                   Version 2.0 Microsoft Word Template
                   for Creating Internet Drafts and RFCs

Abstract

   This document describes the properties and use of a revised Microsoft
   Word template (.dot) for writing Internet Drafts and RFCs. It
   replaces the initial template described in RFC 3285 to more fully
   support Word's outline modes and to be easier to use. This template
   can be direct-printed and direct-viewed, where either is line-for-
   line identical with RFC Editor-compliant ASCII output. This version
   obsoletes RFC 3285.

   The most recent version of this template and post-processing scripts
   are available at http://www.isi.edu/touch/tools.

Status of This Memo

   This document is not an Internet Standards Track specification; it is
   published for informational purposes.

   This is a contribution to the RFC Series, independently of any other
   RFC stream. The RFC Editor has chosen to publish this document at
   its discretion and makes no statement about its value for
   implementation or deployment. Documents approved for publication by
   the RFC Editor are not a candidate for any level of Internet
   Standard; see Section 2 of RFC 5741.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   http://www.rfc-editor.org/info/rfc5385.

Copyright Notice

Table of Contents

1. Introduction

   Internet Drafts and RFCs are predominantly written in embedded-text,
   compile-based formatting systems [1][4][5]. The primary and first
   such system is NROFF, a text-formatting utility based on manual entry
   of embedded configuration commands, such as ".p" for new paragraphs.
   XML is a more recent alternative that uses structure tags instead of
   explicit formatting commands to allow a single file (.xml) to be
   'compiled' into ASCII output, HTML, or a variety of other formats as
   desired [7].

   Although XML adds more modern semantic information to the structure
   tags, neither system supports modern WYSIWYG (what-you-see-is-what-
   you-get) editing. Editors such as Microsoft Word and Corel
   WordPerfect provide not only WYSIWYG editing, but also semantic tags
   as well as outline-mode capabilities. To that end, a Word template
   called 2-Word.template.rtf was created that supports authoring RFCs,
   as described in RFC 3285 [6]. That version succeeded in enabling
   Word-based RFC editing, but did not support Word's outline mode
   renumbering capabilities.

   This document describes the properties and use of a revised Microsoft
   Word template (.dot) file that supports Internet Draft and RFC
   formatting, intended as an update to that of RFC 3285. This version,
   called 2-Word-v2.0.template.dot, addresses a number of issues with
   the preliminary version:

   o redefines basic styles (Normal, Heading1, etc.) rather than
      creating new styles,

   o updates the boilerplate according to BCP 78,

   o uses more conventional methods for autonumbered references and
      figures, including support for name-based references (e.g.,
      "[Tou2005]"), and

   o supports direct output to a printer from the .doc source, as well
      as RFC-3285-style 'print to text' with post-processing on Windows-
      based PCs.

   This document assumes familiarity with Microsoft Windows operating
   systems and the Word application.

2. Use

   To use this template, double-click on it in Windows (it may work in
   MacOS and/or OpenOffice, but this has not been confirmed). The
   result should be a 'new' document. Do NOT open this document from

   within Word, e.g., via the File->Open menu; this will edit the
   template, rather than using the template to create a new template-
   based document.

2.1. Setting up Your Version of Word to Edit RFCs

   Unfortunately, Word does not have a way to save some useful settings
   in a template. It may be useful to configure autoformatting to avoid
   using smart quotes or hyphens. However, this template is compatible
   with these features as its post-processor script translates such non-
   standard character codes into their RFC-compatible ASCII equivalents.

2.2. Editing

   The template provides a number of styles for use (for details, see
   Appendix A). Some are redefined internal styles and some are new, as
   follows. Throughout the document, avoid the use of bold, italics, or
   any other character formatting, as well as any graphics or paragraph
   or table borders. Smart hyphens and quotes need not be avoided and
   will be translated during post-processing.

   The current styles allow existing hyphens to break (wrap) across
   lines, but do not add hyphenation. To insert a non-breaking hyphen,
   type <CTL-_> (control-underscore); this is particularly useful in
   URLs, which are more readable if not line-wrapped.

   Note that it is critical NOT to use any styles other than those
   provided by this template.

   o Redefined internal styles for general use:

      Normal, Heading1-9, Caption, Header, Footer:

         Use in the normal fashion. Tabs can be used as desired.

   o New styles:

      RFC Title:

         For the document title only.

