Example Asciidoc Document

Table of Contents

Chapter 1. Example chapter

1.1. Example chapter section

With some example text. And folowed by an example sub section.

1.1.1. Background to the structure of the document

Each chapter and section can be kept in a separate folder and access can be provided on a book, chapter, or section level. The idea behind this is to be able to provide editor access to parts of the document by third parties. Also each chapter and section can be built as its own standalone document or web page so that parts of the overall document can be distributed as individual documents for review or editing without disclosing the entire document with all chapters.

Advantages:

1. Reduces workload for reviewers
2. Allows granular access to material subject to NDAs
3. Reduces the impact of errors made in sections
4. Allows integration with git so as to have tight version control in an automated fashion
5. Can integrate other documents and material
    a. Text configuration data (JSON, txt, YAML, etc) can be linked as includes
    b. Diagrams and tables are linked as includes
        i. Original CSVs are untouched and stay up to date in the document
6. Does NOT require word as the document is ASCII at heart
    a. Material is easiy tranferred to other documents
    b. Recombining the document and expanding or shrinking it are very simple and do not destroy the removed data
    c. Security for the document is high as any data that is removed is NOT present in the output but CAN be seen in the version control (github)
7. When integrated with git ownership of individual lines of documentation can be viewed
    a. The history of each line of documentation can be followed through each commit
    b. Merging the document with other documents is made simpler
    c. The document CAN be split across sources and repositories

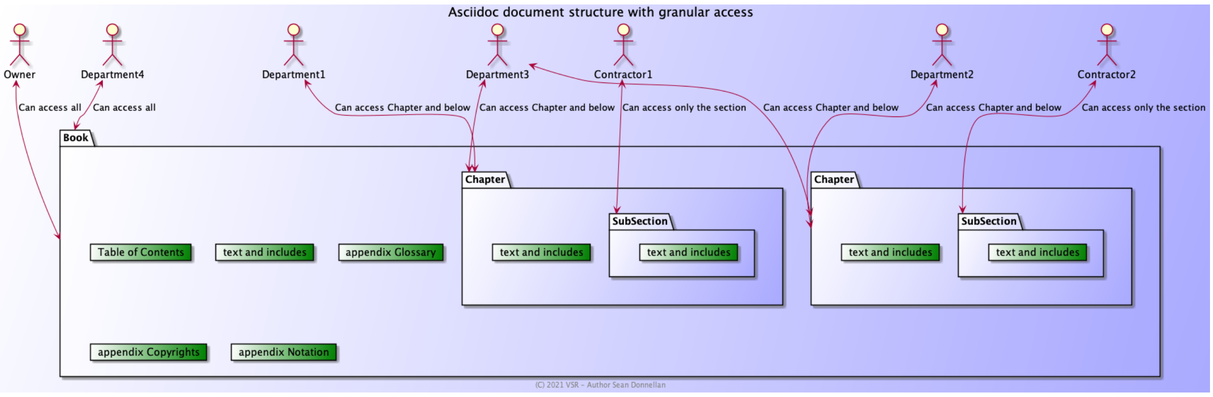

Figure 1. plantuml description of the document structure

Listing 1. source of the above plantuml figure

```
@startuml
title Asciidoc document structure with granular access
footer (C) 2021 VSR - Author Sean Donnellan
skinparam BackgroundColor #fff/aaf
skinparam CardBackgroundColor #fff/green

folder Book {
    card „Table of Contents“ as Book.toc
    card „text and includes“ as Book.text1
    folder Chapter as Book.Chapter1 {
        card „text and includes“ as Book.Chapter1.text1
        folder SubSection as Book.Chapter1.SubSection1 {
            card „text and includes“ as Book.Chapter1.SubSection1.text1
        }
    }
    folder Chapter as Book.Chapter2 {
        card „text and includes“ as Book.Chapter2.text1
        folder SubSection as Book.Chapter2.SubSection1 {
            card „text and includes“ as Book.Chapter2.SubSection1.text1
        }
    }
    card „appendix Glossary“ as Book.glossary
    card „appendix Copyrights“ as Book.copyright
    card „appendix Notation“ as Book.notation
}

actor Owner
actor Department1
actor Department2
actor Department3
actor Department4
actor Contractor1
actor Contractor2

Owner <-> Book : Can access all
Department4 <-> Book : Can access all
Department1 <-> Book.Chapter1 : Can access Chapter and below
Contractor1 <-> Book.Chapter1.SubSection1 : Can access only the section
Department2 <-> Book.Chapter2 : Can access Chapter and below
Contractor2 <-> Book.Chapter2.SubSection1 : Can access only the section
Department3 <-> Book.Chapter1 : Can access Chapter and below
Department3 <-> Book.Chapter2 : Can access Chapter and below

@enduml
```

