            Requirements for an IETF Draft Submission Toolset

Status of this Memo

   This memo provides information for the Internet community. It does
   not specify an Internet standard of any kind. Distribution of this
   memo is unlimited.

Copyright Notice

Abstract

   This document specifies requirements for an IETF toolset to
   facilitate Internet-Draft submission, validation, and posting.

Table of Contents

1. Introduction

   Public Internet-Drafts are the primary means of structured
   communication within the IETF. Current Internet-Draft submission and
   posting mechanisms hinder efficient and timely communication while
   creating an unnecessary load on the IETF Secretariat. The IETF Tools
   team recommends formalization and automation of the current
   mechanisms. This document contains specific automation requirements.

   The IETF Secretariat and many IETF participants have long been
   proponents of automation. This document attempts to reflect their
   known needs and wishes, as interpreted by the Tools team.

2. Scope

   The Draft Submission Toolset discussed in this document is about
   getting a single new version of an Internet-Draft from an IETF
   participant to the IETF draft repository. A single draft version may
   include several formats, and dealing with those formats is in scope
   for the Toolset. Definition and sources of draft meta-information
   (to be used in Secretariat databases and elsewhere) are in scope.
   Submitter authentication and submission authorization are in scope.

   Draft posting may result in various notifications sent to interested
   parties. While this document recommends a subset of notification
   targets, details of notifications are out of scope.

   Creation of new drafts or new draft versions as well as manipulation,
   visualization, and interaction with the drafts already in the
   repository are out of scope. Draft expiration and archiving of old
   draft versions are out of scope.

   The set of requirements in this document is not meant to be
   comprehensive or final. Other IETF documents or procedures may
   require additional functionality from the Toolset. For example, it
   is possible that the Toolset will be required to handle draft source
   formats other than plain text and XML.

3. Notation and Terminology

   The following terms are to be interpreted according to their
   definitions below.

   posted draft: A draft accepted into the public IETF draft repository
      and, hence, publicly available from the IETF web site. Posting of
      a draft does not imply any IETF or IESG review and endorsement.

   draft version: A meant-to-be-public snapshot of an Internet-Draft
      with a meant-to-be-unique version number. Also known as "draft
      revision".

   draft format: Any draft source or presentation format, including
      original and preprocessed XML, original or generated plain text as
      well as PDF, PostScript, and HTML formats.

   primary draft format: The first available draft format from the
      following list: plain text, PDF, PostScript, or XML.

   WG-named draft: A draft for which identifier (a.k.a. filename) is
      known and starts with "draft-SPECIAL-", where SPECIAL is one of
      the following strings: "ietf", "iab", "iesg", "rfc-editor", or
      "irtf". Abbreviated as "WGN draft". Exceptions notwithstanding,
      WG-named drafts are usually controlled by IETF working groups or
      similar IETF-related bodies (and vice versa). The handling of
      such naming exceptions is outside of this document's scope.

   individual draft: A draft other than a WGN draft.

   submitter: A human or software agent initiating submission of an
      Internet-Draft version for validation or posting. In some cases,
      the Secretariat staff does the actual submission, but always on
      behalf of a submitter. In some cases (including but not limited
      to malicious attacks), the submitter is not the draft author.

   expected submitter: A submitter that is authorized by IETF rules to
      post a given draft. This includes a draft author or editor
      (listed in the draft text), a corresponding WG Chair, or an IESG
      member.

   authorized submitter: An expected submitter authenticated by the
      Toolset. Authentication is initially limited to verifying
      submitter access to submitter's email address.

   immediately: Without human interaction or artificial software delays
      and within a few seconds.

The Toolset is specified using a set of normative requirements.
These requirements are English phrases ending with an "(Rnnn/s)"
indication, where "nnn" is a unique requirement number, and "s" is a
single-letter code ("a", "b", or "c") specifying the implementation
stage for the requirement. Implementation stages are documented in
Section 16.

This document specifies the interface and functionality of the
Toolset, not the details of a Toolset implementation. However,
implementation hints or examples are often useful. To avoid mixup
with Toolset requirements, such hints and examples are often marked
with a "Hint:" prefix. Implementation hints do not carry any
normative force, and a different implementation may be the best
choice.

4. Status Quo

   This section summarizes the process for draft submission and posting
   as it exists at the time of writing.

   To get an Internet-Draft posted on the IETF web site, an IETF
   participant emails the draft text to the IETF Secretariat, along with
   an informal note asking the Secretariat to post the draft.
   Secretariat staff reads the note, reviews the draft according to a
   checklist, and then approves or rejects the submission. Draft
   approval triggers the corresponding announcement to be sent to
   appropriate IETF mailing lists. Every 4 hours, approved drafts are
   automatically copied to the IETF drafts repository and become
   available on the IETF web site.

   Collectively, IETF participants submit thousands of Internet-Drafts
   per year (in the year 2000, about 3,000 drafts were submitted; 2002:
   5k; 2004: 7k [secretariat]). About 30-50% of posted drafts are
   WG-named drafts (among some 2,100 drafts, there were about 380 new
   and 290 updated WGN drafts posted in 2003). While no rejection
   statistics are available, the vast majority of submitted drafts are
   approved by the Secretariat for posting.

   It usually takes the Secretariat a few minutes to review a given
   draft. However, since the Secretariat staff does not work 24/7, does
   not work in all time zones, and has other responsibilities, and since
   approved drafts are posted in batches every 4 hours, it may take from
   several hours to several days to get a draft posted. Due to much
   higher demand and fixed processing capacity, postings during the last
   weeks before IETF face-to-face meetings take much longer, creating a
   long queue of unprocessed drafts that are then announced nearly
   simultaneously.

To give IETF face-to-face meeting participants time to review
relevant documents, the Secretariat does not accept Internet-Draft
submissions close to IETF meetings (regardless of whether a draft is
relevant to the upcoming meeting or not).

Many Working Groups have come up with ad hoc solutions to cope with
posting delays. For example, many draft snapshots are "temporarily"
published on personal web sites or sent (completely or in part) to
the group list. Alternative means of publication may effectively
replace official IETF interfaces, with only a few major draft
revisions ending up posted on the IETF web site.

Informal interfaces for submitting and posting drafts discourage
automation. Lack of submission automation increases Secretariat
load, complicates automated indexing and cross-referencing of the
drafts, and, for some authors, leads to stale drafts not being
updated often enough.

Beyond a short Secretariat checklist, submitted drafts are not
checked for compliance with IETF requirements for archival documents,
and submitters are not notified of any violations. As a result, the
IESG and RFC Editor may have to spend resources (and delay approval)
resolving violations with draft authors. Often, these violations can
be detected automatically and would have been fixed by draft authors
if the authors knew about them before requesting publication of the
draft.

