Network Working Group
Request for Comments: 645
NIC: 30899:
Obsolets: 615 (NIC: 21531)

D. Crocker
UCLA-NMC
JUNE 1974

Network Standard Data Specification Syntax

INTRODUCTION

This document defines the basic components of a Network Standard Data Specification (NSDS) syntax. A NSDS is intended to provide a mechanism for specifying all the attributes of a collection of bits.

The definition of a complete NSDS syntax is expected to require an extended effort. Therefore the initial scope of this document has been constrained to provide only a basic syntactic environment.

In order to demonstrate a specific use for the NSDS, this document also provides the complete syntax for specifying the PATHNAME attributes of a collection of bits, to the level of a file. Addition of new subparamters should not be difficult.

In this context, "pathname" referes to that information which specifies the LOCATION of a collection of bits.

The pathname syntax is essentially the same as that proposed in RFC 615 (NIC -- 21531,). Modifications were made in order to allow for graceful addition of other file attributes and to optimize use by humans and by processes.

I would like to thank Jon Postel, Jerry Popek, Vint Cerf, Jim White, Charlie Kline, Buz Owen, Ken Pogran, Jerry Burchfiel and Tom Boynton for their suggestions.

HUMAN AND MACHINE FACTORS

Since computers tend to prefer more highly structured envireonments than do humans, aspects of the NSDS syntax are permitted to be different for computers than they are for humans. Specifically:

For computers (highly-structured mode), keyword fields are fixed length and the variable-length data subfields are prefaced by a byte count. Additionally in highly structured mode, the possible contents of data subfields may be more constrained than for the semi-structured mode.

For humans (semi-structured mode), keyword subfields are variable length and data subfields are surrounded by delimeter characters. A keyword must be long enough to distinguish it from other keywords. That is, partial-name specification is permitted.

STRUCTURE OF THE GENERAL SYNTACTIC ENVIRONMENT

Overview:

A NSDS is prefaced by one or two percent signs, followed by a set of fields subject to context-free interpretation, and terminated with a space. Pathname fields precede any other file attribute specifications.

The BNF:

```
<NSDS>        ::=  <flag> <path> <otherstuff> <sp>

<flag>        ::=  % / %%

<path>        ::=  pathname fields, as described below.

<otherstuff>  ::=  fields for specifying data storage and accesss
                   characteristics, to be defined later.

<sp>          ::=  space.
```

Comments:

The <flag> indicates escape-tp-NSDS-syntax. One percent sign indicates semi-structured syntax, two indicate that highly-structured syntax is being used.

Only <flag> must be considered in relation to any host's current syntax. It is not currently known to conflict with any host's syntax.

Exclamation mark (!) is the only other character that seems permissible (on the assumption that the character should be a graphic). Its use would cause minor problems at Multics; but more importantly as a graphic, it is too similar to the numeral "1".

The basic (highest-level) syntax for individual <path> and <otherstuff> fields is the same, as defined below. The remaining lower-level syntax (including permissible keywords and data subfield contents) for <otherstuff> fields is left for later.

BASIC UNITS OF SUBSTRUCTURE

Overview:

A semi-structured field begins with a varying-length descriptor. The descriptor is followed by a varying-length data subfield, which is surrounded by delimeter characters.

Highly-structured fields have fixed-length descriptors, followed by a data byte-count, followed by the data

BNF for individual fields:

