Paper AD02

Navigate Clinical Study Report Tables and Listings – The iTable & iListing project

Katja Glaß, Bayer AG, Berlin, Germany

ABSTRACT

The generation of extensive PDF documents containing tables and listings for clinical study reports remains a significant challenge in the pharmaceutical industry. The iListing project, first introduced in 2021[1], showcased the transformative potential of a web application designed to present listings at Bayer, effectively addressing the limitations of traditional paper formats. Building on the success of iListing, we have now expanded our toolset to include interactive tables. By seamlessly integrating subject-level data with table metadata, we have created a dynamic environment that allows users to navigate directly from tables to their corresponding listings.

This presentation will delve into the benefits of utilizing an interactive web application, detailing its implementation and practical application within Bayer, leveraging SAS and React. Furthermore, we will explore alternative approaches, including the use of R and dataset-json, to inspire ongoing innovation in our data presentation strategies.

INTRODUCTION

The “Tables, Figures and Graphs” section, along with the “Appendices” containing the listings of a clinical study, constitutes a substantial portion of the clinical study report[2]. These elements are fundamental deliverables typically produced by Statistical Analysts and Statisticians. Given the vast amounts of data collected during clinical trials, listings can become particularly unwieldy, even in the PDF format. For instance, in a Phase 3 study at Bayer, we generated 84 PDF files solely for listings, with 38 of those dedicated exclusively to laboratory data. Each file could span up to 15,000 pages, rendering the management of such documents cumbersome, slow, and often frustrating.

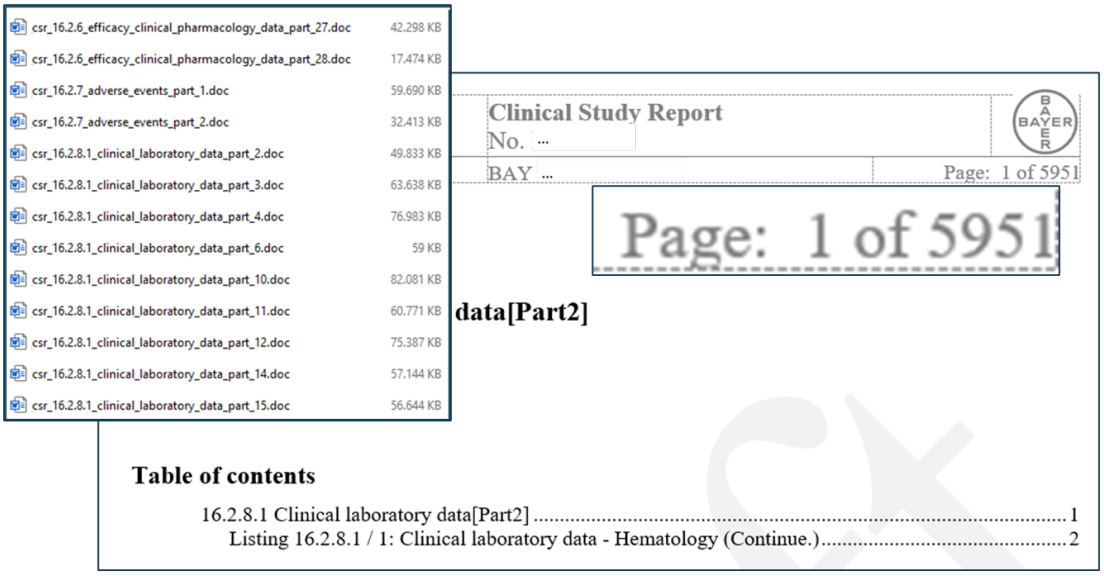

Figure 1: Multiple listing output files, each having multiple thousand pages

We do have to create these files according ICH E3, even though discussions are ongoing on whether at least the listings could be omitted[3]. For this we searched for ways to benefit from the content. To overcome the burden and enable a user-friendly working with these files, we created an interactive web application to display the listings and enabling quick and fluent search, sort and filter options. The iListing Application had been rolled out at Bayer in 2021 and been used ever since.

Despite the necessity of adhering to ICH E3 guidelines for these reports, ongoing discussions have emerged regarding the potential omission of listings[3]. As we still create listings for now, we sought to leverage the content

more effectively. To alleviate the burden of navigating extensive files, we developed an interactive web application designed to display listings while offering intuitive search, sort, and filter functionalities. The iListing Application was successfully rolled out at Bayer in 2021 and has since been a vital tool for our teams.

Given Bayer's statistical analysis tool landscape, we have further enhanced the iListing Application to incorporate interactive tables. This advancement allows for a more comprehensive presentation of data, linking table content directly to corresponding listings, thereby enriching the review process. Reviewers can now seamlessly transition from specific table numbers to related listings, facilitating a more integrated review.

Remark: The data presented in this paper originates from an existing study; however, we have applied anonymization and deduction methods to ensure confidentiality and compliance with regulatory standards.

THE SOLUTION

At Bayer, we have developed two versions of the iListing and iTable solution: a GxP validated version launched in June 2023 and a development version that includes additional features. The development version clearly indicates its unvalidated status, ensuring users are aware of the differences. Depending on the data location, a draft symbol is displayed in the GxP version when the underlying data is in a development environment rather than production.

The application operates as a web-based tool accessible through any browser. Users can specify a particular study via a parameter in the URL; if no study is specified, the application presents a general screen where users can explore example studies or access their personal study through a list of previously opened studies.

