



BERKELEY LAB

Bringing Science Solutions to the World



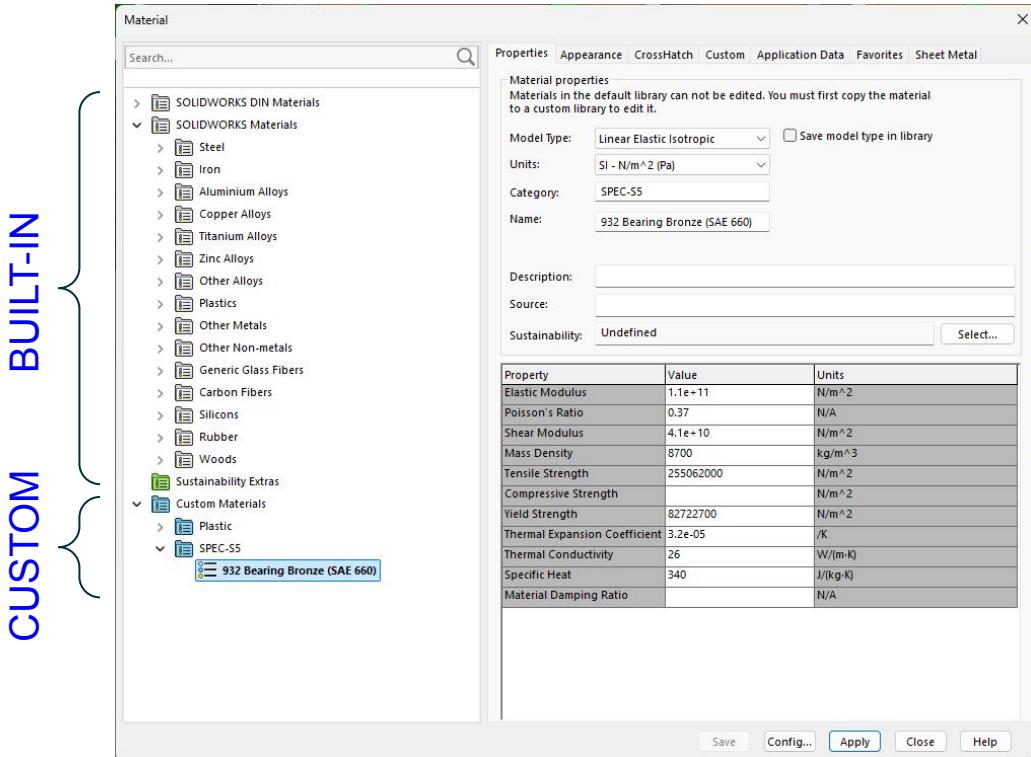
Solidworks material library sharing strategy