      RFC Figure:

         For ASCII-art figures. Single spaced, kept together.

      RFC List Bullet:

         For bulleted lists.

RFC List Numbered:

   For numbered lists. Use 'restart numbering' as desired for new
   list sequences.

RFC References:

   For normative and informative references. Do not restart
   numbering for informative references.

   NOTE: references can use an alternate unnumbered style that
   relies on use of Word's bookmark feature to set cross-reference
   tags.

RFC App:

   For appendix titles, using "Appendix A." format. Starts at the
   top of a new page. If appendices are used, start the
   references (after appendices) at the top of a new page (insert
   'page break').

RFC App H1-5:

   For appendix headings ("A.1." format).

   NOTE: these headings will NOT auto-renumber when
   promoted/demoted in outline mode.

o Redefined internal styles not generally accessed by users:

   TOC1-9:

      For table of contents entries.

   RFC Instructions:

      For instructions to authors. These notes must be deleted.

   RFC H1 - no num:

      For unnumbered headings in end boilerplate text.

   RFC H1 - no TOC no num:

      For unnumbered, unlisted (in TOC) headings in front boilerplate
      text.

2.3. Saving the File

   The template is provided as a Word 97-2003 ".dot" template; Word 2007
   uses an XML-based variant called ".dotx". Word 2007 can use ".dot"
   templates in "Compatibility Mode". If the file is saved in that
   mode, it has a ".doc" suffix; if saved in native Word 2007 mode, it
   has a ".docx" format. It does not matter which saved file format is
   used.

2.4. Generating Output

   This template supports printing similar output both to regular
   printers (8.5 x 11 inch paper) and as Internet-Draft/RFC-compatible
   text files. Printing to a printer is designed for Windows XP and
   Vista and may work with earlier versions of Windows; it has not been
   tested with other OS's. The template is currently in a Word 97-2003
   format (.dot); this can be used in "Compatibility Mode" in Word 2007.

   Printing to text is supported only for Windows at this time. There
   is no current support for XML. Details are provided as follows.

2.4.1. Printing Direct to a Printer

   The template produces output direct to a printer that is line-for-
   line, page-for-page identical with the text-only version, with a few
   minor exceptions:

   o Single and double quotes may be angled (left quote, right quote),
      rather than straight, depending on whether 'smart quotes' are
      enabled in Tools->Autocorrect options.

   o Hyphens may print as an em dash or en dash, depending on whether
      'smart hyphens' are enabled in Tools->Autocorrect options.

2.4.2. Printing the Text File

   Printing to an ASCII text file is currently known to work only on
   Windows-XP and Windows-Vista PCs; appropriate ASCII-output drivers
   for MacOS or Unix boxes running OpenOffice are not currently known.

   To generate .txt output on a Windows-XP or Windows Vista PC, use a
   two-step process. First, generate a .prn file by printing the
   document to a text-only printer. Second, apply post-processing to
   clean up the text and apply 'new page' characters.

Install the "Generic/Text Only" printer, as found under "Generic" in
the available print drivers list. Configure the printer to save to a
file or click 'save to file' when printing. A printed file will have
a .prn file suffix.

The printed output needs to be run through a post-processor to
generate valid Internet-Draft or RFC formatted text. Run the .prn
file through the post-processing as described in Appendix B. This
performs the following operations:

o Omits <CR>s (converts <CR><LF> to <LF> and omits bare <CR>s).

o Converts smart quotes and hyphens to their ASCII counterparts.

o Omits blank lines between the footer of one page and the header of
   the next, and inserts a <CTL-L> (form-feed) between the two.

o Checks for remaining illegal characters (not printable ASCII, CR,
   LF), as required.

2.4.3. XML Support

There is no current support for XML in this template. Although Word
2003 provides XML support, its use is counterintuitive to most Word
users. XML fields are edited using a separate database editor,
rather than in the native Word input screen. Future support for XML
is being considered for a future revision of this template.

3. Changes from RFC 3285

This document and the ".dot" template borrow heavily from RFC 3285
[6]. Notably, all specifics of point sizes, tab locations, and the
automatic date fields are directly from that template. This document
builds on that information as follows.

