- Log in

User menu

- Main page
- About
- Documentation
- Roadmap
- Projects
- Downloads
- Media
- Sponsors
- TV
- License...

DECOR-Publication

From art-decor.org

Contents

Introduction

DECOR definitions may be exported into various formats. You may export into any format and any layout you need. DECOR comes with many predefined exports with the toolset, all conveniently accessible from one main stylesheet, and configurable through a parameter file. This page explains how to set it up and how to make it work for your purposes.

Available formats

The exports that come supported from SourceForge are:

- HTML - documents everything in the DECOR file at conversion time
- Schematron - creates ISO Schematron (http://www.schematron.com) with XSLT 2.0 query binding. These are useful for validating instances that are based on the specification
- DocBook - *unfinished work*. DocBook (http://www.docbook.org) is an XML based standard that allows conversion into many more formats such as PDF.
- PDF - *not yet finalized*. As of now the docbook method is also used to prepare HTML output that is translatable into a PDF booklet. It uses *Prince (http://princexml.com)* as the conversion mechanism as DECOR makes use of some special CSS features that are not correctly transformed by some CSS parsers.

Set up

Exporting DECOR into other formats could be done in more ways than described here. The discussed way is the recommended way.

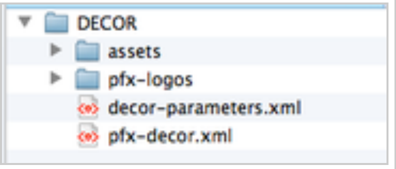
Prerequisites

- DECOR exports are based on XSLT 2.0. To run any export you'll need an XSLT 2.0 capable processor. Development and testing is done based on Saxon-PE/HE/SE 9.x (http://www.saxonica.com) . XSLT 2.0 support is also into built into tools such as <oXygen/> (http://oxygenxml.com) (Windows/Linux/OSX) and XMLSpy (http://altova.com) (Windows).
- Download the DECOR assets folder from Download
- Create a directory layout resembling the layout below
- Download your DECOR file from the DECOR server. Currently you may only do this through direct eXist-db access (*TODO: add page on how to*), but in future versions of ART-DECOR there should be UI options for precompiling and downloading this file.

Directory layout

pfx is your project prefix. **Bold** is a folder.

- **DECOR**
  - **assets** -- from DECOR transformations, contains CSS, Javascript, images etc. Note that you may add your own CSS etc. into this directory if you need to match the styles common to your environment.
  - ***pfx*-logos** -- holds any image configured in DECOR copyright/@logo
  - *pfx*-decor.xml -- your DECOR file
  - decor-parameters.xml -- contains all parameters for exporting. Name is fixed
  - **resources** -- contains more CSS, Javascript, images etc. which are used in dataset overview (especially for the expandable tree view and hideable columns).



Folder layout

decor-parameters.xml

This is the basic setup of the decor-parameters.xml file:

```
<?xml version="1.0" encoding="UTF-8"?>
<decor-parameters xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
xsi:noNamespaceSchemaLocation="http://art-decor.org/ADAR/rv/decor-parameters.xsd"
   <!-- create Schematron with switchCreateSchematron1 or
           don't create (switchCreateSchematron0) -->
   <switchCreateSchematron1/>
   <!-- create Schematron with HL7v3 messaging wrappers includes
           (switchCreateSchematronWithWrapperIncludes1) or don't create
           (switchCreateSchematronWithWrapperIncludes0) -->
   <switchCreateSchematronWithWrapperIncludes0/>
   <!-- create closed qualification quality schematrons (switchCreateSchematronCl
           respect whatever is actually configured (switchCreateSchematronClosed(
   <switchCreateSchematronClosed0/>
   <!-- create Schematrons with includes only for those that are actually part o
           (representingTemplate/@ref and/or @flexibility) -->
   <switchCreateSchematronWithExplicitIncludes1/>
   <!-- create HTML documentation (switchCreateDocHTML1) or
           don't create (switchCreateDocHTML0) -->
   <switchCreateDocHTML1/>
   <!-- create SVG images where applicable (switchCreateDocSVG1) or
           don't create (switchCreateDocSVG0) -->
   <switchCreateDocSVG1/>
   <!-- create DocBook documentation (switchCreateDocDocbook1) or
           don't create (switchCreateDocDocbook0) -->
   <switchCreateDocDocbook0/>
   <!-- use local assets dir ../assets (useLocalAssets1) instead of online
           version (useLocalAssets0) on http://art-decor.org/ADAR/rv/assetsvxx/
           (xx depends on DECOR2Schematron.xsl) -->
   <useLocalAssets1/>
   <!-- use local logos dir ../pfx-logos (useLocalLogos1) instead of online
           version (useLocalLogos0) on /decor/project/reference + /pfx-logos/ --
   <useLocalLogos1/>
   <!-- add a logo to the top right on HTML pages, optionally with a URL
           (useCustomLogo1) or don't add (useCustomLogo0) -->
   <useCustomLogo0 src="../pfx-logos/nictiz.jpg" href="http://pfx.art-decor.org"/
   <!-- If set, decor2html will get overview results using RetrieveTransaction fo
           most recent version in project-decor.xml. If absent, current Retrieve
           results will be used. -->
   <useLatestDecorVersion1/>
   <!-- create artifacts in timestamped directories (inDevelopment0) or
           create artifacts in pfx-runtime-develop / pfx-html-develop (inDevelopm
           while we are just publishing/testing intermediate results-->
   <inDevelopment1/>
   <!-- override /decor/project/@defaultLanguage with a specific language.
           Value ALL activates all languages -->
   <!--defaultLanguage>nl-NL<defaultLanguage-->
   <!-- If switchCreateDatatypeChecks1 all XML elements with a datatype are
           created with an extends in Schematron. Normally left active. -->
   <switchCreateDatatypeChecks1/>
   <!-- Creates pseudo XML instances in HTML folder -->
   <createDefaultInstancesForRepresentingTemplates0/>
   <!-- Customizations to be used in RetrieveTransaction. May contain attribute
           @hidecolums, which contains columns to be hidden when calling Retrieve
   <useCustomRetrieve1/>
</decor-parameters>
```

