HENNEPIN COUNTY
PUBLIC HEALTH

Toward becoming an informatics-savvy local health department: lessons learned

Document information

- Authors: Aaron Peterson, Nate Bean, Dave Johnson
- Contributors: Andrew Murray
- Published: December 2024
- Last updated: July 2025
- See Appendix A: Changelog
- Last reviewed: July 2025
- Contact PublicHealthData@hennepin.us with any questions or comments.

Overview

This document describes the history and lessons learned of the public health informatics unit at Hennepin County Public Health (HCPH). We hope this document will be helpful to public health departments looking to grow their informatics capacity and implement data modernization efforts.

Background

- The public health informatics unit was staffed in March 2022 with a working supervisor (Aaron) and in June 2022 with a data engineer (Nate).
- In addition to the years since the unit was staff, this document describes pre-launch lessons on leadership buy-in, funding considerations, etc.
- The public health informatics team has three primary roles:
    - Data engineering
    - Data and information management
    - Data governance
- The team is responsible for:
    - Data modernization efforts
    - Public health cross-sector data framework
    - Data pipeline development and maintenance

Hennepin

    - Data and information sharing and permissions
    - Documentation organization
    - Data sharing agreements

Before informatics

- Long-standing need identified to modernize data storage and processing lacked funding and staffing
- Post-COVID Federal American Rescue Plan Act (ARPA) funds allocated through county board action to create public health informatics unit, adding 2 full time employees
- Work aligned with countywide focus on data modernization

Data modernization

Data modernization efforts often focus on the IT components (data lake, Databricks, etc). We emphasize that these efforts also include human resources (data engineer staffing, trainings for public health data analysts to leverage the tools), and the legal framework to do the work. We use these values to guide our efforts.

Future ready

- Public health informatics is a rapidly evolving field. The data infrastructure needs to be adaptable to future changes in preferred technology tools, data structure, data analysis needs, increasing data complexity, and increasing dataset size.
- Separating storage and compute (see below) makes it easier to accomplish this.

IT alignment

- The data infrastructure was developed with IT architecture and business intelligence (BI) team members who advised on IT best practices.
- Primary and backup contacts should be identified to troubleshoot IT issues.

Documented

- Thorough and accessible documentation will ensure the data infrastructure meets the needs of stakeholders and complements the replicable value.
- See the Documentation section below for more information on implementing this value.

Standardized and replicable

- Data pipelines, including reporting, should be aligned across analysts.
- Furthermore, with IT support, these processes should be replicable through documentation and cross-training.

Key concepts

Storage vs. Compute

- Modern data infrastructure separates storage resources from compute resources.
- Storage refers to the platform that you use to save, keep, and organize your data.
    - Storage is accomplished through a data lake.
    - A data lake is different than a data warehouse or a database. A data lake can store structured and unstructured data in its original/raw form while a data warehouse and database will contain processed data.
    - You may, also, hear the term data lakehouse. This is a newer concept that combines the advantages of a data lake and a data warehouse. While we see the theoretical advantages to implementing a data lakehouse model, we do not think the additional human and IT resources are worth the effort currently at HCPH.
- Compute refers to the platform that you use to transform and analyze data, usually through code.
    - Compute is accomplished through data orchestration tools that automate movement and transformation of data across platforms. Examples include Data Factory or notebook-based tools like Databricks or Synapse.
    - Advantages of code-based compute solutions are outlined later in the document.
- Be sure to understand connection options between your choice of storage and compute platforms. Connections may be easier and more secure between some platforms than others. Integration between platforms from the same developer (e.g., Microsoft) may be smoother.

Data *and* information management

- When completing the InfoSavvy assessment, we realized informatics encompasses management of both data and information.
- Data can be thought of as quantitative and qualitative data that havetraditionally been defined under the purview of public health surveillance and assessment (e.g., vital records, infectious disease surveillance, immunization records, program evaluation, public health surveys, etc).
- Information management can be thought more generally as knowledge that can inform program and business decisions, including:
    - Documentation (in line with our values)
    - Inventories to track data sources, datasets, applications, Power BI workspaces, etc.
    - Tracking of information such as contractors, customer contacts, etc.
- The difference between data and information may not always be clear.
- Certain information sources may have existing management systems managed at the enterprise level. The ability to connect these data sources to the other data and information systems managed by informatics can lead to better informed decision making.

Tips for success

- Advocate for informatics staff to have IT access.
    - Whatever your IT BI team has for access, you should have, too.
- Complete the InfoSavvy assessment developed by the Public Health Informatics Institute.