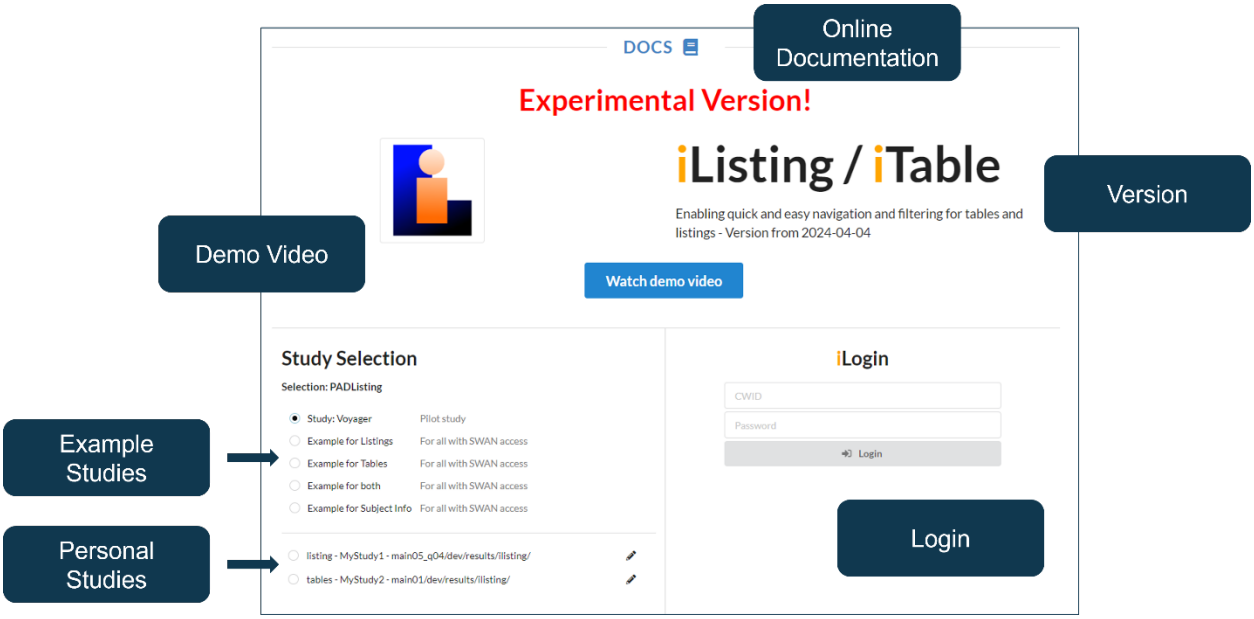

Figure 2: Start Screen of the Application

The introductory screen is designed for user-friendliness, featuring links to online documentation and a demonstration video. It displays the current version, including the release date (e.g., April 2024) and its experimental status for the development version. Users can easily navigate to example studies to play around or their personalized study lists. This list is updated whenever a user opens a specific study link after login. The study structure and the tables are loaded after login through an API, ensuring that only people with access can browse the relevant data. No data is stored locally for security reasons.

Once logged in, users can browse all tables and listings, toggling between them with ease. The content mirrors the deliverables typically found in PDF documents, but the web application offers advanced functionalities such as filtering, sorting, and enhanced performance. For instance, opening a listing in the application is significantly quicker than accessing one or even multiple PDF document.

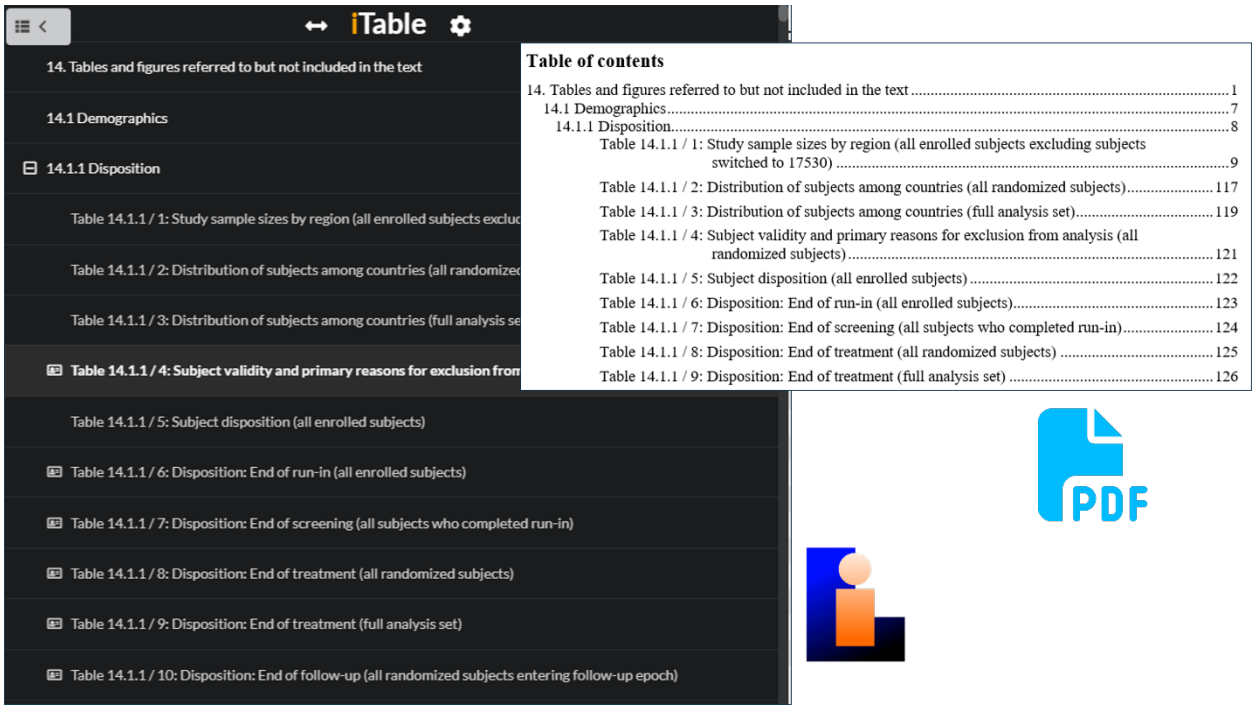

Figure 3: Table of Contents for iTable and the corresponding PDF

When examining a listing, users benefit from a range of interactive features that enhance their experience. In traditional PDF formats, we often utilize a variable at the top of the table, referred to as “PAGE.” While this approach minimizes the number of columns displayed, it presents a challenge: when the variable changes, the table continues onto the next page, disrupting the flow of information.