1.1.2. Example sub section

(This text is, outside of the sub section, in the chapter).

1.1.2.1. Example section

(and this line is in the sub section)

1.1.2.1.1. example sub section

With some test text here.

1.1.2.1.2. example dynamic includes section

Listing 2. exampleJSON.txt

```
{
    "glossary": {
        "title": "example glossary",
        "GlossDiv": {
            "title": "S",
            "GlossList": {
                "GlossEntry": {
                    "ID": "SGML",
                    "SortAs": "SGML",
                    "GlossTerm": "Standard Generalized Markup Language",
                    "Acronym": "SGML",
                    "Abbrev": "ISO 8879:1986",
                    "GlossDef": {
                        "para": "A meta-markup language, used to create markup
languages such as DocBook.",
                        "GlossSeeAlso": ["GML", "XML"]
                    },
                    "GlossSee": "markup"
                }
            }
        }
    }
}
```

Listing 3. exampleinclude.txt

```
This is an example raw text file
the file contains just text and could be a log file or output from some tests
a list of these can be compiled and automatically pulled in
It can also be source code or JSON or siilar and can then even be highlighted by
syntax.
```

Chapter 2. Description of logo

2.1. VSR logo description

VSR is an abbreviation for Virtual Space and Global Communications Research. The above is an imaginary company created by Sean Donnellan so as to facilitate progressing ideas without attachin them to a real company. The below is copyright Sean Donnellan as is Virtaul Space and Global Communications Research itself. Feel free to create another imaginary company and please use a different name for your invention.

The VSR logo was created from a 2d table of 8x8 with 8 data points with only one data point in each row and column. This then was made 3 dimentional and the points are further distributed to be only one in each row and column no matter from which side the 3d cube is viewed.

While it’s entirely possible to just distribute the points down a straight line through the cube they are distributed to make the 8 corners of a further cube like object within the outter cube.

The inner cube is skewed due to the constraints of the placements and it is this shape which is desired.

The overall idea behind this is that the inner cube when subtracted from the outer cube leaves a space within, virtual space, and it is defined by data points, nodes, in a multidimenional table, and connexions between the nodes, edges, that together represent a graph and an object. This represents the communication in Virtual Space and Global Communications Research.

The original logo also has an at symbol in the center which is not shown in the SCAD file.

The idea behind the @ was to make it clear that communications is being referred to but it is redundant because the inner cube/graph represents a network. The @ can ba added again to represent the applications that run across the network in virtual space.

When shown as a wire frame the inner cube should be hollow, and the inner space should be empty. When shown as a solid the inner cube can be assumed to be hollow if one so wishes.

When an @ is added to the logo it passes through both cubes and passes clearly through the inner space that would otherwise be hollow.

The use of the nuber 8 is a bow to the 8 bits that make a byte. The use of the 8x8 and 8x8x8 is desired. Also the use of the 8 data points that comprise the inner cube which is also a graph are desired.

When using only the inner cube for the logo the corners of the cube should be cubes so that the resulting cube has a complex shape and is not just a simple skewed cube but also has a clearly more complex geometry.

If using just the 8 corners of the inner cube for an animation these nodes should be cubes and no edges should be shown between them so that a complex dance of the cubes is visible as they rotate on one or more axis.

