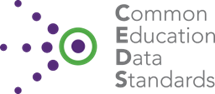
STATUS NOTE: This is nearly-finished draft. If you have any suggestions (things to be added, moved, revised, or removed altogether), please add a comment (Ctrl+Alt+M) in the appropriate location.

NOTE: Colored/highlighted text indicates placeholder text and/or instructions -- areas where information needs to be added/revised later on.

This status note will be updated as we progress through the revision process to indicate next steps. It will be DELETED when we do the final polish of this document.



**Best Practices for All Documentation**

Recommendations from the Scalable Data Use Workgroup  
On Standards for Representations, Reports and/or Dashboards

Created August 2024

As part of the FY23 SLDS Scalable Data Use   
supplemental grant for the CEDS Collaborative Exchange

<https://github.com/CEDStandards>

At one of our meetings, we should discuss if we want an Creative Commons license to include on this document, and if so, which one: <https://chooser-beta.creativecommons.org/>

# **Table of Contents**

[Table of Contents 2](#_piqph78q8mp5)

[Introduction 3](#_wyuzamh2qjai)

[What content to include in all documentation 4](#_yvfry994e6n2)

[Where did this documentation come from?  
Authorship, Ownership, Versioning & Copyright 4](#_2yxsqkko49d)

[Why does this documentation exist?  
Purpose, Audience, Assumptions & Introductions 5](#_h8kixjq4euch)

[How will this impact me/my work?  
The Importance of Vivid Examples and Calling Out Changes 6](#_xs1o4v2c9dfn)

[Structuring and styling the document for ease of use 7](#_ep2yxz2vrnoy)

[Navigation Tools & Document Structure 7](#_dt7l76x6jxqx)

[Language & Grammar 8](#_14zz9eqar6sv)

[Tables & Spreadsheets 9](#_ln497za3b30k)

[Visuals, Graphics, Images, Diagrams, Figures, etc. 10](#_ds131fywta4p)

[File format 12](#_u2s69t2h3e2h)

[Doing the right thing 13](#_x2yxo3v0t9kl)

[Privacy & Security 13](#_1cneiirs22ef)

[Accessibility 14](#_dl9mu0y39s4i)

[Citation of outside sources & Respecting copyright 15](#_jc8x4n78nma6)

[Best Practices specific to sharing on GitHub   
(and within the Open Source Community) 16](#_1fig6nj2v7k3)

[ReadMe 16](#_u38jbn7zmjtx)

[Folder & File Structure on GitHub 18](#_fr7a0xekewzj)

☑️ [Link to Best Practices Checklist in Google Sheets](https://docs.google.com/spreadsheets/u/0/d/1js61mb6Nq__IzdrgJWAHUMttqf97OSYV00w6SV8gRTM/edit)

(A companion resource to this document)

# **Introduction**

We assume that you have been tasked with writing documentation about data -- perhaps about the data itself, perhaps about a data process or data product. We also assume that you want your documentation to be used. We state this assumption at the outset because not all documentation is intended for frequent use -- some is just to check off a legal requirement. We assume you are writing documentation that your coworkers and colleagues across the country will need to reference with some frequency.

As consumers of documentation, we all prefer documentation that is not only correct but also easy to use and easy to understand.

This means that we -- as creators of documentation -- also have a responsibility to make our own documentation easy to use and easy to understand.

“Best Practices for All Documentation” was created through the collaborative work of Michigan’s Center for Educational Performance and Information, South Dakota’s Board of Regents and Department of Education, Kansas’ State Department of Education, Virginia’s Department of Education, and Illinois’ State Board of Education as part of the FY23 SLDS Scalable Data Use supplemental grant.

We are sharing the ways that we strive to create documentation that is user-friendly. These best practices may be used by anyone having to work with data, but our particular audience are state agencies that work with educational data and may be interested in working with the [Open Source Community](https://github.com/CEDStandards) for [Common Education Data Standards](https://ceds.ed.gov/default.aspx).

This document is called “Best Practices for All Documentation” because our workgroup has companion documents that address how to document data architecture and UAT testing. This document serves as a foundation upon which other, more specialized documents can be built.

