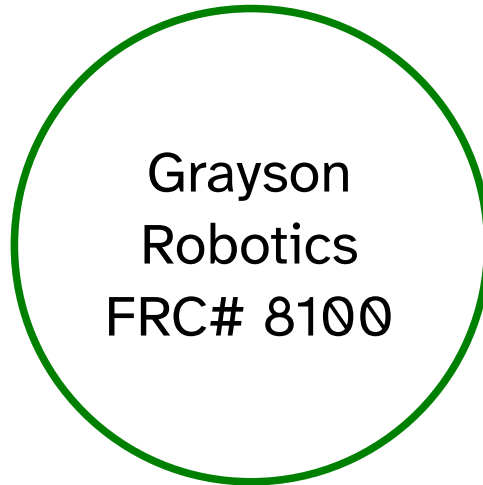


# ORG-1

## Document Control System

2025-04-26



*So let it be written. So let it be Done.*

# Contents

Document Control .....	4
Authors .....	4
Purpose .....	4
Scope .....	4
Terms .....	4
1 Document Control System .....	6
1.1 Controlled Management Documents .....	6
1.2 Controlled Working Files .....	6
1.3 Common Requirements .....	7
1.4 DCS Repositories .....	7
1.4.1 Numbered Document Database (NDD) .....	7
1.4.2 Common DCSR requirements .....	7
1.4.3 Requirements for numbered documents .....	9
1.4.3.1 Standard control information .....	9
1.4.3.2 Document Codes .....	9
1.4.4 Edit & Pull Request process .....	10
1.5 ORG DCSR .....	10
1.5.1 ORG directory structure .....	10
1.6 SDR DCSR .....	11
1.6.1 SDR directory structure .....	12
1.7 TRM DCSR .....	12
1.7.1 TRM directory structure .....	12
1.8 CAR DCSR .....	12
1.8.1 CAR directory structure .....	12
1.9 PRJ DCSR .....	12
1.9.1 PRJ directory Structure .....	12
1.10 Action Item DCSR .....	13
1.10.1 Action Item Titles .....	13
1.10.2 Directory Structure .....	13
2 PRJ documents .....	14
2.1 PRJ document update process .....	15
2.2 Including other files .....	15
3 Action Item documents .....	15
4 SDR documents .....	15

4.1	SDR document structure .....	15
4.1.1	SDR document section usage .....	16
4.1.2	SDR document revision section .....	16
4.1.3	SDR document source signature .....	17
4.2	SDR Approval .....	17
5	ORG documents .....	17
6	Corrective Action Reports .....	17
7	TRM .....	18
7.1	TRM document structure .....	18
7.1.1	TRM disclaimer .....	18
7.2	TRM Approval .....	19
	Revision History .....	19
	V0 - yyyy-mm-dd .....	19
	Source File Signature .....	19

# Document Control

Document Type: ORG

Document #: 1

Status: DRAFT

Version: N/A

Effective Date: 2025-05-03

Source Repository: <https://github.com/>

## Authors

Joshua T. - [joshua.turner@frcteamreboot.com](mailto:joshua.turner@frcteamreboot.com)

## Purpose

- Define requirements for the Document Control System (DCS).
- Distinguish working files from organizing documents.
- Define procedures for documents subject to the DCS:
  - ORGs
  - SDRs
  - TRMs
  - CARs
  - Project Planning Documents (PPD).
  - CWF

## Scope

Sub-Teams: All

Roles: All

## Terms

APA American Psychological Association

CAR Corrective Action Report

CMD Controlled Management Documents

CWF Controlled Working Files

DCS Document Control System

Directory In computing: a location on a hard drive where a files may be stored. Also know as a "folder".

Git The Git version control system

GitHub A website for managing Git repositories

IAW In accordance with

JSA Job Safety Analysis

ORG Organizational Document type code

PPD Project Planning Document: either a Project Document (PRJ) or Action Item

PRJ Project Document type code

SHA3-256 The 256 bit variant of the "Secure Hash Algorithm" SHA-3

# 1 Document Control System

Certain types of documents - owing to their importance - require a Document Control System (DCS). The purpose of a DCS is to ensure documents are:

- correct
- easy to locate
- versioned with a transparent history

Control procedures also ensure that team members are know that the document they are referencing is the most up to date version.

