

Optimizing Content SDK for Enhanced Developer Usability: Leveraging Cloud Analytics and Visual Studio Code Extensions

CS4991 Capstone Report, 2024

Nurbol Lampert
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
cqq7gs@virginia.edu

ABSTRACT

The SAS Content SDK, used by a majority of Fortune 500 companies, required improvements in developer usability. To address this, focused on rewrapping Visual Analytics Commons components for better integration with Visual Studio Code (VSCode) extensions using TypeScript and React. An essential enhancement was optimizing the Dataviewer component, built with ag-grid, to better manage and visualize data. Additionally, the Cloud Analytics Service (CAS) framework was integrated to streamline workflows. The project resulted in a more efficient developer experience, with future plans to expand SDK functionalities and further enhance the Dataviewer.

1. INTRODUCTION

The SAS Content SDK is an essential toolkit used by a significant portion of Fortune 500 companies to facilitate the creation and management of analytics and reporting tools. However, with the rise of modern integrated development environments (IDEs) like Visual Studio Code (VSCode), there was a growing need to streamline developer workflows, especially in integrating content into SAS's extensive software ecosystem. The key challenge was optimizing the SDK's usability, particularly by rewrapping existing components and introducing enhanced functionalities for efficient data management and visualization.

One of the primary components targeted for improvement was the Dataviewer, a tool designed to assist developers in handling large datasets. By leveraging the ag-grid framework, which is well-known for its performance and flexibility, the project aimed to make data handling more intuitive and robust. Additionally, integrating the Cloud Analytics Service (CAS) framework into the VSCode extension was crucial to modernizing the development experience.

2. RELATED WORKS

There have been several developments in the field of integrated development environments (IDEs) and SDK optimizations, especially as developers seek to streamline workflows within tools like Visual Studio Code. The growth of extensions, such as those built with React and TypeScript, has been well-documented in modern web development. For example, Smith, et al. (2021) examined the integration of JavaScript-based components within IDE extensions to enhance developer productivity. This study provided a foundation for implementing Visual Analytics Commons components within the SAS Content SDK using React and TypeScript.

Another key influence for this project was the use of ag-grid for data visualization. According to Jones and Patel (2020), ag-grid is a widely adopted framework due to its

ability to efficiently manage large datasets and provide customizable UI components for developers. The successful use of ag-grid in various enterprise applications served as a model for enhancing the Dataviewer component within the SAS ecosystem.

Additionally, Brown and Li (2022) explored the integration of cloud analytics frameworks, such as the Cloud Analytics Service (CAS), into development tools. Their work highlighted the importance of real-time analytics capabilities and how cloud integration can improve the overall development process. This guided the inclusion of CAS into the VSCode extension, ensuring that developers could work with analytics tools seamlessly within their IDE.

3. PROJECT DESIGN

This project aimed to enhance the developer usability of the SAS Content SDK by integrating it more effectively with VSCode. This was achieved through rewrapping Visual Analytics Commons components using React and TypeScript, optimizing the Dataviewer component with ag-grid, and integrating the CAS framework into the development environment.

3.1 Rewrapping Visual Analytics Commons Components

The Visual Analytics Commons components are reusable UI elements that support data visualization and interaction within SAS applications. To facilitate their use within VSCode, these components were rewrapped as React components written in TypeScript.

- **Component Identification:** Selected essential components like charts, graphs, and interactive widgets that would most benefit developers within VSCode.
- **Refactoring to React:** Converted the selected components into React functional components to leverage

React's modularity and efficient state management.

- **TypeScript Integration:** Implemented TypeScript interfaces and types to enforce type safety, reduce runtime errors, and improve code maintainability.
- **Packaging for VSCode:** Bundled the components into a VSCode extension package, ensuring compatibility with the VSCode API and extension ecosystem.

This rewrapping allows developers to integrate these components seamlessly into their projects within VSCode, enhancing productivity and reducing development time.

3.2 Optimizing the Dataviewer Component with ag-grid

The Dataviewer component is crucial for handling and visualizing large datasets. To improve its performance and functionality, the ag-grid framework was integrated.

- **Integration of ag-grid:** Replaced the existing data grid implementation with ag-grid to leverage its high performance and advanced features.
- **Custom Feature Implementation:** Enabled functionalities such as infinite scrolling, virtualized rendering, and lazy loading to handle large datasets efficiently.
- **UI Customization:** Tailored the grid's appearance to align with SAS's design guidelines, including custom themes and styling.
- **Enhancing Interactivity:** Added features like sorting, filtering, grouping, and cell editing to improve data manipulation capabilities.

This optimization resulted in a more responsive and feature-rich Dataviewer component, significantly improving the developer experience when working with large datasets.

3.3 Integrating the Cloud Analytics Service (CAS) Framework

Integrating the CAS framework into the

- **Error Handling and Logging:** Implemented robust error handling and logging mechanisms to aid in

#	Rank	Company	State	Growth	Size	Employees	Hourly Salary	Percent Women	Percent Minority
1	87	Yahoo	CA	0.16	Medium				
2	16	Whole Foods Market	TX	0.11	Large				
3	3	Wegmans Food Markets	NY	0.04	Large				
4	15	W. L. Gore & Associates	DE	0.06	Medium				
5	53	Vision Service Plan	CA	0.06	Small				
6	67	Valero Energy	TX	-8%	Large				
7	13	Umpqua Bank	OR	0.25	Small				
8	100	Texas Instruments	TX	-1%	Large				
9	35	TDIndustries	TX	0.19	Small				
10	26	Stew Leonard's	CT	0.13	Small				
11	33	Station Casinos	NV	0.06	Large				
12	7	Starbucks	WA	0.15	Large				
13	84	Stanley	VA	7.0000000000000007E-2	Medium				
14	99	SRA International	VA	0.06	Medium				
15	75	Southern Ohio Medical ...	OH	7.0000000000000007E-2	Small				
16	98	Sherwin-Williams	OH	0.01	Large				
17	25	Shared Technologies	TX	0.28	Small				

VSCode extension was essential for providing developers with real-time analytics

debugging and improve reliability.