In contrast, the web application eliminates these width limitations, allowing these variables to be presented as distinct columns. This design choice not only improves readability but also facilitates easier data review.

For listings that exceed the screen's width, users have the option to adjust the width factor, increasing it up to three times. This feature introduces a vertical scrollbar, enabling users to navigate through larger datasets without compromising visibility. As a result, users can comfortably view and analyze extensive listings, ensuring a more efficient and user-friendly experience.

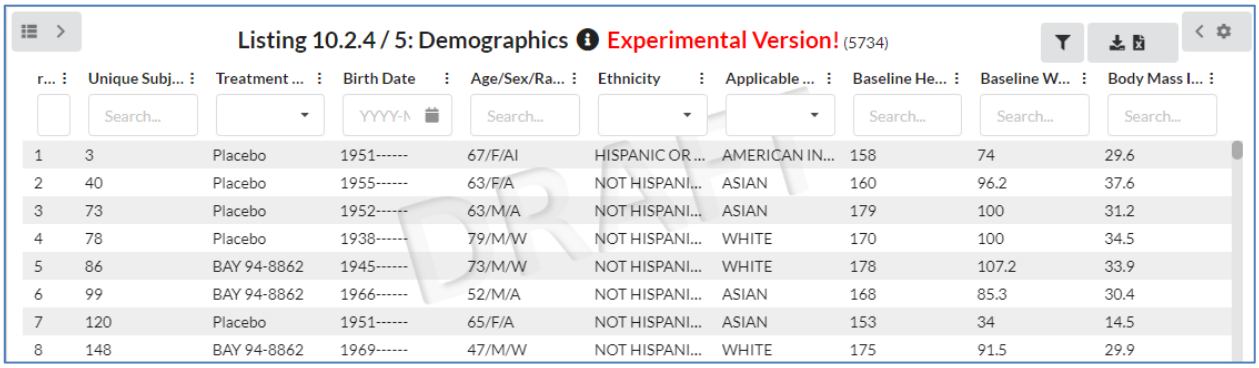

Figure 4: Display of one Listing

Browsing is intuitive, with several special features implemented to enhance user experience. For example, when sorting a column, the application uses an enhanced algorithm that sorts numerically, even if the column is formatted as a character variable. This improvement ensures that values like “10” are correctly placed after “9.” In our implementation, we updated the sorting algorithm to sort numerically by the first number in case the content is formatted like “xx (xxx%)”, which we commonly have available for tables. Sorting can easily be done by clicking on the column title.

The filtering options are robust, offering three types: date filters, dropdowns, and text searches. Users can apply advanced filters, such as greater than or less than, even for numeric values stored as text. This works as well with numbers which are stored as a text variable and special numbered columns like “xx (xxx%)”.

The following table shows an example on different filters for different columns:

Filter	5	<5	<5.001	<100	>100
Example row	5	4.3	5	76 (2.7%)	1815 (63.9%)
Example row	35	3.7	4.3	86 (3.0%)	124 (4.4%)
Example row	5.1	3	3.7	5 (0.2%)	723 (25.4%)

Additional table information, such as footnotes and the origin of the output, can be accessed via an “info” icon next to the table title where also some column options are available, for example to hide columns or change dropdowns to text searches. Users can also customize column widths by dragging the three dots next to each title. Once users have configured their preferred settings for a study, these can be saved for future sessions, enhancing usability.

In listings, where multiple columns are often combined to fit within the constraints of a PDF page, we introduced a feature to split columns by a specified separator. This allows for easy collapsing and expanding, greatly improving the usability of tables with different indent levels for groupings.

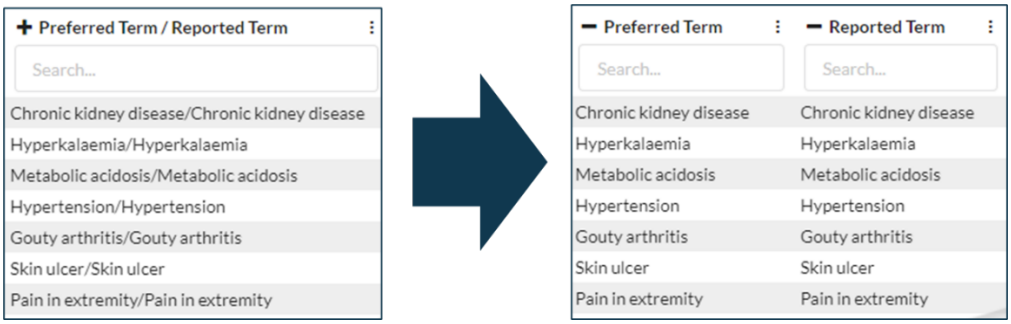

Figure 5: Listing column splitting Option

This column-splitting functionality is particularly advantageous for tables that utilize varying indent levels for groupings. For instance, when comparing different genders across various regions, having the ability to filter by all genders becomes essential. Without this feature, a search for “Sex” would only display the overarching grouping column, omitting the detailed characteristics that lie beneath. By enabling users to split columns, the application ensures that all relevant data is readily accessible, allowing for comprehensive comparisons and deeper insights into the data.

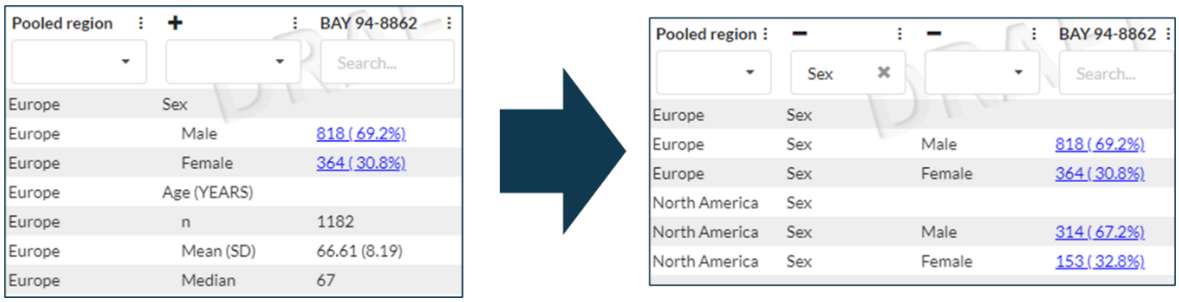

Figure 6: Table column splitting option

Often, users need to compare information across different outputs. For instance, when tracking a specific subject experiencing a adverse event of special interest, the application provides global filters that allow users to display listings exclusively for pre-selected subjects. By applying criteria, such as a particular adverse event, users can filter the listings to showcase only those relevant subjects for all other listings. Moreover, users can access additional listings - such as laboratory data - where only the filtered subjects are displayed, facilitating targeted analysis.