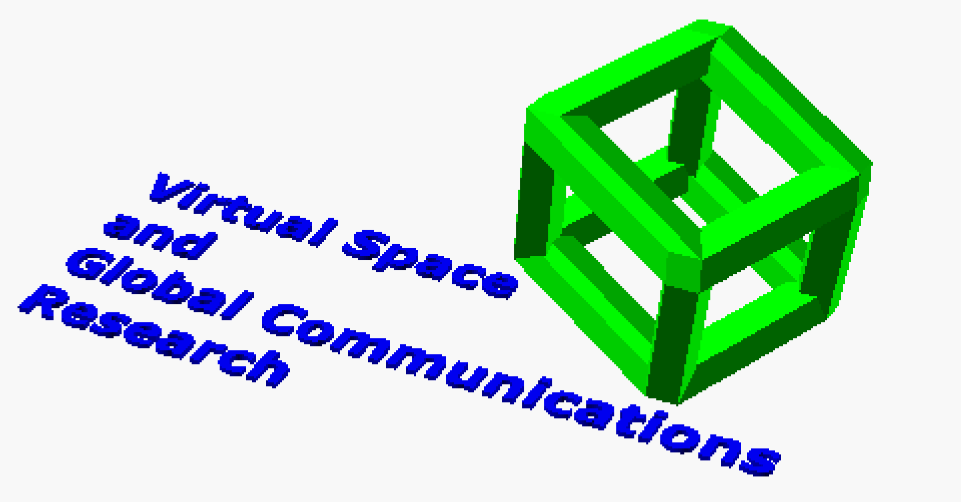

Figure 2. Example of inner logo with text as mix

The above logo was just created ad-hoc for this document and does not signify an actual company. Also of note is that the logo is really just ad-hoc. It is not a final product but an indication of an idea.

Listing 4. VSR logo scad file with diffreent options

```
//Script to create a wire mesh cube with 8x8x8 empty spaces
//Virtual Space and Global communications research department logo base object with 8.2 cm side length
//consisting of the multiplied basic primitives of an x,y,and z axis beam iterate in one dimension in loops
$fn=100;
module x_beam(){
   cube([82,1,1]);
}
module y_beam(){
   cube([1,82,1]);
}
module z_beam(){
   cube([1,1,82]);
}
module vsr_cube(){
   union(){
       //xbeam
       for (xj=[0:10:80]){
           for (xi=[0:10:80]){
               translate([0,xi,xj])
               x_beam();
           }
       }
       //ybeam
       for (yj=[0:10:80]){
           for (yi=[0:10:80]){
               translate([yi,0,yj])
               y_beam();
           }
       }
       //zbeam
       for (zj=[0:10:80]){
           for (zi=[0:10:80]){
               translate([zi,zj,0])
               z_beam();
           }
       }
   }
}
//basic primitive for VSR logo block.
color("#000",.25)
    vsr_cube();
// base cube with centre cube subtracted
*difference(){
   //basic primitive for VSR logo block.
   color("#000",.25)
        vsr_cube();
   //Whole to subtract
   hull() {
       translate ([30,50,70]) x_block();
       translate ([10,10,60]) x_block();
       translate ([70,40,50]) x_block();
       translate ([50,00,40]) x_block();
       translate ([20,70,30]) x_block();
       translate ([00,30,20]) x_block();
       translate ([60,60,10]) x_block();
       translate ([40,20,00]) x_block();
   }
}
//Just the Centre Block as a block
*hull() {
       translate ([30,50,70]) x_block();
       translate ([10,10,60]) x_block();
       translate ([70,40,50]) x_block();
       translate ([50,00,40]) x_block();
       translate ([20,70,30]) x_block();
       translate ([00,30,20]) x_block();
       translate ([60,60,10]) x_block();
       translate ([40,20,00]) x_block();
   }
module x_block(){
    // Cubes as corners for complex wire frame centre
    *color([0,1,0]) translate ([02,02,02]) cube([8,8,8]);
    //spheres as corners for smoothe wireframe centre
    translate ([6,6,6]) sphere (r=1);
```

```
}
//Just 8 centre points as corners
union () {
    //1st set
    hull() {
        translate ([30,50,70]) x_block();
        translate ([10,10,60]) x_block();
    }
    hull() {
        translate ([70,40,50]) x_block();
        translate ([50,00,40]) x_block();
    }
    hull() {
        translate ([20,70,30]) x_block();
        translate ([00,30,20]) x_block();
    }
    hull() {
        translate ([60,60,10]) x_block();
        translate ([40,20,00]) x_block();
    }
    //second set
    hull() {
        translate ([30,50,70]) x_block();
        translate ([20,70,30]) x_block();
    }
    hull() {
        translate ([10,10,60]) x_block();
        translate ([00,30,20]) x_block();
    }
    hull() {
        translate ([70,40,50]) x_block();
        translate ([60,60,10]) x_block();
    }
    hull() {
        translate ([50,00,40]) x_block();
        translate ([40,20,00]) x_block();
    }
    //third set
    hull() {
        translate ([30,50,70]) x_block();
        translate ([70,40,50]) x_block();
    }
    hull() {
        translate ([10,10,60]) x_block();
        translate ([50,00,40]) x_block();
    }
    hull() {
        translate ([20,70,30]) x_block();
        translate ([60,60,10]) x_block();
    }
    hull() {
        translate ([00,30,20]) x_block();
        translate ([40,20,00]) x_block();
    }
}
```

