Marina Compiler Internals

Danny Francisco

December 2025

Contents

Chapter 1

Marina Compiler Internals

This documentation provides a deep technical overview of the Marina (Clipper-2025) compiler and virtual machine. It is intended for contributors, advanced users, and anyone interested in the architecture and implementation details.

1.1 Contents

Each section is a standalone markdown file and can be assembled into a single document using a script.

Chapter 2

Overview

Marina is a modern compiler and virtual machine for the Clipper programming language. This document describes the high-level architecture, main components, and data flow from source code to execution.

- **Source Code** → Lexer → Parser → AST → Compiler → Bytecode → Virtual Machine (VM)

See each section for details on the respective stage.

Chapter 3

Lexical Analysis (Lexer)

The lexer converts raw source code into a stream of tokens. It recognizes keywords, identifiers, literals, operators, and punctuation. Tokens are the input for the parser.

- Input: Source code (text)
- Output: Token stream
- Key file: `src/lexer.rs`

Chapter 4

Parsing

The parser takes the token stream from the lexer and builds an Abstract Syntax Tree (AST) representing the program structure. It uses recursive descent parsing with operator precedence.

- Input: Token stream
- Output: AST
- Key files: `src/parser/`

Chapter 5

AST Structure

The Abstract Syntax Tree (AST) is a hierarchical representation of the program. Each node corresponds to a language construct (e.g., expressions, statements, functions).

- Key file: `src/ast.rs`
- Used by: Compiler for code generation

Chapter 6

Bytecode Generation

The compiler traverses the AST and emits bytecode instructions for the Marina VM. This stage handles variable scope, function calls, and control flow.

- Input: AST
- Output: Bytecode (instructions + constants)
- Key files: `src/compiler/`, `src/bytecode.rs`

Chapter 7

Virtual Machine (VM)

The Marina VM executes bytecode instructions. It manages the stack, call frames, local/global variables, and built-in functions.

- Input: Bytecode
- Output: Program execution
- Key files: `src/vm/`

Chapter 8

Error Handling

Describes how the compiler and VM report errors, including syntax errors, runtime errors, and diagnostics.

- Key files: `src/diagnostics.rs`, error handling in parser/compiler/vm

Chapter 9

Extending the Compiler

Guidelines for adding new language features, opcodes, or built-in functions to Marina.

- Where to add new syntax: `src/parser/`
- Where to add new bytecode: `src/compiler/`, `src/bytecode.rs`
- Where to add new VM behavior: `src/vm/`