<field> ::= <machine> / <human>

```
<machine>     ::=  <stru-field> / <stru-field> <machine>

<stru-field>  ::=  <stru-key> <count> <data>

<stru-key>    ::=  4-character field definition keyword; see
                   below.

<count>       ::=  one-byte binary count of number of bytes of
                   <data>.

<human>       ::=  <h-field> / <h-field> <human>
```



```
<h-field>     ::=  <h-key> <h-rest>

<h-key>       ::=  Variable-length field definition keyword; see
                   below.

<h-rest>      ::=  <l-delim> <data> <r-delim>
                   /  <l-delim> <data> <r-delim> <h-rest>

<l-delim>     ::=  any non-alphabetic printable character that is
                   not in the succeeding <data> subfield and that
                   is acceptable to the object site.  For visual
                   aesthetics and to facilitate human parsing,
                   anything <l-delim> is a left-bracket character
                   (<, [, (, -), <r-delim> must be the
                   complementary right-bracket
                   character (>, ], ), |).

<r-delim>     ::=  either 1) the same character as <l-delim> or 2)
                   if the <l-delim> character is a left-bracket
                   character (<, [, (, -) then its complementary
                   right-bracket (>, ], ), |).

<data>        ::=  any sequence of characters acceptable to the
                   object site.  This is the actual data subfield
                   with the file, directory, device (or whatever)
                   attribute value.
```

Elaboration:

Case is irrelevant to the syntax, though some sites will care about case in <data> subfields.

They key (<stru-key> or <h-key>) indicates what part of the NSDS the next <data> subfield refers to.

<R-delim> amd <l-delim> are used to delimit the beginning and end of the <data> subfield.

<Fields> for pathnames ARE order dependent, but defaulted ones may be omitted. The order is as indicated for <key>s, below. That is, Network, Host, ... Siteparm.

Keywords are used, even though pathname attributes are ordered, to facilitate the addition of new fields and to be consistent with the syntax for <otherstuff> fields which are expected to be unordered.

<Field>s or <h-rest> subfields may be repeated, as permitted by the object site. A series of <h-rest> subfields, without any <h-key> subfields is interpreted as a series of <h-field>s with identical <key>s.

Also, note that since the syntax does not constrain the contents of <data> subfields, compound names within a single <data> subfield are allowed. The delimeter used to separate names within a <data> subfield must be different from <l-delim>/<r-delim> and the same as that used at the object site, since that is the only site which will be able to interpret the <data> subfield.

The validity of any combinations of <field>s is entirely site-dependent. For example, if a site will accept it, an NSDS with a Host field, and nothing more, may be permissible.

The validity of <data> subfields' contents is generally site-dependent. Some exceptions are noted below.

PATHNAME ATTRIBUTES AND VALUES

The basic syntax does not need to be altered, to create the ability

to specify pathnames. Only <key> values need to be defined.

Definitions of Pathname <key>s:

They keyword for semi-structured mode is given first, followed by the keyword for highly-structured mode, if different. For highly-structured mode, keywords that are less than four characters should be padded with blanks at the right.

Semi	Highly	Meaning
NETWORK	NET	Reference to the network (e.g., ARPA) connected to the HOST that contains or will contain the collection of bits.
HOST		Reference to host machine that contains or will contain the collection of bits. Also see section on "Numbers".
PERIPHERAL	PERI	Peripheral device being referred to.
VOLUME ID	VOL	The volume (e.g., specific tape reel or disk pack) associated with the named peripheral device.

DIRECTORY	DIR	Name of directory which contains a pointer to the entity (directory or filename) specified in the following <field>.
FILE		Basic name of the file (data set).
TYPE		Optional modifier to the filename. (Tenex calls it the Extension.)
VERSION	VER	Optional third party to basic filename. Usually used to distibguish updated files. The <data> subfield will usually contain a number.
SITEPARM	SITE	A parameter, such as an access specification

or account number, peculiar to the object site. The contents of the <data> subfield must serve to identify what Siteparm is involved. Each site will be responsible for defining the syntax of Siteparm <data> subfields it will accept. Note that the SITEPARM field allows specification of other than pathname data (e.g., access and account number).

Some reserved PERIPHERAL <data>s:

The alternate forms are merely for typing convenience and are not related to the semi/highly structure modes.

DISK or DSK:	Immediate, direct-access secondary storage.
ONLINE or ONL:	Whatever immediately-accessible (measured in fractions of a second) storage the user accesses by default; usually disk.
TAPE or TAP:	Industry-compatible magnetic tape.
TAPE7 or TP7:	7-Track industry compatible tape.
TAPE9 or TP9:	9-Track industry compatible tape.
DECTAPE or DEC:	DEC Tape.

