# target triples

lies, damned lies, and target triples

target = "machine code can run on"

target triple = 3 components

```
>_ zsh ×  +                    default  Mon Oct 28 10:46  souffle.local

~ via 🐍 v3.9.6
❯ rustc -vV
rustc 1.82.0 (f6e511eec 2024-10-15)
binary: rustc
commit-hash: f6e511eec7342f59a25f7c0534f1dbea00d01b14
commit-date: 2024-10-15
host: aarch64-apple-darwin
release: 1.82.0
LLVM version: 19.1.1

   v3.9.6
```

# **aarch64**-apple-darwin

`aarch64-`**`apple`**`-darwin`

# aarch64-apple-**darwin**

LEANDER KAHNEY SCIENCE MAR 16, 1999 4:15 PM

# Apple Opens OS Code

WIRED

"It's about evolution," said Apple's interim CEO, Steve Jobs, at an unveiling at Apple's campus in Cupertino, California. "We believe that putting it out there ... will help us make it the best technology -- collectively -- we know how to make."

# **x86_64**-apple-darwin


Macintosh
Quadra 605

**powerpc** -unknown-linux-gnu
**powerpc64** -unknown-linux-gnu
**powerpc64le**-unknown-linux-gnu


SONY
SELECT
START
SONY

# More power. Less energy.

Introducing the fastest, most energy-efficient Xserve ever.

**Talos™ II Secure Workstation**
TL2WK2
Talos™ II Secure Workstation

*Starting at $9,977.00*
*Current Status: Full Production*

view details

**Talos™ II 4U Rack Mount Server**
TL2SV2
Talos™ II 4U Rack Mount Server

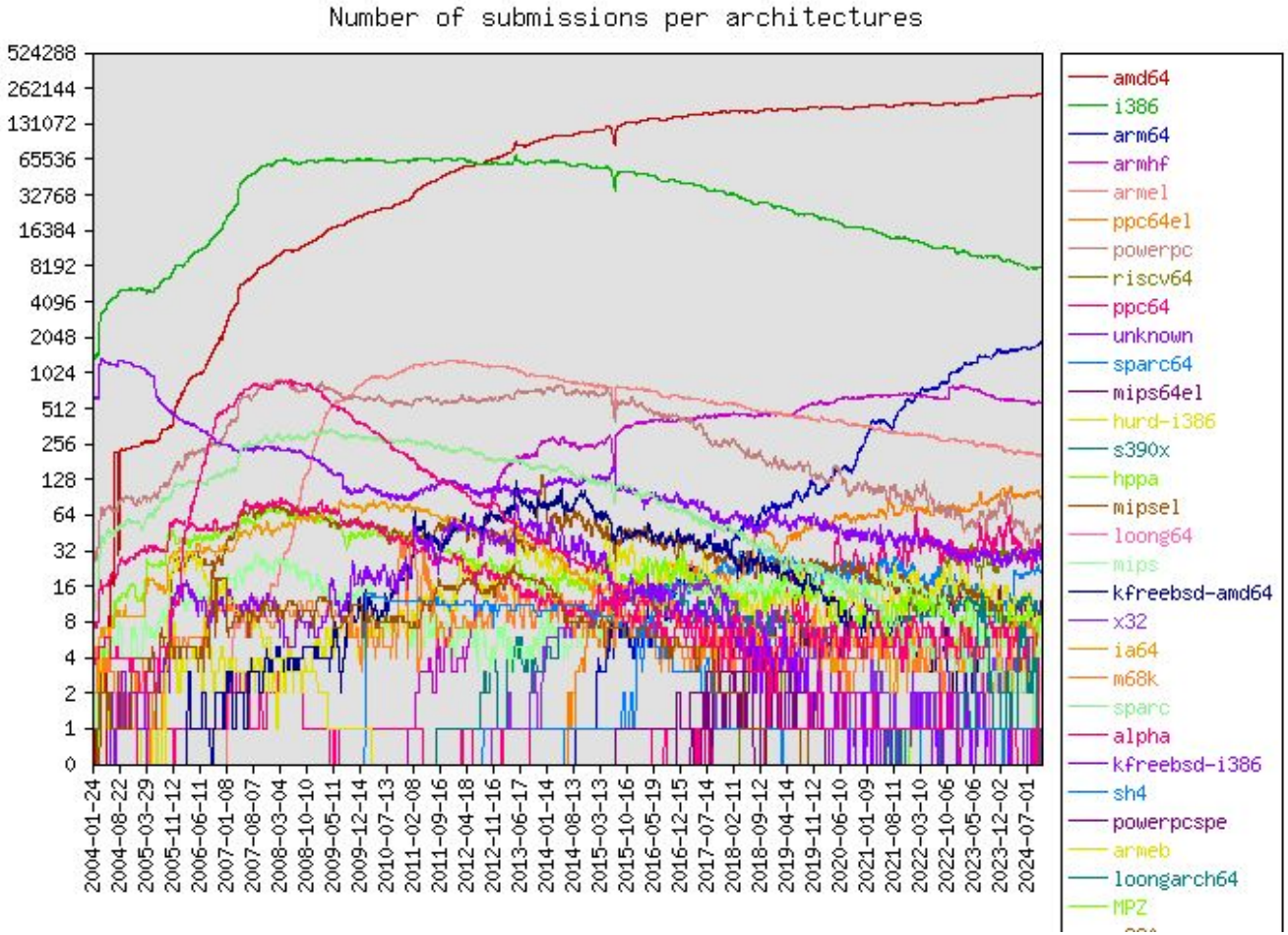
*Starting at $12,226.00*
*Current Status: Full Production*