Change management

- Develop a plan to ensure existing staff properly utilize new, modernized data tools:
    - Training
    - Incentives/directives to use
- The HCPH informatics unit had the advantage of being a new unit, so we could develop policies, procedures, and best practices without historical ways-of-doing-things.
- We recognize moving from years-old practices requires intentionality and time commitment.
- To help analysts migrate to the new technologies described below, we emphasized we were always available to answer questions via Teams chat, scheduled meetings, or email.
- We developed a data lake user manual, Databricks user manual, and a Power BI developer manual.
- Additionally, for Databricks, Nate recorded a tutorial to help people navigate and get started on the platform.
- Each time a new staff member starts using the data lake or Databricks, we ask them to review the videos and manuals, then offer a one-on-one meeting with them.

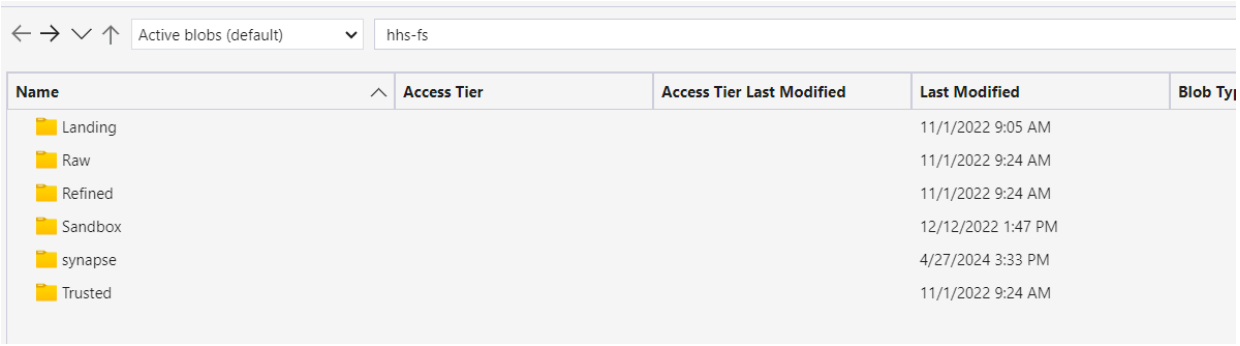
Data lake organization

- We recommend a container for public health separate from those used by other departments.
    - This makes it easier to manage permissions and mount the container to compute platforms.
- Dates should be in the form: YYYY-MM-DD to align with international computer science standards.
    - We started out using YYYY.MM.DD – which was okay, but we switched once we learned about the standard.
- Set general standards for file and field names.
    - File names vary in our data lake but are typically in snake_case and include a date in the above format, when relevant.
    - Raw file names will often be in the format: `topic_[start date]_[end date]_[extracted date]`
        - Example: `childhood_immunizations_2024-01-01_2024-12-31_2025-01-07`.
        - You are adding metadata to the file name that can be extracted and converted into information in the Trusted dataset. More on Raw and Trusted below.
    - By default, all fields in Trusted files have field names converted to snake_case and follow other field naming standards (no punctuation, etc).
    - We started adding standard metadata date fields: record_added, record_last_updated, dataset_added, dataset_last_updated. This has been particularly helpful to track changes when combining weekly birth and death record files into a single Trusted file.
- It's important to establish a process for regularly managing and reviewing data permissions:
    - We use a Databricks script to permission our data lake using Microsoft Entra (formally called Azure Active Directory) security groups. This creates a clear record of the current data lake permissions and record of changes. It also avoids individual-level permissions, which can be messy and increase the risk people retain access to data they should no longer have.

