Network Working Group D. Crocker
Internet Draft <STIF> Silicon Graphics, Inc.
Expiration: 3/94 9 June 1993

Structured Text Interchange Format (STIF)

STATUS OF THIS MEMO

This document is an Internet Draft. Internet Drafts are working documents of the Internet Engineering Task Force (IETF), its Areas, and its Working Groups. (Note that other groups may also distribute working documents as Internet Drafts).

Internet Drafts are draft documents valid for a maximum of six months. Internet Drafts may be updated, replaced, or obsoleted by other documents at any time. It is not appropriate is use Internet Drafts as reference material or to cite them other than as a "working draft" or "work in progress."

Please check the Internet Draft abstract listing contained in the IETF Shadow Directories (cd internet-drafts) to learn the current status of this or any other Internet Draft.

SUMMARY

Various applications need to exchange structured information, such as business-card contact information, bibliographic citations, and structured forms and replies. ASN.1 [ISO87] is a commonly accepted framework for producing binary encoding of information. However, Internet data exchanges often take place in a textual environment, such as electronic mail. In these cases, it would be helpful to have conventions for encoding structured information so that it is entirely legible as text, but sufficiently structured to allow machine processing. This document specifies Structured Text Interchange Format (STIF), a syntax for encoding attribute/value pairs. The pairs can be collected into multi-part sequences and nested sub-lists. The syntax provides for user-defined extensions and for references to data from within sequences and sub-lists. While STIF can be generally compared with ASN.1/BER, it attempts much simpler encoding. In particular, it is strictly text-based and it does not provide for specification of a value’s data type.

Applications for STIF include specialized electronic mail body parts such as enriched header information, organizational exchanges such as for communicating personal contact information, and information/library retrieval descriptions.

TABLE OF CONTENTS

1. INTRODUCTION

Various networking applications require the exchange of structured information, such as lists of attribute/value pairs. ASN.1/BER [ISO87] is a commonly accepted framework for producing binary encoding of such information. However, Internet data exchanges often take place in a textual environment, such as electronic mail. In these cases, it is helpful to have conventions for encoding structured information so that it is entirely legible as text, but sufficiently structured to allow machine processing.

The benefits of textual encoding are counter-intuitive since the method would seem to be highly inefficient for storage and processing. While binary encoding can lead to more compressed representation and more efficient machine processing, a major pressure in environments such as the Internet is for easy interoperability. Textual encoding tends to make software development and debugging much easier and therefore leads to a lower entry cost by independent participants.

This document specifies Structured Text Interchange Format (STIF), a syntax for encoding attribute/value pairs. The pairs can be collected into multi-part sequences and nested sub-lists. The syntax provides for user-defined extensions and for references to data from within lists and sub-lists. While STIF can be generally compared with ASN.1/BER, it attempts much simpler representation and encoding. In particular, it is strictly text-based and it does not provide for specification of a value's data type. That is, the semantic "type" of the data, such as integer or character-string is not explicitly stated in the representation of the data, itself; further only text encoding of the data is supported, rather than allowing different representations for different encoding media. (While all data are encoded as character strings, the definitions of specific data may well define the interpretation of the data as being integers, character strings, etc.)

This specification is of a generic syntax. Hence, the details of its application are left to separate specifications. The benefit of a core representation model of attribute/value is well-established, as are the constructs of sequences and nested (sub-) lists. These three features compose the simple kernel of STIF capability. The basic syntax is kept limited intentionally.

The details of the syntax rules, such as choices for delimiters, were developed from a base of the RFC822 electronic mail addressing syntax. RFC822 enjoys wide implementation, considerably beyond the TCP/IP Internet. As such, it has a large population of existing software and technical developers who are familiar with the syntax. Further, there is extensive experience in moving RFC822-encoded objects throughout the EMail Internet. Hence, it is intended that a syntax which is a (small) derivative from that of RFC822 will be easiest for the Internet community to implement and use.

Applications for STIF include any exchange which needs a simple, stuctured format for labeled information. This includes sending:

- personal contact information, such as is on a business card or a rolodex entry [CROC93],
- bibliographic references & citations [COHE92],
- questionnaire forms, and
- simple query/response transactions.

2. STIF SPECIFICATION

STIF defines data representation to be lists of attribute value pairs. The value portion may be multi-part (a sequence) and sets of pairs may be aggregated into nested, named sub-sets. Although primarily intended for textual representations, STIF permits inclusion of arbitrary data which are encoded as text. Also, text representations are primarily in [US-ASCII], but provision is made for use of alternate character sets. For convenience, STIF also permits data to be simple sequences which are not labeled by attribute name. However, this latter form is not intended as STIF's primary use.