Please note that these are *our* best practices. We have labeled some practices as “‘Essential for all documents’’ and some as “Suggested for many documents” but ultimately these are all just recommendations. Revising documentation can be challenging, and we have found that having a list of best practices by our side is helpful. It becomes a checklist of what we can do to make our documentation as user-friendly as possible. (In fact, if you would like to have a literal checklist of these recommendations, we’ve made [a shareable copy in Google Sheets.](https://docs.google.com/spreadsheets/d/1js61mb6Nq__IzdrgJWAHUMttqf97OSYV00w6SV8gRTM/edit?usp=sharing))

Finally, know that this document is not intended to be the final word on the subject but an invitation to help improve documentation across the board for the common good. We invite you to get involved in the [Open Source Community](https://github.com/CEDStandards) to help grow and improve the [Common Education Data Standards](https://ceds.ed.gov/default.aspx) and its related documentation -- including what best practices you think we should include in the next version of this document!.

# **What content to include in all documentation**

One of the joys of electronic documentation is that it is easily shared. One of the pitfalls is that it can be shared in ways that obscure the context in which it was created. It does take a little time to ensure essential contextual details are included in documentation, but future readers and consumers of the documentation will thank you!

We suggest that all documentation should include:

* Where the documentation came from
* Why the documentation exists
* Vivid examples of who or what will be impacted by the content of the documentation

## **Where did this documentation come from?** Authorship, Ownership, Versioning & Copyright

Who wrote this and when? As readers, we need this information to judge whether documentation is applicable to our own situations. Regardless of whether the author or owner is an individual or an organization, it’s vital information for readers who may have questions about the document and its permitted uses.

### Essential for all documents

**This information should be easy to find.** In a longer document, it could be placed on the title page or on the opening pages (introduction, copyright page, etc.). In short documents, it could be added to the header or footer.

* **Authorship** - Who initially created this document? Credit for authorship should always be given.
* **Ownership** -- Who currently maintains and/or disseminates this document?
* **Versioning** -- When was this document created? When was it last updated? These questions are essential in a world where data and processes are constantly changing.

### Suggested for all documents

* **Contact information** for the current owner (website or email address)
* **Copyright information** -- We suggest using a [Creative Commons license](https://creativecommons.org/) to make it clear how the documentation may be reused and repurposed by others.

## **Why does this documentation exist?** Purpose, Audience, Assumptions & Introductions

Everyone loves saving time! Instead of forcing people to slog through pages and pages of documentation before they realize that it is not applicable to them, give them a gift. State the purpose, intended audience, and assumptions about that audience at the beginning of the documentation.

### Essential for all documents

* **Who is the intended audience?**   
  Ex: Business analysts? Database architects? Researchers? End users? ???
* **How is this documentation intended to be used?**   
  Or - in other words - why was it written?

### Suggested for many documents

* **Assumptions** -- What assumptions have been made about the audience? (ex: Is this documentation intended for experts, novices, or somewhere inbetween? Will it require understanding the jargon of a particular discipline?)
* **Prerequisites** -- What tools, software, or prior steps are needed in order to successfully use this document / product / process?
* **Cautions & Warnings** -- How should this document *not* be used? Which parts are useful but come with limitations or caveats?
* **Background** -- What led to the creation of this document?
* **Context** -- How does this document fit into a larger series of documents or within a larger project? (Ex: Does it follow after a particular document? Should the reader consult other documents in tandem with this one? Is it a preliminary report and the reader may want to seek out the final report elsewhere? )

## **How will this impact me/my work?** The Importance of Vivid Examples and Calling Out Changes

Examples, examples, examples! Most documentation is improved when supported by carefully selected examples. [Human brains are wired to learn and retain information better when it is shared through stories and examples](https://www.harvardbusiness.org/what-makes-storytelling-so-effective-for-learning/).

### Essential for all documents

* **Call out changes -- in data fields, processes, etc. -- and describe them** in enough detail so that users can understand how those changes might impact their work. For example, if a data point changed from holding values 0-4 to holding values 5-9, it should be documented what the new values mean as well as if they align with any of the previous values (i.e. Value 0 was changed to Value 5). Help the user understand what the data is now as well as have some insight into what the data was like before.