Technically, anybody and anything can submit a draft to the
Secretariat. There is no reliable authentication mechanism in place.
Initial submissions of WGN drafts require WG Chair approval, which
can be faked just like the submission request itself. No malicious
impersonations or fake approvals have been reported to date, however.

Lack of authentication is not perceived as a serious problem,
possibly because serious falsification are likely to be noticed
before serious damage can be done. Due to the informal and manual
nature of the submission mechanism, its massive automated abuse is
unlikely to cause anything but a short denial of draft posting
service and, hence, is probably not worth defending against.
However, future automation may result in a different trade-off.

5. Overall Toolset Operation

   This section provides a high-level description for the proposed
   Toolset. The description is meant to show overall operation and
   order; please refer to other sections for details specific to each
   step.

   A typical submitter goes through a sequence of 2-4 web pages and
   associated actions. The number of pages depends on the draft
   validation and meta-data extraction results. For example, validating
   the draft without posting it requires interacting with two web pages:
   Upload and Check. The common case of posting a valid draft without
   manual meta-data adjustments takes three web pages (Upload, Check,
   Receipt).

   Here is a brief overview of pages and actions:

   Upload page: The interface to copy a draft from the submitter's
      computer to the Toolset staging area (Section 6). Multiple
      formats are accepted. The draft is sent to the Check action.

   Check action: Stores the draft in the Toolset staging area, extracts
      draft meta-data, validates the submission (Section 7). Produces
      the Check page.

   Check page: Displays draft interpretation and validation results
      (Section 8). A draft preview may also be given on this page.
      After reviewing the draft interpretation and validation results,
      the submitter has four basic choices: (a) auto-post draft "as is"
      now; (b) make manual corrections and submit the draft to
      Secretariat for manual posting later; (c) cancel submission; or
      (d) do nothing. The automated posting option may not be available
      for drafts with validation errors.

   Automated posting: If the submitter decides to proceed with automated
      posting from the Check page, the system authenticates the
      submitter and may also check whether the submitter is allowed to
      post the draft. If the submitter is authorized, the draft is
      immediately posted, deleted from the staging area, and the
      submitter is notified of the result via email and a Receipt page
      (Section 9).

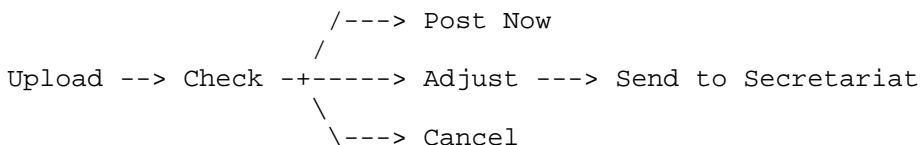   Manual adjustment and posting: If the submitter decides to adjust the
      meta-data, the draft remains in the Toolset staging area, and the
      Adjust action (Section 10) presents the submitter with an Adjust
      page (Section 11). When the submitter makes the adjustments and
      proceeds with manual posting, a pointer to the stored draft and
      its adjusted meta-data is sent to the Secretariat for manual

      processing (Section 12). The submitter is notified of the pending
      Secretariat request via email and a Receipt page.

   Cancellation: If the submitter decides to explicitly cancel the
      submission, the submission state (including the draft) is
      immediately deleted from the Toolset staging area and an
      appropriate Receipt page is generated without further actions
      (R123/a). Cancellation of posted drafts is out of this document
      scope.

   Receipt page: Contains details of a successful or failed draft
      submission and informs the submitter of the next appropriate
      step(s) related to submission result.

   The following informal diagram illustrates the basic submission
   logic:

                        /---> Post Now
                       /
   Upload --> Check -+------> Adjust ---> Send to Secretariat
                       \
                        \---> Cancel

   If the submitter does nothing while the Toolset is expecting some
   response, the abandoned submission times out (R124/a). The timeout
   value depends on the submission state. Hint: A timeout value of one
   hour is probably large enough unless the Toolset is waiting for some
   kind of a 3rd party confirmation (e.g., WG Chair approval). Doing
   nothing is functionally equivalent to explicitly canceling the
   submission, except that explicit cancellation requires immediate
   removal of submission state while the state of submissions marked as
   abandoned is garbage-collected.

   The staging area maintenance algorithms must keep the area in a
   consistent, correct state in the presence of denial-of-service (DoS)
   attacks attempting to overwhelm the area with fake submissions in
   various stages (R67/a). Hint: denial of service to legitimate users
   is acceptable under DoS attack conditions, but corruption of the
   storage area is not.

   The "web pages" this text is referring to are distinct dialogs that
   may be visible to the submitter under the same or different URL and
   that are supported by a single or several server-side programs.

   The Toolset must handle multiple submitters simultaneously submitting
   the same draft (R72/a) and a single submitter simultaneously
   submitting two drafts (R73/a). The latter might happen, for example,
   when the submitter is using several browser windows to submit several

drafts or is submitting drafts via email interface. The term
"simultaneously" means that submission processing times overlap.

Hint: Except for the Upload page, pages contain a submission session
identifier to provide actions with access to stored information. The
identifier is specific to the submission rather than the draft
version or the submitter. While the nature of the web interface
allows the session identifier to be invisible to the submitter, email
communication would need to identify the session so that the
recipient (and Toolset) know the context.

Hint: A single action may correspond to multiple server-side programs
and, vice versa, a single program may implement several actions.
This document does not mandate any specific technology (e.g., Common
Gateway Interface (CGI), PHP, and/or Java servlets) to implement
server-side support. The implementer experience, code reuse across
web and email interfaces, and other factors will determine the right
technology choice.

Hint: Actions preserve and exchange state by storing it along with
the draft. Grouping all submission-specific information in one
subdirectory named using the session identifier may increase
robustness and simplify debugging. Session creation and destruction
can then be logged in a global index.

Ways to partially or completely bypass the Toolset are documented in
Section 14.

It must be possible to transfer the Toolset from one management team
to another, to incorporate work by volunteers, and to allow for
public review of the developed code. To meet these goals, the
Toolset source codes should be publicly available (R152/b) and there
should be an interface to report bugs and request enhancements
(R145/b). Development should be structured to avoid lock-in to
proprietary platforms or backends. The Tools team believes that
developing the Toolset sources under one or more open source licenses
following the Open Source Definition [OSD] would provide an effective
way of meeting these requirements at reasonable cost. Care should be
taken that the licenses selected allow code from different
implementers to be mixed.