Joe Silber
Engineering Division

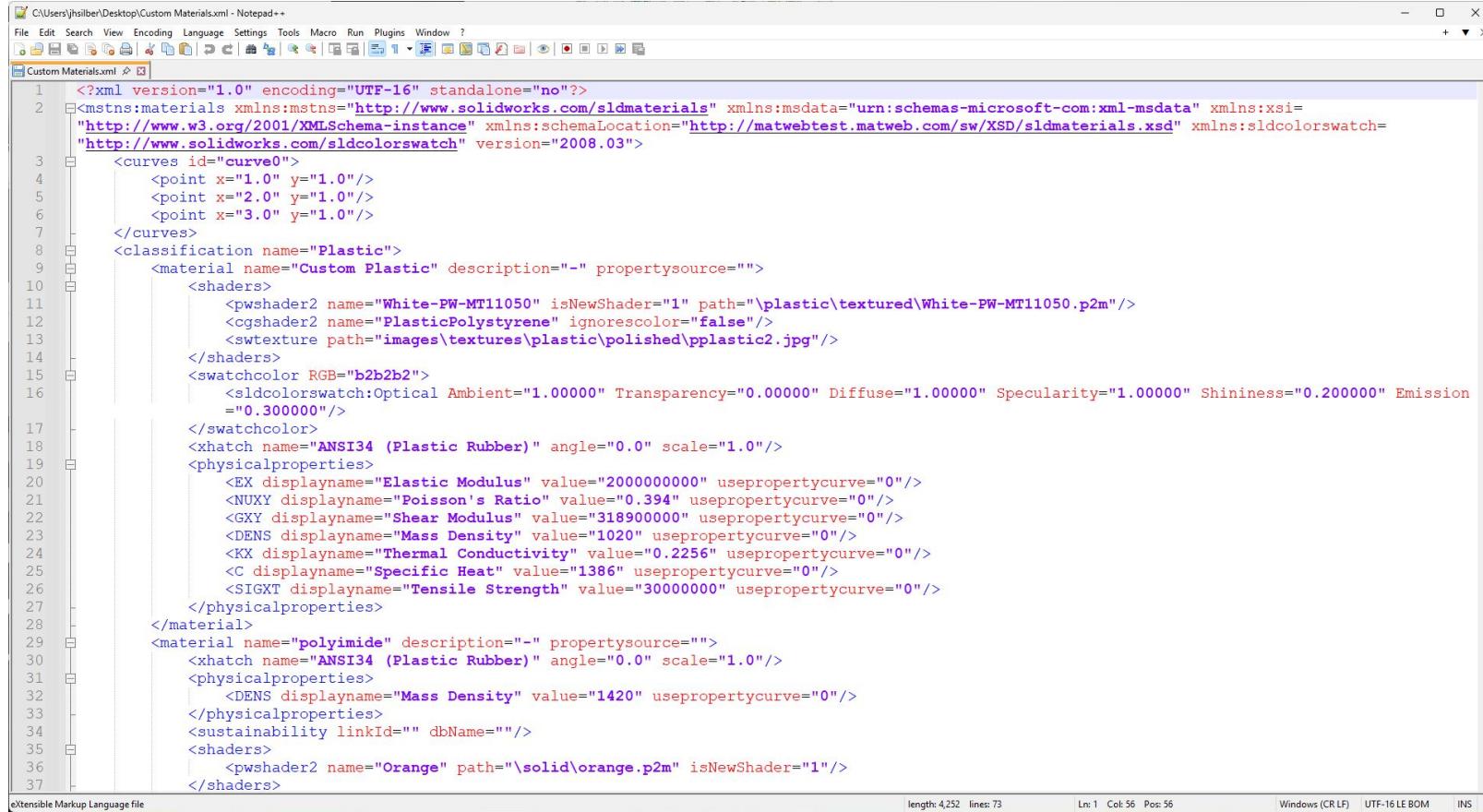
2026-02-18

Background

- Solidworks has both built-in and customizable material libraries
- At LBNL we share CAD data in Windchill
- Windchill support is provided for Solidworks template files, but not material library files
- We have at least three projects in Engineering right now using Solidworks + Windchill:
 - DUNE ND
 - ePIC SVT
 - Spec-S5
- We frequently need to define custom materials that aren't in the standard library
- To date, these material defs live in:
 - text file buried on user's disk
 - the particular part file where a given material is used
- Can we find a reliable, practical, low-friction solution to share custom materials?



Solidworks material files are XML text files like this example



The screenshot shows a Windows desktop environment with a Notepad++ window open. The title bar reads "C:\Users\jhsilber\Desktop\Custom Materials.xml - Notepad++". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, and ?.

The XML code in the editor represents a custom material definition for Solidworks. It includes sections for curves, classification (Plastic), material properties (Custom Plastic), shaders, swatchcolor, xhatch, physical properties, and another material entry for polyimide.

```
<?xml version="1.0" encoding="UTF-16" standalone="no"?>
<mstns:materials xmlns:mstns="http://www.solidworks.com/sldmaterials" xmlns:msdata="urn:schemas-microsoft-com:xml-msdata" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:schemaLocation="http://matwebtest.matweb.com/sw/XSD/sldmaterials.xsd" xmlns:sldcolorswatch="http://www.solidworks.com/sldcolorswatch" version="2008.03">
    <curves id="curve1">
        <point x="1.0" y="1.0"/>
        <point x="2.0" y="1.0"/>
        <point x="3.0" y="1.0"/>
    </curves>
    <classification name="Plastic">
        <material name="Custom Plastic" description="-" propertysource="">
            <shaders>
                <pshader2 name="White-PW-MT11050" isNewShader="1" path="\plastic\textured\White-PW-MT11050.p2m"/>
                <cgshader2 name="PlasticPolystyrene" ignoreColor="false"/>
                <swttexture path="images\textures\plastic\polished\pplastic2.jpg"/>
            </shaders>
            <swatchcolor RGB="b2b2b2">
                <sldcolorswatch:Optical Ambient="1.00000" Transparency="0.00000" Diffuse="1.00000" Specularity="1.00000" Shininess="0.200000" Emission ="0.300000"/>
            </swatchcolor>
            <xhatch name="ANSI34 (Plastic Rubber)" angle="0.0" scale="1.0"/>
            <physicalproperties>
                <EX displayname="Elastic Modulus" value="2000000000" usepropertycurve="0"/>
                <NUXY displayname="Poisson's Ratio" value="0.394" usepropertycurve="0"/>
                <GXY displayname="Shear Modulus" value="318900000" usepropertycurve="0"/>
                <DENs displayname="Mass Density" value="1020" usepropertycurve="0"/>
                <KX displayname="Thermal Conductivity" value="0.2256" usepropertycurve="0"/>
                <C displayname="Specific Heat" value="1386" usepropertycurve="0"/>
                <SIGXT displayname="Tensile Strength" value="30000000" usepropertycurve="0"/>
            </physicalproperties>
        </material>
        <material name="polyimide" description="-" propertysource="">
            <xhatch name="ANSI34 (Plastic Rubber)" angle="0.0" scale="1.0"/>
            <physicalproperties>
                <DENs displayname="Mass Density" value="1420" usepropertycurve="0"/>
            </physicalproperties>
            <sustainability linkId="" dbName="" />
            <shaders>
                <pshader2 name="Orange" path="\solid\orange.p2m" isNewShader="1"/>
            </shaders>
        </material>
    </classification>
</mstns:materials>
```

