

1-26-2017

# Introduction to Scalar

Martin Kass

*Florida International University*, [mkass@fiu.edu](mailto:mkass@fiu.edu)

Follow this and additional works at: <https://digitalcommons.fiu.edu/glworks>

---

## Recommended Citation

Kass, Martin, "Introduction to Scalar" (2017). *Works of the FIU Libraries*. 57.  
<https://digitalcommons.fiu.edu/glworks/57>

This work is brought to you for free and open access by the FIU Libraries at FIU Digital Commons. It has been accepted for inclusion in Works of the FIU Libraries by an authorized administrator of FIU Digital Commons. For more information, please contact [dcc@fiu.edu](mailto:dcc@fiu.edu).

# **INTRODUCTION TO SCALAR**

**MARTIN KASS**

**FIU DIGITAL COLLECTIONS CENTER**

**[MKASS@FIU.EDU](mailto:MKASS@FIU.EDU)**

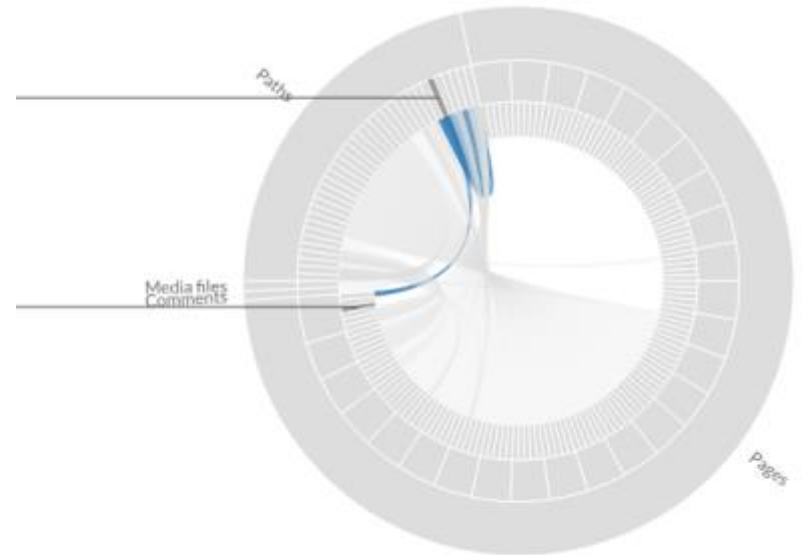
# OVERVIEW

**Scalar is a free open source authoring and publishing platform designed for authors to create digital scholarship online. Scalar allows users to assemble media from multiple sources- and for the creation of structured essay and book length works. A variety of layout options and other add-ons make it ideal for the first-time webpage creator and the experienced web developer alike. It is a project of the Alliance of Networking Visual Culture (ANVC), was released in 2013, and is hosted by University of Southern California.**



# WHAT ARE WE LOOKING AT?

- Flexible platform that requires no Web Design skills to use
- Layout options that are both minimal and ornate
- Strong options for displaying projects involving multiple media
- Rich Dublin Core metadata
- Many ways to visualize the structure of your project (privately or publicly)



# CAPABILITIES



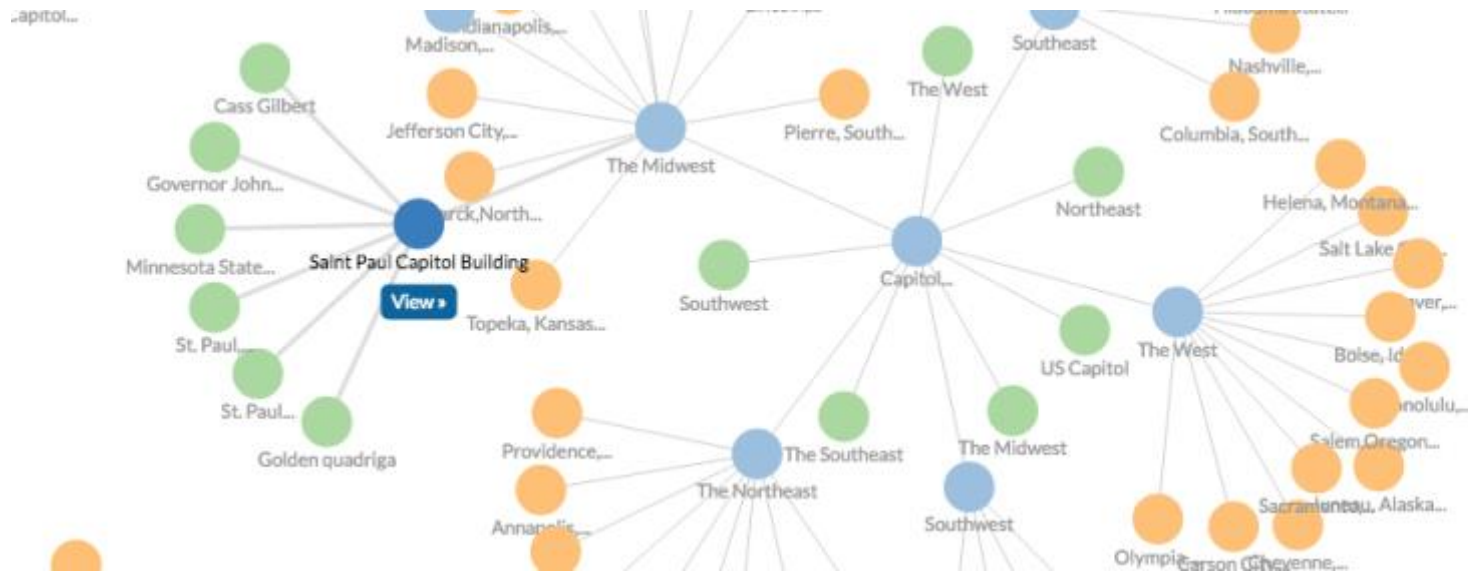
**“Anything can do anything to anything”.**