Hint: Placing the Toolset source repository at an
open-source-friendly project management site like SourceForge.net
would provide the IETF community with a decent, ready-to-use
interface to access the code, documentation, bug reports, and
discussion forums. Establishing and documenting a simple interface
between the Toolset and external software (e.g., the Secretariat
draft posting scripts) would facilitate availability checks.

The Toolset is meant to be compatible with the Secretariat's tools for handling drafts. Hint: Such compatibility can be achieved by appropriately implementing the Toolset or, in some cases, by modifying existing Secretariat tools.

6. Upload Page

To upload a draft, the submitter goes to a well-known page on the IETF web site (R1/b). There, the draft text can be uploaded using an HTML file upload form. This form provides fields to upload the plain text format of the draft (R2/a) and all other formats allowed by IETF draft publication rules (R3/b). At the time of writing, these formats are: XML ([RFC2629] and [writing-rfcs]), PDF, and PostScript.

Submitted forms are handled by the Check action documented in Section 7.

The Upload page also has a validate-only flag, indicating that an uploaded draft must not be posted and may be deleted immediately after the validation (R74/b). Regardless of the validation results, the stored draft meta-data is marked so that validation-only drafts can be identified and deleted first by garbage collector for the Toolset staging area (R75/b). Drafts uploaded in a validate-only mode cannot be posted (R76/b); they would need to be uploaded again, without the validate-only flag, and the validation results page should explain that (R77/b). This flag is useful for tools using online validation, especially for bulk draft processing. Hint: it may be better to implement this flag as a hidden HTML input field to simplify the interface for human submitters.

7. Check Action

The Check action preprocesses a submission, generates plain text format (if needed, see R70), stores the submitted draft (all formats) in the staging area, and then extracts meta-data and validates each format (R6/a). Errors and warnings are indicated to the submitter in the response via computer-friendly tag(s) and human-friendly text (R7/a).

If any error is found, automated posting becomes impossible (R113/a). This rule applies to all errors, even those that do not refer to R113 and do not explicitly prohibit automated posting. If automated posting is not possible, the Toolset still gives the submitter an option of sending the draft for manual validation and posting (R114/a). Since each submission is treated in isolation, the submitter also has an option of correcting the problem and resubmitting the draft for automated posting.

The manual validation and posting route is a Toolset bypass mechanism
(see Section 14) not meant for fixing problems with the draft itself.
The Secretariat does not generally correct submitted drafts. If the
draft needs tweaking to match submitter's intent, then the draft
should be corrected by the submitter and resubmitted.

It is an error to submit a draft that has neither plain text nor XML
source format (R68/a). XML source is acceptable without accompanying
plain text only if the Toolset successfully generates a draft in
plain text format from the XML source, as a part of the processing
step documented below (R69/b). These rules imply that PDF- or
PostScript-only drafts cannot be auto-posted. Moreover, even manual
Secretariat involvement cannot help with posting these drafts, as the
IETF policy is to always require a plain text format in addition to
PDF or PostScript. Furthermore, drafts containing PDF or PostScript
format must not be auto-posted until the Toolset can validate that
their content matches the plain text format (R143/a).

The draft format acceptance rules above are meant to decrease the
chances that multiple posted draft formats for a single draft contain
substantially different documents. With experience, the rules may be
simplified so that, for example, only submissions containing nothing
but XML or plain text sources can be posted without Secretariat
involvement and all other submissions require manual actions to match
formats or extract meta-data.

7.1. Preprocessing

Submitting compressed drafts is a desirable feature, especially for
submitters behind slow or content-altering links. Compressed draft
formats may be accepted (R150/c). Compressed formats, if any, must
be decompressed during the preprocessing step (R151/c) so that other
processors do not have to deal with compressed formats. Hint: While
this specification does not document a list of supported compression
standards, it is expected that such popular methods as "zip" and
"gzip" should be accepted if compression is supported. Accepting a
collection of draft formats within a single compressed archive may
also be desirable.

XML source containing XML processor <rfc? include="..."> instructions
(PIs) is preprocessed to include references (R8/b). This step is
needed to remove external dependencies from XML sources and to
simplify tools processing posted XML. This document refers to such
XML processor instructions as "include PIs".

The XML preprocessor uses public database(s) to resolve PI references
(R85/b). The Toolset documentation specifies what databases are used
and how PIs are mapped to database entries (R86/b). The Toolset must

not rely on PIs' existence (R87/b) because some XML sources will be
preprocessed before the submission or will be written without PIs.
Hint: Local up-to-date copies of Marshall Rose's reference databases
at xml.resource.org can be used.

Both original and preprocessed XML sources may be posted later. The
original source with include PIs may be useful to the RFC Editor and
generation of diffs (against future or past original sources). The
preprocessed source without include PIs becomes the default public
XML source of the posted draft (R10/b). If any of the include PIs
known to the Toolset cannot be handled, an error is recorded (R11/b),
and the submitter is encouraged to do the preprocessing locally,
before submitting the draft (R111/b).

Uncompressed draft formats other than XML are not preprocessed.

7.2. Processing

When no plain text format of the draft is submitted, but XML sources
are available, the Toolset attempts to generate plain text format
from submitted XML sources (R70/b).

If XML sources are available, the Toolset generates HTML draft format
(R112/c). HTML generation failures should result in warnings, not
errors (R115/c). HTML generation is not meant to be implemented
until the Enhancement Stage is reached (R130/a). In general, HTML
generation is desirable because HTML drafts are usually easier to
navigate than plain text drafts due to improved overall readability
and links. As any Enhancement Stage feature, HTML generation may be
dropped or drastically changed to reflect then-current IETF consensus
and the experience of the first two implementation stages.

Hint: The Toolset implementers should not assume that draft formats
generated by the same tool from the same source format have
essentially the same content. The generation tool may have options
that allow authors to generate content exclusive to a specific
generated format. Such options might be abused.

7.3. Storage

The Check action needs to store all draft formats so that
successfully validated drafts can later be auto-posted at submitter
request. The action stores all submitted formats of the draft in a
staging area dedicated to the Toolset (R12/a). If, after garbage
collection, the staging area is full (i.e., the total used size has
reached the configured maximum capacity), the submitter and the
Secretariat are notified of a fatal error (R13/a).

7.4. Extraction

   The Toolset extracts meta-data from the following stored draft
   formats: plain text (R131/a), XML (R132/b), and other (R133/c). If a
   meta-data extraction fails, the Toolset records an error (R15/a).
   Meta-data extraction is necessary to validate and post the draft.
   Extraction from all formats is necessary to validate that all
   meta-data matches across all formats (in addition to and before the
   Toolset can validate that the contents matches as well).

