

# The GNU Go Compiler

---

For GCC version 16.0.0 (pre-release)

(GCC)

Ian Lance Taylor

---

Published by the Free Software Foundation  
51 Franklin Street, Fifth Floor  
Boston, MA 02110-1301, USA

Copyright © 2010-2025 Free Software Foundation, Inc.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, the Front-Cover Texts being (a) (see below), and with the Back-Cover Texts being (b) (see below). A copy of the license is included in the section entitled “GNU Free Documentation License”.

(a) The FSF’s Front-Cover Text is:

A GNU Manual

(b) The FSF’s Back-Cover Text is:

You have freedom to copy and modify this GNU Manual, like GNU software. Copies published by the Free Software Foundation raise funds for GNU development.

## Table of Contents

<b>GNU General Public License</b> .....	<b>1</b>
<b>GNU Free Documentation License</b> .....	<b>12</b>
ADDENDUM: How to use this License for your documents.....	19
<b>1 Invoking gccgo</b> .....	<b>20</b>
<b>2 Import and Export</b> .....	<b>23</b>
<b>3 Compiler Directives</b> .....	<b>24</b>
<b>4 C Interoperability</b> .....	<b>25</b>
4.1 C Type Interoperability .....	25
4.2 Function Names .....	26
<b>Index</b> .....	<b>27</b>











































maps each file name to a full path to the file. This option is intended for use by the `go` command to implement `//go:embed`.

**-fgo-importcfg=file**

Identify a file that provides mappings for import package paths found in the Go source files. The file can contain two commands: `importpath` to rename import paths for vendoring and `packagefile` to map from package path to files containing export data. This option is intended for use by the `go` command.

**-g**

This is the standard `gcc` option (see Section “Debugging Options” in *Using the GNU Compiler Collection (GCC)*). It is mentioned here because by default `gccgo` turns on debugging information generation with the equivalent of the standard option `-g1`. This is because Go programs require debugging information to be available in order to get backtrace information. An explicit `-g0` may be used to disable the generation of debugging information, in which case certain standard library functions, such as `runtime.Callers`, will not operate correctly.