view details

**Talos™ II 2U Rack Mount NVMe Storage Server**
TL2SV3
Talos™ II 2U Rack Mount NVMe Storage Server

*Order online for $13,526.99*
*Current Status: Full Production*

Image Coming Soon


Number of submissions per architectures
524288
262144
131072
65536
32768
16384
8192
4096
2048
1024
512
256
128
64
32
16
8
4
2
1
0
2004-01-24
2004-08-22
2005-03-29
2005-11-12
2006-06-11
2007-01-08
2007-08-07
2008-03-04
2008-10-10
2009-05-11
2009-12-14
2010-07-13
2011-02-08
2011-09-16
2012-04-18
2012-11-16
2013-06-17
2014-01-14
2014-08-13
2015-03-13
2015-10-16
2016-05-19
2016-12-15
2017-07-14
2018-02-11
2018-09-12
2019-04-14
2019-11-12
2020-06-10
2021-01-09
2021-08-11
2022-03-10
2022-10-06
2023-05-06
2023-12-02
2024-07-01
amd64
i386
arm64
armhf
armel
ppc64el
powerpc
riscv64
ppc64
unknown
sparc64
mips64el
hurd-i386
s390x
hppa
mipsel
loong64
mips
kfreebsd-amd64
x32
ia64
m68k
sparc
alpha
kfreebsd-i386
sh4
powerpcspe
armeb
loongarch64
MPZ

# **why** is powerpc64[le] tier 2?

# "Legacy" tier 2 targets have misplaced or absent maintainer docs #113739

Open

workingjubilee opened on Jul 16, 2023 edited by RalfJung · Edits

## Location

This affects our platform support documentation.

It specifically affects targets added before the target tier policy was confirmed, and especially those that are tier 2.

## Summary

It is widely expected that the existing tier 1 targets are of primary concern for the Rust Project in general. As I understand it, the current absence of formally documented maintainers for them is based on the belief we have a large enough surplus of "target maintainers" for them that we can expect these targets to be effectively supported by "whoever picks up the slack".