   Section 16 documents a non-obvious implementation schedule related to
   the above requirements. When only partial support for format
   interpretation is available, only interpreted formats are subject to
   extraction and validation requirements. In other words, if the
   Toolset does not yet support interpretation of a given format, then
   the corresponding information is stored and made available "as is",
   regardless of the actual content.

   The draft interpreter extracts the following meta-data from each
   draft format (R16/a):

   identifier: Also known as draft "filename". For example,
      "draft-ietf-sieve-vacation-13".

   version: A non-negative integer number representing draft version
      number (also known as draft revision number). For example, the
      number 7 in "draft-ietf-sieve-vacation-07". The number is usually
      rendered using two digits, padding with "0" if necessary.

   name: The common part of all draft identifiers for all versions of
      the same draft. In other words, a draft identifier without the
      version component. For example, "draft-ietf-sieve-vacation" in
      "draft-ietf-sieve-vacation-07".

   WG ID: Working Group identifier. For example, "sieve" in
      "draft-ietf-sieve-vacation-07" is a WG ID. The WG ID value is
      empty for drafts that are not WG-named drafts.

   WG flag: True for WGN drafts and false for all other drafts. For
      example, "true" for "draft-ietf-sieve-vacation-13". This flag
      only influences the further handling of initial (version 00) draft
      submissions, as far as the current document is concerned.

   title: A human-friendly draft title. For example, the title of this
      document is "Requirements for an IETF Draft Submission Toolset".

   authors: A list of all draft authors. Each author's name and email
      address are extracted.

abstract: The draft abstract text.

creation date: The draft version creation date.

expiration date: The draft version expiration date.

size: The number of pages and octets in the primary format of the
    draft. The definition of a page depends on the format and may be
    imprecise or arbitrary for some formats.

Failure to extract any field results in error (R95/a).

The Toolset requires author email addresses because they are
essential for notifying co-authors that their draft has been posted.
If there are no such notifications, a submitter adding a co-author to
the draft without the co-author's consent may not be caught for a
while. Such "surprise" co-authorships have happened in the past and
can be quite annoying. However, since the Toolset does not solicit
co-authors' consent to post a valid draft (and such solicitation
would not go beyond email control verification anyway), it is not
possible to stop a malicious submitter from adding co-authors without
their knowledge.

Like other meta-data items above, draft creation and expiration dates
are extracted from the draft; their values do not depend on the
actual submission time (i.e., the time the Check action starts).
However, the validation procedure (see Section 7.5) may declare any
extracted date invalid after taking into consideration current (i.e.,
submission) time, IETF draft expiration rules, and other factors
external to the draft.

7.5. Validation

   Drafts need to be validated to catch broken submissions. Validation
   also helps educate or warn authors of problems that may become
   show-stoppers when the draft is sent for IETF Last Call and IESG
   review. IETF standards have to follow a set of syntax and semantics
   requirements (see the "ID-NITS" document at
   <http://www.ietf.org/ID-Checklist.html>. Most of those requirements
   are not enforced for Internet-Drafts. However, following them may
   improve draft quality, reduce the IESG load, and increase the chances
   of the draft being approved as an RFC.

   When validating a given draft, it is important to distinguish between
   absolute requirements and desirable draft properties. Both
   categories are checked for, but violations have different effects
   depending on the category. The two categories are detailed in the
   following subsections.

When a valid draft is being posted and submitter authorization or
co-author notification is performed, validation results should be
included in the email (R81/b) so that the submitter can see meta-data
extraction and validation warnings. Note that these results cannot
include errors since only valid drafts can be posted.

7.5.1. Absolute Requirements

Violating any of these requirements would prevent a draft from being
automatically posted (R17/a). The offending draft would have to be
fixed or submitted for manual posting, with an explanation as to why
the absolute requirements need to be violated (or why the Validator
mis-detected violations). These explanations may speed up the
Secretariat posting decision and may help the Secretariat to improve
the Toolset implementation.

   1. All available meta-data entries must match across all submitted
       draft formats (R18/a). For example, if the interpreter managed
       to extract a draft title from both the plain text and the PDF
       format, both titles must match. This requirement prevents
       accidental submission of mismatching formats.

   2. Version 00 of a Working Group draft has a corresponding Working
       Group approval (R20/a). This approval can be relayed before or
       after the first draft submission, by a Chair or Secretary of the
       WG. See Section 7.5.4 for related requirements.

   3. The draft ID must be correct (R22/a), including the draft version
       number value and format. Single-digit draft version numbers must
       be left-padded with "0". Draft version numbers must start with
       zero and increase by one with every new version. To satisfy this
       requirement, the Toolset would have to consult the repository of
       already posted drafts, including expired ones. If the IETF
       infrastructure cannot handle version numbers greater than 99, the
       Toolset must reject them (R158/a).

   4. An IETF IPR Statement and other boilerplate required for drafts
       according to [RFC3978] and [RFC3979] (or successors) must appear
       in the draft text (R23/a).

   5. The extracted creation date of the draft version must be within 3
       days of the actual submission date (R159/a). Hint: Implementers
       should be careful to handle creation dates that appear to be in
       the past or in the future, due to possible time zone differences.
       Making the most forgiving/permissive assumption about the time
       zone should suffice.

   6. The draft version expiration date obeys IETF draft expiration
       rules.

   7. No IETF submission blackout period applies. Hint: IETF blackouts
       must be enforced based on submission time, not possible draft
       creation time.

   8. Posting the draft must not result in any DoS attack threshold to
       be crossed (R97/a). Specific thresholds are documented in
       Section 7.5.3.

   9. XML sources (if available) are valid with respect to the XML
       format [XML] (R153/c) and XML Document Type Definition (DTD) for
       IETF drafts (R154/c). Note that during the first two
       implementation stages, the corresponding validation failures
       result in warnings and not errors (see Section 7.5.2).

   The XML DTD for IETF drafts is documented in [RFC2629] with recent
   changes available in [writing-rfcs]. Hint: Bill Fenner's "RFC 2629
   validator" at http://rtg.ietf.org/~fenner/ietf/xml2rfc-valid/ (or its
   derivative) may be useful for XML format and DTD validation.

   Hint: If the extracted meta-data differs in the submitted draft
   formats, the Toolset should use the meta-data from the most "formal"
   format when populating the form entries for manual submission. On
   the other hand, if most extracted entries come from a less "formal"
   format, the Toolset may choose that format instead. For example, XML
   source can be considered more "formal" than plain text format. The
   Toolset may also offer the submitter an option to specify which
   format should be used for populating the form. It is probably a bad
   idea to mix-and-match the conflicting entries extracted from multiple
   formats. Instead, either one format should be chosen when populating
   the form or the form should contain several meta-data sections, one
   for each format. The error messages will contain the exact mismatch
   information.