### Suggested for many documents

* **Templates should be accompanied by an example of a completed template** to provide users with a sense of how much explanation or detail needs to be included in different sections of the template.
* **Data dictionaries should include examples of expected values** in addition to element name, definition, data type, and data length. Adding examples of expected values helps users understand the data better than just saying INT64 or “Alphanumeric - length 60.” Include unusual (but correct) values for each field to help demonstrate the range or variety the user can expect -- for instance, one state had districts enter “VACANT” for first name if a position was unfilled. Many states have families with a surname of “Null” which should not be confused with being a null value!
* **Process documents should include examples, case studies or sample scenarios of how it looks to implement the process** -- especially of how to adapt when there are obstacles or difficulties. Rarely does any project go according to plan 100% of the time. Including examples of how a past team overcame common obstacles or difficulties helps future teams overcome those same areas more quickly.
* **Documentation about visuals, dashboards, etc. should include visual examples.** Documenting that a particular type of dashboard should always use a Vertical 100% Stacked Columns is useful, but also showing some sort of example of what a Vertical 100% Stacked Column chart is more helpful for the person who didn’t learn the names of the types of charts and just know them by what they look like.

# Structuring and styling the document for ease of use

Everyone has experienced the agony of trying to read documentation that contains valuable information but the structure or style is impenetrable - confusing, convoluted, or just plain awkward. Let’s break the cycle by agreeing to take the time needed to make our documentation user-friendly.

## **Navigation Tools & Document Structure**

Using navigation tools and a clear document structure will allow users to easily skip over irrelevant sections and quickly access the information they need. Being able to click on a hyperlink in the table of contents to go directly to a section (or CTRL+F to find a particular section header) results in significant time savings over just scrolling, scrolling, scrolling to find the right section.

### Essential for all documents

* **Descriptive Headings** -- A good descriptive heading in documentation is brief but also crystal **clear about what information is contained within the section (or worksheet).** Readers encountering a section titled “Section 5.10” in a text document or a worksheet tab labeled as “Sheet 3” do not glean any useful information from those headings.
* **Hyperlinks** -- Add hyperlinks to text when referencing other sections within the documentation or online resources.

### Essential for large documents or documents with many sections

* **Table of Contents** -- Ideally each entry is hyperlinked to the relevant section/worksheet.

### Suggested for many documents

* **Glossary** -- When a document is introducing a lot of specialized vocabulary or has a significant number of acronyms, a glossary may be helpful to many readers.
* **Owner’s name in header or footer** (on printed pages)
* **Page numbers** (on printed pages)

## **Language & Grammar**

It takes time to revise documentation -- to transform complex concepts into simple, clear, and accurate documentation. Your time spent revising is a priceless gift to the user -- saving them time later on!

### Essential for all documents

* **Describe acronyms at first use** -- This allows the reader to assimilate what may be a new acronym without interrupting the flow of information by having to consult a glossary or the internet.
* **Keep things short --** Typically shorter sentences, paragraphs and sections are easier to read. Use bullets and numbered lists when appropriate. Be concise (but not at the expense of clarity).
* **Avoid unnecessary jargon, pretentious language, or business babble.** -- Documentation cluttered with unnecessarily complicated language does not get used. It tends to be confusing. Sometimes clarity requires the use of technical terms, but otherwise plain, straightforward language should be the goal.
* **Avoid writing anything in ALL UPPERCASE unless absolutely essential** -- and then only for very small bits of text. For example, a word, phrase, or header might occasionally need to be fully capitalized, but not sentences or paragraphs. Our brains can more easily recognize words written in mixed case; reading passages in all uppercase is actually fatiguing to most brains. Plus, in some cultures, all uppercase text is the equivalent of shouting.
* **Use the spell-checking and grammar-checking tools available to you.** -- If your documentation lives in software that does not have spell-check (like Excel) then consider copying-and-pasting the nearly-final draft into a program that does have spell-check in order to help with proofreading. Small typos can sometimes lead to big problems in comprehension or implementation.