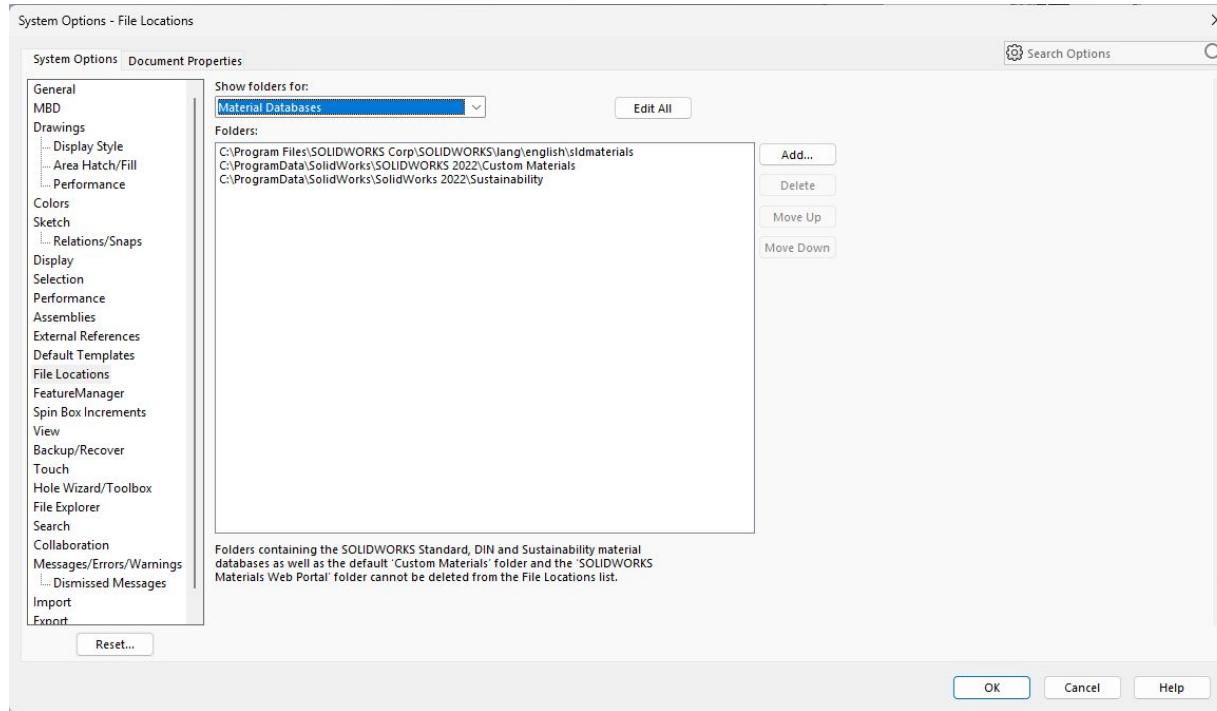
Extensible Markup Language file

length: 4,252 lines: 73

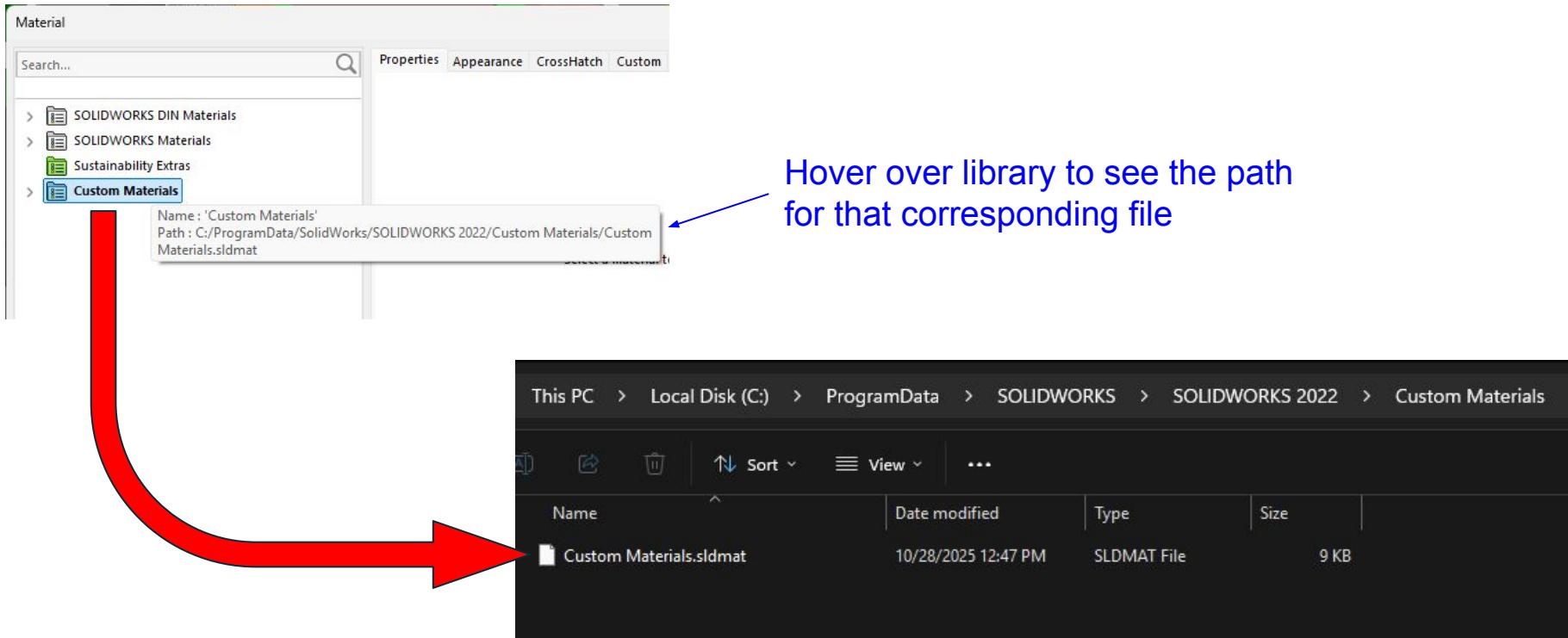
Ln: 1 Col: 56 Pos: 56

Windows (CR LF) UTF-16 LE BOM INS

Multiple folder locations on the user's disk can be specified where Solidworks will look for material database files



In the material picker, “Libraries” correspond 1:1 to database text files on disk



Rough proposal (for discussion)

1. Establish shared Google drive folder
 - a. Or perhaps Github? – more archival, better traceability
2. Make a **single file** for **vetted, shared** properties
 - a. Checked by someone other than the author
 - b. Checks / changes are logged in...
 - i. A spreadsheet in the drive?
 - ii. Custom notes in each specific material?
 - c. Reasonably trustworthy for anyone to use
3. Also make a **folder** for **personal** properties
 - a. More flexible, for immediate sharing
 - b. Not vetted
 - c. Folder can contain any number of files
 - d. Each user can arbitrarily post their own personal properties file
4. Ask that users adopt a common naming convention like **lbnlcommon_2026-02-18.sldmat** and **jhsilber_2026-02-18.sldmat**
5. Finally, provide an **archive folder** to toss old files
6. Users can then download files

Discussion notes

- Peter - make sure if someone directly opens from a Google drive location, that Solidworks doesn't autosave changes to the common file
- Gordon - can we put this on the Windchill common space?
- Joe - feeling less convinced about value of the work-in-progress “personal” files
- Nick - we should understand clearly the refresh behavior
 - does Solidworks autorefresh within a part if the sldmat internals change?
- Nick - does the xml file have a natural location for source notes?
 - and are these easily accessible from the UI?
- Nick - could we write some code to pull properties from a common Ansys material file (perhaps what ALS-U is working on? or just in Windchill common space) and write a Solidworks material file to disk
- Andrew - does collab.lbl.gov token work for X drive access remote? or have to vpn in?
- Andrew - line-by-line comparison features (diff) inherent to a workflow like GitHub could be quite helpful – in depth traceability is built in

Windchill → version control, user trace, no internal trace, CAD department bureaucratic challenge

Google drive → very flexible, easy user control, no traceability really, maybe too malleable

Github → excellent version control, traceability, shareability, independence, write safety, detail checks / approval process (pull requests etc)

Leaning toward github right now...