

**Note: NaCl support [was dropped in Go 1.14](#).**

Go 1.3 includes support for generating x86-32 and x86-64 binaries for Google Native Client (NaCl).

NaCl is a collection of related technologies.

1. NaCl defines a collection of binary formats with specific code layout requirements. There are definitions for 32- and 64-bit x86 and 32-bit ARM.
2. NaCl provides a mechanism for running those binaries in a sandboxed execution environment with a restricted syscall-like interface. That syscall-like interface is not guaranteed to be stable from release to release.
3. To abstract away the syscall-like interface, NaCl defines a stable runtime API called the integrated runtime (IRT).
4. Using the IRT, Google Chrome defines an API called Pepper (aka PPAPI) that NaCl-based plugins use to interact with the browser.
5. To abstract away the machine-specific binary formats, NaCl defines an LLVM-based architecture-independent format called PNaCl.
6. NaCl provides translators from PNaCl format to the architecture-specific formats, invoked automatically, so that you can distribute just a PNaCl binary and execute on all three supported platforms.

Go 1.3 provides support for generating the architecture-specific binaries and using the raw syscall interface (1 and 2 in the list above), and only for the x86 platforms (not for ARM).

There is ongoing work exploring support for the IRT and PPAPI (3 and 4 in the list) but no definite release target. Perhaps they will be in Go 1.4 but perhaps not.

There are no concrete plans to support PNaCl (5 and 6 in the list).

The Go 1.3-generated NaCl binaries can be run using the NaCl SDK `sel_ldr_x86_32` and `sel_ldr_x86_64` programs. They cannot be run directly in Google Chrome. As such, the NaCl support in Go 1.3 is useful only for running sandboxed environments like the [Go Playground](#). The file [misc/nacl/README](#) in the Go distribution explains how to configure your machine so that you can run and NaCl-sandboxed binaries using the `go` command.