To further enhance the review process, the application allows for multiple listings to be viewed simultaneously, one above the other. This vertical display enables users to quickly glance at various listings, transforming the review and investigation experience into a more efficient and insightful endeavor.

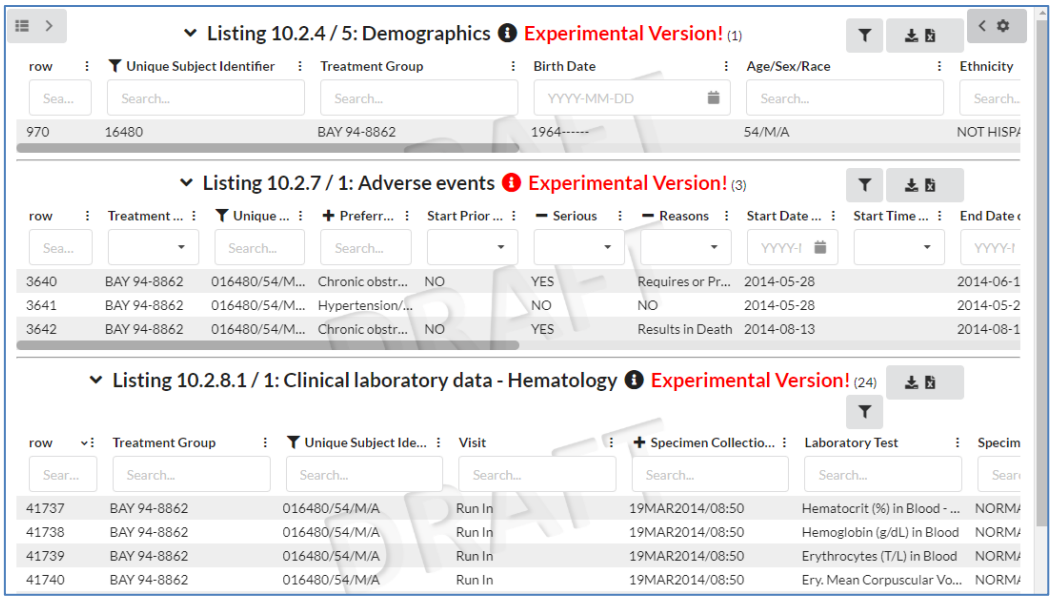

Figure 7: Display multiple listings content for one subject only

As the underlying data is based on the finale clinical study report tables and listings, the subject filter functionality does not extend to the tables themselves. The iListing and iTable solution is designed solely for displaying outputs with various display options; it does not re-derive any frequencies, descriptive statistics, or other statistics. Essentially, it serves as a dedicated output display tool.

At Bayer, we have made significant strides in our metadata collection efforts. Over the past year, we have updated our processes to capture which subjects are associated with specific output cells in certain standard layout tables. This enhancement allows us to display a table and link it to an associated listing that includes filtered subjects. In our new layout, users can click on a table cell to directly access the corresponding listing, filtered to show only the subjects relevant to that cell.

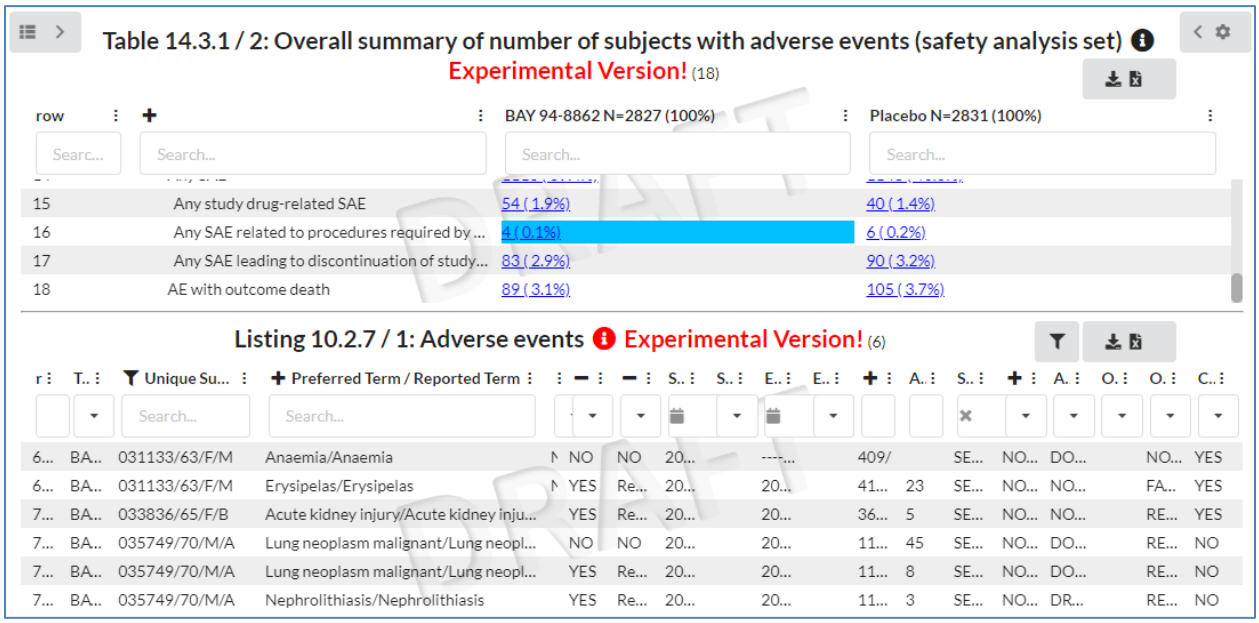

Figure 8: Table cell marked - related AE listing filtered to selected subjects

In addition to these core features, several valuable enhancements have been implemented, including an export functionality designed exclusively for review purposes. We actively seek and incorporate user feedback, and by maintaining a development version of the application, we can rapidly introduce new features and updates. Meanwhile, we continue to support our stable legacy version for GxP tasks, ensuring compliance and reliability in

critical processes.