Any BNF rules which are not defined in this document are taken from the RFC822 [CROC82] or MIME [BORE92] specifications.

2.1. Support for Additional Character Sets

STIF syntax is derived from RFC822 and is identical to RFC822's syntax where possible. One point of departure is the alteration of BNF rules which need to support data in the alternate character set. For simplicity, only one such rule is used in higher-level STIF BNF constructs.

Note: It is possible that some circumstances may make it desirable to have multiple additional character sets within a single STIF header. STIF is not designed to support such fine-grained character set requirements and leaves the solution to the international community, probably through the development of a single, universal character set or through a "character-set switching" convention which can, itself, be treated by STIF (and MIME) as a single character set.

2.2. Data Quoting

When dealing with computer systems, a continuing point of confusion involves the mechanisms that are needed to tell a system not to interpret an otherwise-special character but, instead, to treat this normally-special character as regular, uninterpreted data.

RFC 822 specifies a syntax for structured data. RFC822 also specifies a way to have special RFC 822 characters treated as normal data. SMTP specifies a syntax for data transfer of messages; it also has a "quoting" mechanism, particularly for the special terminating period. MIME specifies a syntax for encoding RFC822 messages (enhanced by MIME) in a way that allows 8-bit user data to be transferred, through its transfer-encoding construct.

The RFC 822 quoting mechanism allows specification of arbitrary US-ASCII 7-bit data, using a "backslash/CHAR" mechanism. Thus, its quoting mechanism serves a dual role. It allows transmission of characters usually treated as special to RFC822, and it allows transmission of non-US-ASCII.

STIF eliminates this latter function, so that the STIF quoting mechanism which is otherwise similar to the quoting mechanism of RFC 822 is to be used **_only_** for passing special RFC 822/MIME/STIF characters as simple data. The MIME `Content-Transfer-Encoding` mechanism is entirely adequate for permitting arbitrary 8-bit data to be passed over limited data channels.

RFC 822 also specifies a string-quoting mechanism. However, STIF specifies only a single-character quoting mechanism. It is valid in any `<eword>` without having to surround the `eword`, itself, with special quotation marks.

2.3. Comment text

As with RFC822, STIF permits text to contain uninterpreted comments. The rules for STIF comments are identical to those for RFC822 and the specification in RFC822 is definitive for STIF.

As a convenience, the rules are summarized here:

> Comments may appear between lexical constructs. Comments consist of a parenthetical phrase, i.e., a left parenthesis, followed by a string of text, followed by a right parenthesis. Comments nest(!). The comment text may contain any character sequence, although the characters which are special to comment lexical analysis must be quoted.

2.4. Line length and line wrapping

STIF permits data text and values of any length. It enforces no limitation on the length of data. However, storage and transmission environments, such as electronic mail transport, often do have limitations. STIF defers all such issues to the environment in which STIF is operating.

For example, if STIF is being processed as a MIME body-part, then MIME's rules for line-length handling and for the wrapping of data shall apply. In particular, MIME provides careful discussion of the handling of newlines and spaces.

2.5. Lexical Constructs

In the manner of RFC822, STIF distinguishes low-level lexical constructs from the STIF-specific parsing constructs. The former define basic character strings. The latter define sequences of tokens into the STIF syntax. STIF's lexical constructs are defined for STIF-related parsing and are independent of any additional pre- or post-processing done by other parsers.

```
ephrase          =  1*ascii-word
                           ;  An enhanced phrase is a series of
                              words in US-ASCII,
                /   ( "[" 1*alt-word "]" )
                           ;  a series of words in the alternate
                              character set, or
                /   1*ephrase
                           ;  a series of such phrases

ascii-word       =  eword

alt-word         =  eword

eword            =  1*( reg-char / equoted-pair )
                           ;  normal data or special characters
                              quoted so as to be treated as
                              normal data

reg-char         =  < any ASCII-CHAR except stif-specials and
                    LWSP-char >

ASCII-CHAR       =  < Any 7-bit character defined in [US-ASCII] >

equoted-pair     =  "\" ( stif-special / LWSP-char )
                           ;  equoted-pair is only  for
                              including printable data which has
                              special STIF interpretation.
                           ;  Also, quoting is only on a per-
                              character basis, and  not for an
                              entire string.

stif-special     =  "\" / "[" / "]" / "<" / ">"
```

```
LWSP-char          =  < As defined in RFC822 >
```

`ascii-word` and `alt-word` appear to be lexically identical. However, the framing of`alt-word` between square brackets defines the enclosed data as being interpreted in the alternate character set that is in force.