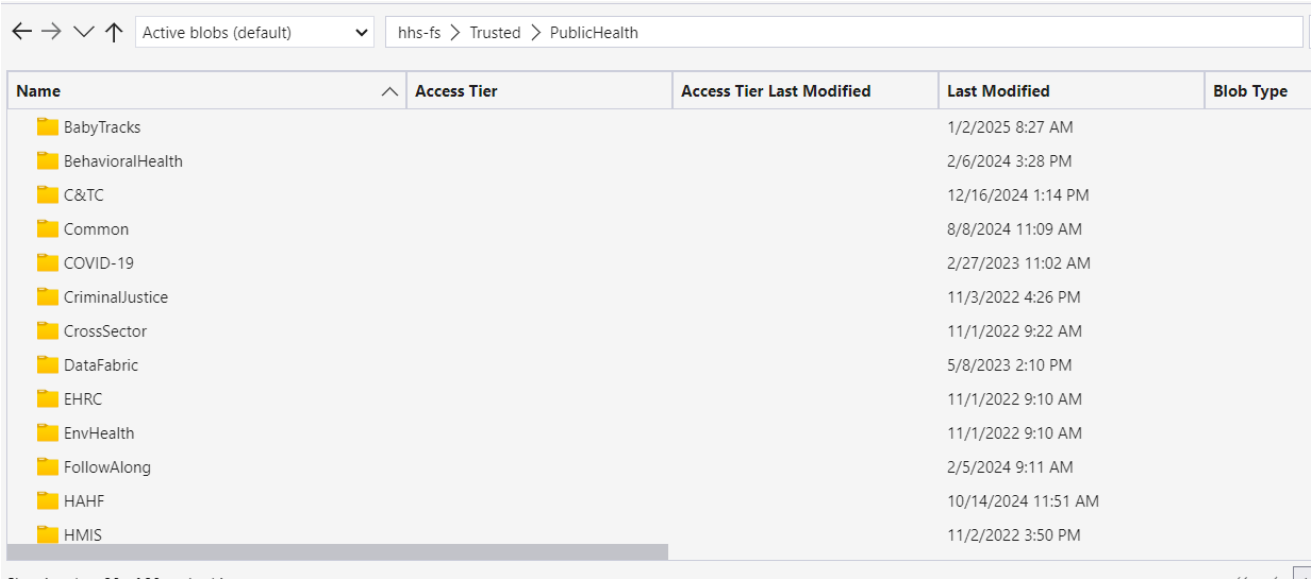
    - We have a quarterly meeting with important stakeholders to review data access where we focus on tasks like removing access for people that have changed roles and making sure current permissions are acceptable for department leaders.

Medallion architecture

- Our data lake is organized into three main levels: Raw, Refined, and Trusted.
    - Raw stores all data as it was originally received or downloaded; we will rename files without modification of the data itself. This data is always retained in case we need to access the original data. This could be necessary if we found an error in the code used to transform this data, or we needed data to fulfill a data request or audit.
    - Refined contains intermediate files, including those that are only partially finalized or aren't meant to be broadly available for reporting or analysis.
    - Trusted includes finalized, quality data. All analysis and reports should only pull from Trusted files, and analysts should be confident in using Trusted data for their reporting.
    - We also have the Landing and Sandbox levels but don't regularly utilize them.



- Data teams across different county departments have somewhat different conceptions of what data belongs in each level of their respective data lakes, so there's some flexibility in what each layer includes. However, we think the key principles that should always be reflected in your data lake architecture include:
    - Data should always be retained in its original format.
    - It should always be clear which state data is in (original, edited, final) to prevent accidentally saving over original data or reporting using data that is incomplete, out of date, or has unexpected transformations.
- We create an identical parent folder below each of the three data lake levels. For instance, there is a *Raw/VitalStatistics* parent folder and there are Refined and Trusted versions of that folder as well. Subfolders below that level vary based on the data present in each level.
- Folder permissions can be aligned across each level (the same permissions for each level of a parent folder) or vary by level. For instance, you may grant a group access to the Raw version of a folder to upload data, but not the data stored in other levels.
- We created a CrossSector folder for when we combine data from different data sources to make clear data privacy rules for all data sources must now be considered.



File types: parquet and csv files as defaults

- In general, we write out a csv and parquet version of a file in Trusted.
- Most analysts are familiar with csv files, and these can be manipulated in Excel. They're a better option for storage than Excel files themselves, because the latter can contain all sorts of formatting and other non-data content. Example: an age group column with a value of 1-3 might be converted to Jan 3.
- Parquet files offer advantages over csv files including:
    - You can define column types. If you read a parquet file into Power BI, these column types will be automatically applied in the data model.
    - You can read in a filtered part of the file using the arrow package when coding in R, Python, etc. This is helpful with large files, and you want to read in a subset of data.
    - Free text is structured in a way that you don't need to worry about commas messing up columns.
- In the Refined folder, we usually write out parquet files, since these will not be used by analysts and only need to be read in the coding environment.
- Chapter 22 of R for Data Science (2e) talks about parquet and arrow.
- Official website: Parquet
- Wikipedia: Apache Parquet - Wikipedia