THE ARCHITECTURE

To facilitate the development of this tool, a comprehensive collection of metadata and data content is essential, all formatted for easy processing. We have decided to store metadata in JSON format, a versatile and flexible text-based standard that allows for efficient data handling. Additionally, content files are stored as comma-separated values (CSV), providing a straightforward structure for data representation. All information resides within our Statistical Computing Environment (SCE), accessed via a robust application programming interface (API). This architecture not only leverages the security protocols of our SCE but also ensures that no clinical metadata or data is stored externally.

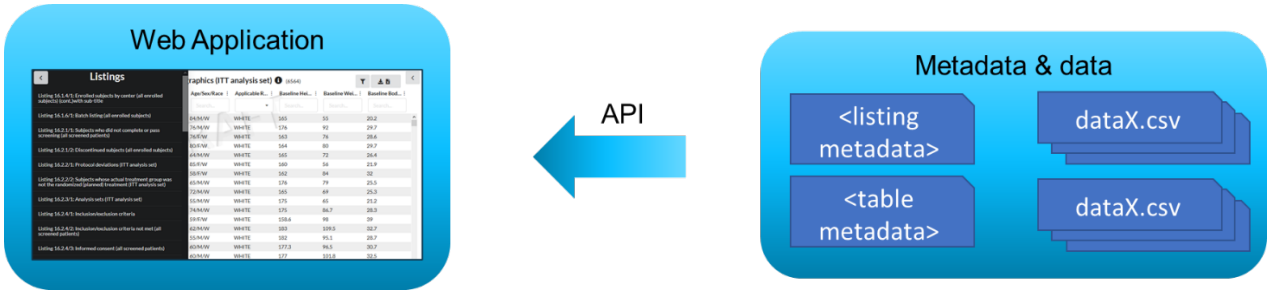

Figure 9: Web Application Architecture using Metadata & Data

The generation of metadata and data content at Bayer is an integrated byproduct of our processes. We have enhanced our SAS macros to automatically capture all necessary information during data processing. Thanks to our programming environment, all outputs ultimately rely on a single macro for final result generation, known as %datalist[4]. This serves as a replacement for PROC REPORT, offering numerous functionalities and optimizations that significantly minimize the effort required for styling and layout adjustments by analysts. Additionally, this macro stores the required content and metadata as SAS datasets.

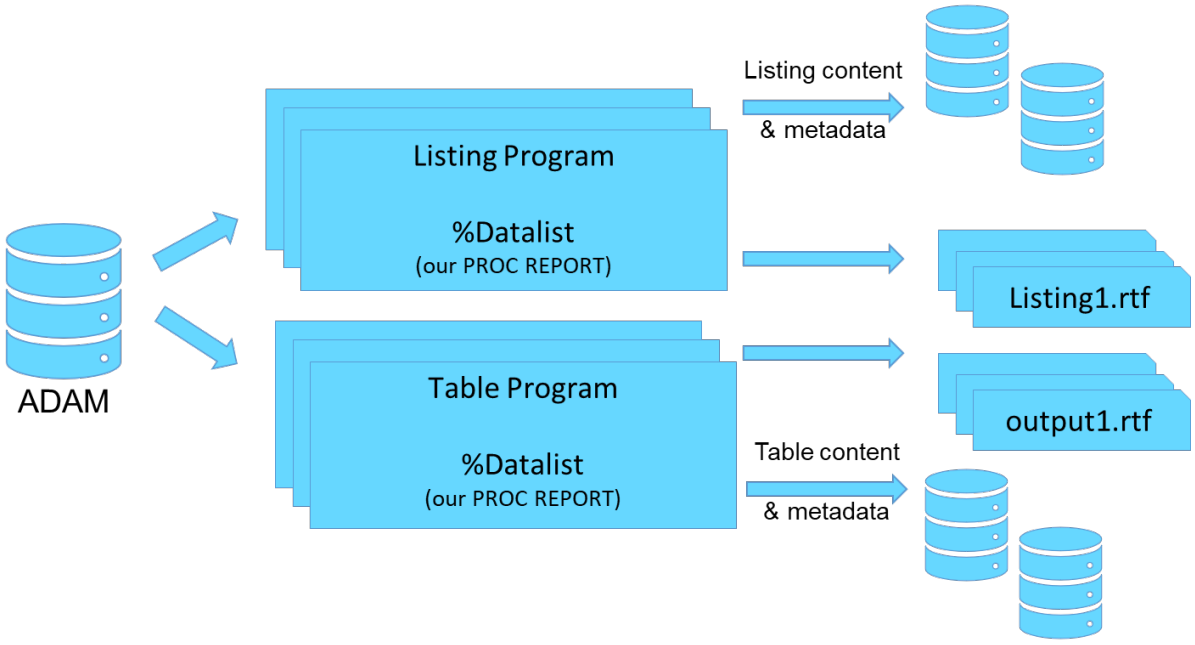

Figure 10: Process of generating general listing and content metadata

Whenever we produce any output, the corresponding data output and metadata are stored alongside the specific output. The final linkage is established through a unique report title, which can span multiple documents. Once all outputs are generated—including final table and listing numbers, as well as Word and PDF documents—we execute

a process to convert the accumulated content and metadata into the required CSV and JSON formats for the frontend. During this step, additional metadata and data can be derived to streamline operations on the frontend.

