A Generic STS Viewer in Your Browser

Wendell Piez

National Institute of Standards and Technology

INFORMATION TECHNOLOGY LABORATORY | COMPUTER SECURITY DIVISION

JATS-Con 2023

13 June 2023

NIST | NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

Disclaimers

The opinions, recommendations, findings, and conclusions represented in this slide deck do not necessarily reflect the views or policies of NIST or the United States Government.

Certain software is identified for historical and descriptive purposes. Such identification is not intended to imply recommendation or endorsement of any product or service by NIST, nor is it intended to imply that the software identified is necessarily the best available for the purpose.

"Test Your Theory"

Garden Tour Demo

During the talk feel free to wander

Open a browser tab here:
https://pages.nist.gov/xslt-blender/sts-viewer/

Take a look!

- Any STS, JATS or BITS data is worth a try
- Expectation: no-namespace JATSy XML
  (um, no external parsed entities...)
- Your file is not uploaded
  this is all in your browser

Public Conversation

Now is your chance to shine.

You have cool data. You have a cool problem. You have a good bug.

Volunteer in the chat and we will look together.

👀 Remember we are being recorded (*doing this for posterity* 😄)

Project goals

Origins

A coincidence of circumstances:

- A high-value, high-touch publication with a revision cycle
- An opportunity to convert “down hill” into STS XML
  - From this XML source (OSCAL), producing NISO STS XML was straightforward
- A need to test and cross-test
  - Making systems work is not always the same as showing them working
- Available system resources and expertise for execution

Short version: we had a large, complex STS document and wished to check its quality, prior to and separately from the publication pipeline in development elsewhere in the organization.

Project Goals - Today

Goals within NIST

- Continue to reap rewards of investments in tools and skills
  - Making STS is the easier part
  - Validating/proofing it is more difficult
- Provide an enabling counter-check for systems in development
  - Testing and proving data, systems, *and requirements*
- Enable broader stakeholder participation in XML-based workflows
  - Especially authors and editors (“owners”)
  - In early stages
  - Where their interventions are most meaningful

Project Stretch Goals

- An application that goes beyond demo
  - Pays for itself soon and repeatedly
  - Can be sustained and supported
    — a light lift for over-committed technical staff
- Something that supports broader STS and JATS/BITS communities
  - Open source (as the best sustainability model for us)
  - Following a replicable pattern, easy to reuse and adapt
  - Complementary to existing tools (network effects of standards)

How does this work

- Open the page
- Load your NISO STS data
- *Et voilà — transformation*
- Invalid is okay too

Tell me again, where's the page?

https://pages.nist.gov/xslt-blender/sts-viewer/

What you need to know

- Transformation runs on your system using your browser's XSLT 1.0 engine
  - no data exposure (“load” is not “upload”)
  - works off line too (once cached)
  - XSLT is limited to version 1.0 (“toy” XSLT)
- XSLTs are custom-written for this application
  - easily adaptable
  - easily replaceable

Why client-side XSLT

This should speak for itself …

(Feel free to raise questions)

Why XSLT 1.0

- Attractive distribution model (*none*)
- Attractive maintenance model (*little*)
- Provides some “natural” (inherent) limits
- Modest risk profile (*browser crash?*)

Kitchen implement analogy 🥣 (this one is already in the drawer)

Working within known limits is good, when we have options

The part that doesn't speak for itself

- This is about data security
- Building a *trust network*
  - Mutually enforcing poles
    - But everyone is their own center
  - Everyone benefits when anyone benefits
- Trust is built on understanding
  - Capabilities, not only intentions
  - *Distributing capability* is the key

NEWS FLASH

New demo application! STS Checker

https://pages.nist.gov/xslt-blender/sts-viewer/checker.html

- Internal link review
- Lists of figures and tables
- Bibliographic tagging preview
- … and more

Demo time

What do you have to show?

Technical details

- Yes, there is some Javascript here along with XSLT
  - Coded in Typescript, compiled for release, and published under the same (public domain) license
  - Uses standard (W3C DOM) APIs to invoke XSLT
  - No external dependencies
- A host project XSLT Blender provides a platform for these scripts
  - Offers a browser self-test for required functionality
  - With a growing set of interesting demonstrations
- These applications are tested using Chromium-based browsers and Firefox
  - But we aim for standards conformance, not any software application *per se*
- Important support is provided by Javascript/ECMAScript features unavailable in 2003
  - Asynchronous processing (**promise** objects)
  - Constraints around CORS (cross-origin resource sharing)
  - Unicode, Unicode fonts and relief of requirements for parsed entities
  - Event and file resource handling
  - And others (array operators, anyone?)

How and when to use the STS Viewer

- Already suitable for “casual” use
  - Sanity checker / cross checker
- Carefully examine dependency management chain
  - For all your software commitments
  - Use Viewer and Checker in that context
- Understand issues of alignment
  - Alignments are investments that start paying for themselves immediately
  - Align expectations, capabilities and the real world, and you will be fine
- Plan for long term too

How and when to fork and modify the STS Viewer

- Use Github to copy (clone or fork) the XSLT Blender in the usual way.
- As U.S. Government work product, this software is placed into the public domain to be used and reused freely by public and private organizations and individuals.
- We cannot indemnify the user or warrant the software in any way, but we can make it available to use, copy, study, and test.
- When you make available a copy, variant or derived product in any form, you must first *remove all NIST and project branding, links and other identifying information* from any page views, as well as any unused resources or dependencies in the project.
    - Your pages should represent the work as yours
    - Acknowledge NIST and the project with links in your documentation and `readme` files
    - Consult the license

Thank you!

portal https://pages.nist.gov/xslt-blender

repo https://github.com/usnistgov/xslt-blender

INFORMATION TECHNOLOGY LABORATORY | COMPUTER SECURITY DIVISION