

First edition
1999-02-01

Corrected and reprinted
2000-12-15

**Information technology — Programming
languages — Fortran —**

**Part 3:
Conditional compilation**

*Technologies de l'information — Langages de programmation — Fortran —
Partie 3: Compilation conditionnelle*
(standards.iteh.ai)

[ISO/IEC 1539-3:1999](https://standards.iteh.ai/catalog/standards/sist/aafbfb1-0ddd-48bd-8f89-6b683f55db54/iso-iec-1539-3-1999)

<https://standards.iteh.ai/catalog/standards/sist/aafbfb1-0ddd-48bd-8f89-6b683f55db54/iso-iec-1539-3-1999>



Reference number
ISO/IEC 1539-3:1999(E)

© ISO/IEC 1999

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

[ISO/IEC 1539-3:1999](https://standards.iteh.ai/catalog/standards/sist/aafbfb1-0ddd-48bd-8f89-6b683f55db54/iso-iec-1539-3-1999)

<https://standards.iteh.ai/catalog/standards/sist/aafbfb1-0ddd-48bd-8f89-6b683f55db54/iso-iec-1539-3-1999>

© ISO/IEC 1999

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.ch
Web www.iso.ch

Printed in Switzerland

Contents

1	General.....	1
1.1	Scope	1
1.2	Normative References	1
2	Overview	2
2.1	Conditional compilation	2
2.2	Clause numbers and syntax rules	2
2.3	Coco program conformance	2
2.4	High level syntax	3
3	Constants, source form and text inclusion	3
3.1	Coco constants	3
3.2	Coco source form	4
3.2.1	Coco commentary	5
3.2.2	Coco directive continuation	5
3.2.3	Coco directives	6
3.3	Source text inclusion	7
4	Coco type declaration directives	8
5	Coco variables, expressions and assignment directive	9
5.1	Coco variables	9
5.2	Coco expressions	9
5.2.1	Coco primary	9
5.2.2	Level-1 expressions	9
5.2.3	Level-2 expressions	9
5.2.4	Level-3 expressions	10
5.2.5	General form of a coco expression.....	10
5.3	Data type and value of a coco expression	10
5.4	Coco initialization expression	12
5.5	Coco assignment directive	13
6	Coco execution control and conditional compilation	13
6.1	Coco blocks.....	13
6.2	Coco IF construct	13
6.2.1	Form of the coco IF construct	13
6.2.2	Execution of an IF construct	14

7	Coco message and stop directives	15
8	Scope and definition of coco variables	16
8.1	Scope of coco variables	16
8.2	Events that cause coco variables to become defined	16
9	The coco SET file	17
	Annex A Examples	20

iTeh STANDARD PREVIEW
(standards.iteh.ai)

[ISO/IEC 1539-3:1999](https://standards.iteh.ai/catalog/standards/sist/aafbfb1-0ddd-48bd-8f89-6b683f55db54/iso-iec-1539-3-1999)

<https://standards.iteh.ai/catalog/standards/sist/aafbfb1-0ddd-48bd-8f89-6b683f55db54/iso-iec-1539-3-1999>

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 1539-3 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 22, *Programming languages, their environments and system software interfaces*.

ISO/IEC 1539 consists of the following parts, under the general title *Information technology — Programming languages — Fortran*: **(standards.iteh.ai)**

- *Part 1: Base language*
- *Part 2: Varying length character strings* [ISO/IEC 1539-3:1999](https://standards.iteh.ai/catalog/standards/sist/aafb1c1-0ddd-48bd-8f89-6b683f55db54/iso-iec-1539-3-1999)
- *Part 3: Conditional compilation*

Annex A of this part of ISO/IEC 1539 is for information only.

Introduction

Programmers often need to maintain several versions of code to allow for different systems and different applications. Keeping several copies of the source code is error prone. It is far better to maintain a master code from which any of the versions may be selected.

This conditional compilation facility has deliberately been kept very simple. The additional lines inserted to control the process and all the lines that are not selected are omitted from the output or are converted to comments. Those that are selected are copied to the output completely unchanged. Which version is selected is controlled by directives in a file known as the SET file.

Examples of the need for such a facility are:

- (1) Parameterized types do not solve all the problems associated with different precisions. Parameterized derived types are not part of Fortran 95.
- (2) A version of a code for complex arithmetic may differ little from the version for real arithmetic.
- (3) The relative efficiency of different algorithms or constructions may vary from processor to processor.
- (4) Versions may be required for different message-passing libraries.
- (5) Additional print statements may be inserted into a program when under development. It may be very helpful to have these readily available in case some unexpected results are found in production use.
- (6) Versions may be required with character constants in different languages (internationalization).
- (7) For OPEN statements, the file naming convention varies between systems.