   Hint: The Toolset should accept dates without the day of the month,
   as long as IETF rules do not prohibit them. The Toolset should make
   the most forgiving/permissive assumption about the actual day of the
   month when validating day-less dates.

7.5.2. Desirable Features

   Violating any of the following requirements does not prevent the
   submitter from auto-posting the draft (R24/a) but results in a
   warning (R160/a). Each warning explains the corresponding violation
   and provides advice on how to comply (R161/b). Hint: To ease
   maintenance and encourage 3rd party updates, detailed explanations

and/or advice may be available as a resource separate from the
Toolset.

1. All automatically testable nits in the "ID-NITS" document at
    <http://www.ietf.org/ID-Checklist.html> (R116/b) and
    automatically testable guidelines at
    <http://www.ietf.org/ietf/1id-guidelines.txt> (R157/b). The
    Toolset should use external tools to check these nits and
    guidelines rather than embed checking code (R117/a). Hint:
    Henrik Levkowetz's idnits tool can be used
    (http://tools.ietf.org/tools/idnits/) and other tools can be
    written or adopted.

2. New draft versions are expected (R21/b). For example, version 00
    of an individual draft is always expected, while posting a new
    version of a draft already under the IESG review should generate
    a warning.

3. If both XML and plain text formats are submitted, the submitted
    plain text matches what can be generated based on submitted XML
    (R146/b).

4. The previous version, if any, was posted at least 24 hours ago
    (R96/b). This warning may prevent some human errors, especially
    when multiple authors may post the same draft.

5. XML sources (if available) are valid with respect to the XML
    format (R155/b) and XML DTD for IETF drafts (R156/b). These
    requirements become absolute after the second implementation
    phase. See Section 7.5.1 for related information.

When comparing generated and submitted plain text formats to satisfy
R146, a standard word-based diff is sufficient for initial Toolset
implementations (R147/b). However, a custom fuzzy matching function
can be developed (R148/c) to minimize false warnings due to, for
example, draft text formatting differences. When differences are
detected, a complete diff may be provided on a separate page
(R149/c), in addition to the warning.

Hint: When comparing generated and submitted plain text formats, the
Toolset may try several recent xml2rfc versions for plain text
generation, to eliminate warnings due to differences among xml2rfc
versions.

7.5.3. DoS Thresholds

   The following table documents DoS attack thresholds for various draft
   categories. Daily limits correspond to all drafts (and all draft
   formats) within the category. Other thresholds may be introduced and
   these initial thresholds may be adjusted as necessary. The
   thresholds are likely to become more smart/dynamic with experience.

                        DoS attack thresholds:

   +--------------------------------+--------------+-----------+
   | category | versions/day | MB/day |
   +--------------------------------+--------------+-----------+
drafts with the same draft name	3	5
drafts with the same submitter	10	15
WGN drafts with the same WG ID	30	45
all drafts	400	200
   +--------------------------------+--------------+-----------+

   The thresholds are meant to limit destructive effects of DoS attacks
   (e.g., full disks cause other tasks to fail), allow for capacity
   planning (e.g., how much storage space the Toolset needs), and limit
   annoying side effects of "too many" drafts being posted (e.g., when a
   person receives posting notifications about a given draft or a given
   working group). The Toolset should warn the Secretariat if total
   submissions are approaching any threshold (R134/b). Hint: Bandwidth
   available for submissions may need to be throttled (on a network
   subnet basis?) to make reaching the daily size quota (with malicious
   intent) difficult.

7.5.4. WG Approval

   For version 00 of a WGN draft, the Toolset checks for an existing WG
   approval (R125/a). If (a) no approval exists, and (b) the Toolset
   does not support the "waiting for WG approval" feature, the Toolset
   records an error (R135/a).

   If (a) no approval exists, (b) the Toolset supports the "waiting for
   WG approval" feature, and (c) the draft cannot be posted even if WG
   approval is received, then the Toolset records a warning that a WG
   approval would be required once all errors preventing draft from
   posting are fixed (R137/b).

   If (a) no approval exists, (b) the Toolset supports the "waiting for
   WG approval" feature, and (c) the draft can be posted if WG approval
   is received, then the Toolset explains the situation to the submitter
   and asks whether an explicit approval from the WG should be solicited
   or expected (R126/b). If the approval should be solicited, it is

   solicited by the software or the submitter. If appropriate, the
   Toolset puts the submission into a "waiting for WG approval" state
   until the expected approval is available (R127/b). Otherwise, the
   Toolset records a "no WG approval is expected" error (R138/b).

   The details of manual or automated solicitation for WG approval is
   outside the scope of this document. Hint: Initially, the submitter
   will be responsible for soliciting a WG Chair approval, but this
   process should eventually be automated.

   Details of the approval recording and access interfaces as well as
   the mechanism to resume the submission upon approval are out of this
   document's scope.

8. Check Page

   The Check page, created by the Check action, displays extracted draft
   meta-data and validation results (R25/a). The purpose of the page is
   to allow the submitter to verify whether the stored draft and
   automatically extracted meta-data match the submitter's intent and to
   be informed of validation problems.

   Meta-data items specified in Section 7.4 that failed validation
   checks must be marked specially (rather than silently omitted or
   ignored) (R26/b). Hint: rendering those items in red, with links to
   corresponding validation errors or warnings, may force authors to pay
   attention.

   Validation messages include both errors and warnings. Each
   validation message refers to normative document(s) containing the
   corresponding validation rules (R27/b).

   The Check page allows the submitter to enter external meta-data
   (Section 8.1) (R28/a). If validation was successful, an
   "automatically post the draft now" button is provided (R29/a).
   Regardless of validation results, "adjust and post manually" and
   "cancel" buttons are provided (R30/a).

   The Check page provides a preview of the draft plain text format
   (R31/a), with a link to see how the entire draft (with all its
   formats) would look if posted (R82/b). Hint: the Check page preview
   should be sufficiently long to let authors detect obvious draft
   mismatch or misinterpretation errors but short enough to avoid
   dominating the page. Displaying the first line of the draft through
   the last line of the abstract may be sufficient.

   For draft updates, the Check page reports the time and the submitter
   of the last update (R83/b). This information is especially useful

   when multiple authors are working on the same draft. The page also
   provides a link to generate a diff against the last posted version
   (R84/c).

