TEP Structure and Keywords

TEP:	1
Group:	Core Working Group
Type:	Best Current Practice
Status:	Final
TinyOS-Version:	All
Author:	Philip Levis

Note

This document specifies a Best Current Practices for the TinyOS Community, and requests discussion and suggestions for improvements. Distribution of this memo is unlimited.

Abstract

This memo describes the structure all TinyOS Extension Proposal (TEP) documents follow, and defines the meaning of several key words in those documents.

1. Introduction

In order to simplify management, reading, and tracking development, all TinyOS Extension Proposals (TEPs) MUST have a particular structure. Additionally, to simplify development and improve implementation interoperability, all TEPs MUST observe the meaning of several key words that specify levels of compliance. This document describes and follows both.

2. Keywords

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in TEP 1.

Note that the force of these words is modified by the requirement level of the document in which they are used. These words hold their special meanings only when capitalized, and documents SHOULD avoid using these words uncapitalized in order to minimize confusion.

2.1 MUST

MUST: This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.

2.2 MUST NOT

MUST NOT: This phrase, or the phrase "SHALL NOT", mean that the definition is an absolute prohibition of the specification.

2.3 SHOULD

SHOULD: This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.

2.4 SHOULD NOT

SHOULD NOT: This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.

2.5 MAY

MAY: This word, or the adjective "OPTIONAL", mean that an item is truly optional. One implementer may choose to include the item because a particular application requires it or because the implementer feels that it enhances the system while another implementer may omit the same item. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)

2.6 Guidance in the use of these Imperatives

Imperatives of the type defined in this memo must be used with care and sparingly. In particular, they MUST only be used where it is actually required for interoperation or to limit behavior which has potential for causing harm (e.g., limiting retransmissions) For example, they must not be used to try to impose a particular method on implementors where the method is not required for interoperability.

3. TEP Structure

TEPs have two major parts, a header and a body. The header states document metadata, for management and status. The body contains the content of the proposal.

All TEPs MUST conform to reStructuredText standards[1] and follow the docutils template, to enable translation from reStructuredText to HTML and LaTeX.

3.1 TEP Header

The TEP header has several fields which MUST be included, as well as others which MAY be included. The TEP header MUST NOT include headers which are not specified in TEP 1 or supplementary Best Current Practice TEPs. The first six header fields MUST be included in all TEPs, in the order stated here.

The first field is "TEP," and specifies the TEP number of the document. A TEP's number is unique.. This document is TEP 1. The TEP type (discussed below) determines TEP number assignment. Generally, when a document is ready to be a TEP, it is assigned the smallest available number. BCP TEPs start at 1 and all other TEPs (Documentary, Experimental, and Informational) start at 101.

The second field states the name of the working group that produced the document. This document was produced by the Core Working Group.

The third field is "Type," and specifies the type of TEP the document is. There are four types of TEP: Best Current Practice (BCP), Documentary, Informational, and Experimental. This document's type is Best Current Practice.

Best Current Practice is the closest thing TEPs have to a standard: it represents conclusions from significant experience and work by its authors. Developers desiring to add code (or TEPs) to TinyOS SHOULD follow all current BCPs.

Documentary TEPs describe a system or protocol that exists; a documentary TEP MUST reference an implementation that a reader can easily obtain. Documentary TEPs simplify interoperability when needed, and document TinyOS service implementations.

Informational TEPs provide information that is of interest to the community. Informational TEPs include data gathered on radio behavior, hardware characteristics, other aspects of TinyOS software/hardware, organizational and logistic information, or experiences which could help the community achieve its goals.

Experimental TEPs describe a completely experimental approach to a problem, which are outside the TinyOS core and will not necessarily become part of it. Unlike Documentary TEPs, Experimental TEPs may describe systems that do not have a reference implementation.

The fourth field is "Status," which specifies the status of the TEP. A TEP status can either be "Draft," which means it is a work in progress, "Final," which means it is complete and will not change. Once a TEP has the status "Final," the only change allowed is the addition of an "Obsoleted By" field.

The "Obsoletes" field is a backward pointer to an earlier TEP which the current TEP renders obsolete. An Obsoletes field MAY have multiple TEPs listed. For example, if TEP 191 were to replace TEPs 111 and 116, it would have the field "Obsoletes: 111, 116".

The "Obsoleted By" field is added to a Final TEP when another TEP has rendered it obsolete. The field contains the number of the obsoleting TEP. For example, if TEP 111 were obsoleted by TEP 191, it would have the field "Obsoleted By: 191".

"Obsoletes" and "Obsoleted By" fields MUST agree. For a TEP to list another TEP in its Obsoletes field, then that TEP MUST list it in the Obsoleted By field.

The obsoletion fields are used to keep track of evolutions and modifications of a single abstraction. They are not intended to force a single approach or mechanism over alternative possibilities.

If a TEP is Best Current Practices or Documentary, then it MUST include an additional field, "TinyOS-Version:," which states what version(s) of TinyOS the document pertains to. This document pertains to all versions of TinyOS, until made obsolete by a future TEP. This field MUST appear after the Status field and before the Author field.

The final required field is "Author," which states the names of the authors of the document. Full contact information should not be listed here (see Section 3.2).

There is an optional field, "Extends." The "Extends" field refers to another TEP. The purpose of this field is to denote when a TEP represents an addition to an existing TEP. Meeting the requirements of a TEP with an Extends field requires also meeting the requirements of all TEPs listed in the Extends field.

If a TEP is a Draft, then four additional fields MUST be included: Draft-Created, Draft-Modified, Draft-Version, and Draft-Discuss. Draft-Created states the date the document was created, Draft-Modified states when it was last modified. Draft-Version specifies the version of the draft, which MUST increase every time a modification is made. Draft-Discuss specifies the email address of a mailing list where the draft is being discussed. Final and Obsolete TEPs MUST NOT have these fields, which are for Drafts only.

3.2 TEP Body

The first element of the TEP body MUST be the title of the document. A TEP SHOULD follow the title with an Abstract, which gives a brief overview of the content of the TEP. Longer TEPs MAY, after the Abstract, have a Table of Contents. After the Abstract and Table of Contents there SHOULD be an Introduction, stating the problem the TEP seeks to solve and providing needed background information.

If a TEP is Documentary, it MUST have a section entitled "Implementation," which instructs the reader how to obtain the implementation documented.

If a TEP is Best Current Practice, it MUST have a section entitled "Reference," which points the reader to one or more reference uses of the practices.

The last three sections of a TEP are author information, citations, and appendices. A TEP MUST have an author information section titled entitled "Author's Address" or "Authors' Addresses." A TEP MAY have a citation section entitled "Citations." A citations section MUST immediately follow the author information section. A TEP MAY have appendices. Appendices MUST immediately follow the citations section, or if there is no citations section, the author information section. Appendices are lettered. Please refer to Appendix A for details.

4. Reference

The reference use of this document is TEP 1 (itself).

5. Acknowledgments

The definitions of the compliance terms are a direct copy of definitions taken from IETF RFC 2119.

6. Author's Address

Philip Levis
358 Gates Hall
Stanford University
Stanford, CA 94305

phone - +1 650 725 9046

email - pal@cs.stanford.edu

7. Citations

Appendix A. Example Appendix

This is an example appendix. Appendices begin with the letter A.

[1] reStructuredText Markup Specification. <http://docutils.sourceforge.net/docs/ref/rst/restructuredtext.html>