In the context of FRC, the concerns that necessitate building a DCS are as follows:

- Find, correct, and prevent future process flaws.
- Manage a complex interdisciplinary project with limited man-hours.
  - Make it easier for mentors to help remotely.
  - Allow all members to work asynchronously.
  - Clearly define roles and expectations between sub-teams.
- Preserve rarely used, but important, processes and techniques.
- Combat the effects of student turnover.

## 1.1 Controlled Management Documents

Standards (SDR), Training Manuals (TRM), Corrective Action Reports (CAR), Project Documents (PRJ), and Action Items together constitute Controlled Management Documents (CMD). CMD are distinct because all team activities are defined, planned, and managed through these documents. All team members will interact with and author some form of CMD throughout normal operations. Grayson Robotics shall maintain CMD pursuant to the requirements outlined in this standard.

## 1.2 Controlled Working Files

For these types of files, each sub-team shall maintain its own internal Policy Documents covering proper handling and version control.

These policies shall adhere to the following requirements:

1. Provide a means of versioning files when they are of a type that may be revised.
2. Define a location and procedure for storing the files electronically.

3. Define when and how to notify relevant parties of changes.
4. TODO: what else?

### 1.3 Common Requirements

For documents written by Grayson Robotics:

1. All documents shall use the Atkinson Hyperlegible Next font when exported or rendered from their source format.
  1. Source code is excluded from this requirement.
2. Authors shall format dates in the ISO 8601 format: yyyy-mm-dd.
3. All documents shall adhere to the Concise Guide to APA Style, seventh Edition, ISBN: 978-1-4338-3273-4, sections 4 and 5.

### 1.4 DCS Repositories

DCS repositories (DCSR) are publicly accessible stores of CMD, authored by Grayson Robotics team members.

#### 1.4.1 Numbered Document Database (NDD)

Numbered documents - ORGs, SDRs, TRMs, CARs, & PRJs - each require a database to keep track of the index. The file shall be a Tab Separated Value (TSV) spreadsheet file encoded in utf-8. The file shall have the following columns:

- document number
- document title
- file name with extension
- document status
- version number
- file name
- date first issued
- date of last change
- SHA3-256 hash of the source file

#### 1.4.2 Common DCSR requirements

1. Team members shall publicly review new documents.
2. For each type of CMD, Grayson Robotics shall maintain a public GitHub repository and control procedure.
3. The IT Dept. shall administer control procedures.

4. The Programming Dept. shall, at the request of the IT Dept., assist with creating any software utilities used to aid in administering DCSR.
  1. The Programming and IT Departments shall write all helper scripts and utilities, stored in a DCSR, in the Nushell language.
  2. The Programming and IT Departments shall ensure that all scripts run correctly on Windows and Linux.
5. At least one teacher at Grayson high school shall own - or be a co-owner of - all GitHub repositories.
6. Team members shall obtain unanimous approval from the relevant sub-teams prior to updating a document.
7. The repository shall be impossible to physically update without at least 2 team members intending to do so.
8. The IT team shall notify all team members of any change to the form, structure, implementation, or location of the repository that goes beyond updating a single controlled document prior to said change.
9. Team members shall create all data in formats that are viewable and editable via free software.
10. Team members shall author all CMD in a text format rather than a binary format.
  1. The intent is to keep content viewable via the built-in rendering feature of GitHub or similar services. For example:
    1. Markdown
    2. HTML
    3. Typst
  2. A controlled document, written in one of the above formats, shall reference content in other formats when that content isn't embedded.
  3. For Typst sources, after the file is approved and released, the author shall export the file to PDF and store it in the corresponding release directory.
11. Authors shall name all files free of spaces
  1. Authors shall use underscores in lieu of spaces
12. Authors shall chose titles that are unique from all other documents of the same type.
13. The IT team shall maintain directory structure and files for each DCSR.



### 1.4.3 Requirements for numbered documents

1. Authors shall number documents sequentially, in the order they're created.
2. A document number shall be unique from all other documents of the same type.