Multi-byte alternate character sets are parsed by the lexical analyzer as a series of single-byte values. Hence, any multi-byte character which has a single-byte bit-patter that matches a `stif-special` must quote that*one* byte as an `equoted-pair`. Interpretation of the multi-byte sequence as an alternate character therefore takes place after the byte sequence is processed by the STIF lexical analyzer. (The lexical analyzer will return a string of bytes which is known to be in the alternate character set but which has not been parsed further.)

> **NOTE:** While STIF borrows various definitions and constructs from RFC822 and is intended for use within a MIME environment, it defines its own lexical environment. In particular, note that the list of `stif-special` characters contains only the characters that are special to STIF. If a particular STIF value has special meaning, such as an RFC822 `addr-spec` (`local@domain`), then the interpretation of that string and parsing for additional special characters takes place after STIF processing extracts the string. In effect, parsing becomes a multi-layer process, with each layer having its own rules.

2.6. STIF Syntax

STIF is used in document segments that are called `parts`, to faciliate their use within MIME body-parts. A part comprises a set of `headers`, similar to the headers in RFC822. Each header comprises a set of `fields`. Each field contains one “unit” of data or it defines an aggregation of data.

STIF defines a simple syntax for specifying complex attribute/value data lists, with nesting, sequencing and structuring. Nesting permits a field to have sub-aggregations of attribute/value sets. Sequencing allows multiple values to be associated with a single attribute. Structuring allows a value to have multiple parts, with an ordered relationship. (Sequencing can be treated as more than one complete value, whereas structuring specifies one value in a structured fashion.) The functioning of structuring could sometimes be accomplished through the use of additional, smaller attribute/value pairs -- and this would make the syntax more concise and "cleaner" -- but it is felt that the use of sequencing will allow simpler user specification.

STIF Headers are similar to RFC822 fields and they are formally specified as a modification to RFC822 BNF. STIF headers follow the typical RFC822 rules for wrapping of field data. The specific rules for valid `STIF-field` detailed syntax and semantics will depend upon the specific STIF header and STIF field definitions, as provided in separate specifications

A STIF Header comprises one or more attribute/value pairs (fields). At its simplest an `attr-val` field attribute/value pair has a name and a value:

```
phone: +1 408 246 8253
```

with a list of such information represented as:

```
phone: +1 408 246 8253; fax: +1 408 249 6205
```

A value actually may be a multi-part `sequence` so that one value is structured into an ordered list of sub-values. For example, specifying a geographic address usually requires city, region and country. While this could be accomplished with three, separate attribute/value pairs, it is economical to specify them together, reflecting common practice:

```
geo: Sunnyvale / CA / US
```

Also, a value may simply be a list of identical values. In STIF, this is represented the same way as a sequence of different *kinds* of values:

```
phone: +1 408 246 1234 / +1 408 249 6205
```

Determining whether a `sequence` specifies values that all are to be used or values that are to be used as alternatives will depend upon the semantics of the attribute. Specifying one value from a list is accomplished with ones-based indexing, so that the first number in the above example is specified as `phone[1]`.

Lastly, it may be appropriate to aggregate information into a hierarchy of attribute/value sets (`nesting`). A classic example of this pertains to information about resources at home and those at work:

```
Contact < work <phone: +1 415 246 1234>
          home <phone: +1 408 246 8253; fax: +1 408 249 6205> >
```

Choosing among nested alternatives is accomplished with a dotted notation (`nest-ref`), so that `contact.work.phone` refers to the first number and `contact.home.phone` refers to the second. Common usage may cause some information eventually to change from a nested notation to one that is called out explicitly. For example, reference to a telephone number is an embedded, or core, requirement. Distinguishing a telephone number used for facsimile could easily be accomplished with `fax.phone`. However, specification of facsimile telephone numbers is sufficiently common that an explicit fax construct may be appropriate.