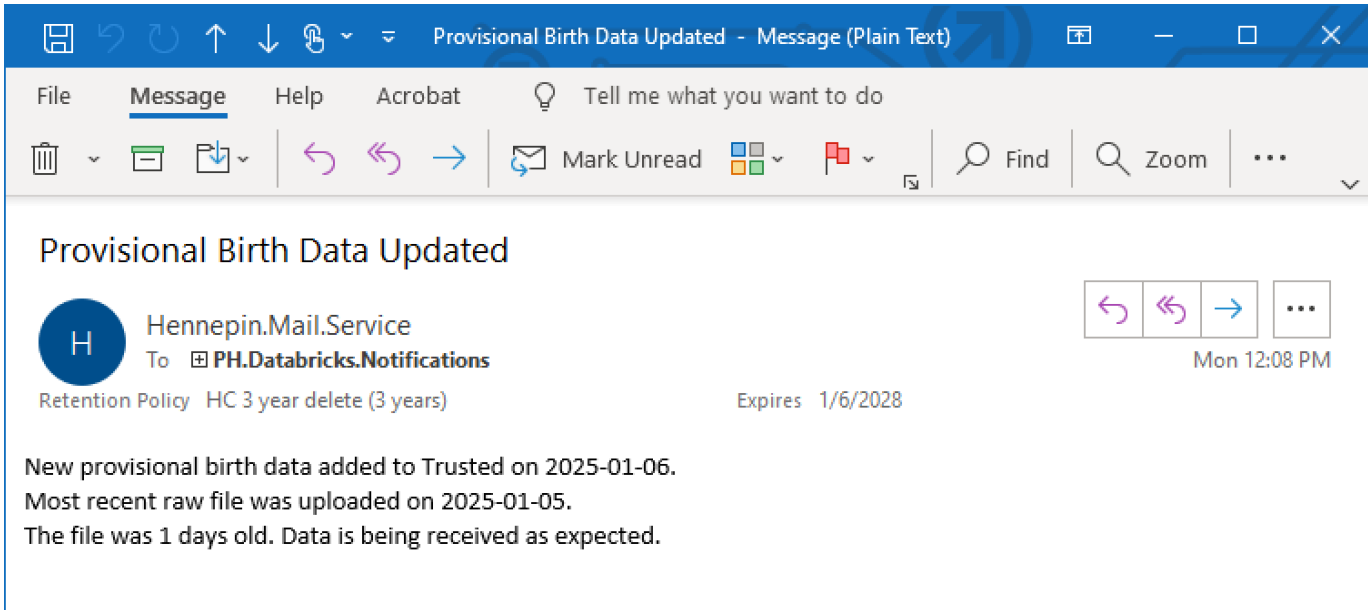
Data migration

- We began the data modernization efforts by migrating birth records, death records, and infectious disease surveillance files from the network drive to the data lake.
- We can provide a migration project document showing how we managed that process.

Helpful technology solutions

Email alerts

- Data Factory, Synapse, Power BI, Databricks, etc. have job/refresh failure notifications.
- Additionally, Nate set up email alerts to catch when files from outside sources are not being uploaded as expected. As the amount of data stored in the data lake increases, it's harder to manually monitor whether data is being received at the correct cadence, so this prevents issues when we attempt to use data and realize we haven't received a new file in several weeks.
- Also, useful for less regular file uploads (ex: client list for outreach workers, for us that was new births meeting criteria for the Follow Along Program)



Data lake inventory Power BI report and avoiding permanent deletion

- Nate wrote a script that monitors when folders and files are added and deleted in the data lake. This list of changes was used to create a Power BI report for folks to see what is available and what has changed.
- When files are deleted from the data lake, they are retained for 30 days. This was *not* the default policy on the data lake container; be sure your IT administrator enables this.
- We have a recurring meeting reminder on Friday where Nate goes through what files were deleted that week to see if anything needs to be restored. Aaron and Dave are backups on the calendar invite for when Nate is out of office to keep the procedure going.

Code-based solutions

We have found data transformations are usually better accomplished with code-based solutions. Power Query in Excel or Power BI can be a good tool for simple data transformation, but code offers more flexibility. Additionally, it's easier to review code than Power Query for collaboration.