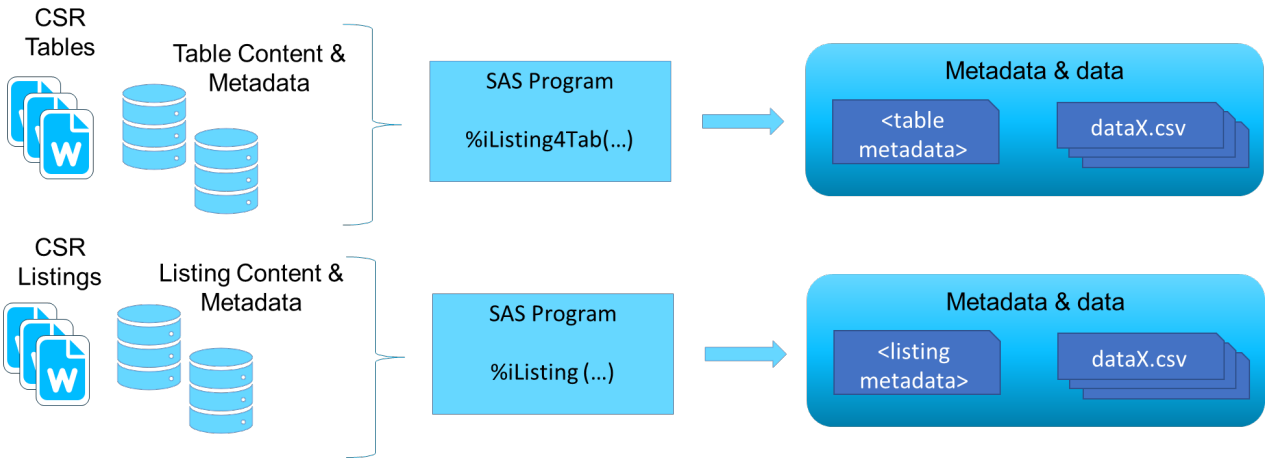

Figure 11: Process to create application metadata and data

THE IMPLEMENTATION

To effectively display data in the frontend, a comprehensive set of metadata is essential. The following list outlines key items that are valuable for metadata storage:

- Generic information
  - Study / pool number
  - Metadata version
  - Type (tables or listings)
  - Location of metadata files
  - Is special subject-linkage available
  - Does subject filters apply (these should only apply for listings)
- Order of tables or listings with section hierarchy if applicable
- Per table / listing
  - Identifier of the content dataset
  - Title (including number)
  - Footnotes
  - Related origin program name
  - Related final document name
  - Display variables
  - Split variables (in case column shrink/enlarge should be available)
  - Subject variables (for subject filters)
  - Number of observations (for completeness checks)
  - Variables
    - Id
    - Label
    - Is it a date variable?
    - Is it a dropdown variable?
    - Does this belong to a “split-group” to enlarge/shrink content

With this information at hand, along with the content datasets containing the actual display data in CSV format, the web application can seamlessly implement the necessary features. This integration enables enhanced functionality and user experience, allowing for efficient data presentation and interaction.

THE “SPLIT” FEATURE

Let’s delve into the “split” feature, which is implemented on both the backend data processing and the frontend

display. Within the SAS macros responsible for generating the content and metadata files, a check is performed to determine whether a column should be able to be split based on indentation levels. When applicable, multiple columns are created, each corresponding to the number of indent levels, with the postfix "_iliN" indicating the specific level.

NAME	_NAME__ILI1	_NAME__ILI2	_NAME__ILI3	_NAME__ILI4
Number (%) of subjects with pre-randomi	Number (%) of subje			
Any AE	Number (%) of subje	Any AE		
Maximum intensity for any AE	Number (%) of subje	Any AE	Maximum intensity for any AE	
MILD	Number (%) of subje	Any AE	Maximum intensity for any AE	MILD
MODERATE	Number (%) of subje	Any AE	Maximum intensity for any AE	MODERATE
Any study drug-related AE	Number (%) of subje	Any AE	Any study drug-related AE	
Any AE related to procedures require	Number (%) of subje	Any AE	Any AE related to procedures requ	
Any AE leading to discontinuation o	Number (%) of subje	Any AE	Any AE leading to discontinuatior	
Any SAE	Number (%) of subje	Any SAE		
Any study drug-related SAE	Number (%) of subje	Any SAE	Any study drug-related SAE	
Any SAE related to procedures requi	Number (%) of subje	Any SAE	Any SAE related to procedures rec	
Any SAE leading to discontinuation	Number (%) of subje	Any SAE	Any SAE leading to discontinuatic	
AE with outcome death	Number (%) of subje	AE with outcome		

NAME

is split up into

_NAME_ILI1
_NAME_ILI2
_NAME_ILI3
_NAME_ILI4

Figure 12: Enhanced content for split up column

In the underlying metadata structure, this table item includes crucial information indicating the availability of a column group. For example:

```
"tf_adae_1" : {
       "id" : "tf_adae_1",
       "titles" : ["Table 14.3.1 / 1: Overall summary of number of subjects …"],
       ...,
       "columnGroupBase" :
            [{ "_NAME_" :
                [ "_NAME__ILI1" , "_NAME__ILI2" , "_NAME__ILI3" , "_NAME__ILI4" ]
            }],
       "vars" : {
           "_name_" : {
               "id" : "_name_",
               "isDate" : false,
               "isDropdown" : true,
               "columnGroupBase" : "_NAME__ILI1 _NAME__ILI2 _NAME__ILI3 _NAME__ILI4",
               "label" : " "
           },
           "_name__ili1" : {
               "id" : "_name__ili1",
               "isDate" : false,
               "isDropdown" : true,
               "columnGroupParent" : "_NAME_",
               "label" : " "
           },
           ...
       }
```

This metadata can then be leveraged in the frontend to display an expand icon when a variable possesses a "columnGroupBase," or a shrink button if it is linked to a "columnGroupParent." Additionally, each dropdown is evaluated on a variable-by-variable basis, allowing for flexibility in determining whether certain groups should be displayed as dropdowns. Ultimately, this results in either a single column or multiple columns being presented based on the data structure and the user interaction.