Chapter 3. Example external links to other material

github link to an jupyter notebook of an example 3d Object

https://github.com/donnels/3d-objects/blob/main/spoolholder.ipynb

The above is an example of dockerizing Juypter while adding an opensaced kernel to be able to describe the story of how an object was created. The ipynb is an extension assigned to jupyter notebooks and the kernel for openscad was added for a raspery Pi by Sean Donnellan so as to give jypter a home there in the VSR network. Normally Jupyter is used with Python or R for example for data analysis while mixing code and documentation. Output can also be PDF. When openen in the proper Jupyter environment it is interactive.

The object created for printing

https://github.com/donnels/3d-objects/blob/main/spool.stl

A link to some ARM software for IOT created before the Raspberry Pi

https://www.auto.tuwien.ac.at/~mkoegler/index.php/eibnslu

The above was created by Sean because he saw the potential of the Arm platform for IoT and began exploring it before the Raspberry Pi existed and before the Iphone existed.

Appendix A: Glossary

ACI

Cisco Spine Leaf Ethernet infrastructure

API

Interface to a program or system

APIC

management console for Cisco ACI

ARP

Address Resolution Protocol - Used primarily on IP networks to dynamically link MAC addresses to IP addresses

BGP

Border Gateway Protocol - Path-Vector Routing Protocol

CIO

Chief Information Officer - At VSR the team responsible for the, primarily green, VSR internal network

DB

Data Base

DCI

Data centre interconnect. A switch that connects multiple DCs

DLSW

Data Link Switching - Normally DLSW+ as DLSW is deprecated - Is used to extend non routed networks across IP networks (example: Enterprise Extender)

DNS

Domain Name Service - resolve hostnames to IP addresses

DR

Disaster Recovery

ECMP

Equal Cost Multi Path - A method used, amongst others, by spine leaf architectures in which all the links to all the spines are used at the same time thereby increasing the overall bandwidth

Edge

Is the connecting link between two Vertices (See Vertex). Also referred to as relationship

Enterprise Extender

A network device used to interconnect non IP communication to Mainframes across IP networks

FICON

A form of cabling used primarily for the Mainframe

GI

Global Infrastructure - The VSR Internal Networks owned by the VSR CIO

GRE

Generic Routing Encapsulation (GRE) is a tunneling protocol developed by Cisco Systems that can encapsulate a wide variety of network layer protocols inside virtual point-to-point links or point-to-multipoint links over an Internet Protocol network

Graph

A mathematical structure used to model pairwise relations between objects (see also Vertex and Edge)

HA

High Availability

HMC

Hardware Management Console - Used to manage IBM P and Z series hardware

ICAP

Intellectual Capital

IPSEC

Internet Protocol Security (IPsec) is a secure network protocol suite that authenticates and encrypts the packets of data to provide secure encrypted communication between two computers over an Internet Protocol network. It is used in virtual private networks (VPNs).