- Code-based solutions create more reproducible workflows, where each step is clearly defined and can be iterated on independently. Transformations done in a coding environment do not directly impact the underlying data, unlike transformations in Excel.
- Code-based solutions can be significantly more elegant for complex analysis and transformations. For instance, there are single-line dplyr functions that handle Excel tasks like complex pivots and joins. The simplicity makes the code easier to work with and also reduces the chance for manual errors.
- R and Python have stronger communities, documentation, and resources relevant to public health data tasks, in our opinion.
- Code-based solutions can be automated as well as applied to new datasets without many changes or duplication of work.
- When working with larger data, code-based solutions are faster (and do not have a row limit) and compute power can be scaled to meet the needs of the task in many development tools.
- Code is better for handling certain types of data, such as JSON or geospatial data

Notebooks

- Most data analytics tools now have notebooks. A notebook is a collection of code and text that can be run as part of the data pipeline.

R vs. Python

- In general, Hennepin County Public Health uses R, specifically the tidyverse libraries, for data transformations.
- We occasionally use Python as well and have identified a few different situations where it's worth using Python over our default coding language
    - When specific packages or resources are only available or significantly better in Python. Some things that come to mind are resources for the FHIR data standard and common machine learning tools.
    - If there is much more available documentation about whatever we're working on in Python than in R. Using Selenium to control an automated web browser was one case where we found this to be true.
    - We're collaborating with other teams that use Python.
    - A project can benefit from Python's faster speed.

R tidyverse

- Tidyverse is a collection of R libraries used for data pipeline development.
- In general, these packages encourage snake_case (lowercase with underscores). Since HCPH uses R tidyverse, we have adopted this as our file and field naming standard as noted above.

Getting started with R tidyverse

- Do a book club with R for Data Science (2e). Be sure to use the second edition.

Migrating from SPSS to R tidyverse

The migration of infectious disease data management and analysis occurred in a stepwise fashion as new tools like Power BI, cloud file storage, and cloud computing were onboarded. Prior to these tools, SPSS scripts were used to run single year analysis. Results were manually entered into excel spreadsheets to create trends over time visuals and single year demographic tables.

Ideally agencies would adopt cloud storage first and then develop R scripts (or preferred language) that standardize variables across file years, append annual files together, and create variables for analysis, culminating in a cleaned analysis dataset. Analysis and visuals can be developed in R or Power BI/Tableau.

Getting started with Python

- Developing Python skills is similar to developing R skills. Our team was part of a book club with other county analysts that used Python for Data Analysis, 3E as an introduction.

Documentation

- In general, we use Word documents to create data source and project specific data and information manuals. We initially had long documents that included related data pipelines and projects (example: vital statistics manual that covered birth, death, maternal mortality, cross-sector projects); we found it's easier to have separate documents for a particular topic, then link between the documents and keep a table of contents document. Usually an analyst is interested in a particular topic, and having a concise manual makes it easier to send them the information they need without saying: "scroll to page 7 under heading X."
- In recent months, we started titling these documents [Topic] data and information manual with the filename being [Topic] data-info manual. This aligns with the discussion above about data and information.

- Screenshots from the table of contents document –

Ordered list

1. Public health data manual.docx – general policies and procedures, overview of data infrastructure
2. Data lake
    a. From IT Enterprise Architecture: Data Lake Design Standards.docx
    b. From IDA: 2022.06.21 EIDS Additional guidance on data lake design standards.docx
    c. Data lake permissions
    d. Data lake technical manual
    e. Data lake user manual
3. Databricks
    a. Databricks user manual
    b. Databricks technical manual
4. General information
    a. Geocoding manual.docx
    b. Data suppression manual.docx – in draft, will publish to PH SharePoint
5. Power BI (PBI)
    a. PHDA/General/Power BI SharePoint folder – when I say a document is internal, it is saved in this folder
    b. Public Health Power BI SharePoint page – landing page, links to relevant countywide SharePoint resources
    c. Public Health Power BI Users SharePoint page
    d. Power BI user guide.docx – internal document; not finished
    e. Power BI tutorial.docx – internal document for when someone gives a PBI tutorial
    f. Public Health Power BI Developers SharePoint page
    g. Power BI report developer manual.pdf – PH SharePoint; last updated Feb 2023
        i. Power BI report developer manual.docx – internal update; publish in early 2025
    h. Power BI public report manual.pdf – PH SharePoint; last updated Nov 2022
        i. Power BI public report manual.docx – internal update; publish in early 2025
    i. Power BI dept contact manual.docx – internal; technical details for PH PBI admins
    j. Power BI workspaces.docx – created by Aaron in Sep 2024 to start keeping track of workspace decisions; not completed as of Jan 2025
6. Birth records
    a. Birth data manual.docx
7. Death records
    a. Death data manual.docx
    b. Death data dictionary.xlsx
8. Infectious disease surveillance
    a. Infectious disease surveillance manual.docx
    b. HIV surveillance manual.docx
    c. Measles surveillance manual.docx (created Oct 2024)
9. HIV
    a. PHDA Documentation / HIV SharePoint folder
    b. HIV data-info manual.docx
    c. HIV data to care 3.0 manual.docx
    d. Cross-sector / HIV outbreak data-info manual.docx
10. Ryan White
    a. Ryan White data manual (phda version).docx
11. Family health
    a. Family health data manual.docx – overview of programs, data sources, and the HHS.PH.FamilyHealth Power BI workspace/app. Individual programs and data sources will have separate documents.
    b. IHVE SharePoint folder
    c. IHVE data manual.docx
    d. HAHF SharePoint folder
    e. HAHF data-info manual.docx
    f. WIC phone report manual.docx