### Suggested for many documents

* If possible, have someone who is new to the content area (or only familiar with it at a surface level) **peer review the documentation.** Someone who is deeply familiar with the material may not notice when you use jargon or gloss over areas, but a newcomer will ask questions that will help pinpoint parts that need revision.
* If your agency has style or branding guidelines, be sure to consult them.

## **Tables & Spreadsheets**

Tables and spreadsheets are ubiquitous in documentation because -- when used properly -- they can make it easier to organize and present data in an easily consumable form. However, tables can impede understanding and frustrate users if they are structured or formatted poorly.

### Essential for all documents

* **Tables should have descriptive names.** -- In text documents, this provides a method of directing the someone’s attention to a specific table. In spreadsheet software, it allows the user to reference that table by name.
* **Set defaults so that headers are always visible.** In a workbook, this can be accomplished by ‘freezing’ the relevant column and row headers. In text documents, make sure to adjust the table settings to print the relevant headers on all pages.
* **Spell out words in columns that are used as flags.** For example, writing “Yes” or “True” rather than just placing an “X” in the column. This helps with clarity for regular readers and for those using screen readers.

### Suggested for many documents

* **Tables with more than 5 columns should use row banding** (alternating colors for rows) to make it easier for users to follow a specific row of data across the table.
* **Use subtle colors for backgrounds and/or borders as the default** so as to reduce visual clutter. Thick borders or ones using intense colors should be used sparingly for emphasis -- such as for headers or to define different sections of a long table.

## **Visuals, Graphics, Images, Diagrams, Figures, etc.**

We’ve all heard that “A picture is worth a thousand words.” Describing something can be helpful, but actually seeing the thing is even better. However, visuals only work when they are clear and comprehensible. We don’t want our users to squint at the documentation, frustrated as they try to discern tiny text on a pixelated screenshot or try to unravel the mysteries of a graph lacking titles or labels.

### Essential for all documents

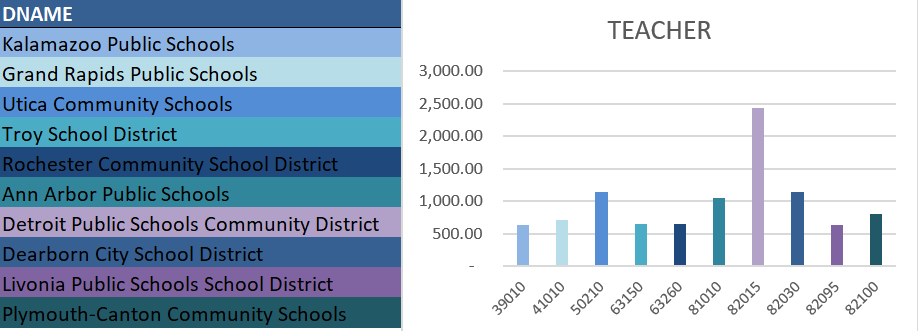
* Graphs and charts should have **appropriate, descriptive titles located at the top left or top center** of the visualization.
* Images, diagrams, or figures that are referenced by the text ideally have appropriate titles.
* **Avoid visualizations that mislead viewers.** For example, 3D graphs or radar charts almost always make users over- or under-estimate some values.
* **Clearly convey units of measurement** **used** (if any). They could be included in the title, on the axes, or as part of a caption.

### Suggested for many documents

* **Visuals/text should be legible and intelligible** unless there is a purpose behind using a blurry or incomprehensible image. Generally speaking, if some of the words or images are so small or pixilated as to be unintelligible, then don’t use it.
* **Non-decorative visuals should still be comprehensible when viewed in greyscale.** -- If the visuals depend so heavily on color that the meaning vanishes when viewed in greyscale, the visual may need revision so that those with colorblindness or using printers can still understand the documentation.
* **Use legends for graphs or charts if the visual itself does not have the necessary labels** directly attached to the relevant bar/line/point/etc.