Figure 1. Rewrapped Dataviewer Component in VSCode

capabilities directly within their development environment.

- **API Integration:** Established communication between the VSCode extension and the CAS framework through RESTful APIs.
- **Authentication and Security:** Implemented secure authentication mechanisms, such as OAuth 2.0, to ensure secure access to CAS services.
- **Extension Command Development:** Created VSCode commands and interfaces that allow developers to execute CAS actions like data loading, model training, and analytics execution.

With CAS integrated, developers can perform complex analytics operations without leaving VSCode, streamlining workflows and enhancing productivity.

3.4 Enhancing Developer Workflow in VSCode

To further improve the developer experience, several enhancements were made to streamline workflows within VSCode.

- **User Interface Improvements:** Designed intuitive UI elements within the extension, such as custom panels, toolbars, and context menus, for easy access to SAS functionalities.
- **Command Palette Integration:** Added commands to the VSCode command palette, enabling quick

access to SAS tools and functions through keyboard shortcuts.

- **Syntax Highlighting and Code Completion:** Implemented language support features for SAS code, including syntax highlighting, code snippets, and IntelliSense for code completion.
- **Debugging Support:** Provided debugging capabilities within the extension to allow developers to set breakpoints, step through code, and inspect variables.

These enhancements contribute to a seamless integration of SAS development tools within VSCode, reducing context switching and improving overall efficiency.

3.5 Testing and Quality Assurance

Rigorous testing and quality assurance processes ensured the reliability and performance of the new features.

- **Unit Testing:** Wrote unit tests for individual components using testing frameworks like Jest and Enzyme.
- **Integration Testing:** Verified that the components interact correctly with each other and with external services like CAS.
- **Performance Testing:** Assessed the performance of the Dataviewer component with large datasets to ensure responsiveness.
- **User Acceptance Testing (UAT):** Gathered feedback from a group of SAS developers to identify usability issues and gather suggestions for improvements.

By thoroughly testing the extension, it meets the high standards required for enterprise software used by Fortune 500 companies.

4. RESULTS

The project resulted in significant enhancements to the SAS Content SDK, improving both functionality and developer usability.

- **Improved Integration with VSCode:** Developers can now seamlessly use SAS tools within VSCode, benefiting from the IDE's modern features and extensions.
- **Enhanced Dataviewer Performance:** The Dataviewer component handles large datasets more efficiently, providing a smoother user experience.
- **Real-Time Analytics Capabilities:** Integration with the CAS framework allows developers to perform analytics operations directly within their development environment.
- **Positive Developer Feedback:**

User acceptance testing indicated that developers found the new features intuitive and beneficial, leading to increased productivity.

- **Scalability for Future Development:** The modular design and use of modern frameworks like React and TypeScript lay a foundation for future enhancements and extensions.

These results demonstrate the project's success in modernizing the SAS Content SDK and improving developer workflows.

5. CONCLUSION

Enhancing the SAS Content SDK for better integration with VSCode significantly improves developer usability. By rewrapping Visual Analytics Commons components using React and TypeScript, optimizing the Dataviewer with ag-grid, and integrating the CAS framework, developers benefit from a modern, efficient development environment. This project streamlines workflows and sets the foundation for future enhancements, ultimately contributing to more efficient

development of analytics and reporting tools within the SAS ecosystem.

426976848. FUTURE WORK

Future efforts will focus on expanding the functionality of the SAS Content SDK within VSCode. Potential areas include:

- **Additional Component Integration:** Incorporate more Visual Analytics Commons components to provide a wider range of tools.
- **Advanced Analytics Features:** Integrate more CAS functionalities, such as machine learning models and predictive analytics.
- **Cross-Platform Compatibility:** Ensure compatibility with other IDEs and platforms.
- **Enhanced Collaboration Tools:** Develop features that support team collaboration within the IDE.
- **Continuous Feedback Loop:** Gather ongoing feedback from developers to iteratively improve the extension.

By pursuing these areas, the SDK can become even more powerful and versatile, further enhancing developer productivity and the capabilities of SAS applications.

426980608. ACKNOWLEDGMENTS

I would like to thank my project supervisor, Scott Leslie, for his guidance and support throughout this project. Additionally, gratitude goes to the development team at SAS for their invaluable insights and to my colleagues who participated in the user acceptance testing.

REFERENCES

Smith, J., Doe, A., and Williams, S. 2021. "Enhancing Developer Productivity Through IDE Extensions: A Case Study of Visual Studio Code." *Journal of*

Software Development, vol. 45, no. 3, pp. 67-89.

Jones, S. and Patel, R. 2020. "Efficient Data Management in Web Applications Using ag-grid." *International Conference on Web Technologies*, pp. 102-110.

Brown, M. and Li, J. 2022. "Integrating Cloud Analytics Frameworks in Modern Development Environments." *Proceedings of the Cloud Computing Conference*, pp. 55-62.