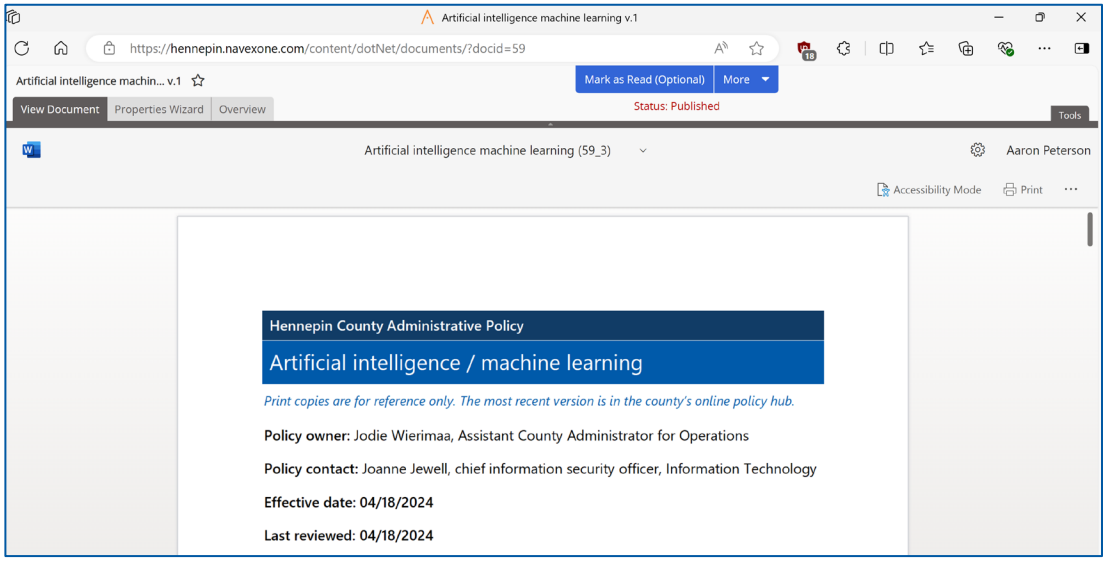
- Document information. In Sep 2024, we started adding this section to the top of each document along with an Appendix A: Changelog (like the one in this document) to track changes.
- Screenshot from the top of the table of contents document --

HENNEPIN COUNTY
PUBLIC HEALTH

Documentation overview

Document information

- Contact: Aaron Peterson
- Created: Sep 2024
- Last updated: Jan 2025
- See Appendix A: Changelog
- Last reviewed: Jan 2025

- We aspire to eventually migrate from Word documents to a browser-based solution, but we have not prioritized that effort. We have tentatively looked at using:
  - SharePoint: other public health teams have created a collection of SharePoint pages as their manual.
  - The eBook, R for Data Science: 2nd edition, was published using Quarto (sort of the successor to R Markdown documents). This would be more sophisticated but an option since informatics team members have experience with R Markdown.
  - Wiki: At least one IT team at the county is looking at creating an internal wiki, and we might be able to piggyback off that effort.
  - Word document wrapper: County administration uses a platform called Navex One to manage some of their policy and procedures while keeping them in Word document format. Screenshot below.
  - 

Appendix A: Changelog

- July 2025: Continued refinement when Aaron and Dave presented 2025 NACCHO360 | PHI*con-Toward becoming an informatics-savvy local health department: lessons from the first three years
- February 2025: Further revisions based on feedback from a CDC Foundation and PHII panel.
- January 2025: Finalized draft and shared with Olmsted County. Aaron did a final readthrough after Dave, Nate, and Andrew added their information.
- December 2024: Started a draft document to share with Olmsted County.