

OpenOffice templates for writing standards and specifications

Draft 0.60

Modified at 11:36 on Feb 7, 2007

Sponsor:

Friendly supporters

Abstract: These templates were developed by volunteers with extensive experience in the development of IEEE Standards. These templates are not mandated; each working group has the freedom to select the most applicable text-formatting tools. These templates may be freely used by other SDOs and/or private organizations.

Keywords: Style, formatting, template

Copyright (c) 2003 JGG. All rights reserved.

Redistribution and use in electronic and printed documents, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of visible template text must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Usage of these templates, with alternative visible text content, must reproduce the above copyright notice, this list of conditions and the following disclaimer in nonvisible text on the first or last page.
3. The name "JGG" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact dvj@alum.mit.edu.

THIS DOCUMENTATION IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JGG OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This Copyright notice can be removed and (as necessary) replaced with the appropriate corporate or Standard's Development Organization (SDO) Copyright notice after print-visible text has been replaced with alternative written material.

Background

This manual describes the use of document templates developed by for use when writing Cypress and IEEE drafts. This document is **preliminary** and **subject to change**.

Contacts

David V. James
3180 South Ct
Palo Alto, CA 94306
Home: +1.650.494.0926
+1.650.856.9801
Cell: +1.650.954.6906
Fax: +1.360.242.5508
Base: dvj@alum.mit.edu

Change history

The following table shows the change history for this user's manual.

Version	Date	Author	Description
—	—	DVJ	Original content.
0.55	2004Oct12	DVJ	Text updated to track FrameMaker style changes.
0.58	2005Nov25	DVJ	Style changes to support annex numbering.

Table of contents

02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

- 1. Overview.....9
- 1.1 Scope and purpose.....9
- 1.1.1 Template contents.....9
- 1.2 Standard clauses.....10
- 2. References.....11
- 3. Terms and definitions.....12
- 3.1 Conformance levels.....12
- 3.2 Glossary of terms.....12
- 3.3 Unimplemented locations.....13
- 3.4 Numerical values.....13
- 3.5 Field notation.....14
- 3.5.1 Field names.....14
- 3.5.2 Fields within figures.....15
- 3.5.3 Multibyte fields within frames.....15
- 3.6 C-code notation.....16
- 3.7 State machines.....17
- 3.8 Field notations.....17
- 3.8.1 Use of italics.....17
- 3.8.2 Field conventions.....18
- 3.8.3 Field value conventions.....18
- 3.9 Informative notes.....19
- 4. Abbreviations and acronyms.....20
- 5. Using the templates.....21
- 5.1 Formatting the final draft.....21
- 5.2 Company document numbers.....21
- 5.3 Paragraph styles.....22
- 5.3.1 Baseline paragraph styles.....22
- 5.3.2 Character styles (such as α).....24
- 5.4 Style tricks.....25
- 5.4.1 Subclause cross references.....25
- 5.4.1.1.1.1.....25
- 5.4.2 Row cross references restarts.....26
- 5.4.3 Row cross references tags.....26
- 6. Clause and subclause headings.....28
- 6.1 Subclause-2 heading.....28
- 6.1.1 Subclause-3 heading.....28
- 6.1.1.1 Subclause-4 heading.....28
- 6.1.1.1.1 Subclause-5 heading.....28
- 6.1.1.1.1.1 Subclause-6 heading.....28
- 6.1.1.1.1.1.1 Subclause-7 heading.....28
- 7. Definitions.....30
- 7.1 Leading second-level indent.....30
- 7.1.1 Leading third-level heading.....30
- 7.1.1.1 Definition5 heading.....30
- 7.1.2 Definition4 heading.....30
- 7.2 Definition3 headings.....30
- 8. Highest level definitions.....31

02

03 9. Formatting styles.....32

04 9.1 Notes.....32

05 9.1.1 Editorial notes.....32

06 9.1.2 Informative notes.....32

07 9.2 Page breaks.....33

08 9.3 Body styles.....33

09 9.3.1 Body.....33

10 9.3.2 BodyTight.....33

11 9.3.3 BodyCenter.....33

12 9.3.4 BodyBlock.....33

13 9.4 Lists.....34

14 9.5 Tables.....34

15 9.5.1 Tables cells.....34

16 9.5.2 Tables row labels.....35

17 9.5.3 Equations.....35

18 9.6 C-code listings.....35

19 9.6.1 Shorter in-line C code.....35

20 9.6.2 Lengthy C code.....36

21

22 10. Constructing OpenOffice drawings.....37

23 10.1 Figures.....37

24 10.1.1 Figure creation.....37

25 10.1.2 Snap grid usage.....37

26 10.1.3 Register layouts.....37

27 10.1.4 Flow-chart styles.....38

28

29 Annexes.....39

30

31 Annex A (informative) Bibliography.....39

32 A.1 Bibliography references.....39

33 A.2 Grammar references.....40

34

35 Annex B (informative) Annex and subannex.....41

36 B.1 Subannex-1 header.....41

37 B.2 Subannex-2 header.....41

38 B.2.1 Subannex-3 header.....41

39 B.2.1.1 Subannex-4 header.....41

40 B.2.1.1.1 Subannex-5 header.....41

41 B.2.1.1.1.1 Subannex-6 header.....41

42 B.3 Annex definitions.....42

43 B.3.1 Subannex-3 header.....42

44 B.3.1.1 Subannex-4 header.....42

45 B.3.2 Subannex-3 header.....42

46 B.4 Definition2 headings.....42

47

48 Annex C (informative) Highest level definitions.....43

49

50 Annex D (normative) Numbered annex styles.....44

51 D.1 Annex figures.....44

52 D.2 Annex tables.....44

53 D.3 Equation numbering differences.....44

54

55 Annex E (informative) Common grammatical issues.....45

56 E.1 Use of the word “the”.....45

57 E.2 Use of the word “a” or “an”.....45

58 E.3 Numbers: numerical or alphabetical?.....45

59

60

E.4 That and which.....46 03
E.5 Gender-neutral language.....46 04
E.6 Use of the second-person form of address.....46 05
06
Annex F (informative) General writing guidelines.....47 07
F.1 Capitalization.....47 08
F.1.1 When to capitalize.....47 09
F.1.2 When to capitalize.....47 10
F.1.3 Noncapitalized definitions.....48 11
F.1.4 Noncapitalized acronyms.....48 12
F.1.5 Improper nouns.....49 13
F.2 Definitions.....49 14
F.2.1 Definition clauses.....49 15
F.2.2 Definition text.....49 16
F.3 Variable names.....49 17
F.3.1 Spaceless variable names.....49 18
F.3.2 Hyphenless variable names.....49 19
F.3.3 Aliased variable names.....50 20
F.4 Spelling checker.....50 21
F.5 Cross references.....50 22
F.6 Indentation.....51 23
F.7 Table cell justification.....51 24
F.8 Punctuation within lists.....51 25
26
Annex G (informative) Writing guidelines.....53 27
G.1 Field formats within figures.....53 28
G.1.1 Aliased variable names.....53 29
G.1.2 Generic field illustrations.....53 30
G.1.3 8-bit field format.....54 31
G.1.4 16-bit field format.....54 32
G.1.5 32-bit field format.....54 33
G.2 Standard subclause formats.....55 34
G.2.1 Common parameters.....55 35
G.2.1.1 Common definitions.....55 36
G.2.1.2 Common variables.....55 37
G.2.1.3 Common routines.....56 38
G.2.2 Variables defined in other clauses.....56 39
G.2.3 State machine definitions.....57 40
G.2.4 Single-queue transmit state machine.....57 41
G.2.5 Single-queue transmit flow chart.....57 42
G.2.5.1 Single-queue transmit flow chart.....57 43
G.2.5.2 Single-queue transmit state machine definitions.....58 44
G.2.5.3 Single-queue transmit state machine variables.....59 45
G.2.5.4 Single-queue transmit state machine routines.....59 46
G.2.5.5 Single-queue transmit state table.....59 47
G.3 Service primitives.....60 48
G.3.1 MA_DATA.request.....60 49
G.3.1.1 Function.....60 50
G.3.1.2 Semantics of the service primitive.....60 51
G.3.1.3 Semantics of the service primitive.....61 52
G.3.1.4 Effect of receipt.....61 53
G.3.1.5 Additional comments.....61 54
55
Annex H (informative) Highest level definitions.....62 56
H.1 Standard subclause formats.....62 57
H.2 Standard subclause formats.....64 58
59
60

02
03 Annex (informative) C code illustrations.....68
04
05 Annex A (informative) Bibliography.....39
06 A.1 Bibliography references.....39
07 A.2 Grammar references.....40
08
09 Annex B (normative) Annex and subannex.....41
10 B.1 Annex headings.....41
11 B.2 Subannex-2 header.....41
12 B.2.1 Subannex-3 header.....41
13 B.2.1.1 Subannex-4 header.....41
14 B.2.1.1.1 Subannex-5 header.....41
15 B.2.1.1.1.1 Subannex-6 header.....41
16 B.2.1.1.1.2 Subannex-7 header.....42
17 B.3 Annex definitions.....42
18 B.3.1 Subannex-3 header.....42
19 B.3.1.1 Subannex-4 header.....42
20 B.3.2 Definition4 heading.....42
21 B.4 Definition2 headings.....42
22
23 Annex C (informative) Highest level definitions.....43
24
25 Annex D (normative) Tables and figures.....44
26 D.1 Annex figures.....44
27 D.2 Annex tables.....44
28 D.2.1 Equation numbering differences.....44
29
30 Annex E (informative) Common grammatical issues.....45
31 E.1 Use of the word “the”.....45
32 E.2 Use of the word “a” or “an”.....45
33 E.3 Numbers: numerical or alphabetical?.....46
34 E.4 That and which.....46
35 E.5 Gender-neutral language.....46
36 E.6 Use of the second-person form of address.....46
37
38 Annex F (informative) Common grammatical issues47
39 F.1 Capitalization.....47
40 F.1.1 When to capitalize.....47
41 F.1.2 Heading capitalization48
42 F.1.3 Noncapitalized definitions.....48
43 F.1.4 Noncapitalized acronyms.....48
44 F.1.5 Improper nouns.....49
45 F.2 Improper nouns.....49
46 F.2.1 Definition clauses.....49
47 F.2.2 Definition text.....49
48 F.3 Variable names.....49
49 F.3.1 Spaceless variable names.....49
50 F.3.2 Hyphenless variable names.....50
51 F.3.3 Aliased variable names.....50
52 F.4 Spelling checker.....50
53 F.5 Cross references.....51
54 F.6 Indentation.....51
55 F.7 Table cell justification.....52
56 F.8 Punctuation within lists.....52
57
58 Annex G (informative) Writing guidelines53
59
60
61

02

G.1 Field formats within figures.....53 03

 G.1.1 Frame format illustrations.....53 04

 G.1.2 Generic field illustrations.....54 05

 G.1.3 8-bit field format.....54 06

 G.1.4 16-bit field format.....55 07

 G.1.5 32-bit field format55 08

G.2 Standard subclause formats.....56 09

 G.2.1 Common parameters.....56 10

 G.2.1.1 Common definitions.....56 11

 G.2.1.2 Common variables.....56 12

 G.2.1.3 Common routines.....57 13

 G.2.2 Variables defined in other clauses.....57 14

 G.2.3 State machine definitions.....57 15

 G.2.4 Single-queue transmit state machine.....57 16

 G.2.5 Single-queue transmit flow chart.....58 17

 G.2.5.1 Single-queue transmit flow chart.....58 18

 G.2.5.2 Single-queue transmit state machine definitions.....59 19

 G.2.5.3 Single-queue transmit state machine variables.....59 20

 G.2.5.4 Single-queue transmit state machine routines.....59 21

 G.2.5.5 Single-queue transmit state table.....60 22

 G.3 Service primitives.....61 23

 G.3.1 MA_DATA.request.....61 24

 G.3.1.1 Function.....61 25

 G.3.1.2 Semantics of the service primitive.....61 26

 G.3.1.3 When generated.....62 27

 G.3.1.4 Effect of receipt.....62 28

 G.3.1.5 Additional comments.....62 29

Annex H (informative) IEEE specifics63 31

 H.1 IEEE references.....63 32

 H.2 IEEE cover pages.....65 33

Annex I (informative) C code illustrations69 35

02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63

List of figures

Figure 3.1—Expanded bit-field descriptions.....	15
Figure 3.2—Frame format illustrations.....	15
Figure 3.3—Flow-control sendA generation overview.....	17
Figure 10.1—OpenOffice expansion and contraction restrictions.....	37
Figure 10.2—Expanded bit-field descriptions.....	37
Figure 10.3—Font sizes within flow-chart illustrations.....	38
Figure D.1—Instruction illustration.....	44
Figure G.1—Frame format illustrations.....	53
Figure G.2—Example of a bit-field illustration.....	53
Figure G.3—8-bit field format.....	54
Figure G.4—16-bit field format.....	54
Figure G.5—32-bit field format.....	55
Figure G.6—Flowchart template.....	58
Figure D.1—Instruction illustration.....	44
Figure G.1—Frame format illustrations.....	53
Figure G.2—Example of a bit-field illustration.....	54
Figure G.3—8-bit field format.....	54
Figure G.4—16-bit field format.....	55
Figure G.5—32-bit field format.....	55
Figure G.6—Flowchart template.....	58

List of tables

Table 3.1—Names of constants, registers and fields.....	14
Table 3.2—C code expressions.....	16
Table 3.3—Flow-control sendA generation details.....	17
Table 3.4—Names of fields and sub-fields.....	18
Table 3.5—wrap field values.....	18
Table 5.1—Paragraph styles.....	22
Table 5.2—Character styles.....	24
Table 5.3—Table-Row formatting example.....	27
Table 9.1—Example table.....	34
Table 9.2—Row-numbered table.....	35
Table D.1—Names of command, status, and CSR values0.....	44
Table F.1—Example table.....	51
Table G.1—Single-queue transmit selection states0.....	59
Table G.2—service_class values.....	60
Table D.1—Names of command, status, and CSR values0.....	44
Table F.1—Example table.....	52
Table G.1—Single-queue transmit selection states0.....	60
Table G.2—service_class values.....	61

Open source styles for standards and specifications (OS4) OpenOffice styles and templates

1. Overview

NOTE—A manual no-page override allows the “Heading 1” style to be used for the first heading.

1.1 Scope and purpose

NOTE—Every document should start off the overview with a scope and purpose statement. Each should consist of a single paragraph outlining, as clearly as possible, the scope and purpose of the document. These should be viewed as executive summaries. The scope is intended to communicate the range of topics covered in the document; the purpose is intended to describe the reasons for generation of the document.

This document is intended to assist corporate engineers in the development of standards, with the scope and purpose listed below:

Scope: This document describes the use of OpenOffice Writer¹ templates for creating ISO/IEC compatible standards. corporate engineering documents may also use these style guidelines.

Purpose: To provided clarity and consistency of corporate documentation developed for internal engineering uses, and to facilitate the transfer of such specifications to standards development organizations (SDOs) for the subset of specifications intended to be standardized.

