Use XMLStarlet to parse XML in the Linux terminal

Become an XML star with XMLStarlet, an XML toolkit for your terminal.

By Seth Kenlon (Team, Red Hat)

July 13, 2021 | 2 Comments | 53 readers like this

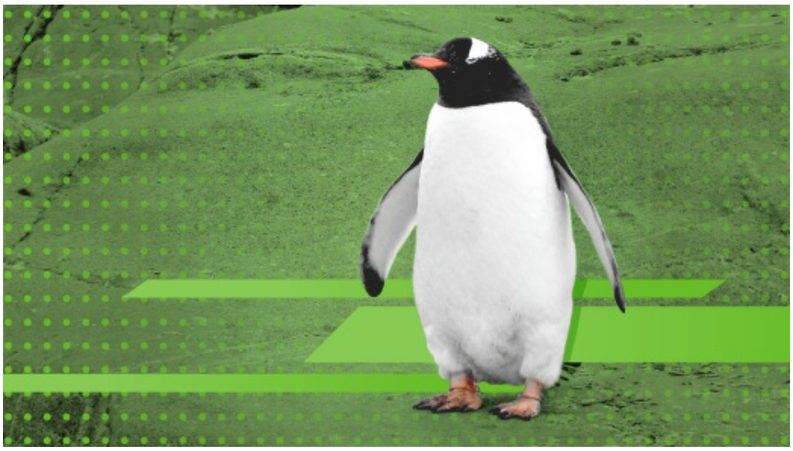


Image by: Opensource.com

Learning to parse XML is often considered a complex venture, but it doesn't have to be. XML is highly and strictly structured, so it's relatively predictable. There are also lots of tools out there to help make the job manageable.

One of my favorite XML utilities is XMLStarlet, an XML toolkit for your terminal. With XMLStarlet, you can validate, parse, edit, format, and transform XML data. XMLStarlet is a relatively minimal command, but navigating XML is full of potential, so this article demonstrates how to use it to query XML data.

Install

XMLStarlet is installed by default on CentOS, Fedora, and many other modern Linux distributions, so just open a terminal and type `xmlstarlet` to access it. If XMLStarlet isn't already installed, your operating system offers to install it for you.

Alternately, you can install the `xmlstarlet` command from your package manager:

```
$ sudo dnf install xmlstarlet
```

On macOS, use MacPorts or Homebrew. On Windows, use Chocolatey.

Should all else fail, you can install it manually from the source code on Sourceforge.

Parsing XML with XMLStarlet

There are many tools designed to help parse and transform XML data, including software libraries that let you write your own parser and complex commands like `fop` and `xsltproc`. Sometimes you don't need to process XML data, though, you just need a convenient way to extract important data from, update, or just validate it. For spontaneous XML interactions, I use `xmlstarlet`, a classic "Swiss Army knife"-style application that does the most common XML tasks. You can see what it has to offer by running the command along with the `--help` option:

More Linux resources

- Linux commands cheat sheet
- Advanced Linux commands cheat sheet
- Free online course: RHEL Technical Overview
- Linux networking cheat sheet
- SELinux cheat sheet
- Linux common commands cheat sheet

```
$ xmlstarlet --help
Usage: xmlstarlet [<options>] <command> [<cmd-
options>]
where <command> is one of:
  ed    (or edit)      - Edit/Update XML
document(s)
  sel   (or select)    - Select data or query XML
document(s) (XPATH, etc)
  tr    (or transform) - Transform XML document(s)
using XSLT
  val   (or validate)  - Validate XML document(s)
(well-formed/DTD/XSD/RelaxNG)
  fo    (or format)    - Format XML document(s)
  el    (or elements)  - Display element structure
of XML document
  c14n  (or canonic)   - XML canonicalization
  ls    (or list)      - List directory as XML
[...]
```

You can get further help by appending `--help` to the end of any of these subcommands:

```
$ xmlstarlet sel --help
  -Q or --quiet             - do not write anything to standard output.
  -C or --comp              - display generated XSLT
  -R or --root              - print root element <xsl-select>
  -T or --text              - output is text (default is XML)
  -I or --indent            - indent output
[...]
```

Selecting data with sel

You can view the data in XML with the `xmlstarlet select` (`sel` for short) command. Here's a simple XML document:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<xml>
  <os>
   <linux>
    <distribution>
      <name>Fedora</name>
      <release>7</release>
      <codename>Moonshine</codename>
      <spins>
        <name>Live</name>
        <name>Fedora</name>
        <name>Everything</name>
      </spins>
    </distribution>

    <distribution>
      <name>Fedora Core</name>
      <release>6</release>
      <codename>Zod</codename>
      <spins></spins>
    </distribution>
   </linux>
  </os>
</xml>
```

When looking for data in an XML file, your first task is to focus on the node you want to explore. If you know the path to the node, specify the full path with the `--value-of` option. The earlier in the Document Object Model (DOM) tree you start to explore, the more information you see:

```
$ xmlstarlet select --template \
--value-of /xml/os/linux/distribution \
--nl myfile.xml
      Fedora
      7
      Moonshine

        Live
        Fedora
        Everything

      Fedora Core
      6
      Zod
```

The `--nl` stands for "new line," and it inserts copious amounts of whitespace to ensure your terminal prompt gets a new line after your results are in. I've removed some of the excess space in the sample output.

Narrow your focus by descending further into the DOM tree:

