are:

- the workspace accepts either uploaded files or pasted snippets
- the quality of the input strongly affects the quality of the parsed output
- workspace states help you know when Atloria is empty, ready, processing, completed, or failed

- parsed results are most useful when each run stays tied to a clear and focused source input

This workflow is especially useful when you are preparing technical documentation for a specific page, feature, or reference section. Instead of loading everything at once, you can choose the right input method for the job, confirm what is visible in the workspace, run parsing, and review the result before moving into writing or revision work.

If your next step is to apply the output inside a project's technical documentation area, continue with [Exploring Technical Documentation Inside a Project](#). If you need to work more deeply with coverage and structure, return to [Reviewing Parsed Code Results and Reference Coverage](#).

Prerequisites

Before you start a parsing run in Atloria, make sure you have the basic materials and access needed to use the workspace effectively.

- You can sign in to Atloria and reach your project workspace. If needed, use [Accessing and Registering Your Atloria Account](#) or [Signing In to Atloria and Solving Access Problems](#).
- You know which project, documentation task, or reference page the parsing result will support.
- You have source material ready in one of these forms:
 - a complete source file for broader parsing
 - a focused code snippet for a smaller, targeted run
- Your code is clean enough to review before upload or paste:
 - complete where possible
 - properly indented
 - not obviously cut off at the beginning or end
- You are prepared to rerun parsing if the first result is too broad, too partial, or tied to the wrong source.

It also helps to decide in advance whether you are documenting a broad code area or a single focused section. That choice affects whether you should use the file upload area or the snippet input box.

If you have not yet checked whether your language or framework is supported, review [Viewing Supported Languages and Parser Coverage](#) before loading code into the workspace.