The templates described by this document contain all the formatting necessary for the cover page, table of contents, list of tables, list of figures, main content, and annexes of your document. No index formatting has been provided, since the editors of this document do not ordinarily have the time to create an index.

1.1.1 Template contents

The templates described by this document contain all the formatting necessary for the cover page, table of contents, list of tables, list of figures, main content, and annexes of your document. These templates provide the editor with the following services:

- a) Boilerplate. Standard boilerplate as well as terms-and-definitions material is provided.
- b) Table of contents. Automatic generation of table of contents, optionally including:
 - 1) List of figures.
 - 2) List of tables.
- c) Autonumbering. All clauses, subclauses, annexes, and subannexes as well as tables, figures, equations, and table rows are automatically numbered.
- d) Commonality. The same look-and-feel templates and common style names are available in Word, Openoffice, and FrameMaker.

The intent of this document is to improve productivity by improving the readability of architecture-specification documents. Perceived benefits to individual companies, that may choose to use these formats for internal specifications, include (but are not necessarily limited to) the following:

¹ All product and company names mentioned in this document are trademarks of their respective holders.

- 03 a) Efficiency. Fewer hours will be consumed communicating necessary information.
- 04 b) Quality. Fewer mistakes will occur due to misinterpreted specifications.
- 05 c) Timeliness. Reviews and product design times will be reduced due to improved efficiency.
- 06 d) Transferability. Corporate text can be readily copied into IEEE and ISO/IEC standards proposals.
- 07 e) Sustainability. Adopting IEEE and ISO/IEC formats eliminates the need for document-style
- 08 committees.
- 09
- 10

12 **1.2 Standard clauses**

13 Document contents are usually constrained by the type of document you are writing, or by documentation
14 standards outlined by whatever agency or office requests or requires your document. However, a few
15 clauses and annexes are expected to take the form described below.

- 16 — Clause 1. Overview shall be the first clause and shall start with scope and purpose subclauses.
- 17 — Clause 2. References shall be the second clause, edited as appropriate.
- 18 — Clause 3. Definitions and notation shall be the third clause, edited as appropriate.
- 19 — Annex A (informative). Bibliography shall appear in every document.
- 20
- 21
- 22
- 23
- 24
- 25
- 26
- 27
- 28
- 29
- 30
- 31
- 32
- 33
- 34
- 35
- 36
- 37
- 38
- 39
- 40
- 41
- 42
- 43
- 44
- 45
- 46
- 47
- 48
- 49
- 50
- 51
- 52
- 53
- 54
- 55
- 56
- 57
- 58
- 59
- 60
- 61

2. References

References are listed here is their content is normative, in that the document would be incomplete without them. Other documents that provide background, but not specification material, should be included in Annex A.

The following standards contain provisions that, through reference in this document, constitute provisions of this standard. All the standards listed are normative references. Informative references are given in Annex A. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below.

- [R1] IEEE Standards Style Manual, October 1996.
- [R2] ANSI/ISO 9899-1990, Programming Language—C.
- [R3] *The Chicago Manual of Style*. Chicago: The University of Chicago Press
- [R4] *Words Into Type*. Englewood Cliffs, NJ: Prentice-Hall, Inc

All the standards listed are normative references. Informative references are given in Annex A. At the time of publication, the editions indicated were valid.

Editors should be aware that approval votes on several draft standards have been delayed due incomplete reference lists. If other standards are referenced, your standard should clearly state which portions apply. Every reference should include a footnote, clearly specifying how this document can be obtained and/or purchased, so that your reviewers have access to these normative references.

Standards a specific draft of a standard may be referenced, such as "ISO 646:1991." This improves the stability of your standard, decoupling it from revisions or extensions in your referenced standard. Alternatively, you may reference the most recent version of a standard, such as "ISO 646." This allows your standard to evolve over time and reduces dependencies on potentially out-of-print documents. Both approaches have their advantages and disadvantages that should be carefully considered by the working group.

03 3. Terms and definitions

06 These subclauses contain examples of specifications that may be included in an IEEE Standard.

10 3.1 Conformance levels

11 Several keywords are used to differentiate between different levels of requirements and optionality, as
12 follows:

14 **3.1.1 expected:** Describe the behavior of the hardware or software in the design models assumed by this
15 specification. Other hardware and software design models may also be implemented.

18 NOTE—The preceding “expected” conformance statement has been found to be useful in some
19 standards. The following “may, shall, should” conformance definitions should be provided by IEEE
20 standards; if provided, their definitions shall be as follows:

22 **3.1.2 may:** Indicates a course of action permissible within the limits of the standard with no implied
23 preference (“may” means “is permitted to”).

24 **3.1.3 shall:** Indicates mandatory requirements strictly to be followed in order to conform to the standard
25 and from which no deviation is permitted (“shall” means “is required to”).

27 **3.1.4 should:** An indication that among several possibilities, one is recommended as particularly suitable,
28 without mentioning or excluding others; or that a certain course of action is preferred but not necessarily
29 required; or that (in the negative form) a certain course of action is deprecated but not prohibited (“should”
30 means “is recommended to”).

33 3.2 Glossary of terms

35 NOTE—The following terms illustrate the types of terms that could be included in your document. Recent
36 standards avoid the use of *half-word*, *word*, or *double-word* to describe register or bus widths, as the
37 meaning of *word* is highly context sensitive and therefore subject to misinterpretation.

39 **3.2.1 byte:** Eight bits of data, used as a synonym for octet.

40 **3.2.2 doublet:** Two bytes of data.

42 **3.2.3 quadlet:** Four bytes of data.

44 **3.2.4 octlet:** Eight bytes of data.

46 Other terms that have special meanings in the context of your document should also be included here.
47 The numbering scheme is necessary for IEEE documents.

3.3 Unimplemented locations

The capabilities of reserved, ignored, and unused values are carefully defined, to minimize conflicts between current implementations and future definitions.

3.3.1 reserved fields: A set of bits within a data structure that is defined in this specification as reserved, and is not otherwise used. Implementations of this specification shall zero these fields. Future revisions of this specification, however, may define their usage.

3.3.2 ignored location: Selected locations or portions of locations are partially implemented and are defined to be ignored (abbreviated as *ign* or *i*). An ignored value has an affiliated storage element, but the value in the storage elements has no side effect.

3.3.3 reserved location: Some locations or portions of locations are not implemented and are defined to be reserved (abbreviated as *res* or *r*). When a reserved value is written, a zero values shall be assumed; when read, the returned value shall be ignored.

3.3.4 unused location: Selected locations or portions of locations may be not implemented or partially implemented and are defined to be unused (abbreviated as *un* or *u*). For unused locations, the selection between reserved and ignored behaviors is implementation dependent.

3.4 Numerical values

NOTE—The notation text, or its equivalent, is required to explain the notation used throughtout. The subscript definition is preferred within English text, although other notations are used within code.

Decimal, hexadecimal, and binary numbers are used within this document. For clarity, decimal numbers are generally used to represent counts, hexadecimal numbers are used to represent addresses, and binary numbers are used to describe bit patterns within binary fields.

Decimal numbers are represented in their usual 0, 1, 2, ... format. Hexadecimal numbers are represented by a string of one or more hexadecimal (0-9,A-F) digits followed by the subscript 16. Binary numbers are represented by a string of one or more binary (0,1) digits, followed by the subscript 2. Thus the decimal number "26" may also be represented as "1A₁₆" or "11010₂".

An alternative would be to use trailing characters to identify number formats, as in the following paragraph.

Decimal numbers are represented in their usual 0, 1, 2, ... format. Hexadecimal numbers are represented by a string of one or more hexadecimal (0-9,A-F) digits followed by the letter 'h'. Binary numbers are represented by a string of one or more binary (0,1) digits, followed by the letter 'b'. Thus the decimal number "26" may also be represented as "1Ah" or "11010b".

These notatational conventions have one exception: MAC addresses and OUI/EUI values are represented as strings of 8-bit hexadecimal numbers separated by hyphens and without a subscript, as for example "01-80-C2-00-00-15" or "AA-55-11".

03 3.5 Field notation

06 3.5.1 Field names

08 NOTE—All documents should describe their naming conventions, with text similar to the following.

10 This document describes values that are in memory-resident or control-and-status registers. For clarity,
11 distinct capitalization conventions are used when naming different components, as illustrated in Table 3.1.

13 **Table 3.1—Names of constants, registers and fields**

Name	Row	Description
MAX_VALUE	1	A defined constant value
StateMachineName	2	A formal state machine name (if required)
parameter_value	3	A Service Primitive parameter
<i>RoutineName()</i>	4	A subroutine name, when referenced within text.
<i>runCommand</i>	5	A referenced control register.
<i>startCode</i>	6	The <i>startCode</i> field
<i>start</i>	7	The <i>start</i> bit
<i>runCommand.startCode</i>	8	The <i>startCode</i> field within the <i>runCommand</i> register
<i>runCommand.start</i>	9	The <i>start</i> bit within the <i>runCommand</i> register

32 NOTE—If ever applied to a variable name, the italics style should always be applied to that variable
33 name, whether contained in tables, figures, or headings (but not C-code, where Courier is used).

36 NOTE—When lengthy descriptions are necessary, row numbers and following clarifications should
37 supplement the descriptive column, as illustrated (for Table 3.1) in the text below.

39 **Row 1:** Constant values are spelled with capital letters; an underscore separates run-together words.

40 **Row 2:** Formal state machine names (when necessary) do not include blank spaces.

41 **Row 3:** Service primitive parameters include underscores, for consistency with the past and to differentiate
42 these parameters from defined field values.

43 **Row 4:** Subroutine names (within normal text) are distinguished by italics and first-capital-letter. The italics
44 formatting convention applies to the name, not associated special symbols, such as ‘(’ and ‘)’.

45 **Row 5, Row 6, Row 7:** Register names, fields, and bit names start with a lower-case letter; each run-
46 together word starts with a capital letter. Run-together names like *runCommand* are preferred because they
47 are more compact than under-score-separated names (like “*run_command*”).

48 **Row 8, Row 9:** When their register location is unclear or ambiguous, the name of a fields includes the
49 name of the register where that field is located.

3.5.2 Fields within figures 02
03
04

NOTE—Numbering of bits within registers causes problems, due to distinct bit numbering conventions adopted by little-endian and big-endian designers. Bit ordering arguments are not easily resolved by using the bit-transmission order, which can be PHY dependent or ambiguous. Arguments and confusions are best resolved by avoiding the numbering altogether, which also simplifies each illustration. 05
06
07
08
09

The location of fields within registers is specified by the cumulative widths of fields within the register, as illustrated in Figure 3.2. The width of each field (in bits) is implied by bottom-line tick marks; the field name is normally contained within its bounding rectangle. When the field name is smaller than its bounding rectangle, lines associate the field’s name with its location (as illustrated for *error*, *mode*, and *phase* bits). 10
11
12
13
14

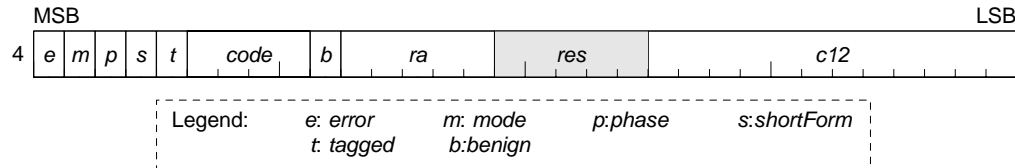


Figure 3.1—Expanded bit-field descriptions 15
16
17
18
19
20
21
22

NOTE—Abbreviations within figures are the only time that special names are spelled differently. Within such figures, the association with the abbreviated and standard name must be specified in the legend. 23
24
25
26
27

3.5.3 Multibyte fields within frames 28
29

Figure 3.2 provides an illustrative example of another possible byte-sequential data representation. These representations are drawn as fields (of arbitrary size) ordered along a vertical axis, with numbers along the left sides of the fields indicating the field sizes in bytes. Fields are drawn contiguously such that the transmission order across fields is from top to bottom. The example shows that *ttl*, *baseControl*, and *da* are 1-, 1- and 6-byte fields, respectively, transmitted in order starting with the *ttl* field first. 30
31
32
33
34
35
36

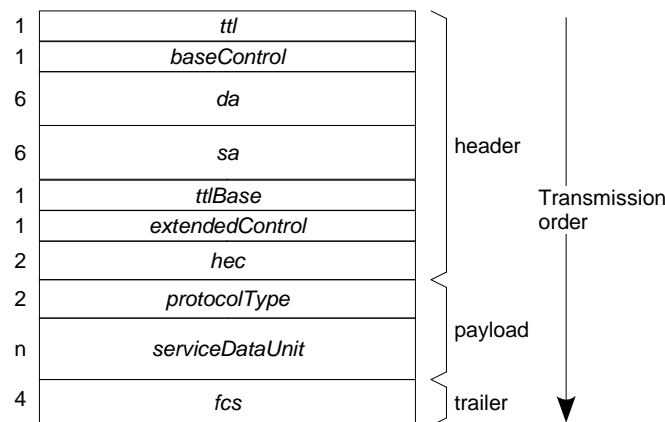


Figure 3.2—Frame format illustrations 37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52

NOTE—The height of the rectangles need not be proportional to the field size, but shall be monotonic with respect to the size of the contained field. Thus, if $SizeOf(b) > SizeOf(a)$, then $Height(b) \geq Height(a)$ where $SizeOf(x)$ and $Height(s)$ represent the byte-count and height associated with rectangle x. 53
54
55
56
57
58
59
60
61

02
03
04 NOTE—The right-side arrow and Transmission order text are useful when describing this notation. In
05 other illustrations, this information would be redundant and distracting, and is therefore not
06 recommended.

09 3.6 C-code notation

10 The behavior of data-transfer command execution is frequently specified by C code, such as Equation 3.1.
11 To differentiate this code from textual descriptions, such C code listings are formatted using a fixed-width
12 Courier font. Similar C-code segments are included within some figures.
13

```
14 // Return maximum of a and b values (3.1)  
15 Max(a,b) {  
16     if (a<b)  
17         return(LT);  
18     if (a>b)  
19         return(GT);  
20     return(EQ);  
21 }
```

22 Since the meaning of many C code operators are not obvious to the casual reader, their meanings are
23 summarized in Table 3.2.

24 **Table 3.2—C code expressions**

Expression	Description
$\sim i$	Bitwise complement of integer i
$i \wedge j$	Bitwise EXOR of integers i and j
$i \& j$	Bitwise AND of integers i and j
$i \ll j$	Left shift of bits in i by value of j
$i * j$	Arithmetic multiplication of integers i and j
$!i$	Logical negation of Boolean value i
$i \&\& j$	Logical AND of Boolean i and j values
$i j$	Logical OR of Boolean i and j values
$i \wedge = j$	Equivalent to $i = i \wedge j$.
$i == j$	Equality test, true if i equals j
$i != j$	Equality test, true if i does not equal j
$i < j$	Inequality test, true if i is less than j
$i > j$	Inequality test, true if i is greater than j

3.7 State machines

NOTE—The following illustrations have been found useful to some IEEE working groups are therefore provided as optional content. This material should be deleted if such state machine specifications are not used within the document.