- **Scalar aims to provide endless opportunities for structuring, displaying, referencing, and citing your content.**
- **Tags, annotations, comments can link any two types of media**
- **Broad support for most popular media formats**
- **Several options for exporting data**
- **Providing multiple layout options within one project**

# VISUALIZATIONS

Visualizations help you and your users navigate through the parent-child relationships in your book, and can also be helpful for debugging.

Can be viewed by page or book level.



# MEDIA

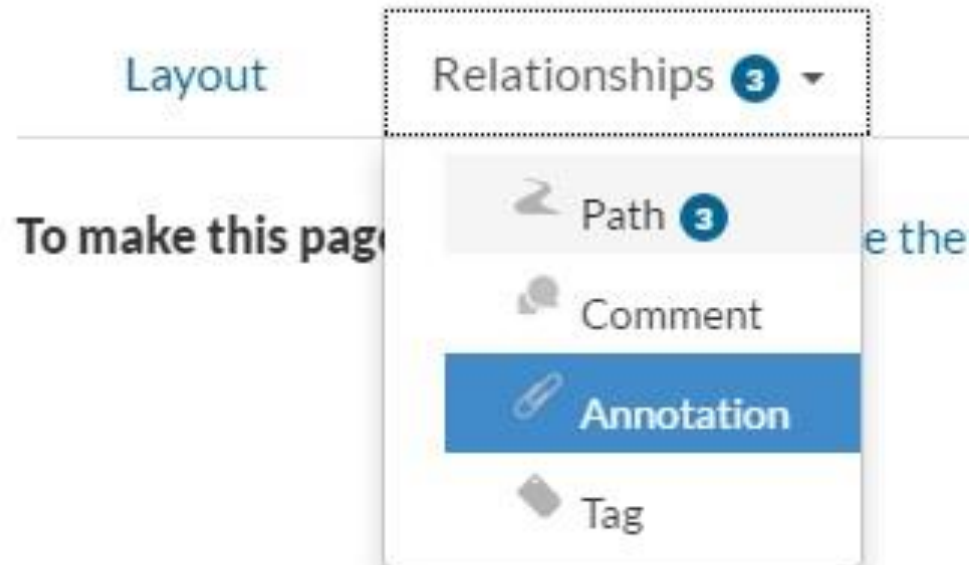
**Files must be less than 2M in size (larger files can be hosted at archive supported by Scalar).**

**Local media files must first be imported to your Scalar site before they are inserted to a specific page.**

**Files can also be imported from the Internet as “Internet Media files”**



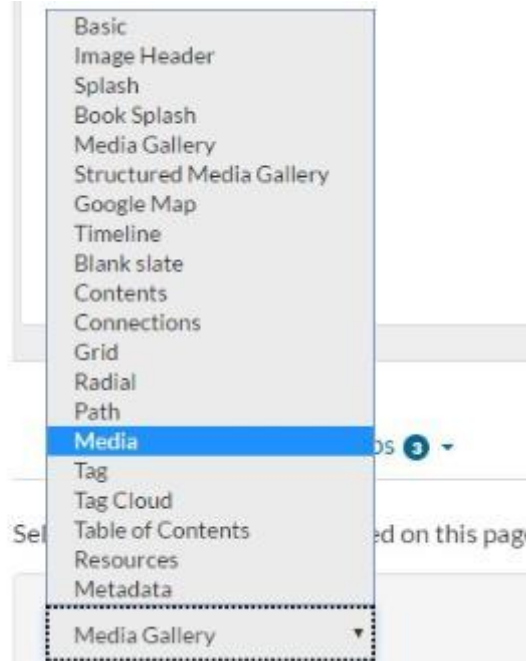
# RELATIONSHIPS



Any page or media file can take on any of these relationships to any other page/ media file.