3.1. Heading Styles

This document redefines Normal, Heading1-9, Header, Footer, and
Caption, rather than defining new RFC-named styles as in RFC 3285.
The use of internal style names is required for proper operation of
outline mode, notably when promoting/demoting sections of text with
subsequent renumbering of headings therein.

Note that this use of redefined standard styles is the common
practice, both in Microsoft-supplied templates as well as templates
from the IEEE and ACM, among others.

   Additional heading styles RFC App and RFC App H1-9 were added to
   support alternate numbering used in appendices, although these styles
   exhibit the previous deficiency of custom styles in outline mode
   (will not auto-renumber on promote/demote).

3.2. References Style

   There are two different reference styles -- RFC References, and RFC
   References Bookmark. The former cross-references using numbers
   (e.g., [1]); the latter uses Word's bookmarking cross-reference
   features to support name/year cross-references (e.g., [Tou2005]).
   The latter is expected to be the preferred format for future I-Ds and
   RFCs.

   RFC References is a body text paragraph style that autonumbers based
   on the "[1]" format. Again, it is common practice, both among
   Microsoft-supplied templates as well as templates from the IEEE and
   ACM, to use autonumbered paragraphs in the body text for references.
   This differs from RFC 3285, in which references were inserted as
   endnotes, rather than as main body text.

   Putting references in endnotes has unexpected behavior. In
   particular, the reference is defined by its first use as a citation
   and is cross-referenced by subsequent citations. Removing the first
   citation removes the reference entirely, despite other cross-
   references. This is not the case with body text autonumbered
   paragraphs, which persist regardless of which cross-references are
   deleted.

   Note that Word does not provide a way to ensure that all RFC
   Reference paragraphs are cross-referenced, i.e., users can insert
   references that are not cited. This can be desired, however, e.g.,
   for bibliographies of supplemental material.

   RFC References Bookmark is a body text paragraph style that does not
   use autonumbering. Currently, authors are expected to manually
   format their references using name-based tags between square brackets
   ("[]"). There is no required format, but a suggested format would
   use the first three letters of the first author, together with the
   four numbers of the document year, e.g., for this document's
   reference "[2]", the reference would now appear as:

   [Bra2004]   Bradner, S., "IETF Rights..."

   The author is expected to select the text of the reference
   "[Bra2004]" and insert a bookmark at that point with whatever name is
   convenient for the author. It can then be cited as usual as a cross-
   reference to the bookmark: [Bra2004]. Note that Word bookmark names
   must start with a letter, and may include numbers but not spaces.

   When two references collide, e.g., for two papers written by Bradner
   in 2004, a trailing lowercase letter should be used to distinguish
   them, e.g., [Bra2004a] and [Bra2004b].

3.3. Boilerplate

   The boilerplate in this template was updated to conform with current
   RFC Editor requirements, notably BCP 78 rights statements, as well as
   pending guidelines for Internet Draft and RFC authors [2][3][5].

   The boilerplate is written in regular text, and can be easily edited
   by authors to keep the template up-to-date as BCP 78 is revised.

3.4. Simplification

   Headers and footers use more conventional tabbing to control
   formatting, rather than tables.

   Users no longer need to avoid the use of smart quotes or hyphens;
   these are automatically translated to RFC-compliant ASCII characters
   during post-processing.

   A number of styles include grouping configuration, designed to
   provide more readable output. In particular, all headings are "keep
   with next" to avoid widowed heading lines, and all list items,
   references, and figure lines are "keep together" to avoid inadvertent
   splitting across page boundaries.

3.5. Ability to Direct Print and Direct View

   This template can print directly to a printer, generating output that
   is line-for-line, page-for-page identical with the compliant ASCII
   text output, excepting minor formatting of hyphens and quotes.

   Further, this template can be previewed in File->Print Preview or
   View->Print Layout, again generating screen images that are line-
   for-line, page-for-page identical with the compliant ASCII text
   output. This allows true WYSIWYG (what-you-see-is-what-you-get)
   editing and printing.

4. Compatibility Issues

   There are no known compatibility issues at this time. This version
   of the template was designed under Windows XP and Word 2002, and has
   also been tested under Windows Vista and Word 2007. It is not yet
   known whether previous versions of Windows/Word are supported using
   this template.