Some of these cases may be addressed within the Fortran code itself by run-time tests, but this will result in a large object code and some run-time overhead. Without conditional compilation, however, most of them can only be solved by maintaining separate versions of the code.

Information technology — Programming languages — Fortran —

Part 3: Conditional compilation

1 General

1.1 Scope

This part of ISO/IEC 1539 defines facilities for conditional compilation in Fortran. This part of ISO/IEC 1539 provides an auxiliary standard for the version of the Fortran language specified by ISO/IEC 1539-1 and informally known as Fortran 95.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

[ISO/IEC 1539-3:1999](https://standards.iteh.ai/catalog/standards/sist/aafbfb1-0ddd-48bd-8f89-6b683f55db54/iso-iec-1539-3-1999)

<https://standards.iteh.ai/catalog/standards/sist/aafbfb1-0ddd-48bd-8f89-6b683f55db54/iso-iec-1539-3-1999>

1.2 Normative References

The following standard contains provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 1539. At the time of publication, the edition indicated was valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 1539 are encouraged to investigate the possibility of applying the most recent editions of the standard indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 1539-1 : 1997, *Information technology – Programming languages – Fortran – Part 1: Base language*.

2 Overview

2.1 Conditional compilation

Conditional compilation (coco) is described in this document as an independent process that yields a source program for a Fortran processor. It is expected that implementations will usually integrate the two processes.

The coco process is controlled by directives that are either omitted from the coco output or are converted to Fortran comments. Coco comments may be introduced to explain the actions and these, too, are either omitted from the coco output or are converted to Fortran comments. Other lines (noncoco lines) are either copied unchanged to the output, omitted, or converted to Fortran comments. There is no requirement that the coco output is a valid Fortran program. The lines of the coco output are in the same order as the corresponding lines of the coco program.

Coco execution is a sequence of actions specified by the coco directives and performed in the order that they appear. The combination of a computing system and the mechanism by which these actions are performed is called a **coco processor** in this part of this standard.

2.2 Clause numbers and syntax rules

The notation used in this part of ISO/IEC 1539 is described in ISO/IEC 1539-1, 1.6. However, item (4) in ISO/IEC 1539-1, 1.6.2 is replaced with: **(standards.iteh.ai)**

- (4) Each syntax rule is given a unique identifying number of the form CCR snn , where s is a one or two-digit clause number and nn is a two-digit sequence number within that clause. The syntax rules are distributed as appropriate throughout the text, and are referenced by number as needed.

2.3 Coco program conformance

A coco program is a standard-conforming coco program if it uses only those forms and relationships herein and if the program has an interpretation according to this part of this standard.

A coco processor conforms to this part of this standard if:

- (1) It executes any standard-conforming coco program and its SET file in a manner that fulfills the interpretations herein, subject to any limits that the processor may impose on the size and complexity of the coco program and its SET file.
- (2) It contains the capability to detect and report the use within the executed part of a coco program and its SET file of an additional form or relationship that is not permitted by the numbered syntax rules or their associated constraints.
- (3) It contains the capability to detect and report the use within the executed part of a coco program and its SET file of source form not permitted by clause 3.
- (4) It contains the capability to detect and report the reason for rejecting a submitted coco program and its SET file.

If a coco program contains a STOP directive that is executed, there is no requirement for the processor to report on any directives that follow the STOP directive.

2.4 High level syntax

This subclause introduces the terms associated with the conditional compilation program.

CCR201 *coco-program* is *pp-input-item* [*pp-input-item*] ...

CCR202 *pp-input-item* is *coco-construct*
or *noncoco-line*

The term *noncoco-line* refers to any line without the characters "??" in character positions 1 and 2.

CCR203 *coco-construct* is *coco-type-declaration-directive*
or *coco-action-construct*

CCR204 *coco-action-construct* is *coco-action-directive*
or *coco-if-construct*

CCR205 *coco-action-directive* is *coco-assignment-directive*
or *coco-message-directive*
or *coco-stop-directive*

Note 2.1

A coco program is not required to contain any coco directives.

ITEH STANDARD PREVIEW
(standards.iteh.ai)

ISO/IEC 1539-3:1999

<https://standards.iteh.ai/catalog/standards/sist/aafbfb1-0ddd-48bd-8f89->

3 Constants, source form and text inclusion

3.1 Coco constants

CCR301 *coco-constant* is *coco-literal-constant*
or *coco-named-constant*

CCR302 *coco-literal-constant* is *coco-int-literal-constant*
or *coco-logical-literal-constant*

CCR303 *coco-int-literal-constant* is *digit* [*digit*] ...