Flow charts are used throughout this document to illustrate high-level functionality, as illustrated in Figure 3.3. Flow charts are typically affiliated with an exact table-structured specification, as illustrated in Table 3.3.

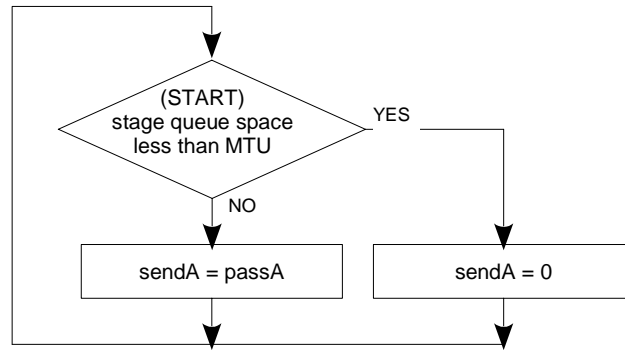


Figure 3.3—Flow-control sendA generation overview

Table 3.3—Flow-control sendA generation details

Current state		Row	Next state	
state	condition		action	state
START	DepthStageBuffer < (sizeStageBuffer-MTU)	1	sendA= 0	START
	—	2	sendA= passA	

Row 1: An MTU of stage-queue storage is required to safely buffer client-supplied frames.

Row 2: Setting *sendA* to *passA* enables transmissions when stage-buffer space is available.

NOTE—The first state within all state machines should be distinctive and uniformly defined. The name START is recommended for this purpose.

3.8 Field notations

3.8.1 Use of italics

All field names or variable names (such as *level* or *myMacAddress*), and sub-fields within variables (such as *thisState.level*) are italicized within text, figures and tables, to avoid confusion between such names and similarly spelled words without special meanings. A variable or field name that is used in a subclause heading or a figure or table caption is also italicized. Variable or field names are not italicized within C code, however, since their special meaning is implied by their context. Names used as nouns (e.g., *subclassA0*) are also not italicized.

02
03 **3.8.2 Field conventions**

04 This document describes values that are packetized or MAC-resident, such as those illustrated in Table 3.4.

05
06 **Table 3.4—Names of fields and sub-fields**

07
08

Name	Description
<i>newCRC</i>	Field within a register or frame
<i>thisState.level</i>	Sub-field within field <i>thisState</i>
<i>thatState.rateC[n].c</i>	Sub-field within array element <i>rateC[n]</i>

09
10
11
12
13
14

15 Run-together names (e.g., *thisState*) are used for fields because of their compactness when compared to
16 equivalent underscore-separated names (e.g., *this_state*). The use of multiword names with spaces (e.g.,
17 “This State”) is avoided, to avoid confusion between commonly used capitalized key words and the
18 capitalized word used at the start of each sentence.

19
20 A sub-field of a field is referenced by suffixing the field name with the sub-field name, separated by a
21 period. For example, *thisState.level* refers to the sub-field level of the field *thisState*. This notation can be
22 continued in order to represent sub-fields of sub-fields (e.g., *thisState.level.next* is interpreted to mean the
23 sub-field *next* of the sub-field *level* of the field *thisState*).

24 Unless specifically specified otherwise, reserved fields are reserved for the purpose of allowing extended
25 features to be defined in future revisions of this standard. For devices conforming to this version of this
26 standard, nonzero reserved fields are not generated; values within reserved fields (whether zero or nonzero)
27 are to be ignored.

28
29
30 **3.8.3 Field value conventions**

31 This document describes values of fields. For clarity, names can be associated with each of these defined
32 values, as illustrated in Table 3.5. A symbolic name, consisting of upper case letters with underscore
33 separators, allows other portions of this document to reference the value by its symbolic name, rather than a
34 numerical value.

35
36 **Table 3.5—wrap field values**

37

Value	Name	Description
0	WRAP_AVOID	Frame is discarded at the wrap point
1	WRAP_ALLOW	Frame passes through wrap points.
2,3	—	Reserved

38
39
40
41
42
43
44

45 Unless otherwise specified, reserved values are reserved for the purpose of allowing extended features to be
46 defined in future revisions of this standard. Devices conforming to this version of this standard do not
47 generate reserved values for fields, and process fields containing reserved values as though the field values
48 were not supported. The intent is to ensure default behaviors for future-specified features.

49
50 A field value of TRUE shall always be interpreted as being equivalent to a numeric value of 1 (one), unless
51 otherwise indicated. A field value of FALSE shall always be interpreted as being equivalent to a numeric
52 value of 0 (zero), unless otherwise indicated.

3.9 Informative notes

Informative notes are used in this standard to provide guidance to implementers and also to supply useful background material. Such notes never contain normative information, and implementers are not required to adhere to any of their provisions. An example of such a note follows.

NOTE—This is an example of an informative note.

02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63

03 4. Abbreviations and acronyms

04 This document contains the following abbreviations and acronyms:

07 NOTE—Editors should update the following list of acronyms based on those used within their drafts.
08 ***OBSERVE: The description is only capitalized when it corresponds to a proper noun.***

- 10 CSR control and status register
- 11 IEEE Institute of Electrical and Electronics Engineers, Inc.
- 12 RAM random access memory
- 13 ROM read only memory
- 14 TCAM ternary content addressable memory.

5. Using the templates

5.1 Formatting the final draft

Unfortunately, OpenOffice has minimal support for the concatenation of clauses and annex. To cope with this constraint, the annex styles have been customized from numbered fields and are therefore distinct.

5.2 Corporate document numbers

All corporate documents should have a distinctive identifier. In the absence of corporate specified identifiers, a default identifier should be used, as illustrated below:

CompanyDVJ2002Jul02

The parameters within this identifier include:

Company—To identify this as a Company authored document.

DVJ—A distinct identifier that identifies the author.

2002Jul02—The year, month, and day when the first draft was written.

02
 03 **5.3 Paragraph styles**

04
 05
 06 **5.3.1 Baseline paragraph styles**

07 The specific paragraph styles that this template provides are listed in Table 5.1.

08
 09 **Table 5.1—Paragraph styles**

10
 11

Style name	Row	Description	Reference
Bibliography	1	For enumerating informative Bibliography references.	Annex A
Body	2	Basic text throughout.	9.1
BodyCenter	3	Centered text.	
Code	4	Courier 132-column code within the annex	Annex H
BodyTight	5	Acronym & abbreviation lists.	9.1
Copyright	6	The copyright notice on the front page footer.	page 1
Definition2	7	Second-level text for definition-inclusion purposes.	8
Definition2Like	8	Second-level text for definition-exclusion purposes.	8
Definition3	9	Third-level text for definition-inclusion purposes.	7.2
Definition3Like	10	Third-level text for definition-exclusion purposes.	7.2
Definition4	11	Fourth-level text for definition-inclusion purposes.	7.1.2
Definition4Like	12	Fourth-level text for definition-exclusion purposes.	7.1.2
Definition5	13	Fourth-level text for definition-inclusion purposes.	7.1.1.1
Definition5Like	14	Fifth-level text for definition-exclusion purposes.	7.1.1.1
EditorNote	15	Editorial note, work-in-progress comments	9.1
EquationMath	16	A numbered equation	9.5.3
EquationCode	17	A numbered C-code equation.	9.6.1
FigureTitle	18	Trailer placed after the figure	10
Heading 1	19	Clause heading	5.3.1
Heading 2	20	Second level subclause	6.1
Heading 3	21	Third level subclause	6.1.1
Heading 4	22	Fourth level subclause	6.1.1.1
Heading 5	23	Fifth level subclause (the last recommended level)	6.1.1.1.1
Heading 6	24	Sixth level subclause	6.1.1.1.1.1
Heading 7	25	Sixth level subclause (the last supported level)	6.1.1.1.1.1.1
List1	26	First-level list: 1) format	9.4
List1More	27	First-level list continued	
List1Dash	28	First-level dash list	
List2	29	Second-level list: a) format	9.4
List2More	30	Second-level list continued	

12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31
 32
 33
 34
 35
 36
 37
 38
 39
 40
 41
 42
 43
 44
 45
 46
 47
 48
 49
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
 63

Style name	Row	Description	Reference
List2Dash	31	Second-level dash list	
List3	32	Third-level list: i) format	9.4
List3More	33	Third-level list continued	
List3Dash	34	Third-level dash list	
Note	35	Informational content	9.1.2
Reference	36	Normative references.	Clause 2
TableCellCenter	37	Center-justified table cell	9.5.1
TableCellCode	38	Centered C-code within a table cell	3.6
TableCellLeft	39	Left-justified table cell	9.5.1
TableCellRow	40	Center justified table-row number	9.5.1
Table Footnote	41	Use this style for footnotes appended to a table.	
TableHeading	42	Used for the headings of tables.	9.5.1
TableTitle	43	Header line placed before the table	9.5.1
Title	44	4-point title	9.6.2
Title10	45	Abstract and keywords text in the title page.	

02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63

02
03 **5.3.2 Character styles (such as α)**

04 The specific paragraph styles that this template provides are listed in Table 5.2.

05
06
07 **Table 5.2—Character styles**

Style name	Row	Description
Bold	1	Bold text modifier.
Emphasis	2	Italics text modifier (assumed by FrameMaker cross references).
Hidden	3	White very narrow font, for complete hiding of reference material.
Invisible	4	White font, for print hiding of reference material.
NameItalic	5	Italic non-hyphenated (no-language) text
NamePlain	6	Plain non-hyphenated (no-language) text
Subscript	7	Subscripted text
Superscript	8	Superscripted text .
Symbol	9	Greek symbols, when included in titles

08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25 **Row 1:** The *Bold* character font applies a bold-font emphasis. This style is invoked within {*Definition2*, ..., *Definition5*} styles, to create a bold paragraph number when the default text is plain.

26
27 **Row 2:** The *Emphasis* character font applies an italics-font emphasis to hyphenate-allowed words.

28 **Row 3:** The *Hidden* character font narrows the font by 99% and applies white coloring for invisibility.

29 **Row 4:** The *Invisible* character font applies white coloring for invisibility.

30 **Row 5:** The *NameItalic* character font applies to non-hyphenated words, such as *thisone*.

31 **Row 6:** The *NamePlain* character font applies to non-hyphenated words, such as *thisone*.

32 **Row 7:** The *Subscript* character font applies to subscript characters, such as *n* in y_n .

33 **Row 8:** The *Superscript* character font applies to superscript characters, such as *n* in y^n

34 **Row 9:** The *Symbol* character font applies to symbol-font characters.

5.4 Style tricks

A couple of tricks are specialized for use by row and section cross references.

5.4.1 Subclause cross references

There is no easy apparent way to cross-reference numbered clauses and subclauses in an automatic way. While insertion of cross references or bookmarks is possible, this would imply management of yet-another set of names or numbers, an unacceptable overhead.

A viable alternative is to insert a tag after the subclause text, allowing that tag to be cross-referenced elsewhere in the document. To avoid hide the tag from the printer and table-of-contents, the tag is placed within a frame. That frame is inserted using the following parameters:

Insert
Frame
Width: 0.10
Height: 0.12
Anchor: As character

With the curser immediately before the frame's paragraph marker, a tag is inserted, as follows:

Insert
Fields
Other
Variables
Type: Number range
Selection: Subclause
Format: Arabic (1 2 3)
Level: 10
Separator: .
Value: 0

The character style "Hidden" is then applied, to make the printed tag small and white (invisible).

A much easier way is to simply cut-and-paste the small square box at the end of the following subclause, placing this immediately in front of the referenced subclause.

5.4.1.1.1.1.1

02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63

03 5.4.2 Row cross references restarts

04 The cross-referencing of row numbers requires a few special adaptations, since cross-references to the same
05 names value (such as “1”) cannot be resolved within the cross-reference listing, where only a single '1' value
06 appears. And, of course, row numbers restart at the beginning of each table, rather than once per clause. The
07 following tricks overcome these limitations.

08
09 First, the row number should be reset. This involves the insertion of a special tag at the end of the table title.
10 The insertion of the frame is described in 5.4.1, but its contents are different. With the cursor immediately
11 before the frame's paragraph marker, a tag is inserted, as follows:

```
12     Insert
13         Fields
14             Other
15                 Variables
16                     Type: Number range
17                     Selection: TableRow
18                     Format: Arabic (1 2 3)
19                     Level: None
20                     Separator:
21                     Value: 0
```

22
23 To eliminate this image from the printed copy, the *Hidden* character style is then applied.

26 5.4.3 Row cross references tags

27 A row cell entry consists of a row count field, generated as follows:

```
28     Insert
29         Fields
30             Other
31                 Variables
32                     Type: Number range
33                     Selection: TableRow
34                     Format: Arabic (1 2 3)
35                     Level: None
36                     Separator: (blank)
37                     Value: (blank)
```

38
39 To support remote as well as local cross references, the visible number is preceded by “a.b-” and followed
40 by “-xx”. The leading “a.b-” characters attaches a table-number identification to the row number, which
41 populates the cross-reference listing with non-aliased names. The trailing “-xx” characters assist in
42 centering of the entry. The “a.b” portion of the prefix is a duplicate of the table number, generated as
43 follows:

```
44     Insert
45         Fields
46             Other
47                 Variables
48                     Type: Number range
49                     Selection: Table
50                     Format: Arabic (1 2 3)
51                     Level: 1
52                     Separator: .
53                     Value: Table
```

54
55 To eliminate this image from the printed copy, the Hidden “Character Style” is then applied; to reduce the
56 bounding-box size, the paragraph style of “Spacing” was applied. The white color of the font eliminates the
57 printed image; the distinct paragraph limits the height of the inserted clear-count entry.

58
59 To avoid these complexities, editors can simply cut-and-paste these portions of Table 5.3.

Table 5.3—Table-Row formatting example

Name	Row	Description
MAX_VALUE	1	A defined constant value
StateMachineName	2	A formal state machine name (if required)
<i>runCommand.start</i>	3	The <i>start</i> bit within the <i>runCommand</i> register

Row 1: This is how the first row would be cross-referenced within immediately following text. Within this reference, the “Numbering” format was selected.

This is how the StateMachine description (TableRow 5.3-2 could be cross-referenced within remote text. Within this reference, the “Category and Number” format was selected.

This is how the StateMachine description (Table 5.3, Row 3) could be cross-referenced within remote text. This style is apparently preferred by the IEEE editorial management staff. Two distinct cross-references are required: a Table reference and a TableRow reference.

02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63

03 **6. Clause and subclause headings**

04
05 The previous *Clause and subclause headings* text was created using the *Heading 1* paragraph style and typ-
06 ing the text *Clause and subclause headings*. Each *Heading 1* heading is preconfigured to start on a new
07 page; standards may modify this style to eliminate the page break or force an odd-page break, depending on
08 the document length and editorial preferences.

10 **6.1 Subclause-2 heading**

11
12 The previous *Subclause-2 heading* text was created using the *Heading 2* paragraph style and typing the text
13 *Subclause-2 heading*. The *Heading 2* heading is configured to force before and after spacing, but no page
14 break.

16 **6.1.1 Subclause-3 heading**

17
18 The previous *Subclause-3 heading* text was created using the *Heading 3* paragraph style and typing the text
19 *Subclause-3 heading*. The *Heading 3* heading is configured to force before and after spacing, but no page
20 break.