> *Simply stated, an attribute/value pair is distinguished by colon separating the pair and a semi-colon between pairs. A nested sub-set is distinguished by surrounding the sub-set in an angle-bracket pair, and a sequence is distinguished by separating the values with forward slashes.*

Formally, the generic syntax for STIF headers is:

```
STIF-fields      =   named-fields / sequence
                            ;  a set of labeled values, or
                            ;  a set of order-dependent,
                               unlabeled values

named-fields      =  attr-values / nestings
                            ;  one named value, or
                            ;  a named set of sub-fields
```

```
attr-values        =  attribute ":" [ sequence ]
                      *[ ";" attr-values ] [ ";" ]
                             ;  An attributes value may be an
                                ordered sequence of values

attribute          =  field-name
                             ;  same syntax as RFC 822

sequence           =  value *( "/" value )
                             ;  a sequence of integrated values

value              =  ephrase

nestings           =  nest-name "<" *named-fields ">"
                             ;  nested data may include one or
                                more  nested attribute/value
                                pairs, such  as for distinguishing
                                home phone  number from work phone
                                number  (nesting is infinitely
                                recursive)

nest-name          =  field-name

attr-el-ref        =  *( nest-name "." )
                      attribute
                      [ "[" el-index "]" ]
                             ;  citation format for an element
                             ;  nest1.nest2.attribute[index]

el-index           =  1*DIGIT

DIGIT              =  < as defined in RFC822 >
```

The context in which STIF-based data are defined will determine any additional encoding, such as distinguishing different sets of header data. The following section discussing a framework that will be used frequently for STIF-based data used within MIME. When STIF-based data represents a single set of information, then simply encoding it as a set of `STIF-fields` is appropriate, with no special labeling between sets of `STIF-fields`.

The `nest-ref` rule provides a standard method of referencing data within STIF fields. Hence, the example of nesting earlier in this section would allow reference to a home phone number as `contact.home.phone`.

`header-name` should be whatever string is appropriate for identifying the collection of information in the set of fields. Within a database context, it will be the value of the primary key for the entire entry. When the key is a person's name, care should be taken to choose a value which is sufficiently canonical. For example, a person's name may include various addenda, such as "Ms." or "Ph.D." or "III" or "Jr.". When the name is part of a database tag, such addenda should be omitted, since they are not likely to be specified during a query.

Portions of data are cited using a dotted notation with array indexing, as appropriate. The dotted notation walks down the tree structure of nested data and the ones-based indexing allows selection from a list of values. If a reference resolves to a sub-tree of nestings, rather than a specific attribute, then it is referring to that entire sub-tree. If a reference resolves to a value list, but does not specify an index, then it is referring to that entire list.

3. STIF USED IN MIME BODY-PARTS

STIF headers are primarily encoded in US-ASCII text. When contained in a MIME message and received by a MIME-aware user agent that does not understand the semantics of the specific STIF-based data, it often will still be reasonable for the user agent to process the body-part as regular `Text`. For such STIF-encoded data, it is therefore reasonable to define the MIME data as a sub-type of `Content-Type:Text`. When the data cannot reasonable be processed as text, then the body-part should be defined under the appropriate `Content-Type`, usually application. STIF, itself, does not have an explicit MIME definition. Rather, it is a characteristic of certain subtypes.

Independent of the top-level `Content-Type` under which the STIF-encoded MIME subtype is defined, the `Content-Type` header may contain a definition of the alternate character set which is used in the body-part. A mechanism is provided for invoking an “alternate” character set. Within MIME, this alternate character set is defined by a "`CHARSET=`" parameter.

Information in the alternate character set generally will need to be transfer-encoded for cross-net communication. The standard MIME `Content-Transfer-Encoding` mechanism may be used for this. It is expected that `Quoted-Printable` will usually be the best method of ensuring the ability to transfer data, while retaining the general ability to view US-ASCII-based structured information, since it will maintain readability of the US-ASCII information. However, the choice of `Content-Transfer-Encoding` mechanism is entirely the choice of the encoding system. No special considerations are required, when encoding STIF-based data.

Support for non-US-ASCII character sets is through the same "`CHARSET=`" parameter as is used for MIME’s `Content-type:Text` mechanism. For `Content-type:Text` all text in the body-part is in the specified character set. For STIF, the basis for structured information always is US-ASCII, in order to facilitate any machine processing which may be appropriate. However, some portions of the structured text are allowed to be either in US-ASCII or in the alternate character set specified with the "`CHARSET=`" parameter. (When STIF is used in non-MIME environments, some other method must be provided for specifying the alternate character set.)

Changing to a different alternate character set is allowed only at STIF header boundaries. Any number of STIF headers may be aggregated into a single STIF-encoded body-part, but each STIF header must be complete.

