

GNU Fortran Internals

For GCC version 16.0.0 (pre-release)

(GCC)

The gfortran team

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

Copyright © 2007-2025 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 the Invariant Sections being “Funding Free Software”, 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.

Short Contents

1	Introduction.....	1
2	Code that Interacts with the User	3
3	Frontend Data Structures	5
4	Internals of Fortran 2003 OOP Features	11
5	Generating the intermediate language for later stages.....	13
6	The LibGFortran Runtime Library	15
	GNU Free Documentation License.....	17
	Index	25

Index.....	25
-------------------	-----------

1 Introduction

This manual documents the internals of **gfortran**, the GNU Fortran compiler.

Warning: This document, and the compiler it describes, are still under development. While efforts are made to keep it up-to-date, it might not accurately reflect the status of the most recent GNU Fortran compiler.

At present, this manual is very much a work in progress, containing miscellaneous notes about the internals of the compiler. It is hoped that at some point in the future it will become a reasonably complete guide; in the interim, GNU Fortran developers are strongly encouraged to contribute to it as a way of keeping notes while working on the compiler.

not an error existed. To check the state of the buffer without changing its state or reporting the errors, the `gfc_error_flag_test` function can be used. The `gfc_clear_error` function will clear out any errors in the buffer, without reporting them. The `gfc_warning_check` and `gfc_clear_warning` functions provide equivalent functionality for the warning buffer.

Only one error and one warning can be in the buffers at a time, and buffering another will overwrite the existing one. In cases where one may wish to work on a smaller piece of source code without disturbing an existing error state, the `gfc_push_error`, `gfc_pop_error`, and `gfc_free_error` mechanism exists to implement a stack for the error buffer.

For cases where an error or warning should be reported immediately rather than buffered, the `gfc_error_now` and `gfc_warning_now` functions can be used. Normally, the compiler will continue attempting to parse the program after an error has occurred, but if this is not appropriate, the `gfc_fatal_error` function should be used instead. For errors that are always the result of a bug somewhere in the compiler, the `gfc_internal_error` function should be used.

The syntax for the strings used to produce the error/warning message in the various error and warning functions is similar to the `printf` syntax, with `'%'`-escapes to insert variable values. The details, and the allowable codes, are documented in the `error_print` function in `error.cc`.

3.2.7 Constant Substring References

`EXPR_SUBSTRING` is a special type of expression that encodes a substring reference of a constant string, as in the following code snippet:

```
x = "abcde"(1:2)
```

In this case, `value.character` contains the full string's data as if it was a string constant, but the `ref` member is also set and points to a substring reference as described in the subsection above.

Index

D

data structures..... 5

F

FDL, GNU Free Documentation License..... 17

G

gfc_code 5
gfc_expr 7

S

statement chaining..... 5
struct gfc_code..... 5
struct gfc_expr..... 7