

**NAME**

**abnff** – parse and nicely format ABNF

**SYNOPSIS**

**abnff** [**files**]

**DESCRIPTION**

**Abnff** is a preprocessor for parsing and formatting ABNF as used in RFCs and Internet Drafts. ABNF is described in RFC 2234.

**Abnff** implements a grammar which differs from the one in RFC 2234:

1. comments which are not a continuation of a rule must not have whitespace before the semicolon (this is necessary to remove an ambiguity in the RFC 2234 grammar).
2. ABNF lines parsed by **abnff** are not required to have an old-fashioned teletype "carriage return" character preceding the newline character marking the end of a line (ABNF found in RFCs and Internet-Drafts does not use a CR).

This software is OSI Certified Open Source Software. OSI Certified is a certification mark of the Open Source Initiative.

ABNF input is contained between lines beginning with .AS and .AE. The .AS line may contain two numeric arguments specifying the indent and line length in characters. Because **abnff** is a preprocessor, the numeric arguments must be specified directly; one cannot refer to troff registers or use troff width escapes, etc.

The .AS line can also take any combination of several one-character alphabetic option flags:

Flag	Description
c	canonicalize case of character literals; type (letter after '%') is lower-case, hexadecimal digit letters are upper-case. Rulename references use the same case as the defining ABNF line.
d	omit display wrappers around output rules, instead separating rules with an empty line. This may be useful if the entire ABNF is placed within a display wrapper.
g	group concatenated elements in parentheses if in an alternation expression
i	implicit definition of core ABNF rules (suppress warnings about lack of definitions)
n	normalize repetition specifications: 0*1foo becomes [foo], M[foo bar] becomes *M(foo bar), etc.
s	Combine adjacent quoted strings into a single quoted string where possible
t	topologically sort rules in a ruleset so that productions are defined before use
r	reverse the sense of topological sorting; use precedes definition (ineffective if the t flag is not specified)

Within a ruleset delimited by .AS and .AE lines, **abnff** recognizes a .BC line, which causes the immediately following rule in the ruleset to be formatted with a block comment. That is, the rule definition and comment text are placed side-by-side with running comment text. The height of definition ABNF and comment text is approximately balanced by **abnff**.

Comment lines in troff style are also permitted within ABNF; they are echoed to output before all formatted ABNF rules.

**EXAMPLE**

The following example demonstrates formatting of some ABNF:

```
.AS 3 72 c g n r s t
script-list = Script *( [ CFWS] ", " [CFWS ] SCRIPT )

content-script = "Content-Script" ":" [CFWS] script-list [CFWS] CRLF

CFWS = comment / FWS
CRLF = %X0d.0a

FWS = 000*([CRLF] WSP)

.BC
script = 4*00004ALPHA ; script tag per ISO 15924:2004;
; script tags are case-insensitive protocol elements
.AE
```

which is processed by **abnff** to produce:

```
.DS L
content-script = "Content-Script:" [CFWS] script-list [CFWS] CRLF
.DE
.DS L
script-list = script *([CFWS] ", " [CFWS] script)
.DE
.DS L
script = 4ALPHA ; script tag per ISO 15924:2004; script tags
; are case-insensitive protocol elements
.DE
.DS L
CFWS = comment / FWS
.DE
.DS L
FWS = *([CRLF] WSP)
.DE
.DS L
CRLF = %x0D.0A
.DE
```

which in turn is processed by the rfc macros to produce output which looks like:

```
content-script = "Content-Script:" [CFWS] script-list [CFWS] CRLF

script-list = script *([CFWS] ", " [CFWS] script)

script = 4ALPHA ; script tag per ISO 15924:2004; script tags
; are case-insensitive protocol elements

CFWS = comment / FWS

FWS = *([CRLF] WSP)

CRLF = %x0D.0A
```

## RETURN VALUE

**Abnff** returns zero unless a serious error occurs.

## BUGS and CAVEATS

Comment-only ruleset lines have no associated rulename, therefore cannot be topologically sorted. If a ruleset contains comment-only lines, has undefined rulenames, and topological sorting is specified, comment-only lines may appear in strange places.

## DIAGNOSTICS

**Abnff** emits warning and error messages on the standard error stream. It indicates errors if the input specification is faulty in a way that precludes sensible parsing; less serious issues result in warnings. Warnings include multiply-defined rules, rules which are defined but not referenced, rulenames which are referenced but are undefined, constructs which map to empty input (e.g. a zero repetition count), and cycles in definitions.

## SEE ALSO

RFC 2234

**AUTHOR**

Bruce Lilly <blilly@erols.com>