   As noted in Sec. 3.1, this template redefines predefined styles,
   which is common practice.

5. Security Considerations

   There are many security issues in the general use of Microsoft
   operating systems and applications. This template is not known to
   expose any new security issues; it contains no macros as developed
   and deployed. The author considered including the MD5 signatures of
   the current versions of the .dot template and .pl post-processor
   files. The current processor .pl file is included in this document
   as an appendix. The .dot file is updated to track the current
   requirements of the IETF boilerplate, so its MD5 signature cannot be
   included here, but is posted on our website for verification.

6. Acknowledgments

   The author acknowledges the substantial efforts of the authors of the
   previous version of the Word template, Mike Gahrns and Tony Hain [6].
   This document is intended to build upon their work. Thanks also to
   Lars-Erik Jonsson for feedback on this template and post-processor
   script, as well as suggestions on making it more generic to support
   earlier versions of Windows, and to Jixiong Dong for finding an
   obscure bug in the formatting.

   This document was prepared using 2-Word-v2.0.template.dot.

Appendix A. Template Configuration Details

   The Word 97-2003 ".dot" template, compatible with Word 2007, consists
   of a set of default configuration settings, a set of modified and
   newly-defined styles, and an initial template of text. This section
   in particular assumes familiarity with Microsoft Word configuration
   and modification of styles. Note that use of the template does not
   require understanding this section; this merely documents the
   settings already in the ".dot" file.

A.1. Configure Styles

   Modify paragraph styles as indicated. In general use, only the
   redefined existing styles Normal, Heading1-9, Header, Footer, and
   Caption, and new styles RFC Figure, RFC References, RFC List Bullet,
   RFC List Numbered, RFC App, and RFC App H1-5 need be used by authors.

   Other styles, such as redefined TOC1-9, RFC Hidden, RFC H1 - no num,
   and RFC H1 - no TOC no num are used by the template, but are
   typically hidden in the styles list, and need not be explicitly
   applied by authors.

   All measurement units below are in points. Change measurement units
   to points to set.

A.1.1. Redefine Existing Styles

   Redefine the Normal style first; all others are based on Normal,
   except as noted.

   1. Normal: font Courier New, font size 12 point, next style Normal,
      line spacing EXACTLY 12 point, spacing before 0 pt, spacing after
      12 pt, indent left 21.6 pts, widow/orphan control, left tabs at
      every 3 spaces (1 space = 7.2 points, given 72 points/inch and 10
      characters/inch): 21.6, 43.2, 64.8, 86.4, 108, 129.6, 151.2,172.8,
      194.4, 216, 237.6, 259.2, 280.8, 302.4, 324, 345.6,367.2, 388.8,
      410.4, 432, 453.6, 475.2, 496.8.

      NOTE: 12 point fonts are 12 points tall, i.e., 6 lines/inch
      vertically; 12-point Courier is 10 characters/inch horizontally.

   2. Heading 1-9: Normal + indent left 0 pt, hanging 21.6 pts, keep
      with next, set autonumbering as "1. ", "1.1. ", "1.1.1. ", etc.
      Note the space after the right-most period; type this in the

      "number format" field, and leave the "follow number with" field as
      'nothing' (click on the 'more' button to see this field if it is
      not already visible).

      NOTE: change the autonumbering ONCE and in ONE format; in that
      format, change all subsequent levels.

   3. TOC 1-9: Normal + paragraph flush, clear tabs, add new Right tab
      at 504 pts that includes a "..." leader. Add indent left as
      follows: TOC 1: 21.6, TOC 2: 43.2, TOC 3: 64.8, TOC 4: 86.4, TOC
      5: 108, TOC 6: 129.6, TOC 7: 151.2, TOC 8: 172.8, TOC 9: 194.4.

   4. Header: Normal + space after 0 pts, next style Header, clear tabs,
      and add centered tab at 252 pts, tab right at 504 pts.

   5. Footer: Header + next style Footer.

   6. Caption: Normal + centered, autonumbered "Figure #".

A.1.2. Add New Styles

   NOTE: "keep lines together" is optional for lists and references; it
   helps avoid breaking individual items across pages.

   1. RFC Figure: Normal + space after 0 pts, keep with next, keep lines
      together, next style RFC figure (also used for authors'
      addresses).