22 **6.1.1.1 Subclause-4 heading**

23
24 The previous *Subclause-4 heading* text was created using the *Heading 4* paragraph style and typing the text
25 *Subclause-4 heading*. The *Heading 4* heading is configured to force before and after spacing, but no page
26 break.

28 **6.1.1.1.1 Subclause-5 heading**

29
30 The previous *Subclause-5 heading* text was created using the *Heading 5* paragraph style and typing the text
31 *Subclause-5 heading*. The *Heading 5* heading is configured to force before and after spacing, but no page
32 break.

33
34 This is the deepest level allowed by the IEEE! Thus, your should revise any IEEE document that requires to
35 the use of any level-6 or lower levels. This formatting style is intended to force conformance to good
36 document-style conventions, rather than providing the author with an unlimited range of ill-conceived
37 nesting depths.

38
39
40 **NOTE**—This is the deepest level allowed by the IEEE Style Manual. Thus, you should revise any IEEE
41 document that requires the use of level-6 or lower levels. This formatting style is intended to force
42 conformance to good document-style conventions, rather than providing the author with an unlimited
43 range of ill-conceived nesting depths.

44 **6.1.1.1.1.1 Subclause-6 heading**

45
46 The previous *Subclause-6 heading* text was created using the *Heading 6* paragraph style and typing the text
47 *Subclause-6 heading*. The *Heading 6* heading is configured to force before and after spacing, but no page
48 break.

50 **6.1.1.1.1.1.1 Subclause-7 heading**

51
52 The previous *Subclause-7 heading* text was created using the *Heading 7* paragraph style and typing the text
53 *Subclause-7 heading*. The *Heading 7* heading is configured to force before and after spacing, but no page
54 break.

Note this is the deepest level that is supported by these formats! Thus, you should revise any document that requires to the use of any level-7 or lower levels. This formatting constraint results from having 9 number levels, with three of these reserved for figure-title, table-title, and equation-title purposes.

02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63

03 7. Definitions

04
05 Paragraphs of definitions may be distinctively numbered to facilitate their cross referencing by other parts
06 of a document. This should only be done at the lowest level, to avoid discontinuities in the heading
07 numbering.

09 7.1 Leading second-level indent

12 7.1.1 Leading third-level heading

15 7.1.1.1 Definition5 heading

16 Within fourth-level subclauses, a fifth-level field distinctively numbers paragraphs, as illustrated in
17 7.1.1.1.1 and 7.1.1.1.2. Although the number is provided by the Definition field, use of distinct definition
18 paragraph styles is recommended for post-processing purposes.

19
20 **7.1.1.1.1 definedValue5a:** The *Definition5* paragraph style was applied to this paragraph. This definition
21 would be automatically included in the IEEE dictionary.

22
23 **7.1.1.1.2 definedValue5a:** The *Definition5Like* paragraph style was applied to this paragraph. This
24 definition would be automatically included in the IEEE dictionary.

26 7.1.2 Definition4 heading

27 Within third-level subclauses, a fourth-level field distinctively numbers paragraphs, as illustrated in 7.1.2.1
28 and 7.1.2.2.

29
30 **7.1.2.1 definedValue4a:** The *Definition4* paragraph style was applied to this paragraph. This definition
31 would be automatically included in the IEEE dictionary.

32
33 **7.1.2.2 definedValue4a:** The *Definition4Like* paragraph style was applied to this paragraph. This definition
34 would be automatically included in the IEEE dictionary.

36 7.2 Definition3 headings

37
38 Within second-level subclauses, a third-level field distinctively numbers paragraphs, as illustrated in 7.2.1
39 and 7.2.2.

40
41 **7.2.1 definedValue3a:** The *Definition3* paragraph style was applied to this paragraph. This definition would
42 be automatically included in the IEEE dictionary.

43
44 **7.2.2 definedValue3a:** The *Definition3Like* paragraph style was applied to this paragraph. This definition
45 would be automatically included in the IEEE dictionary.

8. Highest level definitions

Within clauses, a second-level field distinctively numbers paragraphs, as illustrated in 8.1 and 8.2.

8.1 definedValue1a: The *Definition2* paragraph style was applied to this paragraph. This definition would be automatically included in the IEEE dictionary.

8.2 definedValue1b: The *DefinitionLike* paragraph style was applied to this paragraph. This definition would not be included in the IEEE dictionary.

02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63

03 9. Formatting styles

04
05 The following subclauses contain information for understanding the use of these templates. Though not
06 comprehensive, they provide guidance for preparation of formal documents.

08 9.1 Notes

11 9.1.1 Editorial notes

13
14 **Editor’s note:** To be removed prior to publication.
15 Editor notes may be added to identify editing work that remains to be done.

17
18 **NOTE**—Comments within this document provide out-of-band explanations without inserting paragraph
19 styles that would not normally be provided.

20 The *EditorNote* style is associated with the editorial note above. These are intended to be included within
21 drafts for the purpose of communicating information between the editor and readers. Information may
22 clarify the intent of recent changes, or speculate on additional text needed to complete the document.

25 9.1.2 Informative notes

26 Informative notes can be inserted throughout the document, as illustrated by the following two-paragraph
27 note. Notes are *not* official parts of the specification—they are merely informative. Contrast this with
28 informative annexes, which are set off in separate sections of the document. This is the reason for setting
29 notes in a different font size.

30
31 **NOTE**—The *Note* style (used in this paragraph) is used on note paragraphs that follow the note style, although notes
32 are rarely longer than one paragraph.

33 The *Note* style can also be used without the leading “NOTE—“ typed characters, for continued notes, although notes
34 are rarely longer than one paragraph.

9.2 Page breaks

Page breaks can be manually inserted by using the Insert/Break pulldown menu, as was done before the preceding subclause heading. This convention has been found to be convenient and avoids the need to override default paragraph styles.

9.3 Body styles

9.3.1 Body

The *Body* style is associated with the basic text blocks in this document, including this example.

The *Body* style is 10-point Times New Roman, black, with right and left justification. Times New Roman is the serif font that is used primarily throughout this document. Exceptions are Arial and Courier fonts: Arial is used for headings and figure/table/equation titles; Courier is used for C code.

9.3.2 *BodyTight*

The *BodyTight* style is similar, but less paragraph-to-paragraph spacing is specified. Thus, such styles are appropriate for acronym listings, as listed below although normally utilized within 3.7.

- NSE network search engine
- RPR resilient packet ring
- SRAM static random access memory

9.3.3 *BodyCenter*

The *BodyCenter* style is used for centered text, such as written below.

This style is provided for use on IEEE front-cover pages.

Note that centering should be done using a distinct style instead of by using a style override, so that the centering property will survive automatic reapplication or importation of the styles by subsequent editors.

9.3.4 *BodyBlock*

The *BodyBlock* style forces text to remain on one page, rather than split across pages, as used below.

```
MA_DATA.request
(
    parameter1,
    parameter2,
    parameter3
)
```

03 9.4 Lists

04
05 In general, lists are used to display information that does not require explanation or that is offered by way of
06 explanation. Lists that are written in phrases should not allow each item to end in a period or other closing
07 punctuation; rather, only the last item of such a list should allow a period at the end. If however each item in
08 the list is a sentence or a series of sentences, use closing punctuation for each item. Also, it is not necessary
09 to repeat subject information in each list item, nor is it necessary to precede every list with a paragraph
10 ending in a semicolon. It is best to end the paragraph above a list with a period (especially with sentence
11 lists), or with no punctuation at all (with phrase lists).

12 The first *BodyCount* 2nd level indent generated this text block.

13 The next *BodyCount* 2nd level indent generated this text block.

14 a) The first *BodyCount* 3rd level indent generated this auto numbered list.

15 b) The next *BodyCount* 3rd level indent generated this auto numbered list.

16 The first *BodyCount* 4th level indent generated this auto numbered list.

17 The next *BodyCount* 4th level indent generated this auto numbered list.

18 — The first *BodyCount* 5th level indent generated this auto numbered list.

19 — The next *BodyCount* 5th level indent generated this auto numbered list.

20 1) The first *BodyCount* 6th level indent generated this auto numbered list.

21 2) The next *BodyCount* 6th level indent generated this auto numbered list.

22 The first *BodyCount* 7th level indent generated this auto numbered list.

23 The next *BodyCount* 7th level indent generated this auto numbered list.

24 — The first *BodyCount* 8th level indent generated this auto numbered list.

25 — The next *BodyCount* 8th level indent generated this auto numbered list.

26 i) The first *BodyCount* 9th level indent generated this auto numbered list.

27 ii) The next *BodyCount* 9th level indent generated this auto numbered list.

28 The first *BodyCount* 10th level indent generated this auto numbered list.

29 The next *BodyCount* 10th level indent generated this auto numbered list.

30 Note that the same indentation spacing is used for all number styles, but the dash versions use the *em* dash,
31 instead of numerated values. The ISO/IEC specification doesn't allow the use of bulleted lists, so dashes are
32 used at all levels.

33 9.5 Tables

34 Tables are intended to summarize the use of multiple items, not to describe them in detail. A row-reference
35 column can be added to elaborate specific rows of the table. The following subclauses describe the methods
36 used to create tables.

37 9.5.1 Tables cells

38 You should insert tables by cut-and-pasting a previously created table, since the author does not yet
39 understand how to specify table formats. For example, you could start with a three-columns and three-rows
40 table of Table 9.1.

41 **Table 9.1—Example table**

Heading A	Heading B	Description
ValueA	NameA	DescriptionA
ValueB	NameB	DescriptionB
ValueC	NameC	DescriptionC

Use the **Table** menu to modify the number of rows and columns, or to straddle rows/columns as desired. 03

The *TableTitle* style was applied to the text above the table, although the field within is called *Table*. 04

The *TableHeading* style is used on the top row, which sets the style as {bold and centered}. The *TableCellCenter* style is used to center-justify table entries, as illustrated in the left columns. The *TableCellLeft* style is used to left-justify table entries, as illustrated in the right column. 06
07
08
09

9.5.2 Tables row labels 11

Tables should contain small items, not paragraphs of explanatory text. A row-number cross-reference can be used to cross-reference a row with detailed following comments, as illustrated in Table 9.2. The *TableCellRow* style is used to establish the row count within the Row headed column. 12
13
14
15

Table 9.2—Row-numbered table 16

Heading A	Heading B	Row	Description
ValueA	NameA	1	DescriptionA
ValueB	NameB	2	DescriptionB
ValueC	NameC	3	DescriptionC

Row 1: A detailed description of a row can be placed after the table. This has the benefit of maintaining a concise table, while allowing each row to be described in detail. 17
18
19
20
21

Row 2, Row 3: Multiple rows can be described together, by placing both cross-referenced before the combined description, as was done here. 22
23
24
25
26
27

9.5.3 Equations 31

An equation involves the use of the *EquationMath* format, as illustrated in Equation 9.1. The equation may be left-justified, indented, or centered based on the number of left-side tabs. A rightside autonumber specifier provides the right-side equation number, not the paragraph style itself. 32
33
34
35
36

$$v = 2^7 \left(-b_0 + \sum_{i=1}^l b_i 2^{-i} \right) \tag{9.1}$$

An unnumbered equation involves the use of the same *EquationMathMore* format, but without the rightside autonumber specifier, as illustrated below. 37
38
39
40
41
42

$$v = 2^7 \left(-b_0 + \sum_{i=1}^l b_i 2^{-i} \right)$$

02
03 **9.6 C-code listings**
04

05
06 **9.6.1 Shorter in-line C code**

07 A few lines of C code can be expressed as an equation, as illustrated by 9.2. The *EquationCode* style is used
08 on the first line, to specify the proper C-Code font and provide the numerical label. The
09 *EquationCodeMore*

```
11     crcCheck= CrcStep32(crcValue);                               (9.2)  
12     if (crcCheck!=crcValid)  
13         error|= 1;  
14     crcCheck= CrcStep32(crcValue);
```

15
16 **9.6.2 Lengthy C code**
17

18 Lengthy C code (which is typically defined as code listings covering more than ½ page) should be placed
19 in an annex, such as Annex I. An annex can support a wider 132-character width while allowing the code to
20 be more easily extracted by the reader.

10. Constructing OpenOffice drawings

10.1 Figures

10.1.1 Figure creation

The simplest way to create a new figure is to cut-and-paste an existing figure, picking a figure that has similar graphics elements. If the original figure is not grouped, you will need to use the arrow selection tool to select all drawing elements before copying. Figures with useful register templates are provided in the annex and remainder of this subclause.

Since copies often lose the shading details of the original, shaded colors should be limited to three: white, light gray, and black. When black shading is used, the text color should be white. As examples, in Figure 10.2, the *res* field is shaded gray.

10.1.2 Snap grid usage

When drawing graphics, we STRONGLY suggest that the snap grid be used, so that Graphics can be more easily modified at a later date. A snap-grid value of 3 points (1/24 of an inch) is recommended, for compatibility with simple 72 dots-per-inch software and modern 600 dots-per-inch printers.

If an old drawing is no longer on the snap grid, bring it back before continuing with the editing session: select everything within the image, being careful to select a rectangle (or other known-to-be-aligned-with-grid object) last. Then move the selected objects to the desired snap-grid location.

The reference point within an inserted drawing is the TBD corner. To maintain the snap-grid, this reference point should never be changed. Thus, expansion (or contraction, not shown) of a OpenOffice drawing should be made by using a click-and-slide motion on the TBD1, TBD2, or TBD3 handles only, as illustrated in Figure 10.1.

TBD

Figure 10.1—OpenOffice expansion and contraction restrictions

10.1.3 Register layouts

The location of fields within registers is specified by the cumulative widths of fields within the register, as illustrated in Figure 10.2. The width of each field (in bits) is implied by bottom-line tick marks; the field name is normally contained within its bounding rectangle.

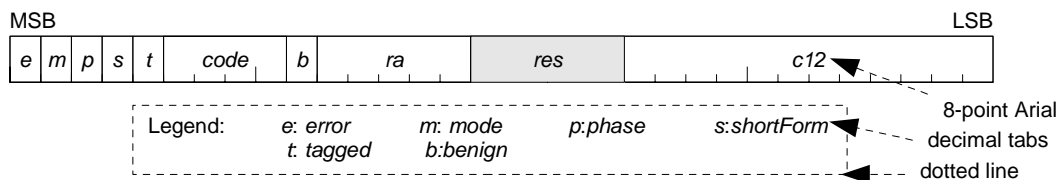


Figure 10.2—Expanded bit-field descriptions

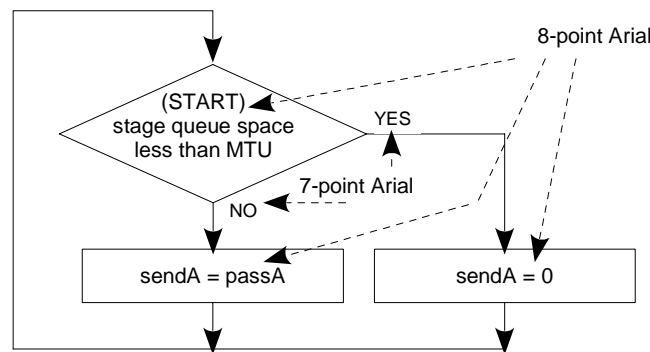
Since copies often lose the shading details of the original, shaded colors should be limited to three: white, gray, and black. When black shading is used, the text color should be white.

