Virtual Room #1

Hosted By: Wendell Piez, IT Security Specialist, NIST/ITL/CSD

Disclaimer: Portions of the event may be recorded and audience Q&A or comments may be captured. The recorded event may be edited and rebroadcasted or otherwise made publicly available by NIST. By attending this event, you acknowledge and consent to having your conversation recorded.

NIST

oscal2022@nist.gov
conferences@nist.gov
OSCAL Tools

Open-source XSLT for OSCAL

(and more)

Wendell Piez
NIST / ITL / CSD
One resource among many

Too many OSCAL Tools sites is a good problem to have....

... we do need a way to make the work available ...

Today we talk about https://github.com/usnistgov/oscal-tools

Other OSCAL Tools initiatives are listed on https://pages.nist.gov/OSCAL/tools/
Origins and purpose

Started as a repository for code not normative for OSCAL, but supporting it
Built internally to support various efforts and as demonstrations
But was also made generalized and customizable
So why not make it available
  - Show what can be done
  - Catalyze other initiatives coding to OSCAL

Did anyone mention web sites with git repositories in back?

... a static web site is also a platform for demonstration ...
Current and planned scope

XSLTs for styling OSCAL began as spinoffs from NIST project work / proofs-of-concept

Repository is also useful as a clearing house for utilities

Has also hosted other (NIST) projects before they find permanent homes – and might do so again

Aspiring to an "always finished, always growing" maintenance model
XSLT for download

Why XSLT? (Long topic)
tldr: XSLT 3.0 is not your grandma's XSLT
Performance, versatility, safety features

Record of success encapsulating and managing the "static presentation" problem
Declarative language suitable for a library supporting local customizations
Works in wide range of architectures / environments
Scalable – both throughput and complexity
Delivers functionality on a commodity freeware stack
Produces well-understood, useful outputs
  HTML, CSS, PDF... CSV, plain text, structured code...
Formal presentation for web and print

Interpreting semantic encoding for human readers

Elapsed runtimes for 5.5MB OSCAL catalog:
(1189 controls and enhancements)

HTML: 0.6sec
PDF: 17.1sec (1.1 sec transformation, 16sec layout)
Commodity tools / no optimization
This PDF has 766pp.

XSLT core use case: producing human-readable pages from XML source data
Other capabilities

CSS for OSCAL authoring or lightweight browser display
XSLT UUID refresher utility
  assigns new (random) UUIDs
XSLT OSCAL "blank" document generator
  catalog, profile, SSP, SAP, or SAR (so far)
  valid but empty; fresh timestamp

... and more to come ...
Link checkers
Network discovery
Semantic analysis, mapping, exposition
(Please contribute ideas)
usnistgov Pages site

https://pages.nist.gov/oscal-tools/

Web site accompanying the software repository
Just finalized! February 2022
Expected to grow
  Updates to current projects
  New projects
  Cover/describe projects in other repositories
CSX demonstrations


Started as experiments in a personal repo, then grew

Demonstrates a concept: XSLT transformations in the browser

   But this time, you provide the data

   (It is processed for display and shown, but not passed anywhere.)
Further work / feedback

What should we be doing?

Github Issues
Public Gitter chat channel
We also have an oscal/xslt-etc 'room' in Gitter

Web site – tutorials; hints and ideas?
Tools – for online or offline use?