   2. RFC List Bullet: Normal + custom bulleted, "o" bullet style,
      aligned at 21.6 pts, bullet tab after 43.2 pts, bullet indent at
      43.2 pts, next style RFC List Bullet, keep lines together.

   3. RFC List Numbered: Normal + custom numbered, "1. " number format,
      aligned at 21.6 pts, bullet tab after 43.2 pts, bullet indent at
      43.2 pts, next style RFC List Numbered, keep lines together.

   4. RFC References: Normal + hanging 43.2 pts, outline level body
      text, remove tabs at 21.6 and 43.2 pts, custom numbering with
      format "[1]", numbering left aligned 21.6 pts, number tab space
      after at 64.8 pts, number text indent at 64.8 pts, next style RFC
      References, keep lines together (used for both normative and
      informative references).

   5. RFC References Bookmark: Normal + hanging 72 pts, no num, remove
      tabs less than 72 pts, keep lines together (used for both
      normative and informative references).

   6. RFC Title: Normal + space after 24 pts, centered (used for
      document title).

   7. RFC App: RFC H1 - no num + page break before, custom outline
      numbered, outline number style "Appendix A.", outline level 1,
      follow number with nothing, outline number at 0 pts, outline
      indent text at 0 pts, (used for Appendix titles).

   8. RFC App H1-5: RFC H1 - no num + paragraph level 2-6, custom
      outline numbered, outline number style "A.1." - "A.1.1.1.1.1.",
      follow number with nothing, outline number at 0 pts, outline
      indent text at 0 pts, (link with RFC App at level 2-6), (used for
      Appendix heading levels).

A.1.3. Hidden Styles

   These are used for pre-formatted components or instructions and are
   configured to be hidden from the list of available styles.

   1. RFC H1 - no num: Normal + indent left 0 pts, outline Level 1, keep
      with next (used for base template trailer headers -- Copyright,
      etc.).

   2. RFC H1 - no TOC nonum: Normal + indent left 0 pts, (outline level
      body text, as with Normal), keep with next (used for base template
      front matter headers -- Abstract, Status, TOC header, etc.).

   3. RFC Instructions: Normal, character bold. This style is not
      normally shown and is used for instructions that should be removed
      before publication.

A.2. Define Page Layout

   Configure Page Setup as follows:

   1. Margins:

      Portrait orientation.

         Top: 72   pts
         Bottom: 60   pts (72 + 60 +_660 [55 lines] = 792 [11 in])
         Left: 36   pts (5 characters at 7.2 pts/char)
         Right: 57.6 pts (8.5"-7.2" = 1.3in = 93.6 pts - 36 for left)
         Gutter: 0   pts
         Header: 0   pts
         Footer: 0   pts

      If the error message "One or more margins are outside the
      printable area" appears, select Ignore. This may depend on the
      printer currently selected.

   2. Paper size:

      Paper size "letter".

      Width: 612 pts
      Height: 792 pts (55 lines/page + 132 for top and bottom margins)

   3. Layout:

      Different headers and footers on the first page.

      Header: 72 pts from edge
      Footer: 60 pts from edge

A.3. Insert Boilerplate

   See below for definitions of {ACDt}, {ACDy}, {AEM}, {AEY}, {AP}.

   First page header:

      {blank line}
      {blank line}
      <Working Group Name>{tab}<Initial. Lastname>
      Internet Draft{tab}<Affiliation>
      Intended status: <e.g., Informational>{tab}{ACDy}
      Expires: {AEM}{AEY}
      {blank line}
      {blank line}

   Footer (same on first and subsequent pages):

      {blank line}
      {blank line}
      {blank line}
      <Lastname>{tab}Expires: {AEM} {ACDy}, {AEY}{tab}[Page {AP}]
      {blank line}

   The front text and end text are as specified in BCP 78 [2]. For
   Internet Drafts, the sentence declaring the expiration dates uses
   automatic fields as needed, as with the headers and footers. Note
   that some of these fields will not be updated properly until the
   document has been saved and/or printed.

   A basic document outline with examples is provided in the Word
   template to demonstrate the use of automatic fields, including a
   table of contents.

A.4. Automatic Fields

   Most of these are self-explanatory; the expiry month is set 6 months
   in advance. The expiry year rolls-over (+1), if needed. The expiry
   day is a copy of the current date.