#### 1.4.3.1 Standard control information

Where applicable, as defined later in this document, authors shall include the following control information:

- Document Type: ORG, TRN, etc.
- Document Number
- Status
  - Issued: Approved and in use.
  - Draft: New or update suitable for review.
  - Depreciated: Previously issued but no longer for active use.
- Version Number
- Effective Date
- Source Repository (A link to its DCSR).

#### 1.4.3.2 Document Codes

Document Codes uniquely identify all numbered documents. The document code is the combination of the Type Code and Document Number with a single hyphen in between. For example: "ORG-1" is the Document Code for this document.

Document codes are constructed as follows:

- Three letter Type Code:
  - ORG: Organizational Standard
  - SDR: Standard
  - TRM: Training Manual
  - CAR: Corrective Action Report
  - PRJ: Project Document
- Hyphen character: "-"
- Document Number:
  - Sequential integer number.
  - Unique between documents of the same type.

All team members shall write, type, or otherwise represent document codes with all letters capitalized.

#### 1.4.4 Edit & Pull Request process

Authors shall use GitHub Pull Requests (PR) to submit documents and updates to each DCSR:

1. Fork the GitHub repository
  1. Clone the fork to your computer locally
2. Make edits
  1. change and add content
  2. check structure against this standard and ORG-0.
  3. update hash file
  4. update NDD
3. Commit
  1. Describe changes in message
4. push to fork
5. Submit PR
  1. reference the PPD
6. Discuss and edit using GitHub's PR comment system

At the discretion of the IT Dept., the Author or the IT Dept. shall "squash" and Rebase commits on the fork prior to merging the PR.

### 1.5 ORG DCSR

1. The GitHub repository is named: ORG-DCS.
2. The URL is: TBD
3. Team members shall author documents using the given template: draft/template.typ shown in Section 1.5.1.
4. The IT Dept. shall ensure that the template automatically sets up the document IAW: Section 4.1.

#### 1.5.1 ORG directory structure

All file names are case sensitive.

The IT Dept. shall maintain the structure of the root directory of the repository as follows:

- Team logo: 8100-logo.svg

- General instructions file: `README.md`
- CC0 license for all documents: `LICENSE`
- NND spreadsheet: `ORG-NDD.tsv`
- Helper script to compile Typst source files: `compile.nu`
- Helper script to generate a hash of the template: `hash-template.nu`
- Helper script to create a new document draft: `init-document.nu`
- Gitignore file (used to tell Git not to track certain files): `.gitignore`
- Git directory (Directory used by the Git software. Not to be edited directly by team members): `.git/`
- Directory containing document sources, example, and template: `draft/`

The `draft/` directory contains the working versions of ORG documents in their source format. The IT Dept. shall maintain the structure of the `draft/` directory as follows:

- Template for ORG documents: `template.typ`
- Hash of template file: `template.sha256.txt`
- Example document directory: `example/`
  - Example Typst document source file: `ORG-X_Example.typ`
  - Hash of `ORG-X_Example.typ`: `sha256.txt`
- Document directories (one per document). For example: `ORG-999_some_title/`
  - Typst source file for the document with the name of the directory: `ORG-999_some_title.typ`
  - The draft of document, exported to PDF format. For example: `ORG-999_some_title.pdf`
  - Hash of the Typst source file in this directory: `sha256.txt`
  - Any other files the author needs such as data tables or images.

The released directory contains the approved & released versions of all ORG documents in PDF format.

## 1.6 SDR DCSR

1. The GitHub repository is named: SDR-DCS.
2. The URL is: TBD

### 1.6.1 SDR directory structure

Same as ORG DCSR but every instance of the “ORG” document type code shall appear as the “SDR” code.

## 1.7 TRM DCSR

1. The GitHub repository is named: TRM-DCS.
2. The URL is: TBD

### 1.7.1 TRM directory structure

Same as ORG DCSR but every instance of the “ORG” document type code shall appear as the “TRM” code.

## 1.8 CAR DCSR

1. The GitHub repository is named: CAR-DCS.
2. The URL is: TBD

### 1.8.1 CAR directory structure

Same as ORG DCSR but every instance of the “ORG” document type code shall appear as the “CAR” code.

