

# Naming Rules

---

Metadata	Value
Date	2021-08-17
Author	@scottf
Status	Approved
Tags	server, client

## Context

This document describes naming conventions for these protocol components:

- Subjects (including Reply Subjects)
- Stream Names
- Consumer Names
- Account Names

## Prior Work

Currently, the NATS Docs regarding [protocol convention](#) says this:

Subject names, including reply subject (INBOX) names, are case-sensitive and must be non-empty alphanumeric strings with no embedded whitespace. All ascii alphanumeric characters except spaces/tabs and separators which are "." and ">" are allowed. Subject names can be optionally token-delimited using the dot character (.), e.g.: A subject is comprised of 1 or more tokens. Tokens are separated by "." and can be any non space ascii alphanumeric character. The full wildcard token ">" is only valid as the last token and matches all tokens past that point. A token wildcard, "\*" matches any token in the position it was listed. Wildcard tokens should only be used in a wildcard capacity and not part of a literal token.

Character Encoding: Subject names should be ascii characters for maximum interoperability. Due to language constraints and performance, some clients may support UTF-8 subject names, as may the server. No guarantees of non-ASCII support are provided.

## Specification

dot	= "."
asterisk	= "*"
lt	= "<"
gt	= ">"
dollar	= "\$"
colon	= ":"
double-quote	= "[\""]
fwd-slash	= "/"
backslash	= "\"

```

pipe           = "|"
question-mark  = "?"
ampersand      = "&"
dash          = "-"
underscore     = "_"
equals         = "="
printable      = all printable ascii (33 to 126 inclusive)
term           = (printable except dot, asterisk or gt)+
limited-term    = (A-Z, a-z, 0-9, dash, underscore, fwd-slash, equals)+
limited-term-w-sp = (A-Z, a-z, 0-9, dash, underscore, fwd-slash, equals, space)+
restricted-term = (A-Z, a-z, 0-9, dash, underscore)+
prefix         = (printable except dot, asterisk, gt or dollar)+
filename-safe  = (printable except dot, asterisk, gt, fwd-slash, backslash)+
maximum 255 characters

message-subject = term (dot term | asterisk)* (dot gt)?
reply-to        = term (dot term)*
stream-name     = filename-safe
durable-name    = filename-safe
consumer-name   = filename-safe
account-name    = filename-safe
queue-name      = term
js-internal-prefix = dollar (prefix dot)+
js-user-prefix  = (prefix dot)+
kv-key-name     = limited-term (dot limited-term)*
kv-bucket-name  = restricted-term
os-bucket-name  = restricted-term
os-object-name  = (any-character)+

```

## Notes

### filename-safe

The **filename-safe** is designed with unix operating systems in mind. If the server is running on Windows, the **filename-safe** is too lenient and will operate like:

```

filename-safe = (printable except dot, asterisk, lt, gt, colon, double-quote, fwd-slash, backslash, pipe, question-mark)+ maximum 255 characters

```

The server will reject names (return an API error) when it cannot build a valid path.

### kv-key-name

Keys starting with **\_kv** are limited to internal use.

### Client Validation

This note is simply to capture that currently the Go client simply validates that stream and consumer/durable names are not empty, and do not contain **..**. Any other constraints are left to the server. NATS cli, performs additional validations, **., \*, >, /** and **\** are not allowed.