LAN

Local Area Network - Generally a layer 2 ethernet Environment

LDAP

Lightweight Directory Access Protocol - Method often used to access credential databases

LISP

Locator/Identifier Separation Protocol - A protocol that allows L3 mobility to be implemented

- commonly used by AT&T to migrate networks - Major limitations include inability to suport layer 2 and also problems with overlap and segments with multiple subnets and load balancers amongst others

LPM

Live Partition mobility - IBM technology to move Virtual machines between systems without interruption

MAC

Media Access Control - Usually a layer 2 Ethernet Address linked to an ethernet NIC

MF

MainFrame

MPLS

Multi Protocol Label Switching - Transport technology used primarily by WAN carriers

NAT

Network Address Translation

NFV

Network Function Virtualisation - Virtual Firewalls, routers, switches etc. - Often within a device that thereby then combines multiple modular functions

NIC

Network Interface Card

Neo4J

A particular graph DB

Node

See Vertex

OSA

An Ethernet NIC used in the Mainframe

OSPF

Open Shortest Path First - Routing Protocol for dynamic routing (often required in conjunction with Mainframes and VIPA)

OTV

Overlay transport virtualization (OTV) is a Cisco proprietary protocol for relaying layer 2 communications between layer 3 computer networks

PAT

Port Address translation - a form of NAT

PBR

Policy based Routing - Normal routing is destination based and policy based routing may involve other things like source or type

Python

A programming language

QSFP

Quad Small Form Pluggable - Removable network interface for switches and devices

Relationship

See Edge

SDDC

Software Defined Data Centre - A cloud like DC that is centrally managed

SDWAN

Software Defined WAN - A general Term for a centrally controlled and configured WAN - Usually with mixed Internet and MPLS instead of redundant MPLS

SFP

Small Form Pluggable - Removable network interface for switches and devices

SQL

Sequel Query Language - Database API

Spine Leaf

Network Architecture that connects leaf switches with access ports to central switches called spines. The spin switches are not connected to anything but leafs.

VIPA

Mainframe redundancy mechanism that involves dynamic routing of host routes.

VLAN

Virtual LAN - VLANS are used to create separate layer 2 domains within a switch or network environment - Used primarily for customer separation.

VMotion

VMware technology to move Virtual machines between systems without interruption

VPN

A virtual private network (VPN) extends a private network across a public network and enables users to send and receive data across shared or public networks as if their computing devices were directly connected to the private network

VRF

Virtual Routing Facility - essentially a router within a router used for routing separating between customers etc.

Vertex

Is a fundamental unit of graphs, usually labeled and denoted by a circle. Also referred to as a node

WAN

Wide Area Network - The network between remote sites

YAML

YAML (a recursive acronym for "YAML Ain't Markup Language") is a human-readable data-serialisation language

YANG

YANG (Yet Another Next Generation) is a data modelling language for the definition of data sent over network management protocols such as the NETCONF and RESTCONF

Appendix B: Notation

This section contains examples of the notation used in the overall document.

B.1. Warnings and Callouts

The style can be any one of the admonition labels:

ℹ	Note Benne
💡	This is a tip
⚠	This is a warning.
🔥	let me **caution** you.
❗	This is important.

B.2. Sidebars

Related information

This is aside text.

It is used to present information related to the main content.

B.3. Lists

A numbered list of items

1. spoon
2. fork
    a. spork
        i. spock
            I. star treck

A bulletted list

- spoon
- fork

        - spork
            - spock
                - Star treck

A TODO list

☑ Learn the AsciiDoc syntax

☑ Install AsciiDoc

☑ Write my document in AsciiDoc

B.4. Links, Quotes and Listings

A typical Link

https://asciidoctor.org

A Quote

> Don’t Panic (Yet)
>
> — Douglas Adams

Listing 5. A Listing

```
ssh -J sean@9.154.199.101 sean@192.168.2.2
```

VSR

VSR, the VSR logo are trademarks of Virtual Space and Global Communications Research, registered in no jurisdictions worldwide. Other product and service names might be trademarks of VSR or other companies. A current list of VSR trademarks is available on the Web at “Copyright and trademark information” at www.vsr.local.

This document is current as of the initial date of publication and may be changed by VSR at any time. Not all offerings are available in every country in which VSR operates.

THE INFORMATION IN THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON INFRINGEMENT. VSR products are warranted according to the terms and conditions of the agreements under which they are provided.