We have created some examples of effective and ineffective visuals. Look at Figure 1 and 2 on the following page. Which one is easier to understand? Which one is more accessible for those with visual disabilities?

**Figure 1**



**Figure 2**

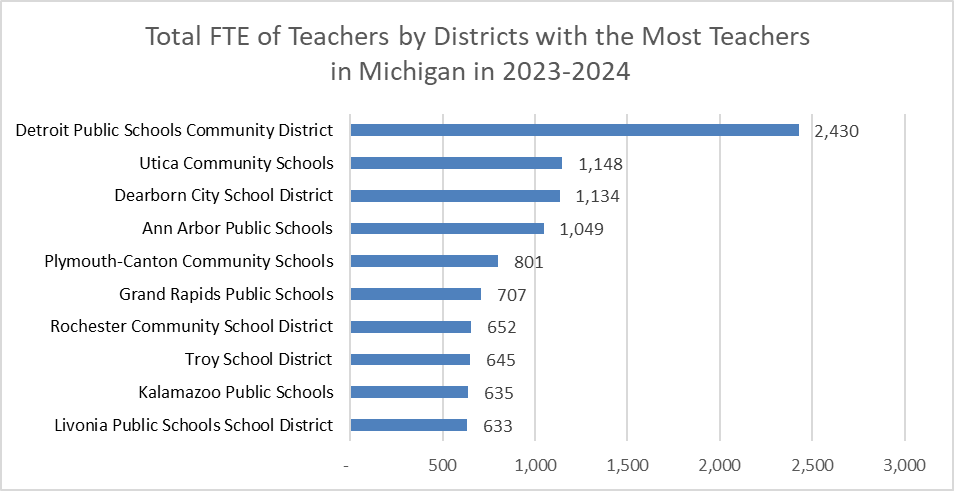


Figure 1 and Figure 2 were created using the Registry of Educational Personnel Full-time Equivalency Data Set for the 2023-2024 School Year collected and published by Michigan’s Center for Educational Performance and Information. Downloaded from <https://www.mischooldata.org/additional-staffing-data-files/> on August 29, 2024.

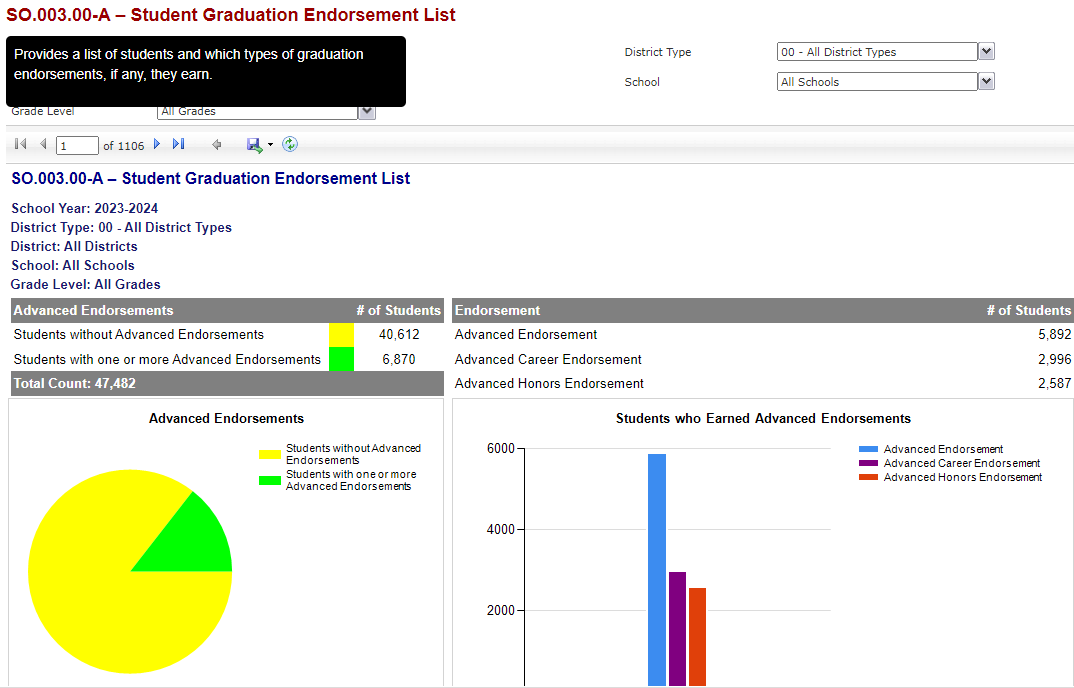
**What are some of the ways that Figure 2 is more effective than Figure 1?**