   NOTE: expiry date can generate invalid dates, e.g., April 31,
   February 30, etc.

   o {AEM} means auto expiry month, and is a field code =

      { IF { SAVEDATE \@ "M" * MERGEFORMAT } = 1
      July
        { IF { SAVEDATE \@ "M" * MERGEFORMAT } = 2
        August
        { IF { SAVEDATE \@ "M" * MERGEFORMAT } = 3
          September
          { IF { SAVEDATE \@ "M" * MERGEFORMAT } = 4
            October
            { IF { SAVEDATE \@ "M" * MERGEFORMAT } = 5
              November
              { IF { SAVEDATE \@ "M" * MERGEFORMAT } = 6
                December
                { IF { SAVEDATE \@ "M" * MERGEFORMAT } = 7
                  January
                  { IF { SAVEDATE \@ "M" * MERGEFORMAT } = 8
                    February
                    { IF { SAVEDATE \@ "M" * MERGEFORMAT } = 9
                      March
                      { IF { SAVEDATE \@ "M" * MERGEFORMAT } = 10
                        April
                        { IF { SAVEDATE \@ "M" * MERGEFORMAT } = 11
                          May
                          { IF { SAVEDATE \@ "M" * MERGEFORMAT } = 12
                            June
                            "Fail" *\ MERGEFORMAT
   } *\ MERGEFORMAT } *\ MERGEFORMAT } *\ MERGEFORMAT } *\ MERGEFORMAT }
   *\ MERGEFORMAT } *\ MERGEFORMAT } *\ MERGEFORMAT } *\ MERGEFORMAT
   } *\ MERGEFORMAT } *\ MERGEFORMAT } *\ MERGEFORMAT }

   o {ACDt} means auto current date, and is a field code =

      { SAVEDATE \@ "MMMM d, yyyy" }

   o {ACDy} means auto current day, and is a field code =

      { SAVEDATE \@ "d " }

     o {AEY} means auto expiry year, and is a field code =

        { IF { SAVEDATE \@ "M" * MERGEFORMAT } < 7
            { SAVEDATE \@ "YYYY" * MERGEFORMAT }
            { IF { SAVEDATE \@ "M" * MERGEFORMAT } > 6
                { = { SAVEDATE \@ "YYYY" * MERGEFORMAT } + 1
                 * MERGEFORMAT }
              "FAIL" * MERGEFORMAT * MERGEFORMAT
        } * MERGEFORMAT
      }