# LAYOUTS



Each page in a book can have a different layout.

# SUPPORTED MEDIA FORMATS

	Image	Audio	Video	Document
Supported on recent versions of Chrome, Internet Explorer, Safari, and Safari (iOS)	DZI, GIF, JPEG, PNG	MPEG-3 (MP3)	MPEG-4	HTML, Plain Text
Supported only on some of the above browsers		OGG, WAV	FLV, M4V, OGG, QuickTime, WebM	PDF

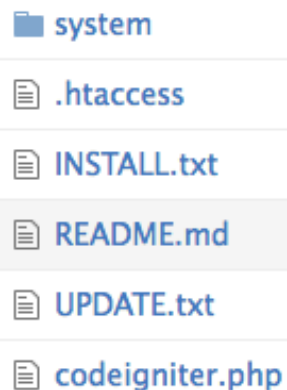
# HOST ON YOUR SERVER

- Scalar allows you to host your own version of Scalar by downloading it from this link.

<https://github.com/anvc/scalar>

- Scalar recommends that you keep track of the changes to its config files, as updates would likely need to be made.

<https://github.com/anvc/scalar/wiki/Changes-to-config-files-over-time>



A screenshot of a file directory listing. The items are listed vertically, separated by thin horizontal lines. The items are: a folder icon followed by 'system', a document icon followed by '.htaccess', a document icon followed by 'INSTALL.txt', a document icon followed by 'README.md' (which is highlighted with a light gray background), a document icon followed by 'UPDATE.txt', and a document icon followed by 'codeigniter.php'.

- system
- .htaccess
- INSTALL.txt
- README.md
- UPDATE.txt
- codeigniter.php

# BULK IMPORTING DATA

Transfer tool in Import/Export in the Dashboard is used for migrating Scalar books from one place to another- (as in one Scalar install to another)

It can be used to

- bulk import pages and media from a spreadsheet
- create an RDF-JSON file that contains all the pages and relationships in your book as a backup

**Export**

The following link will create an RDF-JSON file that contains all pages and relationships contained in this book, which can be used as a backup of your book content or for importing at a later date. Loading the link might take a while depending on the amount of content.

<http://scalar.usc.edu/works/capitolbuildings/rdf/instancesof/content?format=json&rec=1&ref=1>  
Or, download as RDF-XML.

**Import**

The tool below can be used to import pages and relationships from a public Scalar book. Simply place the URL of the source book into the Source Book field. Alternatively, you can import snippets of a Scalar book using the Paste RDF tab ([learn more](#)).

Book URL   Paste Text   Upload File   [List of fields & media that aren't transferred](#)

**Upload File**

Select a file from your computer of type: Scalar-formatted RDF-JSON, individual RDF-JSON nodes, or CSV with header fields from Scalar ontologies — see [Scalar's API Reference](#).

[Browse...](#)

**Destination Scalar Book**

Enter the URL to the destination book below. You need to be logged in to this book already, so we'll also need your login email address.

<http://scalar.usc.edu/works/guide2/bulk-importing-spreadsheets-using-the-transfer-tool?path=advanced-topics>

# SCALAR API

- Combine your content with other data sources
- Create completely new interfaces
- Create your own visualizations
- Receive data through free JavaScript library that does not require interaction with RDF/XML or RDF-JSON.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <script src="http://code.jquery.com/jquery-1.8.1.min.js"></script>
5     <script type="text/javascript" src="http://scalar.usc.edu/works/system/application/view
6     <script type="text/javascript">
7       $( document ).ready( function() {
8
9         scalarapi.setBook( "http://scalar.usc.edu/works/guide" );
10
11        if ( scalarapi.loadNode( "index", true, handleSuccess, handleFailure, 1 ) == "1
12          handleSuccess();
13        };
14
15        function handleSuccess() {
16          var node = scalarapi.getNode( "index" );
```

# LINKS

<http://libguides.fiu.edu/c.php?g=557430&p=4204428>

**Libguide with more information on Scalar**

<http://scalar.usc.edu/scalar/showcase/> **Projects authored with  
Scalar**

<http://scalar.usc.edu/scalar/webinars/> **Future Scalar webinars**