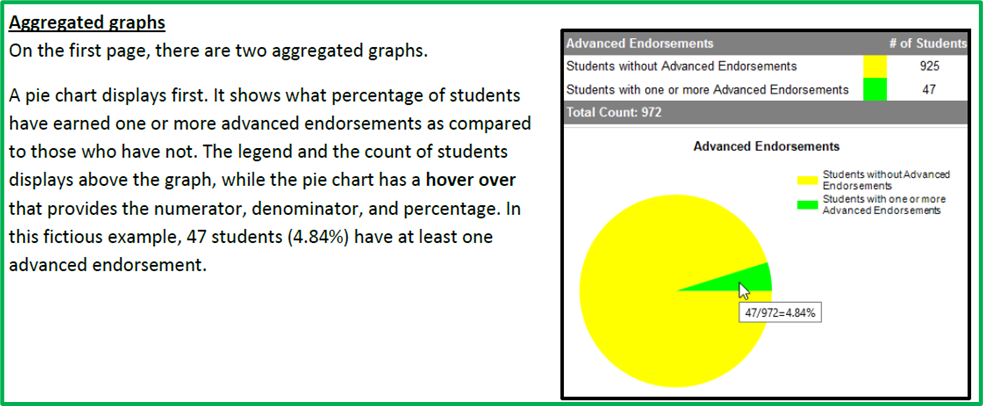
1. The descriptive title “Total FTE of Teachers by Districts with the Most Teacher in Michigan in 2022-2023” gives important context:
   1. when and where the data was collected,
   2. the unit of measurement (Full Time Equivalent rather than headcount), and
   3. that the ten districts represent the districts with the *most* teachers in Michigan (as opposed to a random sample, the bottom 10, or a selection based on some other metric).
2. A horizontal bar chart removes the need to have a color-coded legend because the district names can be next to the bars instead of just the district codes. This eliminates the frustrating experience of having to look back and forth between the legend and the bars over and over again. It also means that the chart is just as meaningful in greyscale as it is in color.
3. Additionally, horizontal bar charts are almost always easier for users to read because the axis text can be horizontal as well.. Vertical bar charts are really only best for time-based data series.
4. The data has been sorted into descending order -- making comparisons easier and reducing misconceptions.
5. Data labels on the ends of the bars make it easy for the user to know values instead of having to estimate based on the axis labels..
6. The FTE has been rounded to the nearest whole number since public FTE reports rarely need two digits of precision and whole numbers are quicker and easier for viewers to grasp.
7. All the text is high-contrast with its background, and meaningful alternative text helps viewers using a screen reader to understand what the visual represents.

Most of these guidelines also apply for creating dashboards and interactive displays. On the next page is a screen capture from South Dakota (Figure 3). It sets up the visualizations to be effective and clear by providing:

1. A header at the top that identifies the topic for the dashboard (“Student Graduation Endorsement List”)
2. A header tooltip with additional context if needed (“Provides a list of students and which types of graduation endorsements, if any, they can earn.”)
3. Clearly visible filters showing the context for the data -- such as the school year and which district types, schools and/or grades are included in the report.
4. Shaded headers to help define different ‘spaces’ on the dashboard.
5. Colors used in the graphics are still distinguishable when viewed in greyscale.