     o {AP} means auto page, and is a field code =

        { Page }

Appendix B. Post-Processor Script (Perl)

```
#!/local/bin/perl
# -------------------------------------------------------------------
#                         2-Word-post-v2.0
# Perl post-processor for MS Word RFC/Internet-draft template output
#
#                             J. Touch
#                           touch@isi.edu
#                       http://www.isi.edu/touch
#
#             USC Information Sciences Institute (USC/ISI)
#                 Marina del Rey, California 90292, USA
#                       Copyright (c) 2004-2009
#
# Revision date: May 5, 2009
# -------------------------------------------------------------------
#
# usage:
#         2-Word-post-v2.0.pl [inputfile.txt] > [outputfile.txt]
#
# function:
#     removes indent on each line (blank print margin, typ. 5 chars)
#        converts cr/lf to cr
#     converts 'smart quotes' to regular quotes (single and double)
#     converts 'smart hyphens' (em dash, en dash) to regular hyphen
#     omits blank lines between footer and next-page header
#     inserts formfeed (ff) between footer and next-page header
#     removes end-of-line whitespace
#     checks for illegal chars (not printable ASCII, cr, lf, ff)
#     checks for page lengths exceeded
#     checks for line lengths exceeded
#     prints errors indicating page and line on that page
#
#         illegal character errors are posted to STDERR
#
#         returns the logical OR of codes indicating errors found:
#                 0x00 no error
#                 0x01 if any illegal characters found
#                 0x02 if any page length exceeds $maxpagelen
#                 0x04 if any line length exceeds $maxlinelen
#
# -------------------------------------------------------------------
```

```perl
    # ---------------------------------------------------------------------
    # VARIABLES
    # ---------------------------------------------------------------------

    $pagenum = 1;           # start on page 1, not 0

    $maxpagelen = 66;       # max lines per page

    $maxlinelen = 72;       # max chars per line

                            # specific error codes
    %codes = (
              'none' => 0x00,
              'char' => 0x01,
              'page' => 0x02,
              'line' => 0x04,
            );

    %codestrings = (
                    'none' => '(no error)',
                    'char' => 'invalid character code',
                    'line' => 'exceeded $maxpagelen lines per page',
                    'page' => 'exceeded $maxlinelen chars per line',
                  );

    $errorcode = $codes{'none'};

    $indentlen = -1;        # how many spaces to eat from the beginning
                            # of each line; ought to be 5. negative flag
                            # means it is not yet initialized

    $indentstr = "     ";   # until known otherwise, assume 5 spaces

    $killwhite = 1;         # flag kills space between footer, header
                            # start in 'between footer and header' mode,
                            # so eats all whitespace before the first line

    # ---------------------------------------------------------------------
    # ERROR SUBROUTINE
    # ---------------------------------------------------------------------
    sub printerr ($) {
      my ($errstring) = shift;

      print STDERR "ERROR: $codestrings{$errstring} ",
        "on line $linenum on page $pagenum of text input file\n";
      $errorcode |= $codes{$errstring};
      return;
    }
```

```
   # -------------------------------------------------------------------
   # MAIN
   # -------------------------------------------------------------------

   while ($line = <>) {
     $line =~ s/\r//g;          # remove Unix-style end-of-line
     # if this line is NOT empty, start printing again (see below)
     if ($line !~ /^\s*$/) {
       $killwhite = 0;
       if ($indentlen < 0) {
         # discover margin indent
         $line =~ /^((\s)*)/;
         $indentstr = $1;
         $indentlen = length($indentstr);
       }
     }
     # remove the margin indent
     $line =~ s/^($indentstr)//;
     # change special hyphens, quotes to regular ones
     $line =~ tr/\221\222\223\224\226\227/\'\'\"\"\-\-/;
     # omit end-of-line whitespace
     $line =~ s/\s+\n/\n/g;
     # print unless we're between the end of one page
     # and the beginning of the next
     if ($killwhite != 1) {
       # check to see if we have any invalid characters left
       # 012 = new line, 014 = form feed, 015 = carriage return
       # 040-176 = printable ASCIIs
       if ($line !~ /^([\012\014\015\040-\176])*$/) {
         printerr('char');
         # note -- we don't stop here, so we can find all the
         # unprintable characters in one pass
       }
       $linenum++;
       if ($linenum > $maxpagelen) {
         printerr('page');
       }
       if ($#line > $maxlinelen) {
         printerr('line');
       }
       print $line;
     }
     # check to see if this is the end of a page;
     # if so, then print a form feed (ctl-L), and
     # kill the printing of subsequent empty lines
     if ($line =~ /\[Page \d+\]\s+$/) {
       print "\f\n";
       $killwhite = 1;
```

```
      $linenum = 0;
      $pagenum++;
    }
  }
  exit($errorcode);
```

Informative References

   [1]   Reynolds, J., Ed., and R. Braden, Ed., "Instructions to Request
          for Comments (RFC) Authors", Work in Progress, August 2004.

   [2]   Bradner, S., Ed., and J. Contreras, Ed., "Rights Contributors
          Provide to the IETF Trust", BCP 78, RFC 5378, November 2008.

   [3]   Bradner, S., Ed., "Intellectual Property Rights in IETF
          Technology", BCP 79, RFC 3979, March 2005.

   [4]   RFC Editor, "Formatting RFCS",
          http://www.rfc-editor.org/formatting.html.

   [5]   IETF, "Guidelines to Authors of Internet Drafts". Available as
          1id-guidelines.txt at http://www.ietf.org.

   [6]   Gahrns, M. and T. Hain, "Using Microsoft Word to create
          Internet Drafts and RFCs", RFC 3285, May 2002.

   [7]   Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June
          1999.

Author's Address

   Joe Touch
   USC/ISI
   4676 Admiralty Way
   Marina del Rey, CA 90292-6695
   U.S.A.

   Phone: +1 (310) 448-9151
   Fax: +1 (310) 448-9300
   EMail: touch@isi.edu