
Workgroup: Network Working Group
Internet-Draft: draft-whited-static-shell-00
Published: 22 November 2020
Intended Status: Informational
Expires: 26 May 2021
Authors: S. Whited C. Paul
Individual

The "Static" Shell Scripting Language

Abstract

We describe a statically typed scripting language with a focus on type safety and shell integration.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 26 May 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Table of Contents

- 1. Introduction
 - 1.1. Overview
 - 1.2. Terminology
 - 1.3. Methodology
- 2. Source Code Representation
 - 2.1. Characters
 - 2.2. Letters and digits
- 3. IANA Considerations
- 4. Security Considerations
- 5. References
- Appendix A. Acknowledgements
- Authors' Addresses

1. Introduction

1.1. Overview

This is a description of the Static shell scripting language.

Static is a scripting language designed to be executed by a terminal emulator and used as the users shell. It is statically typed and has explicit support for standard streams and for executing external programs.

1.2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119], [RFC8174] when, and only when, they appear in all capitals, as shown here.

Many important terms used in this document are defined in [RFC6365], and [Unicode].

1.3. Methodology

The syntax is specified using Augmented Backus-Naur Form (ABNF) as specified in [RFC5234].

2. Source Code Representation

Source code is Unicode text encoded with [UTF-8 \[RFC3629\]](#). The text is not canonicalized.

Implementation restriction: For compatibility with other tools, a compiler may disallow the NUL character (U+0000) in the source text.

Implementation restriction: For compatibility with other tools, a compiler may ignore a UTF-8-encoded byte order mark (U+FEFF) if it is the first Unicode code point in the source text. A byte order mark may be disallowed anywhere else in the source.

2.1. Characters

The following terms are used to denote specific Unicode character classes:

```
newline      = /* the Unicode code point U+000A */ .
unicode_char = /* an arbitrary Unicode code point
                except newline */ .
unicode_letter = /* a Unicode code point classified as
                  "Letter" */ .
unicode_digit = /* a Unicode code point classified as
                  "Number, decimal digit" */ .
```

In The Unicode Standard 8.0, Section 4.5 "General Category" defines a set of character categories. Static treats all characters in any of the Letter categories Lu, Ll, Lt, Lm, or Lo as Unicode letters, and those in the Number category Nd as Unicode digits.

2.2. Letters and digits

The underscore character `_` (U+005F) is considered a letter.

```
letter       = unicode_letter | "_" .
decimal_digit = "0" ... "9" .
octal_digit  = "0" ... "7" .
hex_digit    = "0" ... "9" | "A" ... "F" | "a" ... "f" .
```

3. IANA Considerations

This document requires no interaction with the Internet Assigned Numbers Authority (IANA).

4. Security Considerations

TODO

5. References

- [Go]** "The Go Programming Language Specification", <<https://golang.org/ref/spec>>.
- [RFC2119]** Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3629]** Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <<https://www.rfc-editor.org/info/rfc3629>>.
- [RFC5234]** Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <<https://www.rfc-editor.org/info/rfc5234>>.
- [RFC6365]** Hoffman, P. and J. Klensin, "Terminology Used in Internationalization in the IETF", BCP 166, RFC 6365, DOI 10.17487/RFC6365, September 2011, <<https://www.rfc-editor.org/info/rfc6365>>.
- [RFC8174]** Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [Unicode]** The Unicode Consortium, "The Unicode Standard", <<http://www.unicode.org/versions/latest/>>.

Appendix A. Acknowledgements

Some text in this document was copied from [Go].

Authors' Addresses

Sam Whited

Individual

Email: sam@samwhited.com

URI: <https://blog.samwhited.com/>

Cameron Paul