OFFLINE or OFF:	Any tertiary storage; usually tape, though "devices" like the Datacomputer are permissible. The user should expect to wait minutes or hours before being able to access OFFLINE files.
LINE PRINTER or LPT:	Any available line-printer.

DOCUMENT PRINTER or DOC: Upper/lower case line printer, preferably with 8 1/2" X 11" unlined paper.

PAPER TAPE READER or PTR: Paper tape reader.

PAPER TAPE PUNCH or PTP: Paper tape punch.

CARD PUNCH or PUN: Standard 80-column card punch.

CARD READER or RDR: Standard 80-column card reader.

OPERATOR or OPR: System Operator's console.

CONSULTANT or CON: On-line consultant.

DEFAULTS FOR PATHNAME <DATA> SUBFIELDS:

Often, the appropriate default will be the last-used value. However, defaults will generally be context dependent. Consequently, the following defaults are offered only as guidelines:

Network: ARPA.

Host: The host interpreting the NSDS.

Peripheral: ONLINE (DISK).

Volume id: Catalogued system space.

Directory: The user's current "working" directory, usually set by the logon process.

Filename: None.

Type: None.

Siteparm: None.

NUMBERS

The following scheme is recommended for specifying numbers in <h-field> data subfields:

> A sequence of numberic characters, optionally followed by a character indicating the radix. The default radix is ten. "H" indicates hexadecimal; "O" (oh) indicates octal; "B" indicates binary; and (gratuitously) "D" indicates decimal.

In <stru-field> data subfields, the number should be pure binary. Therefore, reference to a host on the Arpanet would require one 8-bit byte.

GENERAL COMMENTS

The syntax is intended to be adequate for all hosts, so any given portion of it may be inappropriate for any given host.

> A site is expected to permit specifications in a given field if that site already has a way of accepting the same information.
>
> Having two modes of specification (highly- and semi-structured) may prove to be unnecessary. They are defined here merely as a convenience for experimentation.

I believe that modifications to the syntax will be graceful additions, rather than wholesale redesign, and thus can be deferred for a while. Currently, any undefined attributes must be specified in a Siteparm field.

The first version of the syntax was a mix of Tenex and Multics conventions. That is:

> (Network)[Host]Peripheral:Directory>Filename.Type;Sireparm

Through visually more attractive and generally quicker to type, it lacks extensibility. For example, adding a version number as a standard field would be difficult.

It is asserted (conceded) that, as long as extensibility is kept as a design goal, no standardized [semi-structured] syntax will be as pleasant to use as currently exists on some systems.

SOME SAMPLE PATHNAMES

Pathnames in NSDS that occupy more than one line, below, do so only because they are too long for a single line. Bracketed numbers (e.g., <8>) indicate a single byte with the number as its decimal value. Blanks (spaces) are indicated by <sp>.

My message file at ISI (<DCROCKER>MESSAGE.TXT;P770404):

Semi-structured

%H[ISI]D<DCROCKER>F(MESSAGE>T(TXT)S/P770404/<sp>

Highly-structured

%%HOST<1><86>DIR<sp><8>DCROCKERFILE<7>MESSAGETYPE<3>TXTSITE<7>P770404<sp>

ARP061.LAD.DOCUMENT at UCLA-CCN. (Note the use of multiple Directory fields):

Semi-structured

%H[65]DIR[ARP061][LAD]F[DOCUMENT]<sp>

Highly-structured

%%HOST<1><65>DIR<sp><6>ARP061DIR<sp><3>LADFILE<8>DOCUMENT<sp>

>udd>CompNet>Map>Mail at Mit-Multics. (Note that the initial NSDS Directory <data> subfield is empty, in keeping with Multics' method of starting at the top of its directory structure):

Semi-structured

%h(540)DI[]DI[udd][CompNet]D(Map)FIL(Mail)<sp>

Highly-structured

%%HOST<1><44>DIR<sp><0>DIR<sp><3>uddDIR<sp><7>CompNetDIR<sp><3>MapFILE<4>Mail<sp>