04 NOTE—The fields of Figure 10.2 were planned to illustrate several of the larger byte-boundary tick
05 marks. Depending on the field alignments, these are oftentimes hidden by the field's bounding box.
06

07 Each field name that would extend beyond its bounding box is abbreviated; a legend associates the local
08 abbreviation with the formal field name. The legend entries are ordered, in that a left-to-right scan of the
09 fields corresponds to a left-to-right-then-top-to-bottom scan of the legend entries. A space should be
10 included after the colon. The dotted line around the legend is a distinct graphics rectangle, which can be
11 sized appropriately.
12

14 10.1.4 Flow-chart styles

15 Within flow charts, an Arial 8-point centered font is typically applied to text within conditional-branch
16 components, as illustrated in Figure 10.3. The YES and NO constants are larger and would be distracting if
17 done with an 8-point Arial font; a more compact 7-point Arial font is therefore preferred.
18



32 **Figure 10.3—Font sizes within flow-chart illustrations**

Annexes

Annex A (informative) Bibliography

NOTE—The Annex document starts with an Annexes label, followed by a Heading 1 style. A manual override is necessary to avoid the normal paragraph break before the Heading 1 paragraph.

NOTE—The first annex is normally the informative bibliography, an example of which follows.

The annex always starts with the Annexes style followed by the aHeading1First, aHeading1Text, and aHeading1Title styles. The first annex is always the bibliography, which lists informational references that provide useful background information for understanding the specification.

The most recent editions of the following texts are recommended by the IEEE as guides on points of editorial style and usage:

[B1] *IEC Multilingual Dictionary of Electricity, Electronics, and Telecommunications*, Amsterdam: Elsevier Science Publishers.

[B2] *IEEE-SA Standards Board Bylaws*, New York: Institute of Electrical and Electronics Engineers, Inc.

[B3] *IEEE-SA Standards Companion*, New York: Institute of Electrical and Electronics Engineers, Inc.

[B4] *IEEE-SA Standards Operations Manual*, New York: Institute of Electrical and Electronics Engineers, Inc.

[B5] IEEE100, *The Authoritative Dictionary of IEEE Standards Terms*, Seventh Edition, New York, Institute of Electrical and Electronics Engineers, Inc.

[B6] Miller, C., and Swift, K. *The Handbook of Nonsexist Writing*. New York: Harper Collins.

[B7] *Webster’s New Collegiate Dictionary*. Springfield, MA: Merriam-Webster, Inc.

A.1 Bibliography references

The following is a list of on-line locations that are useful resources for creating a bibliography that contains multiple types of references. This list is by no means comprehensive, but it will serve to answer most questions about the creation and development of bibliographies, and will in some cases provide pointers to other sites of interest.

[B8] <http://www.spaceless.com/WWWVL/>¹

[B9] <http://www.lmu.ac.uk/lss/lis/docs/Harvard/bib.htm>²

[B10] <http://www.lmu.ac.uk/lss/lis/docs/Harvard/types.htm>³

[B11] <http://standards.ieee.org/catalog/olis/index.html>⁴

¹ The Electronic References & Scholarly Citations of Internet Sources

² The Harvard Style of Referencing, “Cite References in a Bibliography”

³ The Harvard Style of Referencing, “Source Types and Examples”

⁴ The Institute of Electrical and Electronics Engineers Standards Online.

02
03 **A.2 Grammar references**
04

05 The following is a list of resources that provide information and pointers for further research into the
06 intricacies of the English language. Most common questions may be answered in the FAQ sections of these
07 online sites; the more esoteric points of interest may be resolved with only a little looking around.

08 [B12] <http://ccc.commnet.edu/grammar/>⁵
09

10 [B13] <http://www.ohiou.edu/esl/english/index.html>⁶

11 [B14] <http://englishplus.com/grammar/>⁷
12

13 [B15] <http://www.edunet.com/english/grammar/index.cfm>⁸
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55

56 ⁵Guide to Grammar and Writing

57 ⁶Ohio State University English as a Second Language, "Resources for English Language and Culture"

58 ⁷English Grammar Slammer (n.b.: although this site is for a particular product, it provides a great deal of information
59 and help in the area of English language and grammar)

60 ⁸Education Net: Language, "The Online English Grammar."
61

Annex B (informative) Annex and subannex

B.1 Subannex-1 header

An annex begins with the *aHeading1* paragraph style, which serves as the header for the annex.

Note the presence of special hidden characters on the leftside of the “Annex B” and “Annex and subannex” lines. On the “Annex B” line, these characters are responsible for initializing field values that could be used within the following annex. On the “Annex and subannex” line, these characters are included in the table of contents. Editors should cut-an-paste the same fields within additional annexes.

As is true for clauses, five heading styles are supported in annexes – *aHeading2*, *aHeading3*, *aHeading4*, and *aHeading5*. Only *aHeading1Title*, *aHeading2*, and *aHeading3* normally appear in the table of contents.

B.2 Subannex-2 header

The previous heading applied the *aHeading2* paragraph style to the *Subannex-2 header* text. The *aHeading2* heading forces before and after spacing, but no page break.

B.2.1 Subannex-3 header

The previous heading applied the *aHeading3* paragraph style to the *Subannex-3 header* text. The *aHeading3* heading forces before and after spacing, but no page break.

B.2.1.1 Subannex-4 header

The previous heading applied the *aHeading4* paragraph style to the *Subannex-4 header* text. The *aHeading4* heading forces before and after spacing, but no page break.

B.2.1.1.1 Subannex-5 header

The previous heading applied the *aHeading5* paragraph style to the *Subannex-5 header* text. The *aHeading5* heading forces before and after spacing, but no page break.

This is the deepest level allowed by the IEEE! Thus, your should revise any IEEE document that requires to the use of any level-6 or lower levels. This formatting style is intended to force conformance to good document-style conventions, rather than providing the author with an unlimited range of ill-conceived nesting depths.

B.2.1.1.1.1 Subannex-6 header

The previous *Subannex-6 heading* text was created using the *aHeading6* paragraph style and typing the text *Subannex-6 heading*. The *aHeading6* heading is configured to force before and after spacing, but no page break.

B.2.1.1.1.1.1 Subannex-7 header

The previous *Subannex-7 heading* text was created using the *aHeading7* paragraph style and typing the text *Subannex-7 heading*. The *aHeading7* heading is configured to force before and after spacing, but no page break.

00
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63

02
03 Note this is the deepest level that is supported by these formats! Thus, your should revise any document that
04 requires to the use of any level-7 or lower levels. This formatting constraint results from having 9 number
05 levels, with three of these reserved for figure-title, table-title, and equation-title purposes.

06 07 08 **B.3 Annex definitions**

09 10 11 **B.3.1 Subannex-3 header**

12 The previous heading applied the *Annex3* paragraph style to the *Subannex-3 header* text. The *Annex3*
13 heading forces before and after spacing, but no page break.

14 15 16 **B.3.1.1 Subannex-4 header**

17 Within fourth-level subclauses, a fifth-level field distinctively numbers paragraphs, as illustrated in
18 B.3.1.1.1 and B.3.1.1.2.

19
20 **B.3.1.1.1 DefinedValue5a:** The *Definition5* format was applied to this paragraph. This definition would be
21 automatically included in the IEEE dictionary.

22
23 **B.3.1.1.2 DefinedValue5a:** The *Definition5Like* formats was applied to this paragraph. This definition
24 would be not be included in the IEEE dictionary.

25 26 27 **B.3.2 Subannex-3 header**

28 Within third-level subclauses, a fourth-level field distinctively numbers paragraphs, as illustrated in B.3.3.1
29 and B.3.4.2.

30
31 **B.3.3.1 DefinedValue4a:** The *Definition4* format with a Definitions caption was applied to this paragraph.
32 This definition would be automatically included in the IEEE dictionary.

33
34 **B.3.4.2 DefinedValue4a:** The *Definition4Like* format with a Definitions caption was applied to this
35 paragraph. This definition would not be included in the IEEE dictionary.

36 37 38 **B.4 Definition2 headings**

39 Within second-level subclauses, a third-level field distinctively numbers paragraphs, as illustrated in B.4.1
40 and B.4.2.

41
42 **B.4.1 definedValue3a:** The *Definition3* formats were applied to this paragraph. This definition would be
43 automatically included in the IEEE dictionary.

44
45 **B.4.2 definedValue3b:** The *Definition3Like* formats were applied to this paragraph. This definition would
46 not be included in the IEEE dictionary.

Annex C
(informative)
Highest level definitions

Within annexes, a second-level field distinctively numbers paragraphs, as illustrated in C.1 and C.2.

C.1 definedValue1a: The *Definition2* paragraph style was applied to this paragraph. This definition would be automatically included in the IEEE dictionary.

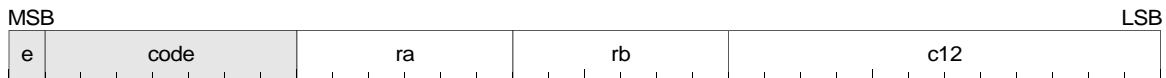
C.2 definedValue1b: The *Definition2Like* paragraph style was applied to this paragraph. This definition would not be included in the IEEE dictionary.

02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63

03 Annex D 04 (normative) 05 Numbered annex styles 06 07 08

09 D.1 Annex figures 10

11 Within the annex, a distinct figure autonumber coding is used, as illustrated in Figure D.1. The distinct style
12 allows the coupling of figure and annex heading styles. Although the caption coding is different, cross-
13 referencing is done in a similar fashion.
14



15
16
17
18
19 **Figure D.1—Instruction illustration**

20 21 22 D.2 Annex tables 23

24 Within the annex, the tables are done in the same fashion. However, the distinct outline numbering produces
25 different results, as illustrated in Table D.1.
26

27 **Table D.1—Names of command, status, and CSR values**

Name	Row	Description
<i>Mover.Csr.control</i>	1	The mover’s control register.
<i>Command.code</i>	2	The code field within a command entry.
<i>Status.count</i>	3	The count field within a status entry

28
29
30
31
32
33
34
35 The caption numbering rules also apply to code tables and equations (not illustrated).

36
37 **Row 1, Row 2:** An example of how to describe the first two rows.

38
39 **Row 3:** An example of how to describe the final row.
40

41 D.3 Equation numbering differences 42

43 Within the annex, the equations are done in the same fashion. However, the distinct outline numbering
44 produces different results, as illustrated in Equation D.1.
45

$$46 \quad v = 2^7 \left(-b_0 + \sum_{i=1}^l b_i 2^{-i} \right) \quad (D.1)$$

Annex E (informative) Common grammatical issues

This annex is designed to point out and clarify common grammatical issues that affect many writers of English that have English as a secondary language. This section is presently small, but will grow over time to support the needs of specification writers.

E.1 Use of the word “the”

The word “the” is often misplaced in English sentences. Though there about 11 rules for using this word, only seven of them are directly applicable to writing our specifications.

- a) Showing that something or someone has already been spoken or written about, or is already known to the reader.
“The function block shown above shows...”
not “Function block above shows...”
- b) Used with things because they are well known.
“The 61883-1 document states...”
not “61883-1 document states...”
- c) Showing that something is the only one of its kind.
“The READ DESCRIPTOR command reads data from the Subunit Identifier Descriptor”
not “READ DESCRIPTOR command reads the data from Subunit Identifier Descriptor”
- d) Showing the reader that the writer means one particular thing.
“The printer shall respond to the command by...”
not “Printer shall respond to command by...”
- e) Used for particular sets of things.
“All of the subunit’s descriptor length fields shall be...”
not “All of subunit’s descriptor length fields shall be...”
- f) with comparison
“The larger the filesize, the longer it takes to download.”
not “With larger of filesize, longer it takes to download.”
- g) with superlative
“This shall be the largest data structure.”
not “This shall be largest data structure.”

E.2 Use of the word “a” or “an”

The word *a* or *an* when preceding a word that starts with a vowel) is also often misplaced and misused in English sentences. Though there about 6 rules for using this word, only several of them are really applicable to writing our specifications.

- a) One among many
“When the button is pressed, a new picture is taken.”
not “When button is pressed, the new picture is taken.”
- b) One in particular
“The output is a color picture”
not “The output is color picture”
- c) General one
“The target sends an FTP packet”
not “Target sends FTP packet”

E.3 Numbers: numerical or alphabetical?

02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63

In general, any number lower than ten shall be written alphabetically, unless it is used in an equation or in a sequence. As examples:

- Command A can process ten pictures. Command B can process 100;
- The range of values shall be between 5 – 15.

E.4 That and which

(Extracted from IEEE Standards Style Manual)

The words *that* and *which* are commonly misused; they are not interchangeable. *That* is best reserved in essential (or restrictive) clauses, *which* is appropriate in nonessential (or nonrestrictive), parenthetical clauses. Simply stated, if a comma can be inserted before the word *that* or *which*, the word should be *which*. If a comma would not be used, the word to use is *that*.

Examples:

- a) Defining the inputs and outputs provides a better understanding of the steps *that* are necessary to complete the process.
- b) Defining the inputs and outputs provides a better understanding of these steps, *which* are explained in 5.1 through 5.9.

E.5 Gender-neutral language

(Extracted from [R1] IEEE Standards Style Manual)

In order to reflect the changing practices in language usage, the IEEE Standards Department uses, in as many cases as possible, generic titles (such as chair rather than chairman) in the body of the standard. The following practices shall apply:

- a) When writing in the third person, the phrase *he or she* shall be used. The male or female pronoun alone or the variations *he/she* or *s/he* shall not be used. Also, the pronoun *they* shall not be used as a singular pronoun.
- b) If a particular sentence becomes cumbersome when *he or she* is used, the sentence should be rewritten in the plural or completely rewritten to avoid using pronouns. The indefinite pronoun *one* should be avoided. In references to a company, the pronoun *it*, not *we* or *they*, should be used.

E.6 Use of the second-person form of address

(Extracted from [R1] IEEE Standards Style Manual)

The second-person form of address (*you*) should not be used or implied in standards, e.g.,

You should avoid working on lines from which a shock or slip will tend to bring your body toward exposed wires.

This should be rewritten to identify the addressee, as follows:

Employees should avoid working on lines from which a shock or slip will tend to bring their bodies toward exposed wires.

Annex F (informative) General writing guidelines

F.1 Capitalization

Engineers seem to Capitalize Anything that could possibly be important, including Acronyms, Application-Specific Terms, and Variable Names, as illustrated in this sentence. The intent is oftentimes to distinguish between normal English and words with specialized meaning. The effect, however, is that excessive (and oftentimes inconsistent) use of capitalization complicates parsing of the sentence, leading to loss of clarity and increased numbers of technical ambiguities.