**Figure 3**



**Some additional design points to note: Figure 4**   


1. The dashboard does not rely on just visualizations OR tables in order to convey data, but uses both so that users can have both the precision of seeing values in a table while also visualizing the relative values via a pie chart or bar chart. When data labels would make the visualization too cluttered, this is an effective design choice.
2. Hover-over tooltips help provide additional details for the visualizations - such as identifying the numerator, denominator and exact percentage for the pie chart slices (see Figure 4 - created using fictitious data)

## **File format**

While technical files typically will have a change history and revision numbering at the top of each document to reflect the last time the policy or standard was reviewed, modified and approved, using consistent naming standards for the file itself is extremely beneficial to allow for easier searching if key words are used. Some organizations use a hierarchical approach to naming conventions, such as the example below:

A sample file name: SD_DOE_SD-STARS_Report_Standards.docx

Each part is labeled 
1 = SD
2 = DOE
3 = SD-STARS
4 = Report
5 = Standards



1. Primary organization
2. Department
3. Program
4. Document Type
5. Purpose

If your organization doesn’t already have a standardized way to name files or some naming conventions, consider starting that conversation.

### Essential for all documents

* **Use a filename that will help newcomers understand what file contains.** Optimize for human readability and understandability, to avoid possible confusion when using a term that might appear in different contexts or across different domains.
* For documentation, **use file formats that can be opened across platforms and devices.** For example, diagrams shared as .JPEG, .PNG or .PDF can be opened by virtually anyone, but a Visio file would either require access to Visio/Microsoft 365 or force users to take an extra step of downloading the Visio Viewer.

### Suggested for many documents

* Include “CEDS” and/or the CEDS domain addressed in your report as part of your naming conventions, increasing searchability.

# Doing the right thing

Addressing concerns of privacy, security, accessibility and copyright laws aren’t necessarily fun topics -- but they are very important ones. Most agencies have their own rules and regulations in these areas, and we encourage you to familiarize yourself with your own organization’s policies.

## **Privacy & Security**

Technology and data documentation often contains sensitive information, such as system architectures (e.g., server number, port number, network traffic), data schemas, and proprietary algorithms. Unauthorized access to this information can lead to data breaches, intellectual property theft, and misuse of confidential information. It is essential that privacy and security needs be addressed when documentation is created and shared. Doing that not only helps protect against system vulnerabilities and compromises, but also helps build trust in relationships with stakeholders and protect the organization’s reputation.

### Essential for all documents

* If the document is restricted to specific audiences, explicitly identify those audiences.
* Where possible, mask or redact sensitive information within the documentation.
* Use anonymized data in documentation to prevent the identification of individuals (such as in data visualization documentations that show student-level identifying information).
* Ensure the documentation follows the established privacy and security guidelines (including auditing, training, etc.)

### Suggested for many documents

* Implement role-based access control to ensure only authorized personnel can access specific documentation.
* Use encryption to protect documentation both in transit and at rest.
* Use secure document management systems to offer features such as version control, access logs, and permission management.

## 

## **Accessibility**

All documentation should be accessible and comply with the Americans with Disabilities Act guidelines. Our goal should be for *everyone* to be able to use these resources as easily as possible.

Note that the best practices listed below are not an exhaustive list. Most public agencies have specific criteria they need to meet before an output (report, visualization, documentation) is ready to be public-facing.

### Essential for all documents

* No part of the document depends solely upon color in order to create understanding. (For example, headings are not just a different color, but make use of the heading tag and a different size/style of font than the regular text. Line graphs also have labels or symbols.)
* All non-decorative text has a contrast ratio of 4.5:1 for normal text and 3:1 for large text.
* All non-decorative visuals have alternative text.
* Have reasonable margins for the intended purpose of the document. Never have margins under 0.25” in case someone needs to print the document.

### Suggested for many documents

* Use whatever accessibility tools your software provides to make the document more accessible for screen readers (ex: Word has heading tags embedded in the Header style, PowerPoint allows you to determine tab order for elements on the page, etc.)

## 

## **Citation of outside sources & Respecting copyright**

It is often tempting to ‘borrow’ language or images from outside sources to use in our documentation.

In some cases, that’s not a problem -- for example, if the source is in the public domain or if the creator explicitly permits reproduction of their work via a creative commons license. In those cases, we have a moral and legal responsibility to acknowledge the use of their work, but we *are* free to use their work.