8.1. External Meta-Data

   The Check page solicits the following meta-data from the submitter.
   This information must be supplied by submitter because it cannot be
   extracted from the draft:

      Submitter email address (R32/a). When submitter is not an
      expected submitter (see Section 3), automated posting is not
      possible and the draft has to go through the Secretariat (R98).
      Hint: A set of checkboxes next to extracted author names along
      with a "none of the above" checkbox with an input field would
      suffice.
      A list of drafts replaced by this draft (R33/c). This is useful
      to make replaced drafts invisible. This document does not specify
      any actions necessary to actually replace an existing draft
      because existing draft manipulation is out of scope, and because
      security concerns and other complications of such actions would be
      better addressed by a separate specification.
      Primary email address for discussion of this draft (R71/b). Hint:
      The Toolset can suggest the WG mailing list address for WGN
      drafts, (submitting) author address for individual drafts, or even
      the first email address in draft text. Offering a few likely
      addresses instead of relying exclusively on user input would also
      reduce the number of typos.

   Except for the submitter email address, external meta-data is
   optional (R109/a).

   If a given submitter email address belongs to an expected submitter
   (i.e., belongs to one of the categories below), the Toolset performs
   submitter authentication during a Post Now action (R19/a).
   Otherwise, an error is reported (R118/a).

   The following possible expected submitters are identified by the
   Toolset, without any Secretariat intervention:

      For version 00 of a draft, any submitter (R119/a).
      For version N+1 of a draft, an author of version N of the same
      draft (R120/a). This requirement only needs to be satisfied for
      drafts for which Nth version was posted using the Toolset; other
      drafts may not have the meta-information available that is
      required to reliably get a list of authors.
      For a WGN draft, a Chair of the corresponding WG (R121/b).
      For any draft, an IESG member (R122/c).

9. Post Now Action

   The Post Now action checks that the draft has been successfully
   validated (R34/a), validates external meta-data (including submitter
   email address) (R35/a), and posts the draft (R36/a). The submitter
   is notified of the action progress and the final result (R37/a).

   The external meta-data contains the submitter's email address. As a
   part of the validation procedure, the Post Now action authorizes the
   submitter. The initial action implementation checks that the
   submitter has access to email sent to that address (R38/a).
   Eventually, the Toolset should accept client certificates signed by
   IETF, PGP-signed email, and/or other forms of client-side
   authentication to eliminate the weak and annoying email access check
   (R110/c). If submitter authentication fails, the submission
   eventually and silently times out (R39/a).

   The Toolset provides both web (R99/a) and email (R139/b) interfaces
   for confirming email access. Hint: To check submitter's access to
   email, the tool can email a hard-to-guess cookie or token to the
   submitter's address. To continue with the submission, the submitter
   is requested to paste the cookie at the specified URL, go to the
   token-holding URL, or respond to the email.

   Immediately after sending an email to the submitter, the Post Now
   action generates an intermediate Receipt page that explains Toolset
   expectations and provides the submitter with the submission ID
   (R100/a). That number allows the Secretariat to troubleshoot stuck
   submissions (R101/a) and can also be used for checking submission
   status without Secretariat involvement (R140/b).

   Immediately after posting the draft, the Toolset notifies all authors
   (with known email addresses) of the posting (R102/a). The
   notification email contains the information available on the
   "successful posting" Receipt page described below (R103/a).

   If draft posting is successful, the submission state is marked as
   available for deletion (R105/a) so that the garbage collection
   routine eventually deletes it.

9.1. Receipt Page

   A successful Post Now action reports at least the following
   information on the final Receipt page (R104/a):

   o the draft ID and a link to the draft status page.

   o the draft title, authors, and abstract.

    o   the submission ID.

    o   a link to the draft submission status page (when status queries
        are supported, see R140).

    o   the submitter's name and email address.

   The primary purpose of the Receipt page is to inform all draft
   authors that (supposedly) their draft has been posted. The secondary
   purpose is to let authors create a permanent record of the event and
   troubleshoot postings. The same information should be sent to other
   parties interested in the draft (e.g., to the WG mailing list), but
   3rd-party notification specifics are out of this document's scope.

10. Adjust Action

   The Adjust action generates the Adjust page (R40/a), populating it
   with available extracted meta-data and external meta-data, as well as
   validation results and a preview. Some information may be missing,
   depending on draft interpretation and the success of preview
   generation.

11. Adjust Page

   The Adjust page includes the same information as the Check page, but
   allows the submitter to adjust all extracted draft meta-data (and,
   naturally, external meta-data) at will (R41/a). Such adjustment is
   necessary when automated extraction failed to extract correct
   information. To avoid any mismatch between draft and its meta-data,
   adjusted drafts cannot be automatically posted and require manual
   validation by the Secretariat (R42/a). Secretariat staff can post
   drafts with adjusted meta-data as described in Section 14.

   The Adjust page allows the submitter to enter an informal comment
   explaining why adjustments are necessary and automated posting mode
   cannot be used (R48/a). Such comments may be essential for the
   Secretariat in their efforts to troubleshoot the problem.

   The "post manually" and "cancel" buttons are provided (R43/a). The
   former is backed by the Post Manually action (Section 12).

12. Post Manually Action

   The Post Manually action sends adjusted meta-data and a draft pointer
   to the Secretariat for manual validation and posting (R44/a). A
   receipt page is generated, instructing the submitter to wait (R45/a).
   The Secretariat will notify the submitter once the draft is posted or
   rejected. This notification is sent by the Toolset if the
   Secretariat is using the Toolset to post the draft (R46/a).

13. Receipt Page

   The Receipt page is generated by various actions to inform the
   submitter of the current submission status and further actions. The
   contents of the page is likely to be highly dependent on the action
   and state for which receipt is being generated. This section
   documents general requirements applicable to all actions and states.

   The Receipt page should give the submitter a Uniform Resource
   Identifier (URI) or another identifier that can be used by
   Secretariat for manual troubleshooting of the submission (R63/a).
   The identifier should be perpetual (R64/a) even though the associated
   details are likely to be eventually lost (e.g., draft submission data
   and logs are deleted from the staging area as a part of the garbage
   collection routine). Hint: Tools should distinguish old identifiers
   from invalid ones; when a given identifier is referring to deleted
   data, the tools accepting the identifier should inform their users
   that the identified submission is recognized, but the related
   information has expired.

   The Receipt page should give the submitter a Secretariat
   point-of-contact to report submission problems (R65/a).

14. Bypassing the Toolset

   A buggy Toolset implementation or unusual circumstances may force a
   submitter to submit a draft to the Secretariat for manual processing.
   This can be done by choosing the "manual posting" route supported by
   the Toolset (R47/a) or, as a last resort, by emailing the draft
   directly to Secretariat. In either case, an informal "cover letter"
   has to accompany the draft. The letter should explain why the
   automated interface cannot be used.