A series of STIF-headers may be separated into multiple body-parts, using the `multipart/mixed` mechanism. This permits redefinition of the alternate character set for STIF.

STIF alternate character sets are the same as the list of `charsets` used in MIME.

Thus, a typical STIF-compatible message which uses more than one alternate character set may exist within an RFC822 and MIME framework, and have the general form:

```
--Boundary-1
Content-Type:     MULTIPART/MIXED; boundary=Boundary-2

--Boundary-2
Content-Type:     TEXT/x-xxx; charset=US-ASCII

(initial part of content, with no special character set
requirements)

--Boundary-2
Content-Type:     TEXT/x-xxx; charset=ISO-8859-1
Content-Encoding: Quoted-Printable

(remaining part of content, with character set supporting some
european languages

--Boundary-2--
--Boundary-1
Content-Type:     Text/plain
Content-Encoding: Quoted-Printable

(Other content, not using the STIF STRUTEXT content type)

--Boundary-1--
```

The specific context in which MIME is used will determine any surrounding syntax or labeling. Frequently, it will be appropriate for STIF data to appear in a series of segments which are partitioned in a manner similar to headers in RFC822. That is, the data will be divided into a series of headers, each with its own header label. In such cases, the STIF-based MIME body-part data format will be:

```
STIF-part        =   *STIF-header

STIF-header      =   header-name ":" STIF-fields CRLF
                             ;  essentially the same as RFC822

header-name      =   field-name

field-name       =   < As defined in RFC822>
                             ;  Any field name which is defined in
                                RFC 822 or later standards
                                document, and which is not
                                explicitly defined in this
                                specification is automatically
                                incorporated as a STIF Header
                                name, using the RFC 822 syntax,
                                but with any interpretation
                                changes as dictated by STIF's
                                redefinition of the <text> and
                                <phrase> constructs >
```

4. STIF Definitions Sent as STIF Encodings

Since STIF is intended for the carriage of structured, text-encoded objects, it should be possible to use STIF to send descriptions of STIF entitities. That is, it should be possible to define specific STIF objects in terms of the STIF syntax. For expedience, the definitions of STIF use of STIF syntax is left for further study.

4. Examples of STIF Usage

This section offers some examples of possible STIF usage. These examples do not represent any sort of formal or accepted use of the syntax, and are intended merely to be representative.

4.1. Personal contact information entry

The following example is taken from PCI [CROC93].

It is common for Internet mail headers to contain extensive information about the author of the mail. For example:

```
From:    "Ole J. Jacobsen" <ole@Csli.Stanford.EDU>
Or:     +1 415 550-9427 (Home) or +1 415 990-9427 (Cellular)
Direct: +1 415 962-2515 (Office) +1 415 998-4427 (Pager)
Fax:    +1 415 949-1779 (Interop) +1 415 826-2008 (Home)
X-Comment: Ignore error messages for "ole@radiomail.net"
   Ole J Jacobsen, Editor & Publisher ConneXions--The
        Interoperability Report
   Interop Company, 480 San Antonio Road, Suite 100, Mountain View,
        CA 94040,
Phone:  (415) 962-2515  FAX: (415) 949-1779
Email:  ole@csli.stanford.edu
```

When encoded as a PCI header, this becomes:

```
Ole J Jacobsen:
  name:    Ole J\. Jacobsen
  email:  ole@csli.stanford.edu;
  work   <title: Editor & Publisher;
          org:    Interop Company;
          dept:   Connexions -- The Interoperability Report;
          street: 480 San Antonio Rd\.\, Suite 100;
          geo:    Mountain View / CA / US;            code: 94040;
          phone:  +1 415 962 2515;           fax: +1 415 949 1779>
  home    <phone: +1 415 550 9427;           fax: +1 415 826 2008>
  mobile  <phone: +1 415 990 9427; pager<phone: +1 415 998 4427> >
  note:   Ignore error messages for "ole@radiomail.net"
```

Note that the entire entry is tagged with a canonical form of the person’s name. This tag is different from the formal presentation string for that person's name. That is, `name` is different from the `header-name` for the header containing the `name` field. Leading and trailing information, such as “Dr.” or “Ph.D.” and “Jr.” should be removed from the `header-name`, to faciliate use of this string as a database storage and search key.