## 1.9 PRJ DCSR

1. The GitHub repository is named: PRJ-DCS.
2. The URL is: TBD

### 1.9.1 PRJ directory Structure

All file names are case sensitive.

The IT Dept. shall maintain the structure of the root directory of the repository as follows:

- Team logo: 8100- logo.svg
- General instructions file: README.md
- CC0 license for all documents: LICENSE
- For each year, a directory containing all projects created that year. For example: 2025/

For each year, authors shall place new PRJ documents in a directory with the four-digit representation of the year as it's name. The IT Dept. shall maintain the structure of each yearly directory as follows:

- Document directories, one per document. For example:  
PRJ-999\_some\_title/
  - Markdown source file document with the name of the directory:  
PRJ-999\_some\_title.md
  - Any other files the author needs such as data tables, notes, or images.

## 1.10 Action Item DCSR

1. The GitHub repository is named: ACT-DCS.
2. The URL is: TBD

### 1.10.1 Action Item Titles

Authors shall name Action Item files as follows:

1. the date in ISO 8601 format: yyyy-mm-dd
2. underscore character “\_”
3. the title, using underscores in lieu of spaces
4. the files extension “.md”

For example: “2025-04-26\_Recycle\_old\_batteries.md”

1. Authors shall chose a title that is unique from all other Project documents.
2. Authors shall write project documents in Markdown.

### 1.10.2 Directory Structure

All file names are case sensitive.

The IT Dept. shall maintain the structure of the root directory of the repository as follows:

- Team logo: 8100-logo.svg
- General instructions file: README.md
- CC0 license for all documents: LICENSE
- For each year, a directory containing all Action Items created that year. For example: 2025/

For each year, authors shall place new Action items in a directory with the four-digit representation of the year as it’s name. The IT Dept. shall maintain the structure of each yearly directory as follows:

- Markdown source files for documents. For example:  
2025-04-26\_Recycle\_old\_batteries.md

## 2 PRJ documents

Authors shall write project documents in the Markdown format.

Project Documents shall have the number and title as part of the file name in the following format:

1. The letters “PRJ” to denote the document type as a Project Document.
2. A dash character: “-”
3. The document number
4. An underscore character: “\_”
5. The document title with any spaces replaced with underscores
6. The file extension: “.md”

Given an example PRJ document number five (5) with the title “Battery Recycling”: “PRJ-5\_Battery\_Recycling.md”

Type	Dash	Number	Underscore	Title	Extension
PRJ	-	5	_	Battery_Recycling	.md

The text of a PRJ document shall, at minimum, contain the following information in this order:

1. Document number
2. Title
3. The date created
4. List members proposing the project
5. List concerned working groups and roles
6. List measurable project objectives
7. List members working on the project
8. Outline a plan for achieving objectives
9. Outline funding requirements
10. Outline material & parts requirements

Sections specific to the project, shall come after these in the Project Document.

The PRJ document shall have a change log with the following information recorded for each update placed at the end of the document (after the project specific sections):

1. Change date

2. Change author
3. Change summary
4. New status (Draft, Open, Closed)

Authors shall place the change log at the end of the PRJ document.

## 2.1 PRJ document update process

1. Discuss the update with other members of the project to obtain unanimous consent for the update.
2. One of the members of the project will follow the PR process: Section 1.4.4 to update the document.
3. At least one member of the IT Dept. shall check the PR meets the requirements of this standard.
4. IT Dept. or another project member shall accept the PR.

## 2.2 Including other files

## 3 Action Item documents

Authors shall write Action Item documents in the Markdown format.

The text of an Action Item shall, at minimum, contain the following information in this order:

1. Title
2. The date created
3. List members proposing the project
4. List concerned working groups and roles
5. List measurable objectives
6. List members working on the project
7. Outline material & parts requirements

## 4 SDR documents

1. Authors shall write SDR documents in the Typst format.
2. Authors shall export SDR documents in PDF format after drafting edits or releasing them.

