# `target-name-here`

**Tier: 3**

One-sentence description of the target (e.g. CPU, OS)

## Target maintainers

- Some Person, `email@example.org`, https://github.com/...

## Requirements

Does the target support host tools, or only cross-compilation? Does the target support std, or alloc (either with a default allocator, or if the user supplies an allocator)?

Document the expectations of binaries built for the target. Do they assume specific minimum features beyond the baseline of the CPU/environment/etc? What version of the OS or environment do they expect?

Are there notable `#[target_feature(...)]` or `-C target-feature=` values that programs may wish to use?

What calling convention does `extern "C"` use on the target?

What format do binaries use by default? ELF, PE, something else?

## Building the target

If Rust doesn't build the target by default, how can users build it? Can users just add it to the `target` list in `config.toml`?

## Building Rust programs

Rust does not yet ship pre-compiled artifacts for this target. To compile for this target, you will either need to build Rust with the target enabled (see "Building the target" above), or build your own copy of `core` by using `build-std` or similar.

## Testing

Does the target support running binaries, or do binaries have varying expectations that prevent having a standard way to run them? If users can run binaries, can they do so in some common emulator, or do they need native hardware? Does the target support running the Rust testsuite?

## Cross-compilation toolchains and C code

Does the target support C code? If so, what toolchain target should users use to build compatible C code? (This may match the target triple, or it may be a toolchain for a different target triple, potentially with specific options or caveats.)