F.1.1 When to capitalize

In general, capitalization is only necessary for proper nouns and the first word of headings: other uses are strongly discouraged. Alternative techniques are available to delineate words that have standard-specific requirements, as follows:

- a) Variables can be easily distinguished from normal English words by any of the following naming conventions:
 - 1) A command register or *startTest* field can be distinguished from English words by using an italics style, with a capital at the start of each following run-together word.
 - 2) An *ExecutionRoutine()* or group of *CommonControl* registers can be distinguished by capitalizing the first run-together word.
 - 3) A *source_address* parameter can be distinguished by the underscore that separates run together words.

Most importantly, a variable or field name should not be split into separate words separated by spaces and tied together by the common use of Capital Letters.
- b) Constants can be easily distinguished from normal English words by using all capitals within the name. Such conventions normally apply to a RESET constant or an enumerated START_FAST value.
- c) Names with special meaning, such as ringlet or port, can be defined in an early *Terms and definitions* clause, and therefore need not be capitalized when used within the text.

Many standards have incorrectly used capitals in an attempt to delineate the boundaries of a special variable name, such as a Negotiation Control Register. However, the following questions must be addressed for the sentence to be correctly parsed, or the document correctly searched for alternate descriptions:

- a) Does this refer to a specific *negotiationControl* instance of a register?
- b) Does this refer to a specific *negotiationControlRegister* location?
- c) Does this refer to generic negotiation-control registers?
- d) Does this refer to a negotiation register called *control*?
- e) Does this refer to a control register called *negotiation*?

While these distinctions may be moot to the editor or familiar readers, they severely restrict document comprehension by unfamiliar engineers and/or non-native English speakers. Do not do this!

F.1.2 When to capitalize

The first word of a clause, subclause, table, table-heading, figure, or figure insert is nominally capitalized, such as:

P.1.2 Shared control and status registers (CSRs)

An exception is when the heading starts with a special variable name, such as:

P.1.3 command register

02
03 Only proper nouns, such as California, are capitalized within the other words of a clause, subclause, table,
04 table-heading, figure, or figure insert. Adjacent words are not necessarily capitalized, as in:

05 **P.1.4 Habits of California wildlife**

06
07 **F.1.3 Noncapitalized definitions**

08
09 Terms within a glossary or definitions of terms are not capitalized, unless these are proper nouns. As
10 examples:

11 **3.2.29 bridge:** A functional unit interconnecting two or more networks on ...

12 **3.2.30 congestion domain:** The set of contiguous links associated ...

13 **3.2.31 cyclic redundancy check (CRC):** A specific type ...

14 **3.2.32 loop round trip time (LRTT):** The time that it takes for a control frame...

15 **3.2.33 medium access control (MAC) sublayer:** The portion of the data link layer...

16
17 **F.1.4 Noncapitalized acronyms**

18
19 Words within acronyms are not capitalized, unless these are proper nouns. As examples:

20
21 CRC cyclic redundancy check

22 FEC forward error correction

23 LRTT loop round trip time

24 MSB most significant bit

25 LSB least significant bit

26 SF signal fail

27 SDH synchronous digital hierarchy

28 WAN wide area network

29 WTR wait to restore

30 USA United States of America

31
32 **NOTE**—Within other clauses, uncommon acronyms should be spelled out on first usage:

33 **Correct:** the cyclic redundancy check (CRC) is computed

34 **Incorrect:** the CRC (cyclic redundancy check) is computed.

F.1.5 Improper nouns

The term improper noun describes a noun that is not a proper noun, but is (typically for historical or confusion reasons) capitalized like a proper noun. To reduce improper noun usage, consider the following when classifying your proper nouns.

- a) Is this a particular person, place, or thing? The United States of America and California are particular places or names; the distinct counties of California are not.
- b) Does this avoid confusion that cannot be solved by the appropriate standard-specific definition?
- c) Is this an acronym? By itself, this is insufficient justification for capitalization (see F.1.4). The capitalization of a cyclic redundancy check is unaffected by its CRC acronym abbreviation.
- d) Is this a title? Only the first title word is normally capitalized, so this is insufficient (see F.1.2).
- e) Is this a definition? A defined term is not necessarily a proper noun (see F.1.3).

Note that many standards have generated improper nouns due to improper interpretations of (c,d,e) usage. You may have to sometimes cope with such conventions, but don't propagate them.

F.2 Definitions

F.2.1 Definition clauses

From the style manual: Acronyms shall not be defined in the definitions clause. Instead, a following clause should be used. However, standards commonly place related content within a definitions clause, expanding the header text as follows:

3. Terms, definitions, and notation

F.2.2 Definition text

From the style manual: The terms should not be used in its own definition. Correct usage is as follows:

4.1.2 doublet: Two bytes.

Incorrect usage is as follows:

4.1.3 doublet: A doublet is two bytes of data.

F.3 Variable names

F.3.1 Spaceless variable names

The preceding (see) capitalization conventions are consistent with C programming conventions. Descriptive code can be easily used to specify formal definitions, as illustrated by Equation

```
errorCount += (protocolErrors + incomingCount.crcBad + clientCount.typeBad); (F.1)
```

This is clear and concise, when compared to the pseudo-defined pseudo code of Equation . The lack of extra spaces also reduces the equation lengths, so more of the equations will be able to fit on one line. The use of C-type conventions, to name subfields by their function (as opposed to position) adds to clarity, while simplifying text maintenance by defining bit-field positions in only one place.

```
(Error Count) ← ((Protocol Errors) + (Incoming Count) [15:14] + (Client Count) [12:11]) (F.2)
```

F.3.2 Hyphenless variable names

Using run-together hyphenless variable names also has the advantage that descriptive code can be easily used to specify formal definitions, as illustrated by Equation F.3.

```
errorCount += (protocolErrors + incomingCount.crcBad + clientCount.typeBad); (F.3)
```

02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63

For example, a hyphenated name could have the subtraction-operator ambiguity associated with Equation F.4.

```
errorCount += (protocol-errors + incomingCount.crcBad + clientCount.typeBad);    (F.4)
```

F.3.3 Aliased variable names

Never reference the same variable using two separate names, except within the context of a figure legend (see 3.5.2). As an example, the interchangeable use of *DA* and *Destination Address* terms raises the following concerns:

- a) The term *DA* could refer to either *destination address* (a generic term) or *Destination Address* (a specific field within frames).
- b) If the *Destination Address* is placed in the transmitted frame, and the *DA* is checked in the received frame, the reader will be confused.
- c) An capitalized abbreviation could easily be confused with a constant, such as PASS or FAIL.

F.4 Spelling checker

Always run a spelling checker on your text **before** submitting to Sponsor Ballot. To simplify spelling-checker and/or character-pattern searches, the following guidelines have been found to be useful:

- a) Ellipsis. The special ellipsis character ‘...’ should be used, not three consecutive periods. This eliminates spurious duplicate-period spell-checker reports.
- b) Spaces. Only one space should be used between words and sentences. This allows a search to readily find duplicate-space errors.
- c) Styles. Special character styles are appropriate for special names, variable names, and field names. For example, styles used within this paper have the following properties:
 - 1) The *NamePlain* style is applied to special names (e.g., StateMachineName) with the effect of inhibiting end-of-line hyphenation and spurious spelling-checker reports.
 - 2) The *NameItalic* style is applied to field and variable names (e.g., *thisValue.thatField*) to force an italic style, while inhibiting end-of-line hyphenation and spurious spelling-checker reports.

F.5 Cross references

Cross references should be clear and concise, with the following guidelines:

- a) The term subclause, not section, is used to describe distinctively numbered portions of a clause.
- b) A proper cross reference, such as “(see F.4)”, does not include the word “subclause” or “section”.
- c) The simple “(see F.4)” text is usually preferred to using multiple or complex alternatives.
 - “, as considered in F.4.”
 - “, as constrained by F.4.”
 - “, as described in F.4.”
 - “, as discussed in F.4.”
 - “, as illustrated in F.4.”
 - “, as shown in F.4.”
 - “, as specified in F.4.”
 - “, as standardized in F.4.”
 - “, as mandated by F.4.”
- d) Never type the numbers within “(see 4.4)”. Explicit cross-reference markers are preferred, because:
 - 1) Numbers are automatically updated as subclauses are inserted, deleted, or removed.
 - 2) Word processors can generate valuable hyperlinks from marked values.

F.6 Indentation

Most textual styles have tabs set at regular 18-point intervals, allowing standard styles to be used for special purposes, such as the service primitive listing below:

```
MA_DATA.request
(
    destination_address,
    service_class,
    mac_protection          // optional
)
```

Older standards have attempted to space the arguments further to the right, as illustrated below. The original intent was to mimic a distinctive hierarchical C-coding style, with a first-heading dependent indentation. Unfortunately, the correct tab distance is difficult to maintain (without per-instance customization of tab settings), and the inconsistency of tab'd data is distracting to the reader. This usage should be deprecated.

```
MA_DATA.request (
    destination_address,
    service_class,
    mac_protection          // optional
)
```

F.7 Table cell justification

Table cells are sometimes centered and sometimes left justified, as illustrated in Table .

Table F.1—Example table

Value	Name	Description
0	TEST_START	Start of the built-in self test.
1	TEST_STOP	Stops the command-invoked self test after the next subtest completion.
2,3	TEST_ABORT	Aborts the command-invoked self test immediately.

Text alignment in table cells is based on the following rules. There is flexibility in the application of these rules, in that long variable names could be classified as either (a) or (b). However, the same justification rules should be applied to all entries within the same column.

- a) If the text is a sentence, or resembles a sentence, then the text is left-justified.
- b) If the text is not a sentence (e.g. it is a phrase of one or two words), then the text is centered.

F.8 Punctuation within lists

A variety of punctuation conventions have been used within lists. Rather than propagate such diversity, the following recommendations are provided:

- a) Lists described with a preceding *as follows* or *following* text end with a colon (see above).
- b) Lists or sublists of sentences or sentence-like text are punctuated as sentences.
 - 1) These start with a capital letter.
 - 2) These end with a period.
- c) Lists or sublists of items or values are capitalized as within normal sentences, as listed below.
 - 1) error counts
 - 2) consistency checks
 - 3) CRC
 - 4) books

- 03 5) *errorCount* fields
- 04 6) United States of America

05 Variable lists have similar properties, as illustrated below. The text after the em-dash is either capitalized
06 and punctuated as a sentence (automatic and *sendReady* items), or listed as isolated words (*rxFilter* items).
07 No spaces should be included on either side of the em dash.
08

09 *automatic*

- 10 Indicates how the frame should be transmitted.
- 11 TRUE—Transmit in the MAC-specified direction(s).
- 12 FALSE—Transmit in the client-specified direction.

13 *sendReady*

- 14 Indicates wheter the sender is ready to accept another data frame for transmission.
- 15 TRUE—The sender is ready.
- 16 FALSE—(Otherwise.)

17 *rxFilter*

- 18 The type of filtering done when receiving frames destined for the client.
- 19 BASIC—host clients
- 20 FLOOD—bridge clients

21
22 For the automatic control signal, the actions triggered by TRUE and FALSE conditions are both described.
23 For the *sendReady* status signals, the TRUE condition suffices; there is no need (or value) in defining the
24 complementary FALSE condition, so consistently abbreviated “(Otherwise.)” text is provided.
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61

Annex G (informative) Writing guidelines

The following guidelines have been found useful within the IEEE P802.17 Resilient packet ring (RPR) Working Group. Other groups are welcome to utilize and or all of these documentation conventions, based on which ones are found to meet their requirements.

G.1 Field formats within figures

G.1.1 Aliased variable names

Figure G.1 is an example of how frame formats should be illustrated (in this context, frame connotes the media-independent portion of a transmitted packet). Names and numbers use an 8-point Arial font (as is true for all figures). Each name is horizontally centered in its box, byte numbers are right justified, and header/payload/trailer names are left justified. The byte-size and field names are vertically centered in their box. Similarly, the header/payload/trailer names are vertically centered with their spanning line.

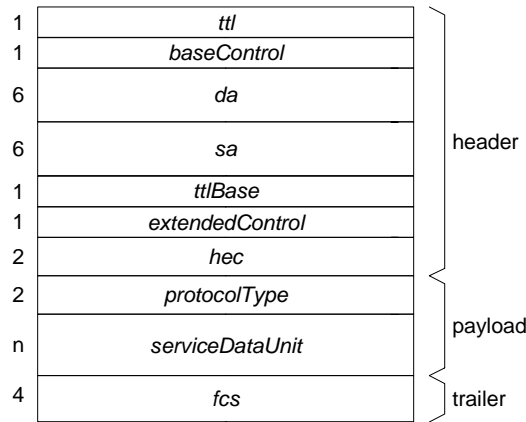


Figure G.1—Frame format illustrations

The fields are stacked vertically and have adjacent byte-size labels. The size of boxes should vary monotonically with it's byte size:

$\text{if } \text{SizeOf}(\text{fieldA}) > \text{SizeOf}(\text{fieldB}) \text{ then } \text{Height}(\text{fieldA}) \geq \text{Height}(\text{fieldB}).$

G.1.2 Generic field illustrations

Figure G.2 illustrates how subfield components of multibyte fields should be produced. For clarity, short and longer tick marks are used to delineate bit and byte positions respectively.

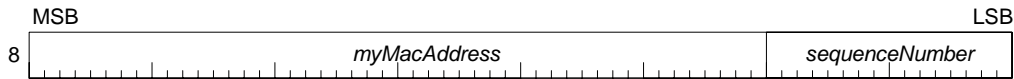


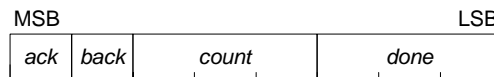
Figure G.2—Example of a bit-field illustration

The field names, byte-size numbers, and MSB/LSB labels use an 8-point Arial font. Each field name is horizontally centered in its box; the byte number and LSB label are right justified; the MSB label is left justified. The byte-size and field names are vertically centered in their box, then moved upward one point.

02
03 Depending on the width of field illustrations, the spacing between tick marks may be changed. Such tick-
04 mark-spacing changes are most conveniently performed by grouping tick marks and box from an existing
05 example (such as Figure G.1) and then stretching the grouped image. To simplify current and future editing,
06 care should be taken to ensure that the final box is snapped to the 3-point grid and the tick-mark spacing is
07 an exact multiple of the 3-point grid size.
08

09 **G.1.3 8-bit field format**

10
11 Figure G.3 illustrates how subfield components of a 1-byte field should be illustrated. Only short tick marks
12 are used, since there is no need to delineate byte locations. The font and alignment associated with field
13 names, byte-size numbers, and MSB/LSB labels is specified in 10.3.1. Abbreviations are used when the
14 field name would extend beyond its bounding box; see 3.5.2 for details.
15



16
17
18
19
20 **Figure G.3—8-bit field format**

21
22 **10.1.4.1 *x.ack*:** A bit that has been illustrated on the left.

23
24 **10.1.4.2 *back*:** A bit that has been illustrated near the left.

25
26 **10.1.4.3 *count*:** A 3-bit field that has been illustrated near the center.

27
28 **10.1.4.4 *done*:** A 3-bit field that has been illustrated near the right.