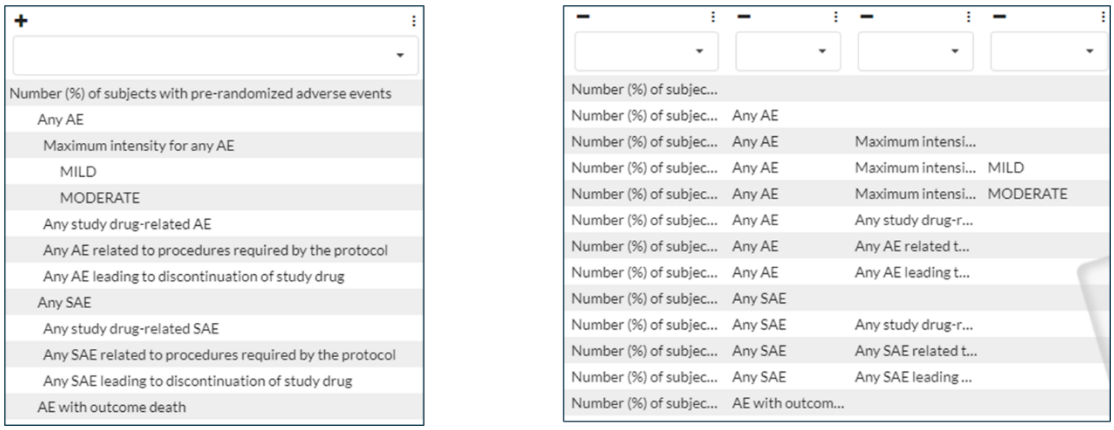

Figure 13: Shrinked or enlarged display of a group column

VIRTUALIZATION OVER PAGINATION

Another critical aspect to consider is performance. The way data is stored and processed can significantly impact the application's speed. When dealing with large volumes of observations, such as those found in listings, traditional pagination methods are often employed, allowing users to view a limited number of rows at a time and click "next" to load additional data. However, this approach can lead to a frustrating user experience, as it disrupts the flow of information.

To enhance performance and user satisfaction, we have utilized a technique known as virtualization. This method ensures that only the data currently visible on the screen is rendered, along with a few additional rows above and below for context. By limiting the rendering to a small subset of data, we achieve faster load times, allowing the application to respond more swiftly to user interactions. As a result, even during scrolling, the rendering process remains quicker than the user's movement, creating a seamless and fluid experience. This approach not only minimizes delays but also significantly enhances the overall usability of the application, ensuring that users can navigate large datasets effortlessly and without distraction.

SORTING OPTIMIZATIONS

When dealing with large displays, the way data is stored significantly impacts performance. To ensure quick sorting and a smooth user experience, we have implemented a strategy that utilizes a simple integer list as an index for accessing rows during display. Instead of sorting the entire data, we sort and store only the index, which allows us to efficiently display the rows in the desired order.

Initially, the index might look like this: [1, 2, 3, 4]. If we sort it in reverse order, the updated index becomes [4, 3, 2, 1]. Each row contains multiple cells, which may include strings or numbers. For display purposes, we track the last sorted column. When the same column is clicked again, we can easily revert the order. This process of reverting an indexed array is highly efficient, and sorting the column content remains rapid since we are only manipulating the index.

The sorting mechanism is implemented as follows:

- Revert sorting
- Date sorting
- “Standard” sorting

Date sorting requires special handling, as dates are stored as strings and must first be converted to date objects. If a date is missing or invalid, we replace it with a value of "zero" for logical comparisons.

Standard sorting is nearly straightforward, as numeric values are automatically converted from strings to numbers, allowing for seamless comparisons. However, we also encounter specialized content formats, such as “xx (yy%)”. To sort these numerically, we identify this format, extract the relevant numeric portion (removing any “<” characters), and convert it to a number. Additionally, we ensure that missing values are appropriately handled to

avoid skewing the sorting process when single missing values are present. Ultimately, our sorting function remains exceptionally fast, even with vast amounts of observations.

Here's a glimpse of our "standard" sorting function:

```
const sortSimple = (a, b, sortBy, data) => {
  let valueA = data[a][sortBy]
  let valueB = data[b][sortBy]
  if (valueA === null || valueA === '') {
    return -1
  } else if (valueB === null || valueB === '') {
    return 1
  }

  if (valueA.toString().includes('(') && valueA.toString().includes('%')) {
    valueA = parseFloat(valueA.substr(0, valueA.indexOf('(') - 1).replace('<', ''))
  }
  if (valueB.toString().includes('(') && valueB.toString().includes('%')) {
    valueB = parseFloat(valueB.substr(0, valueB.indexOf('(') - 1).replace('<', ''))
  }

  if (valueA < valueB) {
    return -1
  }
  if (valueA > valueB) {
    return 1
  }
  return 0
}
```

FILTER OPTIMIZATIONS

Utilizing an index as our display list greatly enhances the efficiency of applying filters. By iterating through the index list and performing comparisons, we can effectively reduce the final display list to include only the observations that meet the filter criteria. When sorting a filtered list, the sorting algorithm is executed solely on this subset, further optimizing performance.

The filtering algorithm operates similarly to the sorting algorithm, checking various conditions and applying different methods based on the specific filter type. Our tool supports three primary types of filters: free-text filters, dropdown filters, and date filters.

- **Free-Text Filter:** This filter performs case-insensitive searches for occurrences of specified values. It allows for the use of range operators, such as "<" or ">," which can be applied to both numbers and text. Additionally, the filter can identify all missing values when the keyword *missing* is used.
- **Dropdown Filter:** This filter searches for all occurrences that match the selections made in the dropdown list exactly, ensuring that users can easily refine their results based on predefined categories.
- **Date Filter:** This filter can accommodate either a single date or a date range, allowing for flexible temporal queries.

By implementing these optimized filtering techniques, we ensure that users can swiftly navigate and analyze large datasets with precision, enhancing the overall user experience.