In other cases, ‘borrowing’ can actually be plagiarism or copyright theft. As your high school English teacher would say, “Just don’t do it. It isn’t worth the consequences.”

### Essential for all documents

* **Visuals and text must be obtained through legal methods** (created, licensed, or in the public domain).
* **Give credit to the original owner/creator if visuals or text were borrowed instead of being created.** Attribution can happen near the visual/text, in a footnote, or at the end of the document.

### Suggested for many documents

* **Remember text or visuals generated by AI are not necessarily in the public domain.** The AI may have plagiarized copyrighted sources. Do your due diligence before using anything generated by AI.

# Best Practices specific to sharing on GitHub (and within the Open Source Community)

Sharing data and report resources with partners and other state agencies fosters a culture of collaboration and creativity that transcends state and organizational boundaries. Common repositories like GitHub allow efficient sharing, access and version control with very little effort on the part of any involved party. To maximize benefit to all and efficiency in maintenance, it is recommended that the following documentation best practices are considered when setting up and curating a GitHub repository.

## **ReadMe**

A ReadMe file is the first stop for new visitors to your GitHub. It provides the first opportunity to introduce participants to the purpose and proper use of content within the repository. A properly constructed ReadMe file will contain similar introductory information as other documentation summarized into smaller, digestible statements linking to more detail within the repository when brevity cannot be achieved.

### Essentials for ReadMe files

* **Project Title and Project Description**

A ReadMe file should first and foremost contain a project title, prominently located for consumers to easily identify what they are interacting with. A short description will usually follow providing a high-level summary of pertinent details of the project.

* [**Purpose, Audience, Assumptions & Introductory Information**](#_h8kixjq4euch)

A short section outlining the purpose, intended audience and any contextual information necessary to properly consume and/or use the contents should be included. After reading this section, the reader should feel confident in assessing the project’s value to their own work effort.

* [**Authorship, Ownership, Versioning & Copyright**](#_2yxsqkko49d) **information**

A ReadMe should always provide the reader with details answering the following questions:

* Who created the contents?
* Who is responsible for the contents?
* Have the contents changed over time (and if so, how)?
* If there are questions or proposed revisions, who should be contacted?
* Can the contents be used with or without a license and what copyright implications exist?
* **Technical Dependencies**

The ReadMe should give readers a sense of any technical dependencies that exist in using the contents. This may include software, hardware, data modeling or other infrastructure needs. The goal of information provided in this section is to ensure that users don’t attempt to use the contents being shared before it is actually possible in their local context. For some contents, this may require a hyperlink to more detailed information

* **Table of Contents**

When possible, a table of contents can be a great tool to provide quick bits of information hyperlinked to other sections of the ReadMe or even to contents throughout the repository. At a minimum, it should give the reader an indication of organization of the ReadMe, the repository folder structure/contents, or both.

### Suggested for many ReadMe files

* A description of how to best navigate/use that space in GitHub
* Quick links to reference material, sub-folders or other common/frequent use locations

## **Folder & File Structure on GitHub**

Organizing your folder and file structure on GitHub effectively is crucial for maintaining a clean, understandable, and scalable project. Here are best practices for setting up your repository structure.

### Essential for all GitHubs

* **Logical Grouping**

Group related files and directories logically to make it easy for users and contributors to navigate the project.

* **Consistency**

Maintain a consistent naming convention and structure throughout the project.

* **Clarity**

Use clear and descriptive names for files and directories to indicate their purpose and content.

### Suggested for many GitHubs

* **GitHub Structure**

It is recommended to follow a GitHub structure consistent with your organization. A repository may be appropriate at the report level or group/owner level (e.g., business unit). Either way, it is recommended to pick one approach and consistently set up repositories accordingly.

* **ZIP Files**

Be strategic about which files are shared as ZIP files. If a collection of files must all be downloaded together (such as with a software install), grouping the files together in a ZIP folder makes sense. If the files are so large that compression is required, then using a ZIP folder makes sense. If your computer automatically zips files together when uploading a group of files, but visitors to GitHub may not need or want to look at all the files that you are sharing, then take the time to load them individually so that visitors can pick and choose which files to read.