29 **NOTE**—The preceding text illustrates how the Definition4Like style is used within figure-field definitions.
30

31 **G.1.4 16-bit field format**

32
33 Figure G.4 illustrates how subfield components of a 2-byte field should be illustrated. Short tick marks and
34 one longer tick mark are used to delineate bit and byte locations. The font and alignment associated with
35 field names, byte-size numbers, and MSB/LSB labels is specified in TBD.
36



37
38
39
40
41
42 **Figure G.4—16-bit field format**

43
44 Abbreviations are used when the field name would extend beyond its bounding box; see TBD for details.
45

G.1.5 32-bit field format

Figure G.5 illustrates how subfield components of a 4-byte field should be illustrated. Short tick marks and three longer tick marks are used to delineate bit and byte locations. The font and alignment associated with field names, byte-size number, and MSB/LSB labels is specified in TBD. Each field name that would extend beyond its bounding box is abbreviated; a legend associates the local abbreviation with the formal field name.

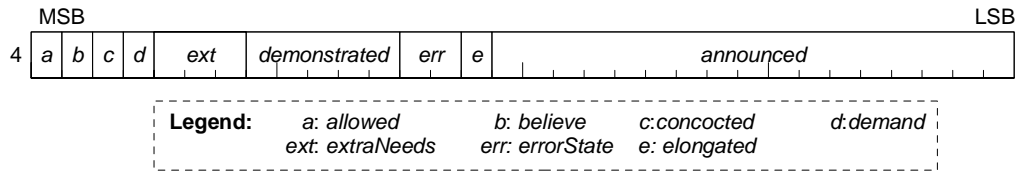


Figure G.5—32-bit field format

Below the fields, a legend is provided. The legend entries are ordered, in that a left-to-right scan of the fields corresponds to a left-to-right-then-top-to-bottom scan of the legend entries. A space should be included after the colon. The dotted line around the legend is a distinct graphics rectangle, which can be sized appropriately.

NOTE—The fields of Figure G.5 were planned to illustrate several of the larger byte-boundary tick marks. Depending on the field alignments, these are oftentimes hidden by the field’s bounding box.

G.2 Standard subclause formats

G.2.1 Common parameters

NOTE—Each clause or annex should use the second subclause to identify common values used within that clause, as illustrated below.

G.2.1.1 Common definitions

The following definitions are used multiple places within this clause.

GROUP_BIT

A constant value specified by the following C-code definition.

```
((uInt8)1) << 40) (G.1)
```

STOMP_CRC

A constant value specified by the following C-code definition.

```
(0xFFFFFFFF) (G.2)
```

NOTE—The fields of Figure 10.2 were planned to illustrate several of the larger byte-boundary tick marks. Depending on the field alignments, these are oftentimes hidden by the field’s bounding box.

05 **G.2.1.2 Common variables**

07 The following state machine inputs are used multiple times within this clause.

08 *hopsToCongestion*

09 An input hop-count value generated by the fairness protocols.

10 *myEdgeState*

11 Indicates the adjacent edge condition.

12 NORMAL—The datapath is operating without a span failure.

13 INTO_EDGE—The transmit side of the datapath is associated with a failed span.

14 FROM_EDGE—The receive side of the datapath is associated with a failed span.

17 NOTE—All enumerated values should be described, even when the definition is obvious to the editor.

19 *myRingletID*

20 An input associated with the ringlet identifier of this datapath entity.

21 RINGLET_0—Transmitting onto ringlet0.

22 RINGLET_1—Transmitting onto ringlet1.

25 NOTE—The enumerated values should be preceded by a description, even when obvious to the editor.

27 *rateC*

28 *rateCC*

29 Input rate values generated by the fairness protocols (see 9.3.1).

30 *tickTime*

31 A variable representing the start time of the current tick processing interval.

32 *time*

33 An input that represents a constantly incremented elapsed time value.

38 **G.2.1.3 Common routines**

39 *Parity(frame)*

40 Returns the parity of the referenced bits in idle or fairness frames, as described in x.y.

43 NOTE—Subroutine descriptions should always include their argument, or two tiny spaces between
44 parenthesis when no arguments are provided. The intent is to readily distinguish between routines (which
45 are called) and variable values (which are read and written).

46 NOTE—The subroutine names and arguments are italics, but parenthesis or special characters are not.

48 *SendSize(frame)*

49 A routine whose behavior is defined by Equation G.3.

50
$$\begin{aligned} &(\text{frame.ri} == \text{myRingletID} \mid \mid \backslash \\ &(\text{myWrapMethod} == \text{CENTER_WRAP} \ \&\& \ \text{myEdgeState} == \text{INTO_EDGE})) \end{aligned} \quad (\text{G.3})$$

52 *SizeOf(frame)*

53 Returns the number of bytes in the frame.

56 NOTE—The left edge of the in-line C code is indented using tabs, but nonbreaking spaces are used for
57 line-to-line indent purposes thereafter.

G.2.2 Variables defined in other clauses

NOTE—The use of remote variables should be summarized in a distinct subclause, so that the dependencies between clauses can be easily identified.

This clause references the following global variables defined in Clause 6:

hopsToCongestion

rateC

rateCC

This clause references the following global variables defined in Clause 10:

myEdgeState

myWrapMethod

G.2.3 State machine definitions

State machine definitions should contain introductory text, an optional flow-chart, variable definitions, and a normative state table, as illustrated in the following subclauses.

G.2.4 Single-queue transmit state machine

An introduction of the state machine belongs here.

G.2.5 Single-queue transmit flow chart

G.2.5.1 Single-queue transmit flow chart

The flow-chart for a state machine oftentimes includes a flow chart, as illustrated in Figure . Text is represented using an 8-point font, except for the 7-point YES/NO labels.

00
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63

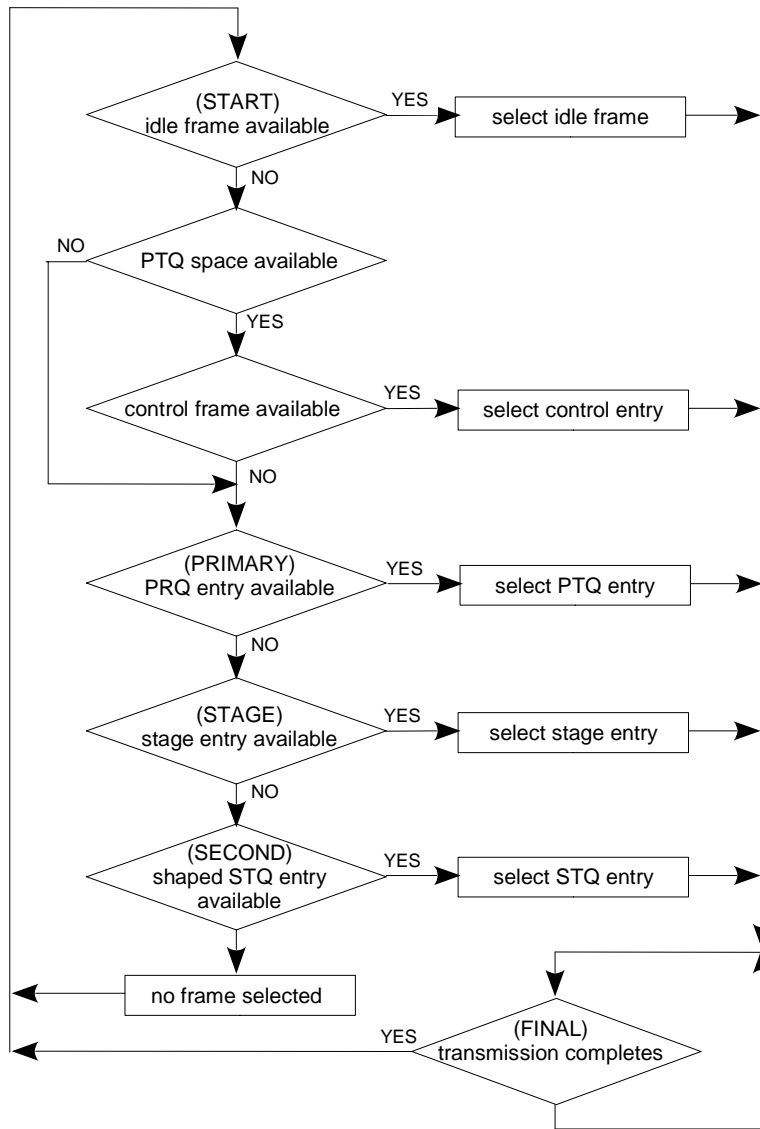


Figure G.6—Flowchart template

NOTE—All names used within the state machine should be described in the immediately preceding subclauses. These terms may cross-reference values set-by and defined-with other state machines, or may be defined at the beginning of the clause and cross-reference by multiple state machines.

G.2.5.2 Single-queue transmit state machine definitions

CLASS_A0
See 6.2.2.

(...)

Q_TX_STAGE
See 6.2.1.1.

G.2.5.3 Single-queue transmit state machine variables

creditI
See 6.2.1.2.
(...)
frame
See 6.2.1.2.

G.2.5.4 Single-queue transmit state machine routines

Dequeue(queue)
See 6.2.4.
(...)
SpaceInPTQ()
See 6.2.4.

G.2.5.5 Single-queue transmit state table

The single-queue transmit frame selection state machine specified in Table G.1 implements the functions necessary for selecting which frame to transmit in a single-queue MAC. (...).

Table G.1—Single-queue transmit selection states

Current state		Row	Next state	
state	condition		action	state
START	PHY_READY.inidication() == READY	1	—	CONT
	—	2	—	START
CONT	(frame == Dequeue(Q_TX_IDLE)) != NULL	3	creditI -= SizeOf(frame); Enqueue(Q_TX_COUNT, frame);	START
	(frame = DequeueControl(Q_TX_STAGE)) == NULL	4	—	PTQ
	SizeOfMacControl() > SpaceInPTQ()	5	—	START
	—	6	creditM -= SizeOf(frame); Enqueue(Q_TX_COUNT, frame);	START
PTQ	(frame == Dequeue(Q_TX_PTQ)) != NULL	7	—	COUNT
	—	8	—	STAGE
COUNT	frame.sc != CLASS_A0	9	creditD -= SizeOf(frame);	START
	—	10	Enqueue(Q_TX_COUNT, frame);	START
STAGE	(frame == Dequeue(Q_TX_STAGE)) != NULL	11	Enqueue(Q_TX_COUNT, frame);	START
	—	12	classC	START

NOTE—
 1) Input conditions should not be OR'd, but split into two rows.
 2) White space should, when possible, be placed between operators and text.
 If such white space would result in a (oftentimes more confusing) line wrap, it can be eliminated.
 3) Common rows should be straddled for clarity within the figure and described in close-together text.

02
03
04 **Row 1:** The physical layer has indicated that it is ready.
05 **Row 2:** Wait for the physical layer to be ready.

06
07
08 **NOTE**—A paragraph (not line feed) is appropriate between descriptions relating to distinct states.
09

10 **Row 9:** Transit frames unreserved bandwidth is subtracted from the downstream shaper credits.

11 **Row 10:** Transit frames using reserved bandwidth do not subtract from any shaper credits.

12
13 **Row 11:** The stage queue is selected when an entry is available in the queue.

14 **Row 12:** No frame is selected when no frame transmissions are possible.
15

16 **G.3 Service primitives**

17
18 The following abbreviated subclause illustrates styles used when specifying service primitives.
19

20 **G.3.1 MA_DATA.request**

21 **G.3.1.1 Function**

22
23
24
25 The MA_DATA.request primitive defines the transfer of data from a MAC client entity to a single peer
26 entity, or to multiple peer entities in the case of group addresses.
27

28 **G.3.1.2 Semantics of the service primitive**

29
30 The semantics of the primitives are as follows:

```
31 MA_DATA.request  
32 (  
33     destination_address,  
34     service_class,  
35     mac_protection           // optional  
36 )
```

37
38 The parameters of the MA_DATA.request are described below:

```
39 destination_address  
40     Specifies either an individual or group MAC address, different from the local MAC address, to be  
41     used to create the da (destination MAC address) field of the transmitted frame, ... .  
42  
43
```

44
45 **NOTE**—A table is sometimes necessary to adequately describe the enumerated service primitive
46 parameters, as illustrated for the service_class parameter below.

```
47 service_class  
48     Indicates the class of service requested by the MAC client, as described in Table G.2, which is  
49     used by the MAC entity to select the value of the sc field, which is described in 5.2, and to indicate  
50     the requested MAC treatment of the transmitted frame, as described in 7.3.  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61
```

Table G.2—service_class values

service_class value	Corresponding sc value	Description
SC_CLASSA	CLASS_A0 or CLASS_A1	classA
SC_CLASSB	CLASS_B	classB
SC_CLASSC	CLASS_C	classC

NOTE—Simple descriptions of enumerated values should be appended as a tab-indented list, as is done with the mac_protection specification below.

mac_protection

Indicates a choice of whether the MAC provides protection for the frame, as described in 4.3.

If the mac_protection parameter is omitted, a TRUE value is assumed.

TRUE—The MAC shall provide protection for the frame.

FALSE—The MAC shall not provide protection for the frame.

G.3.1.3 Semantics of the service primitive

The MA_DATA.request primitive is invoked by the client entity whenever data is to be transferred to a peer entity or entities.

G.3.1.4 Effect of receipt

The receipt of the MA_DATA.request primitive causes the MAC entity to create a data frame or an extended data frame, fill in the fields whose values are given or determined by the parameters of this request, and pass the properly formed frame to the transmit state machines (see Figure 6.5), for transfer to the peer MAC sublayer entity or entities.

G.3.1.5 Additional comments

The MAC does not reflect frames back to the client. If a client issues an MA_DATA.request primitive with a destination_address value equal to its local MAC address, the request is rejected.

00
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63

02
03 **Annex H**
04 **(informative)**
05 **Highest level definitions**
06
07

08
09
10 **H.1 Standard subclause formats**

11 So that common references can be included within this document, cross references are provided for cut-and-paste uses:

12
13
14 AEIC publications are available from the Association of Edison Illuminating Companies, 600 N. 18th Street,
15 P. O. Box 2641, Birmingham, AL 35291-0992, USA.

16
17 ANSI publications are available from the Sales Department, American National Standards Institute, 11
18 West 42nd Street, 13th Floor, New York, NY 10036, USA.

19 API historical materials can be obtained (for a fee) from the American Petroleum Institute Library, 1200 L
20 Street NW, Washington, DC 20005, USA.

21
22 API publications are available from the Publications Section, American Petroleum Institute, 1200 L Street
23 NW, Washington, DC 20005, USA.

24 ARINC publications are available from ARINC Research Corporation, Document Section, 2551 Riva Rd.,
25 Annapolis, MD 21401.

26
27 ASHRAE publications are available from the Customer Service Department., American Society of Heating,
28 Refrigerating and Air Conditioning Engineers, 1791 Tullie Circle, NE, Atlanta, GA 30329, USA.

29 ASME publications are available from the American Society of Mechanical Engineers, 22 Law Drive, Fair-
30 field, NJ 07007, USA.

31
32 ASTM publications are available from the American Society for Testing and Materials, 100 Barr Harbor
33 Drive, West Conshohocken, PA 19428-2959, USA.