THE USE CASES & USERS

The primary users of the iListing and iTable application include Medical Writers, Reviewers, and Quality Checkers. Each of these user groups benefits significantly from the application's capabilities. The most valuable time saving part comes during the quality control (QC) process. The ability to browse and filter outputs facilitates a much smoother workflow, allowing users to efficiently cross-check results against each other, rather than navigating through numerous cumbersome PDF documents.

Figure 14: Using iListing instead of PDF documents for quality checks

For Medical Writers, the iListing and iTable application significantly streamlines the process of adding detailed information to the Clinical Study Report. They can effortlessly access comprehensive listing data related to specific subjects of interest, offering insights that extend beyond what is typically captured in narrative summaries. This enhanced accessibility not only allows them to verif and refine their summaries quickly by applying relevant filters, but it also empowers them to gain deeper insights from tables. By following links to detailed subjects in connected listings, they can explore related data more thoroughly, enriching the overall quality and accuracy of their reports.

While the current implementation primarily serves internal users, there is significant potential for external reviewers, such as regulatory authorities, to benefit from this tool. Such access would enable them to browse outputs more effectively. However, this would necessitate a higher level of application validation and the establishment of various security measures. Although Bayer is not currently pursuing this use case, it presents an exciting opportunity for cross-industry collaboration in the future.

Our use case currently displays the listing and tables section, as we produce and deliver this to Medical Writing and that is the content Medical Writers work with. It would also be possible to change from the listing to a dataset display. Unless we omit creating listings, using this tool to browse datasets for Medical Writers is not necessary, as listings present the datasets. The Statistical Analysts do have another powerful dataset display which provides much additional functionality, which we investigated too complex, irritating and powerful for the purposes of Medical Writers.

THE OPPORTUNITIES

The choice of technology for the iListing and iTable application is flexible. At Bayer, we currently utilize the React web application framework for the frontend; however, alternative technologies, including RShiny apps, could also effectively support this tool. For backend file generation, we rely on SAS to create metadata and content files in CSV and JSON formats. Thanks to our standardized, macro-driven programming approach, we can easily transform collected metadata to meet specific requirements with little effort.

It might not be sufficient to have the analysis results metadata (new industry standard) available as additional information is required. As we need to display the outputs, we do also need the content of this, ideally in a dataset format rather than relying solely on final rendered tables, which may vary in layout and might be difficult to read in. Additionally, maintaining clear linkages between tables and their corresponding listings is essential for full traceability. The R projects “cards” and “cardx” might be an ideal starting point.

Figure 15: R Packages to create/work with CDISC Analysis Results Data Sets

Implementing an Application Programming Interface (API) between the frontend and backend is highly recommended. While it is technically feasible to package final content files with the application, this would necessitate stringent security management due to the handling of clinical data, which falls under a different software classification. By utilizing an API and refraining from storing any data within the application, we can significantly reduce security risks and enhance overall application safety. However, this approach does mean that only users with appropriate access rights within the environment can review a study. Fortunately, the overarching security system ensures that access remains strictly authorized.

CONCLUSION

The iListing and iTable solution has proven to be highly beneficial within Bayer, addressing the limitations of traditional paper formats like PDF, which are often outdated and lack user-friendliness and performance. Interactive tools like these significantly enhance the user experience, allowing for quicker access to information and better information gathering and saves a lot of time.

Performance is particularly crucial in this context, as even the largest listings must remain easily browsable and responsive. The ability to interact with extensive displays seamlessly ensures that users can efficiently locate and analyze the information they need, ultimately leading to more informed decision-making and streamlined workflows.

To enable authorities to use such interactive visualizations, an industry collaboration would be extremely valuable in discussing the security setup and a standardized way to deliver interactive applications to authorities. Some initiatives have already started, as we see in the R Pilot Submission projects[5].

User satisfaction with the iTable and iListing solutions has been overwhelmingly positive. It is crucial to involve statistical analysts—who create the reports—in the development of these tools. Their insights and perspectives are essential for ensuring that the tools effectively meet practical needs. Importantly, the implementation of these solutions should not impose additional burdens on statistical analysts. At Bayer, our streamlined metadata collection and standardization processes when creating any kind of deliverables enable the generation of these outputs in under four working hours. This results in substantial time savings for Medical Writing and quality checks, while presenting a low burden for analysts to produce.

In addition to understanding the specific needs of various departmental groups, it is crucial to be aware of the tools and circumstances that each group utilizes. This comprehensive understanding allows us to derive optimal solutions tailored to their requirements. Continuous feedback is another vital component, enabling us to enhance the tool rapidly. For instance, during the initial planning for sorting, we overlooked the fact that many numeric columns were formatted as characters to ensure proper alignment in printed materials. Furthermore, special columns, such as “xx (xx%),” should ideally be sorted numerically. Based on user feedback, filtering comparison functions have been incorporated, including options for "greater than or equal to" and "less than or equal to".

Overall, the effectiveness of the solution has significantly improved thanks to the continuous feedback from users, ensuring it adapts to their evolving needs. To maximize the value of user insights, it is essential to establish a quick and easy feedback loop that encourages ongoing communication and engagement.

REFERENCES

[1] AD02: iListing – Get More Out of Listings, PHUSE presentation (https://www.lexjansen.com/phuse/2021/ad/PRE_AD02.pdf)

[2] ICH E3 Guideline – Structure and content of clinical study reports (https://www.ema.europa.eu/en/ich-e3-structure-content-clinical-study-reports-scientific-guideline)

[3] PHUSE Whitepaper - Data Listings in Clinical Study Reports (https://phuse.s3.eu-central-1.amazonaws.com/Deliverables/Safety+Analytics/WP061.pdf)

[4] A Macro to replace PROC REPORT!? – PHUSE paper (https://www.lexjansen.com/phuse/2012/ts/TS03.pdf)

[5] R Submissions working group – website (https://rconsortium.github.io/submissions-wg/)

SOURCE CODE LICENSE

CONTACT INFORMATION

Katja Glaß
Bayer AG
Müllerstraße 178
13353 Berlin, Germany
Katja.Glass@bayer.com