### 4.1 SDR document structure

Authors shall start ORG documents with the following in this order:

1. title page with:
  1. document type code: SDR, ORG, or TRM, a hyphen, and the document number (e.g. ORG-1)
  2. document title
  3. date
  4. team logo
  5. team motto: "So let it be written. So let it be done."
2. table of contents
3. "Document Control" section with all information listed in Section 1.4.3.1.
4. "Authors" section with, at least, first and last initials, and up to full name and email address.
5. "Purpose" section
6. "Scope" section
7. "Terms" section

#### 4.1.1 SDR document section usage

1. The author shall use the Purpose section to describe what they intend to accomplish by writing the document.
2. The author shall use the Scope section to show what sub-teams and roles are affected by the content.
3. The author shall list all acronyms and domain specific terms in the Terms section.
4. After the Terms section, the author shall write the content of the Policy.
5. Finally, after the content sections, all Policies shall end with the following in this order:
  1. Revision history section
  2. Source file signature section

#### 4.1.2 SDR document revision section

The revision history section is a list where each item starts with a second level heading. The heading for each list item takes has the form:

1. the letter "V"
2. the version number
3. a dash surrounded by spaces " - "
4. the date the revision was approved.



For example: V2 - 2025-04-18. After the heading, the author shall write a summary of changes.

#### 4.1.3 SDR document source signature

The source file signature section shall contain the SHA3-256 hash of the source document and the template. This is automatically done so long as the directory structure defined by: Section 1.5.1 is maintained, and the author uses the `compile.nu` script.

### 4.2 SDR Approval

1. Find one other member on the relevant sub-team that agrees that a new document or update is needed.
2. Create Action Item and write goals for new document.
3. Get approval for update from rest of affected sub-teams.
4. Create PRJ for new documents. Action Items are sufficient for updates.
5. Author and IT Dept. shall use the PR process in: Section 1.4.4.
6. Project members shall review updates on fork.
7. Author shall sync the fork with the original and incorporate any updates.
8. Run compile script against new document for both the draft.
9. Author shall submit the PR to the DCSR.
10. All affected members shall comment approval or disapproval in the PR request.
11. If the document is approved, the IT Dept. shall accept the PR
12. IT Dept. shall build the document using the compile script in release mode.

## 5 ORG documents

ORG document have the same structure and follow the same procedure as SDRs, as defined by: Section 4, but the entirety of Grayson Robotics must approve any change to an ORG document.

## 6 Corrective Action Reports

CARs have the same structure and contain control information as SDRs (as per: Section 1.4.3.1) with the exception that the status may be either Open or Closed:

- Use *Open* after the document has been created and before the CAR is closed.
- Use *Closed* only after the CAR is closed.

In addition, CAR documents contain the following sections: . Incident Description . Incident Statements: Statements from individuals with knowledge of the incident or potential causes. . Supporting Content: Pictures, diagrams, code, etc. . List of Causes . Resolution Actions: List of PPDs opened to address the subject of the CAR. . Progress Tracking: Track the progress of any related PPD.

## 7 TRM

1. Authors shall write and export TRM documents in the same formats as SDRs as defined by: Section 4.

### 7.1 TRM document structure

1. Authors shall write TRM documents with the same structure as SDRs, as defined by: Section 4.1,

but with additions.

1. Disclaimer statement: see Section 7.1.1
2. Job Safety Analysis (JSA)
3. Intended Outcomes: Whether it takes the form of a tutorial or instructions for a teacher lead class,

summarize the intended outcome for the learner upon completing the training outlined by the document.

1. Training Scope: Detail explicitly what the training is to cover, and note any similar or related topics that

the training is not intended to cover.

#### 7.1.1 TRM disclaimer

Authors shall include the following statement in all TRM documents:

The author(s), maintainer(s), and distributor(s) of this document shall not be liable for any damages, losses, injury, costs or expenses, direct, indirect or

incidental, consequential or special, arising out of, or related to the use of information within this document, or any material it references.

## 7.2 TRM Approval

TRM documents shall use the same approval process as SDRs, as defined by: Section 4.2.

## Revision History

V0 - yyyy-mm-dd

- initial release

## Source File Signature

ORG-1 SHA256:

46c07f6e3adc567959f9a927e9b62230ffec9296d5f6ec19c4bb30721c891a6b

Template SHA256:

3590e22cd456c913b0545813703ed668770cbd00f83d661bf037af14e681e35a