Recommended Procedure

The recommended procedure for publication is as follows:

- Make sure your DECOR file is consistent. This may sound obvious, but better to overstate it.
- Create a version or release from ART. (Project, tab Versioning.) If one or more languages are checked after 'Compile-save', a compiled version will be stored in eXist (apps/version/{project}/version-{date}. This will make your project is self contained.
- Retrieve the decor file (for instance with RetrieveTransaction) for local generation of schematron, HTML docs etc. Use:
  - decor-parameters.xml -- useLocalAssets1, useLocalLogos1, inDevelopment0
- When you intend to publish intermediate output to test you could opt to switch to a development build using
  - decor-parameters.xml -- inDevelopment1
- When your DECOR file references any Building_Block_Repository_(BBR) you need to compile your project before transforming (which is done automatically by making a version from within ART).
  - Call decor/services/modules/RetrieveProject.xquery?prefix=**pfx-**&mode=compiled&language=**en-US** on your server
- Transform your *pfx-decor.xml* file using DECOR2schematron.xsl.

Example Java call (Windows):

```
java -jar %saxonPath%/saxon9.jar -xsl:http://art-decor.org/ADAR/rv/DECOR2schematron.xsl -o:pfx-decor.html
```

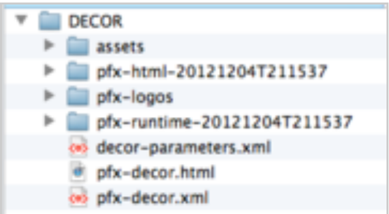
Example Java call (OSX/Unix/Linux):

```
java -jar ${saxonPath}/saxon9.jar -xsl:http://art-decor.org/ADAR/rv/DECOR2schematron.xsl -o:pfx-decor.html
```

Example output:

```
*** Started 21:15:37.964+01:00
*** Reading DECOR parameter file
*** Creating schematrons based on scenario transaction representing templates with wrapper includes if ava
*** Benchmarking indicator: 2
*** Creating schematrons
*** Creating terminology files
*** Creating documentation html + svg
*** Finished
*** Total processing time 0h 0m 14.128s
```

If you created HTML, you will have a file *pfx*-decor.html next to your *pfx*-decor.xml file that essentially redirects to the *pfx*-html-<dateTime>/index.html. If you created Schematron, this will in the folder *pfx*-runtime-<dateTime> with the main Schematrons being named according to their transaction/@label.



```
DECOR
  assets
  pfx-html-20121204T211537
  pfx-logos
  pfx-runtime-20121204T211537
  decor-parameters.xml
  pfx-decor.html
  pfx-decor.xml
```

Folder layout after export

If (and only if) there is a RESTURI for DS/HTML defined in your project: <restURI for="DS" format="HTML">...uri...</restURI> an 'overview' of dataset will be generated for each dataset and transaction, and shown on the 'Dataset' tab of HTML output.

If useLatestDecorVersion1 is set in decor-parameters.xml, a compiled project will be retrieved with RetrieveTransaction from db/apps/version and used to make the overview. If useLatestDecorVersion0 is set, the current projectr will be retrieved. useCustomRetrieve1 can be used to fine-tune the columns in dataset overview output.

Retrieved from "http://art-decor.org/mediawiki/index.php?title=DECOR-Publication&oldid=915"

- This page was last modified on 1 November 2013, at 09:40.

The ART-DECOR expert group

2010-2013

Who we are **Contact**