   When processing manual submissions, the Secretariat may be able to
   use the Toolset. A Manual Check page similar to the default Check
   page provides authenticated Secretariat staff with editable meta-data
   fields and a "force posting" action (R50/b). The forced posting
   action accepts meta-data fields "as is", does not verify submitter
   access to email or WG draft authorization, and posts the draft as if

no validation errors were found (R51/b). The Manual Check page
should still contain all the errors and warnings identical to those
seen by ordinary submitters (R106/b) so that the Secretariat knows
what the Toolset is unhappy about (if anything).

Using manual processing may result in significant posting delays.
Generated submission receipts or notifications ought to give the
submitter an expected processing time estimate (R53/a).

The intent of this mode is to provide a way for submitters to bypass
bugs or limitations of the automated mechanisms in order to post an
"unusual" draft or to post a draft under "unusual" circumstances.
One example would be a draft that does not contain standard IETF
boilerplate but has a special IESG permission to post the draft with
the experimental boilerplate. Another example is a draft that fails
automated validation tests due to a validator bug.

The bypass mode is also likely to be used (effectively) by the
majority of submitters during the Trial stage of the Toolset
implementation, when few submitters know about (or are allowed to
use) the Toolset.

15. Email Interface

The Toolset should have an email interface for automated posting of
valid drafts (R55/b). While virtually every documented Toolset
functionality can, technically, be implemented behind an email
interface, features other than posting of valid drafts are believed
to be prohibitively awkward to implement or use via email.

The email interface accepts a draft as a set of email part(s) (one
per draft format) (R56/b). For example, the plain text format can be
submitted in the "body" of the email message, while XML source format
can be optionally sent as an "attachment" of the same email message.
Each part can either contain the actual format data (R141/b) or a
single URL pointing to it (R142/c). In the latter case, the Toolset
has to fetch the format data. Details of the URL-fetching option are
not documented here, but it is assumed that HTTP URLs are supported
(at least), and fetching errors are reported. This document does not
specify how the format of each email part is determined, but it is
assumed that MIME type and content would need to be analyzed.

After accepting the draft, the Toolset uses the sender's email
address to select the submitter identity (R57/b), checks the
submission (R58/b), and posts the draft if the check is successful
(R59/b). The submitter should be notified of the outcome of the
draft submission via email (R60/b). Other requirements for the web
interface (including requirements on submission preprocessing, draft

   validation, submitter authentication, draft posting, and
   notification) apply to the email interface.

   Therefore, a typical successful submission via email interface may
   result in the following exchange of messages ("T" is for "Toolset",
   "S" is for "submitter", and "A" is for "all authors and submitter"):

      S-->T: the draft version

      S<--T: a challenge to verify email access

      S-->T: a response to the challenge

      A<--T: warnings and the receipt

   where the message containing the challenge may include warnings as
   well.

   When draft validation fails, the following emails may be exchanged:

      S-->T: the draft version

      S<--T: errors and receipt

   Email parts/attachments that are not recognized as draft formats are
   not considered as draft formats. Such parts are ignored by the
   Toolset (R107/b), except that a warning is generated for each
   unrecognizable part containing more than whitespace (R108/b). These
   two requirements are meant to make the interface robust in the
   presence of email signatures and other parts outside of the submitter
   control.

   Hint: Toolset actions can be implemented to support email and web
   interfaces without code duplication.

   While both web and email interfaces allow for fast posting of valid
   drafts, there are significant differences between the two interfaces.
   Primary advantages of the email interface are:

   off-line mode: A submitter can do all the manual work required to
      submit a draft while being disconnected from the network. The
      email client actually submits the draft when connectivity is
      regained.

   poor connectivity: Email systems are often better suited for
      automated transmission and re-transmission of emails when network
      connectivity is poor due to high packet loss ratios, transmission
      delays, and other problems.

   convenience: Some IETFers consider email interfaces as generally
      "more convenient".

   Primary advantages of the web interface are:

   confirmation: A submitter is given a chance to verify that automated
      extraction of meta-data produced reasonable results. Other useful
      confirmations are possible (e.g., "Are you sure you want to post a
      version of the draft that was updated 30 seconds ago by your co-
      author?").

   validation: A submitter can validate the draft without posting it.

   quality: Non-critical warnings may prompt the submitter to postpone
      posting to improve draft quality.

   manual adjustments: The submitter can adjust extracted meta-data and
      ease Secretariat work on manually posting an unusual draft.

   meta-data: The submitter can specify optional external meta-data
      (that cannot be extracted from the draft itself). For example, an
      email address for draft discussion can be specified.

   context help: The web interface makes it easy to provide links to
      extra information about input fields, errors, posting options,
      deadlines, etc.

   opaqueness: Files submitted via the web interface are arguably less
      susceptible to various in-transit transformations and
      misinterpretation than emails. Emails are often mutated by mail
      agents (e.g., automated disclaimers added by senders and extra
      line feeds added by recipients).

   convenience: Some IETFers consider web interfaces as generally "more
      convenient".

16. Implementation Stages

   This section defines the Toolset implementation stages or phases.
   There are three consecutive stages, marked with letters "a", "b", or
   "c". Earlier-stage requirements must still be satisfied in later
   stages. All requirements need to be interpreted and evaluated in the
   context of the current stage and the currently implemented features.
   For example, requirement R68 applies to the first stage but refers to
   XML draft format that may not be supported until the second stage. A
   correct interpretation of R68 until XML support is added is "it is an
   error to submit a draft without a plain text format".

Unless otherwise noted, requirements listed in later stages may be
covered in earlier stages, but do not have to be. If the
implementers decide to add some functionality from a future stage,
they have to be very careful to satisfy all requirements related to
that functionality. Unfortunately, there is no reliable, pragmatic
way to identify "all requirements" related to a given feature.

(a) Trial Stage: Initial basic implementation to test major concepts
    and relieve the Secretariat from handling the most common
    submission case. This stage focuses on plain text draft
    submission via the web interface. The trial stage should take a
    dedicated professional about 45 calendar days to finish (i.e., to
    comply with all the listed requirements).

(b) Production Stage: Support for all major features. Once this
    stage is completed, the Secretariat should only handle unusual
    draft submissions. This stage should take about 100 calendar days
    to finish. Gradual release of implemented features is possible
    and expected. Specifically, the XML support is expected before
    email interface support.

(c) Enhancement Stage: A never-ending stage focusing on sophisticated
    features (e.g., draft interpretation or validation) that improve
    the overall quality of the Toolset. This stage is documented
    primarily to highlight the overall direction of the Toolset; its
    requirements are often imprecise and many are expected to change.

Implementation experience is likely to result in changes of the
Toolset requirements. Such changes should be documented as a part of
stage evaluation activities.