4.2. Reference citations

Possibly deriving nomeclature from [COHE92], literature citations,such as:

```
[BORE92]  Borenstein, N. & Freed, N., "MIME (Multipurpose Internet
          Mail Extensions): Mechanisms for specifying and
          describing the format of Internet Message Bodies.
          March, 1992, Network Information Center, RFC 1341.

[CROC93]  Crocker, D., “Evolving the System”, in Internet System
          Handbook, Lynch & Rose (eds.); Reading, Mass., Addison-
          Wesley Publishing Co. (1993)
```

could be represented in STIF as:

```
Borenstein-Freed-MIME-92:
  author:   N. Borenstein, N. Freed;
  title:    MIME \(Multipurpose Internet Mail Extensions\)\:
            Mechanisms for specifying and describing the format of
            Internet Message Bodies;
  date:     1992 / March / ; id: RFC 1341;
  org:      Network Information Center

Crocker-Evolving-93:
  author:   D. Crocker;              title: Evolving the System;
  in:       Internet System Handbook;  editor: D. Lynch, M. Rose;
  geo:      Reading / Mass / ;
  org:      Addison-Wesley Publishing Co.;        date: 1993 / /
```

5. REFERENCES

[BORE92] Borenstein, N. & Freed, N., "MIME (Multipurpose Internet Mail Extensions): Mechanisms for specifying and describing the format of Internet Message Bodies". March, 1992, Network Information Center, RFC 1341.

[COHE92] Cohen, D. "A Format for E-Mailing Bibliographic Records". July, 1992, Network Information Center, RFC 1357.

[CROC82] Crocker, D., "Standard for the format of ARPA Internet text messages". August, 1982, Network Information Center, RFC 822.

[CROC93] Crocker, D. "Encoding for Personal Contact Information (PCI)". Draft, May 1993.

[ISO87] ISO 8824, Information Processing -- Open Systems Interconnection -- Specification of Abstract Syntax Notation One (ASN.1), Melbourne, 1987

[US-ASCII] Coded Character Set--7-Bit American Standard Code for Information Interchange, ANSI X3.4-1986

6. SECURITY CONSIDERATIONS

This specification covers data encoded within a portion of an email message. It contains addressing and source-identification information which is not specially authenticated. No special provision is made for confidentially of the data nor for maintaining data integrity. No other security-related concerns apply.

7. ACKNOWLEDGMENTS

The idea for STIF is a direct outgrowth from the efforts of IETF working groups to extend the functionality of the Internet's mail service. This specification resulted from a series of discussions with Marshall Rose, Ned Freed, Steve Crocker, and Keith Moore. Additional review and comments were provided by Nathaniel Borenstein and Erik M. van der Poel

8. CONTACT


```
name:      David H. Crocker;
work       <email: dcrocker@sgi.com;
           org:    Silicon Graphics, Inc.;
           street: 2011 N. Shoreline Blvd.;
           box:    7311;
           geo:    Mountain View / CA / US;       code: 94039-7311;
           phone:  +1 415 390 1804;            fax: +1 415 962 8404>
```


APPENDIX: RFC 822 and MIME Rules Used by STIF

A number of BNF rules used by STIF are taken from RFC822 or MIME. In order to maintain consistent definition, this document does not offer explicit definition of those rules, instead referring the reader to the source specifications.

For convenience, a copy of those rules -- taken from the source documents -- is included here.

THE FOLLOWING RULES ARE NOT TO BE CONSIDERED DEFINITIVE.

IN THE CASE OF INCONSISTENCY BETWEEN THE RULES LISTED HERE AND THOSE IN THE SOURCE DOCUMENT -- OR IN LATER VERSIONS OF THOSE SOURCE DOCUMENTS -- THE RULES LISTED HERE ARE TO BE DEPRECATED. ONLY THE SOURCE VERSIONS OF THE RULES ARE TO BE CONSIDERED CORRECT.

```
CHAR        =  <any ASCII character>
                             ; (  0-177,  0.-127.)

DIGIT       =  <any ASCII decimal digit>
                             ; ( 60- 71, 48.- 57.)

field-name  =  1*<any CHAR, excluding CTLs, SPACE, and ":">

LWSP-char   =  SPACE / HTAB
                             ; semantics = SPACE



CTL         =  <any ASCII control character and DEL>
                             ; (  0- 37,  0.- 31.)
                             ; (    177,     127.)

SPACE       =  <ASCII SP, space>
                             ; (     40,      32.)

HTAB        =  <ASCII HT, horizontal-tab>
                             ; (     11,       9.)
```