34 AWEA publications are available from the American Wind Energy Association, Standards Program, 777
35 North Capital Street NE, #805, Washington, DC 20002, USA.

36
37 CCITT publications are available from the International Telecommunications Union, Sales Section, Place
38 des Nations, CH-1211, Genève 20, Switzerland/Suisse. They are also available in the United States from the

39 U.S. Department of Commerce, Technology Administration, National Technical Information Service
40 (NTIS), Springfield, VA 22161, USA.

41
42 CFR publications are available from the Superintendent of Documents, U.S. Government Printing Office,
43 P.O. Box 37082, Washington, DC 20013-7082, USA.

44 CISPR documents are available from the International Electrotechnical Commission, 3, rue de Varembe,
45 Case Postale 131, CH 1211, Genève 20, Switzerland/Suisse. They are also available in the United States
46 from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New
47 York, NY 10036, USA.

48
49 CSA publications are available from the Canadian Standards Association (Standards Sales), 178 Rexdale
50 Blvd., Etobicoke, Ontario, Canada M9W 1R3.

51 ECMA publications are available from the European Computer Manufacturers Association, 114 rue du
52 hone, CH-1204, Geneva, Switzerland/Suisse.

53
54 EGSA publications are available from the Electrical Generating Systems Association, 10251 W. Sample d.,
55 Suite B, Coral Springs, FL 33065, USA.

56
57 EIA publications are available from Global Engineering, 1990 M Street NW, Suite 400, Washington, DC,
58 IPS publications are available from the National Technical Information Service (NTIS), U. S. Dept. of
59 ommerce, 5285 Port Royal Rd., Springfield, VA 22161.

ICEA publications are available from ICEA, P.O. Box 411, South Yarmouth, MA 02664, USA. 03

ICRU publications are available from the National Council on Radiation Protection and Measurements, 7910 Woodmont Avenue, Suite 800, Bethesda, MD 20814, USA. 04
05

IEC publications are available from IEC Sales Department, Case Postale 131, 3, rue de Varembe, CH-1211, Geneva 20, Switzerland/Suisse. IEC publications are also available in the United States from the Sales 07
08

Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA. 09
10

IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA. 11
12
13

Internet Requests for Comments (RFCs) are available from the DDN Network Information Center, SRI International, Menlo Park, CA 94025 USA. They are also available on the World Wide Web at the following URL: <http://www.internic.net/ds/rfc-index.html> 14
15
16
17

IPC publications are available from the Institute for Interconnecting and Packaging Electronic Circuits (IPC), 7380 N. Lincoln Ave., Lincolnwood, IL 60646. 18
19

ISO publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembe, CH-1211, Geneva 20, Switzerland/Suisse. ISO publications are also available in the United States from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA. 20
21
22
23
24

JEDEC publications are available from JEDEC, 2001 I Street NW, Washington, DC 20006, USA. 25

MIL publications are available from Customer Service, Defense Printing Service, 700 Robbins Ave., Bldg. 4D, Philadelphia, PA 19111-5094. 26
27
28

NASA publications are available from NASA Center for Aerospace Information (CASI), ATTN: Document Orders, 800 Elkridge Landing Rd., Linthicum Heights, MD 21090-2934. 29
30
31

NBS publications are available from the Superintendent of Documents, U.S. Government Printing Office, P.O. Box 37082, Washington, DC 20013-7082, USA. 32
33

NCRP publications are available from the National Council on Radiation Protection and Measurements, 7910 Woodmont Avenue, Suite 800, Bethesda, MD 20814, USA. 34
35
36

NEMA publications are available from the National Electrical Manufacturers Association, 1300 N. 17th St., Ste. 1847, Rosslyn, VA 22209, USA. 37
38

NFPA publications are available from Publications Sales, National Fire Protection Association, 1 Batterymarch Park, P.O. Box 9101, Quincy, MA 02269-9101, USA. 39
40
41

UL standards are available from Global Engineering, 1990 M Street NW, Suite 400, Washington, DC, 20036, USA. 42
43

US Regulatory Guides are available from the Superintendent of Documents, US Government Printing Office, P.O. Box 37082, Washington, DC 20013-7082, USA. 44
45
46

This authorized standards project was not approved by the IEEE Standards Board at the time this went to press. It is available from the IEEE Service Center. [NOTE: the reference must include the P-number, title, revision number, and date.] 47
48
49

ANSI XXX-19XX has been withdrawn; however, copies can be obtained from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA. 50
51
52

The numbers in brackets, when preceded by the letter B, correspond to those in the bibliography in Section XX. 53
54

The numbers in brackets correspond to those of the references in XX. 55
56

IEEE Std XXX-19XX has been withdrawn; however, copies can be obtained from the IEEE Standards Department, IEEE Service Center, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA. 57
58
59

02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63

As this standard goes to press, IEEE Std SSS-199X is not yet published. It is, however, available in manuscript form from the IEEE Standards Department, (908) 562-3800. Anticipated publication date is XXX 199X, at which point IEEE Std XXX-199X will be available from the IEEE Service Center, 1-800-678-4333.

This standard will be available from the Institute of Electrical and Electronics Engineers, Service Center, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA, in early 199X.

Approved 1394TA documents can be ordered through: 1394 Trade Association, 221 West Sixth Street, Suite 1520, Austin, TX 78701; by contacting taadmin@1394TA.org; or by fetching a pdf copy from the 1394TA web site: <http://www.1394TA.org/abouttech/specifications/techspec.html>.

H.2 Standard subclause formats

Cover pages for IEEE documents are provided in the remainder of this annex. When editing IEEE specifications, these pages should be moved to the front of the document.

Draft Standard ...

Sponsor:
Committee of the IEEE Society

Abstract:

Keywords:

Copyright © <current year> by the Institute of Electrical and Electronics Engineers, Inc.
Three Park Avenue
New York, New York 10016-5997, USA

All rights reserved. This document is an unapproved draft of a proposed IEEE Standard. As such, this document is subject to change. USE AT YOUR OWN RISK! Because this is an unapproved draft, this document must not be utilized for any conformance/compliance purposes. Permission is hereby granted for IEEE Standards Committee participants to reproduce this document for purposes of IEEE standardization activities only. Prior to submitting this document to another standards development organization for standardization activities, permission must first be obtained from the Manager, Standards Licensing and Contracts, IEEE Standards Activities Department. Other entities seeking permission to reproduce this document, in whole or in part, must obtain permission from the Manager, Standards Licensing and Contracts, IEEE Standards Activities Department.

IEEE Standards Activities Department
Standards Licensing and Contracts
445 Hoes Lane, P.O. Box 1331
Piscataway, NJ 08855-1331, USA

00
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63

02
03 **Introduction**
04

05 (This introduction is not part of IEEE Std xx-200X, title.)
06

07
08 At the time this standard was completed, the working group had the following membership:
09

10 **name**, Chair
11
12 **name**, Secretary
13

14
15

16 The following three columns of names is implemented as a three-column table, with a white color applied 17 to the between-column lines. When names are available, approximately 1/3 should be placed in each of 18 these three table cells.
--

19

20
21 To Be Supplied By IEEE Etc. Etc.
22
23
24
25
26
27
28
29

30 The following members of the balloting committee voted on this standard. Balloters may have voted for
31 approval, disapproval, or abstention.
32

33
34

35 The following three columns of names is implemented as a three-column table, with a white color applied 36 to the between-column lines. When names are available, approximately 1/3 should be placed in each of 37 these three table cells.
--

38

39
40 To Be Supplied By IEEE Etc. Etc.
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

When the IEEE-SA Standards Board approved this standard on XX Month 200X, it had the following membership: 03
04

name, Chair 05
name, Vice Chair 06
name, Secretary 07
08

The following three columns of names is implemented as a three-column table, with a white color applied to the between-column lines. When names are available, approximately 1/3 should be placed in each of these three table cells. 09
10
11
12
13

To Be Supplied By IEEE Etc. Etc. 14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31

*Member Emeritus 32

Also included is the following nonvoting IEEE-SA Standards Board liaisons: 33

Satish K. Aggarwal, NRC Representative 34
Alan H. Cookson, NIST Representative 35
Donald R. Volzka, TAB Representative 36

Editor's name here 37
IEEE Standards Project Editor 38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61

Annex I

(informative)

C code illustrations

```
// The following illustrate how code can be presented in a landscape fashion
//
//          1          2          3          4          5          6          7          8          9          1          1          1          1
//2345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012
```

```
#include <assert.h>
#include <stdio.h>

typedef unsigned char    uInt1;    // 1-byte unsigned integer
typedef unsigned short  uInt2;    // 2-byte unsigned integer
typedef unsigned int    uInt4;    // 4-byte unsigned integer
typedef unsigned long long uInt8;  // 8-byte unsigned integer

typedef signed char      sInt1;    // 1-byte signed integer
typedef signed short    sInt2;    // 2-byte signed integer
typedef signed int      sInt4;    // 4-byte signed integer
typedef signed long long sInt8;   // 8-byte signed integer

// Illustrations of useful ASCII-art comments follow
//
// Ethernet packet format:
// +-----+
// |          |l|m|          destinationMacAddress          |
// +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
// |          |l|m|          sourceMacAddress              |
// +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
// |          vlanCode          | pri |c|          vlanIdentifier          |          typeCode          |
// +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
// |version| ihl |
// +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

```

// IP-V4 packet format
// +-----+-----+-----+-----+-----+-----+
//h2|          vlanCode          | pri |c|          vlanIdentifier          |          typeCode          |
// +-----+-----+-----+-----+-----+-----+
//h3|version| ihl | pre |D|T|R|C|r|          totalLength          |          identification          | fls |          fragmentOffset |
// +-----+-----+-----+-----+-----+-----+
//h4| timeToLive | protocol |          headerChecksum          |          sourceIpAddress          |
// +-----+-----+-----+-----+-----+-----+
//h5|          destinationIpAddress          |          (ipOptions)          |
// +-----+-----+-----+-----+-----+-----+
//h6|          sourcePort          |          destinationPort          |          sequenceNumber          |
// +-----+-----+-----+-----+-----+-----+
//h7|          acknowledgementNumber          | offs | reserved | bits |          window          |
// +-----+-----+-----+-----+-----+-----+
//h8|          checkSum          |          urgentPointer          |          (tcpOptions)          |
// +-----+-----+-----+-----+-----+-----+
//h9|
//          applicationData
//
//          checkSum          |          (extension)
// +-----+-----+-----+-----+-----+-----+

```

```

// IP-V4 packet format
// +-----+-----+-----+-----+-----+-----+
//h0|          |l|m|          destinationMacAddress          |
// +-----+-----+-----+-----+-----+-----+
//h1|          |l|m|          sourceMacAddress          |
// +-----+-----+-----+-----+-----+-----+
//h2|          vlanCode          | pri |c|          vlanIdentifier          |          typeCode          |
// +-----+-----+-----+-----+-----+-----+
//h3|version| ihl | pre |D|T|R|C|r|          totalLength          |          identification          | fls |          fragmentOffset |
// +-----+-----+-----+-----+-----+-----+
//h4| timeToLive | protocol |          headerChecksum          |          sourceIpAddress          |
// +-----+-----+-----+-----+-----+-----+
//h5|          destinationIpAddress          |          (ipOptions)          |
// +-----+-----+-----+-----+-----+-----+
//h6|          sourcePort          |          destinationPort          |          sequenceNumber          |
// +-----+-----+-----+-----+-----+-----+
//h7|          acknowledgementNumber          | offs | reserved | bits |          window          |
// +-----+-----+-----+-----+-----+-----+
//h8|          checkSum          |          urgentPointer          |          (tcpOptions)          |
// +-----+-----+-----+-----+-----+-----+
//h9|
//          applicationData
//
//          checkSum          |          (extension)
// +-----+-----+-----+-----+-----+-----+

```

```

// Options quadlets (assumed to be aligned)
// +-----+
//p0| ip4NULL | ip4OpType | ip4OpLength + ip4OpPointer |
// +-----+
//p1| routeAddress[0] |
// +-----+
// | routeAddress[1] |
// +-----+
// | (etc) |
// +-----+
// | routeAddress[n] |
// +-----+

// IP-V6 packet format
// +-----+
//h0| |l|m| destinationMacAddress |
// +-----+
//h1| |l|m| sourceMacAddress |
// +-----+
//h2| vlanCode |pri|c| vlanIdentifier | typeCode |
// +-----+
//h3|version| trafficClass | flowLabel | payloadLength | nextHead | hopLimit |
// +-----+
//h4| sourceIp6AddressHi |
// +-----+
//h5| sourceIp6AddressLo |
// +-----+
//h6| destinationIp6AddressHi |
// +-----+
//h7| destinationIp6AddressLo |
// +-----+
//h8| nextHead | opLength |
// | hopByHopOptionParameters |
// | (comes first) |
// +-----+
// | nextHead | opLength |
// | otherOptionParameters |
// +-----+
// | nextHead | opLength |
// | sourceRouteParameters |
// +-----+
//i0| sourcePort | destinationPort | sequenceNumber |
// +-----+
//i1| acknowledgementNumber | offs | reserved | bits | window |
// +-----+
//i2| checkSum | urgentPointer | (tcpOptions) |
// +-----+
// |
// | applicationData |
// +-----+
// |
// | checkSum | (extension) |
// +-----+

```

```

// IP-V6 source-route option format
// +-----+-----+-----+-----+-----+-----+-----+-----+
//s0| nextHead | opLength | opType=0 | segment | reserved |
// +-----+-----+-----+-----+-----+-----+-----+-----+
//s1|
// |
// | address [n]
// |
// +-----+-----+-----+-----+-----+-----+-----+-----+
//s3|
// |
// | address [n-1]
// |
// +-----+-----+-----+-----+-----+-----+-----+-----+
// |
// | ...
// |
// +-----+-----+-----+-----+-----+-----+-----+-----+
// |
// | address [0]
// |
// +-----+-----+-----+-----+-----+-----+-----+-----+

// IPX packet format
// +-----+-----+-----+-----+-----+-----+-----+-----+
//h0| |l|m| destinationMacAddress
// +-----+-----+-----+-----+-----+-----+-----+-----+
//h1| |l|m| sourceMacAddress
// +-----+-----+-----+-----+-----+-----+-----+-----+
//h2| vlanCode |pri|c| vlanIdentifier | typeCode
// +-----+-----+-----+-----+-----+-----+-----+-----+
//h3| checksum | packetLength | control | type | destinationNetworkHi
// +-----+-----+-----+-----+-----+-----+-----+-----+
//h4| destinationNetworkLo | destinationNode
// +-----+-----+-----+-----+-----+-----+-----+-----+
//h5| destinationSocket | sourceNetwork | sourceNodeHi
// +-----+-----+-----+-----+-----+-----+-----+-----+
//h6| sourceNodeLo | sourceSocket
// +-----+-----+-----+-----+-----+-----+-----+-----+
//h7|
// |
// | applicationData
// |
// +-----+-----+-----+-----+-----+-----+-----+-----+
// |
// | checksum | (extension)
// |
// +-----+-----+-----+-----+-----+-----+-----+-----+

```