17. Testing

Before letting the Toolset go live, thousands of posted drafts can be
used to test the meta-data extraction algorithms. Such testing can
minimize the number of drafts being sent on for manual handling
because of meta-data extraction failure.

Other Toolset features may also be testable using posted drafts. A
simple pair of scripts can be used to test basic functionality of the
web and email interfaces.

Hint: The IESG may require test results before accepting the initial
implementation. If automated, the above approach can be used for
regression testing as well.

18. Security Considerations

   Removing humans from the draft submission and posting process (a.k.a.
   automation) requires adding features to make the Toolset reliable in
   the presence of denial-of-service (DoS) attacks and attempts to
   corrupt the draft repository. Ideally, the Toolset needs to resist
   both premeditated malicious actions and good-intent accidents.

   This document contains specific requirements to minimize the impact
   of DoS attacks (e.g., R97). The requirements are designed with the
   assumption that it is acceptable for the Toolset to block valid
   submissions during a DoS attack as long as the Toolset maintainers
   are notified and already posted drafts are not damaged.

   This document also contains many specific requirements related to
   detection of drafts violating IETF posting rules. Those requirements
   help reduce the number of "bad" drafts posted by mistake but do not
   offer reliable protection from submitters with malicious intent:
   Since automated tools do not truly understand drafts (and will not do
   so in the foreseeable future), it is technically possible to post a
   rogue draft violating IETF posting rules. For example, a draft may
   contain abstract text that makes the IETF-approved IPR statements
   following the abstract meaningless or legally non-binding.

   Stronger submitter authentication may be required to deter malicious
   submitters. The documented authentication mechanism (i.e., read
   access to one's email) is deemed appropriate for deployment of the
   first versions of the Toolset, under close Secretariat supervision.
   Hint: to increase chances of detecting problems early enough, it may
   be a good idea to automatically inform a designated human of every
   posted submission (during initial deployment of the Toolset).

19. Compliance

   A Toolset implementation is compliant with this specification if it
   satisfies all normative requirements (i.e., the phrases marked with
   "Rnnn" as defined in Section 3). Compliance should be evaluated for
   each implementation stage as some requirements do not apply to some
   stages.

   The IESG evaluates implementations and interprets requirements as
   necessary.

Appendix A.   Comparison with Current Procedures

   This section summarizes major differences between the draft
   submission approach currently in use by IETF and the proposed
   Toolset, including violations of the current IETF rules.

   o The Toolset allows posting of XML and PDF draft formats. The XML
      format is not currently accepted by the Secretariat, and legality
      of PDF acceptance by the Secretariat has been questioned. XML
      sources should be accepted to enable IETF tools and participants
      to have access to raw draft meta-data and content. They are also
      useful to the RFC Editor and, hence, it is a good idea to validate
      and get them "into the system" early. The latter argument applies
      to PDF drafts as well, although the first Toolset versions are not
      expected to interpret PDF drafts.

   o The Toolset may eventually generate HTML draft formats from XML
      draft sources (see R112). Currently, IETF does not provide HTML
      draft formats -- the Secretariat does not accept HTML sources and
      no HTML is generated from accepted draft sources. Note, however,
      that this document does not suggest that the Toolset should
      eventually accept drafts in HTML format.

   o The Toolset defines "WGN draft" as a draft whose name starts with
      "draft-ietf-". All other drafts are treated as individual drafts.
      Currently, an IETF WG does not have to follow a single WG draft
      naming format. Thus, the 00 version of a draft that the WG
      considers a WG draft can be posted by the Toolset without WG
      consent. Affected WGs would have to deal with the consequences of
      their decision not to use a common naming format. The Tools team
      suggests that IETF requires WGs to name their drafts using a
      single format to minimize confusion. Hopefully, there are no
      humans named "Ietf" or, at least, none of them wants to auto-post
      individual drafts.

   o For some drafts, the Toolset verifies that the submitter is
      "expected" (e.g., an author of the previous draft version or WG
      Chair). Currently, the Secretariat does virtually no such
      verification, but an email submission interface and a human
      presence in the submission loop have apparently been sufficient to
      prevent massive automated attacks. The change is needed to
      prevent a simple script from using the web interface to overwrite
      posted IETF drafts with junk. Hopefully, the IETF will eventually
      have a decent authentication scheme making the submitter checks
      simpler, less rigid, and more transparent.

   o The Toolset will automatically notify authors of posted drafts.
      Currently, neither the submitter nor any of the co-authors are
      explicitly notified when the draft is posted. Notification is
      meant, in part, to allow co-authors to detect cases where their
      name is put on the authors list without permission. Eventually,
      there will be a general IETF mechanism to allow 3rd parties such
      as ADs, chairs, or reviewers to register for notifications about
      draft postings.

   o The Toolset may eventually accept compressed drafts (see R150).
      Currently, the Secretariat does not accept "zip" archives due to
      virus contamination concerns. A proper implementation of the
      Toolset must address such concerns, while the Secretariat may
      still need to reject certain formats if they are submitted via the
      manual route.

Appendix B. Acknowledgements

   The author gratefully acknowledges the contributions of Harald Tveit
   Alvestrand (Cisco), Brian E. Carpenter (IBM), Frank Ellermann, Bill
   Fenner (AT&T), Barbara B. Fuller (Foretec), Bruce Lilly, Henrik
   Levkowetz (Ericsson), Larry Masinter (Adobe), Keith Moore (University
   of Tennessee), Pekka Savola (Netcore), Henning Schulzrinne (Columbia
   University), and Stanislav Shalunov (Internet2).

   Special thanks to Marshall Rose for his xml2rfc tool.

Normative References

   [RFC2629]   Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629,
                  June 1999.

   [RFC3978]   Bradner, S., "IETF Rights in Contributions", BCP 78,
                  RFC 3978, March 2005.

   [RFC3979]   Bradner, S., "Intellectual Property Rights in IETF
                  Technology", BCP 79, RFC 3979, March 2005.

   [XML]   World Wide Web Consortium, "Extensible Markup Language
                  (XML) 1.0", W3C XML, February 1998,
                  http://www.w3.org/TR/1998/REC-xml-19980210.

Informative References

   [writing-rfcs] Rose, M., "Writing I-Ds and RFCs using XML (revised)",
                  Work in Progress, April 2004.

   [secretariat] "Private communication with the IETF Secretariat",
                  2004.

   [OSD]   "The Open Source Definition, version 1.9", Open Source
                  Initiative, 2005, available at
                  http://www.opensource.org/docs/definition.php.

Author's Address

   Alex Rousskov
   The Measurement Factory

   EMail: rousskov@measurement-factory.com
   URI: http://www.measurement-factory.com/