CCR304 *coco-logical-literal-constant* is *.TRUE.*
or *.FALSE.*

CCR305 *coco-char-literal* is ' [*rep-char*] ... '
or " [*rep-char*] ... "

CCR306 *coco-named-constant* is *name*

Constraint: *coco-named-constant* shall have the PARAMETER attribute.

CCR307 *name* is *letter* [*alphanumeric-character*] ...

Constraint: The maximum length of a *name* is 31 characters.

CCR308 *alphanumeric-character* is *letter*
 or *digit*
 or *underscore*

CCR309 *underscore* is *_*

Each *digit* is one of the digits

0 1 2 3 4 5 6 7 8 9

and each *coco-int-literal-constant* is interpreted as a decimal value.

Each *letter* is one of the upper-case letters

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

or one of the lower-case letters

a b c d e f g h i j k l m n o p q r s t u v w x y z

Each *rep-char* is a character in the processor-dependent character set, which includes the letters, the digits, the underscore, the blank, the currency symbol, and the characters

= + - * / () , . ' : ! " % & ; < > ?

In a *coco* directive, a lower-case letter is equivalent to the corresponding upper-case letter except in a *coco* character literal.

iTeh STANDARD PREVIEW
 (standards.iteh.ai)

The delimiting apostrophes or quotation marks are not part of the value of a *coco* character literal.

An apostrophe character within a *coco* character literal delimited by apostrophes is represented by two consecutive apostrophes (without intervening blanks); in this case, the two apostrophes are counted as one character. Similarly, a quotation mark character within a *coco* character literal delimited by quotation marks is represented by two consecutive quotation marks (without intervening blanks) and the two quotation marks are counted as one character.

3.2 Coco source form

A *coco* program is a sequence of one or more lines, organized as *coco* directives, *coco* comment lines (3.2.1) and non*coco* lines. A *coco* directive is a sequence of one or more *coco* lines. A *coco* line is a line with the characters "??" in character positions 1 and 2. These characters are not part of the *coco* directive. A non*coco* line is a line that does not begin in this way.

A **keyword** is a word that is part of the syntax of a *coco* directive. Examples of keywords are IF, INTEGER, LOGICAL, and MESSAGE.

A *coco* comment may contain any character that may occur in a *coco* character literal. Outside commentary, a *coco* directive consists of a sequence of *coco* lexical tokens. Each token is a keyword, a name, a literal constant, an operator (see Table 2), a comma, a parenthesis, an equals sign, or the separator ::.

In *coco* source, each source line may contain from zero to 132 characters.

In *coco* source, blank characters shall not appear within *coco* lexical tokens other than in a *coco* character literal. Blanks may be inserted freely between tokens to improve readability. A sequence of blank characters outside of a *coco* character literal is equivalent to a single blank character.

A blank shall be used to separate names, constants, or *coco-char-literals* from adjacent keywords, names, constants, or *coco-char-literals*.

Blanks are optional between the following pairs of adjacent coco keywords:

```
ELSE IF
END IF
```

3.2.1 Coco commentary

Within a coco directive, the character "!" in any character position initiates a coco comment except when it appears within a coco character literal. The coco comment extends to the end of the source line. If the first nonblank character on a coco line after character positions 1 and 2 is an "!", the line is a coco comment line. Coco lines containing only blanks after character positions 1 and 2 or containing no characters after character positions 1 and 2 are also coco comment lines.

Note 3.1

An example of the use of a coco comment in a coco IF construct (6.2) is:

```
?? IF (DEVELOPING) THEN
?? ! The following output statement was used when
?? ! developing the code
    WRITE(UNIT=*,FMT=*) 'The value of A is', A
?? END IF
```

iTeh STANDARD PREVIEW

(standards.iteh.ai)

3.2.2 Coco directive continuation

The character "&" is used to indicate that the current coco directive is continued on the next line that is not a coco comment line. This line shall be a coco line. Coco comment lines shall not be continued; an "&" in a coco comment has no effect during coco execution. Comments may occur within a continued coco directive. When used for continuation, the "&" is not part of the coco directive. After character positions 1 and 2, no coco line shall contain a single "&" as the only nonblank character or as the only nonblank character before an "!" that initiates a coco comment.

3.2.2.1 Continuation other than of a coco character literal

In a coco directive, if an "&" not in a coco comment is the last nonblank character on a line or the last nonblank character before an "!" that initiates a coco comment, the coco directive is continued on the next line that is not a coco comment line. If the first nonblank character after character positions 1 and 2 on the next coco-noncomment line is an "&", the coco directive continues at the next character following the "&"; otherwise, it continues with character position 3 of the next coco-noncomment line.

If a coco lexical token is split across the end of a line, the first nonblank character after character positions 1 and 2 on the first following coco-noncomment line shall be an "&" immediately followed by the successive characters of the split token.