However, tier 2 targets are trickier. Many are more niche, harder to find and run code on, and require specialized developer knowledge. These realities are part of *why* we expect targets to have target maintainers. Yet we have several without any documented support because they predate the target tier policy. This has recently led to us being forced to respond to exigent circumstances by interrupting our usual support because they implicitly violated the other side of the target support "contract": they impeded development of all the

Assignees

No one assign

Labels

A-docs A

C-tracking-i

Type

No type

Projects

No projects

Milestone

No milestone

**powerpc** -unknown-linux-gnu
**powerpc64** -unknown-linux-gnu
**powerpc64le**-unknown-linux-gnu

```
x86_64-unknown-linux-gnu
x86_64-unknown-linux-musl
```

```
>_ zsh ×  +                                   default  Mon Oct 28 12:47  souffle.local

amos in 🌐 hell in ~/musl-samples
❯ ldd ./cargo-dist-x86_64-unknown-linux-{gnu,musl}/cargo-dist
./cargo-dist-x86_64-unknown-linux-gnu/cargo-dist:
        linux-vdso.so.1 (0x00007ffea1bdb000)
        libgcc_s.so.1 => /lib/x86_64-linux-gnu/libgcc_s.so.1 (0x00007f88f805a000)
        libpthread.so.0 => /lib/x86_64-linux-gnu/libpthread.so.0 (0x00007f88f8055000)
        libm.so.6 => /lib/x86_64-linux-gnu/libm.so.6 (0x00007f88f7f76000)
        libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2 (0x00007f88f7f71000)
        libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f88f6c1f000)
        /lib64/ld-linux-x86-64.so.2 (0x00007f88f8081000)
./cargo-dist-x86_64-unknown-linux-musl/cargo-dist:
        statically linked

amos in 🌐 hell in ~/musl-samples
❯
```

# Update the existing musl targets to be dynamically linked. #422

New issue

Closed

pnkfelix opened on Apr 1, 2021 edited by estebank · Edits

## Update the existing musl targets to be dynamically linked.

We should change the `-musl` targets to be consistently dynamically linked to their `libc`, rather than statically linking the `musl` libc into them at compile-time. If a MUSL end-user wants the musl-libc to be statically-linked into their program, they will need to specify it explicitly (e.g. via `-Ctarget-feature=+crt-static`)

Assignees

No one assigned

Labels

T-compiler major-change major-change-accepted

```
-Ctarget-feature=+crt-static
```

# **i686**-pc-windows-gnu

# i586-pc-windows-gnu

x86_64-**uwp** -windows-**gnu**

x86_64-**uwp** -windows-msvc

x86_64-**win7**-windows-msvc

**aarch64**-apple-darwin
**aarch64**-unknown-linux-gnu
**aarch64**-pc-windows-msvc

**arm** = ARMv6
**armv7** = ARMv7
**aarch64** = ARMv8

armv7**a**

armv7**r**

arm**eb**v7**r**

thumb**v6m**
thumb**v7m**
thumb**v7em**
thumb**v8m.base**
thumb**v8m.main**

# **arm64_32**-apple-watchos

`aarch64_be-unknown-linux-`**`gnu_ilp32`**

# **m68k**-unknown-linux-gnu

**loongarch64**-unknown-linux-*

aarch64-**fuchsia**

x86_64-**fuchsia**

# `aarch64-unknown-`**`uefi`**

# aarch64-unknown-**redox**

# aarch64-nintendo-switch-**freestanding**

armv6k-**nintendo-3ds**
armv7-**sony-vita**-newlibeabihf

`arm-unknown-linux-gnu`**`eabi`**

`arm-unknown-linux-gnu`**`eabihf`**

# nv**ptx**64-nvidia-cuda

# **s390x**-unknown-linux-gnu

**wasm**32-unknown-unknown
**wasm**32-wasi

```
arm64_32-apple-watchos
 aarch64-apple-watchos
 aarch64-apple-watchos-sim
  x86_64-apple-watchos-sim
```

`x86_64-apple-ios-`**`macabi`**