```
$ xmlstarlet select --template \
--value-of /xml/os/linux/distribution/name \
--nl myfile.xml
Fedora
Fedora Core
```

Conditional selects

One of the most powerful tools for navigating and parsing XML is called XPath. It governs the syntax used in XML searches and invokes functions from XML libraries. XMLStarlet understands XPath expressions, so you can make your selection conditional with an XPath function. XPath features a wealth of functions, and it's documented in detail by W3C, but I find Mozilla's XPath documentation more concise.

You can use square brackets as a test function, comparing the contents of an element to some value. Here's a test for the value of the `<name>` element, which returns the release number associated only with a specific match.

Imagine for a moment that the sample XML file contains all Fedora releases beginning with 1. To view all release numbers associated with the old name "Fedora Core" (the project dropped "Core" from the name from release 7 onward):

```
$ xmlstarlet sel --template \
--value-of '/xml/os/linux/distribution[name = "Fedora Core"]/release' \
--nl myfile.xml
6
5
4
3
2
1
```

You could view all codenames for those releases, too, by changing the `--value-of` path to `/xml/os/linux/distribution[name = "Fedora Core"]/codename`.

Matching paths and getting values

An advantage of viewing XML tags as nodes is that once you find the node, you can think of it as your current "directory" of data. It's not really a directory, at least not in the filesystem sense, but it is a collection of data that you can query. To help you keep your destination and the data "inside" it separate, XMLStarlet differentiates between what you're trying to match with the `--match` option and the value of the data you want with a `--value-of` option.

Suppose you know that the `<spin>` node contains several elements. That makes it your destination. Once you're there, you can use `--value-of` to specify which element you want a value for. To look at all elements, use a dot (`.`) to represent your current location:

```
$ xmlstarlet sel --template \
--match '/xml/os/linux/distribution/spin' \
--value-of '.' --nl myfile.xml \
Live
Fedora
Everything
```

As with navigating the DOM, you can use XPath expressions to limit the scope of what data is returned. In this example, I use the `last()` function to retrieve just the last element in the `spin` node:

```
$ xmlstarlet select --template \
--match '/xml/os/linux/distribution/spin' \
--value-of '*[last()]' --nl myfile.xml
Everything
```

In this example, I use the `position()` function to select a specific element in the `spin` node:

```
$ xmlstarlet select --template \
--match '/xml/os/linux/distribution/spin' \
--value-of '*[position() = 2]' --nl myfile.xml
Fedora
```

The `--match` and `--value-of` options can overlap, so it's up to you how you want to use them together. These two expressions, in the case of the sample XML, do the same thing:

```
$ xmlstarlet select  --template \
--match '/xml/os/linux/distribution/spin' \
--value-of '.' \
--nl myfile.xml
Live
Fedora
Everything

$ xmlstarlet select --template \
--match '/xml/os/linux/distribution' \
--value-of 'spin' \
--nl myfile.xml
Live
Fedora
Everything
```

Getting comfortable with XML

XML can seem over-verbose and unwieldy at times, but the tools built to interact with it consistently surprise me. If you're looking to take advantage of XML, then XMLStarlet could be a good entry point. The next time you're about to open an XML file to look at structured data, try using XMLStarlet and see if you can query that data instead. The more comfortable you get with XML, the better it can serve you as a robust and flexible data format.

What to read next

What is XML?

Get to know XML, a strict yet flexible markup language used for everything from documentation to graphics.

Seth Kenlon
(Team, Red Hat)

July 12, 2021

Write in XML with the NetBeans text editor

NetBeans is a popular Java IDE, making it a handy text editor for developers already using it.

Seth Kenlon
(Team, Red Hat)

December 14, 2020

Tags: LINUX

Seth Kenlon

Seth Kenlon is a UNIX geek, free culture advocate, independent multimedia artist, and D&D nerd. He has worked in the film and computing industry, often at the same time. He is one of the maintainers of the Slackware-based multimedia production project Slackermedia.

More about me

2 Comments

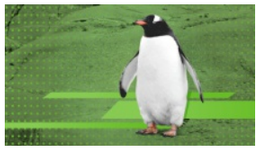
These comments are closed, however you can Register or Login to post a comment on another article.

madtom1999 | July 16, 2021

Anything similar for JSON etc?

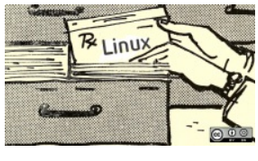
> Seth Kenlon | July 16, 2021
>
> jq is very popular, and very nice to use: https://opensource.com/article/20/6/modern-linux-command-line-tools#jq
>
> There are also so Python libraries that can help you design your own solutions. https://www.redhat.com/sysadmin/yaml-tips but I think jq is probably the best all-rounder.

Related Content



Lengthen the life of your hardware with Linux

3 ways to take screenshots on Linux



A sysadmin's guide to network interface configuration files

This work is licensed under a Creative Commons Attribu

Location of your Mastodon instance:

TOOT

-- Please Select --

ABOUT THIS SITE

The opinions expressed on this website are those of each author, not of the author's employer or of Red Hat.

Opensource.com aspires to publish all content under a **Creative Commons license** but may not be able to do so in all cases. You are responsible for ensuring that you have the necessary permission to reuse any work on this site. Red Hat and the Red Hat logo are trademarks of Red Hat, Inc., registered in the United States and other countries.

A note on advertising: Opensource.com does not sell advertising on the site or in any of its newsletters.

CONTACT

- Follow us @opensource.com on Twitter
- Like Opensource.com on Facebook
- Watch us at Opensource.com
- Follow us on Mastodon
- RSS Feed

opensource.com

Copyright ©2022 Red Hat, Inc.

Privacy Policy |
Terms of use |
Contact