CASE STUDY

Building a Self-Service JDE Data Environment with QuickLaunch for Power BI

Haskell delivers Architecture, Engineering, Construction (AEC) and Consulting solutions to assure certainty of outcome for complex capital projects worldwide.

Haskell is a global, fully integrated, single source design-build and EPC firm with 1,300+ highly specialized, in-house design, construction and administrative professionals across industrial and commercial market sectors. With $1B+/- in annual revenue and 20+ office locations around the globe, Haskell is a trusted partner for global and emerging clients. Quick access to trusted data across all departments is critical to the organization's ability to operate efficiently and make data-driven business decisions.

Evolving With JD Edwards

JD Edwards has been in use at Haskell for over 20 years. While the underlying JDE system continued to meet their ERP needs over the years, their legacy BI and data management system did not.

Reports were static and business users relied on IT to write reports and create ad hoc queries. Users couldn't access the data needed for specific projects, nor did they have a way to easily bring together JDE and non-JDE data. Enterprise data did not reside in a data warehouse and tools were lacking overall.

The small Business Analytics team wanted to free up resources for more strategic projects—including in-depth data analysis—rather than spending all their time developing reports for the rest of the organization.

Leapfrogging Ahead With QuickLaunch and Power BI

After an evaluation process, the Haskell IT group selected Preferred Strategies QuickLaunch for Microsoft Power BI to create a self-service data environment for their hosted JDE system. Key functions include; General Ledger, Accounts Payable, Accounts Receivable, Job Cost, Purchasing, and HR/Payroll.

Working closely with the Preferred Strategies team, the core system was implemented in just weeks—versus the two years that similar projects had taken in the past—by leveraging QuickLaunch accelerators and pre-built data models, templates, and reports.

COMPANY SNAPSHOT

HASKELL

Company: Haskell

Industry: Architecture, Engineering and Construction

Challenge: Unify data in a data warehouse to enable faster reporting and analytics from JD Edwards and other corporate data sources

Solution: Implement Preferred Strategies QuickLaunch™ for Microsoft® Power BI and Enterprise Data Warehouse solution

Result: QuickLaunch has transformed how Haskell business users interact with data by enabling self-service BI with IT governance

"Throughout the entire process, Preferred Strategies has been a true partner to us for our data warehousing and reporting needs."

SEAN BROWNING, MANAGER OF BUSINESS ANALYTICS, HASKELL

Following the initial implementation, Haskell undertook to convert from their legacy reporting platform to Power BI. Most of that effort focused on moving reports to the QuickLaunch Tabular Model, rather than direct query from JD Edwards. "The Preferred Strategies team was very responsive to any issues we encountered along the way. Throughout the entire process, Preferred Strategies has been a true partner to us for our data warehousing and reporting needs," said Sean Browning, Manager of Business Analytics, Haskell.

"QuickLaunch has leap-frogged us ahead in terms of reporting and visualization." said Donald Horne, Director IT, Haskell. "We achieved in weeks what it took years to do at other organizations."

Self-Service Data Access Under IT Governance

Preferred Strategies makes it simple for IT to govern which users get access to specific datasets, ensuring a secure self-service environment. "In our old data environment, security was associated with the reporting layer, not the data layer," said Browning. "Now we give users access to specific data, not specific reports, so the data is secure at the data warehouse level." Another important aspect of the QuickLaunch solution is the ability for Haskell to govern specific calculated values such as Job Profit Margin or Labor Job Costed Hours. This level of governance ensures consistency in analytics and reporting across the organization.

Users previously relied on IT to query data for them. They would then manipulate the data offline in Excel everytime they needed insights. Now business users across the organization are able to access the data they need and easily create automated reports and dashboards—saving time, gaining insights, and making essential KPIs easily available to executives. And, importantly, users have access to only the data that they should have access to as governed by IT-centered data models—resulting in both a streamlined workload and peace of mind for the data team.

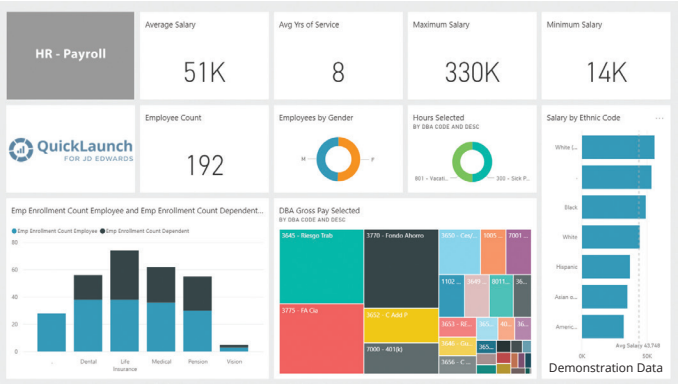
Agile Reporting Delivers Company-Wide Benefits

Executives and business users across the organization are realizing the benefits of the new self-service data environment, dubbed "Haskell BI". For example, one VP in central operations used to have to wait weeks for human resources to run specific reports. Now he has access to the data at his fingertips, when he needs it. Accounting has dramatically decreased the amount of time required to conduct monthly and year-end close. Business analysts can make changes to reports on their own, whenever needed, allowing them to operate in a more agile and responsive way.

"QuickLaunch has leap-frogged us ahead in terms of reporting and visualization. We achieved in weeks what it took years to do at other organizations."

DONALD HORNE, DIRECTOR IT, HASKELL

Manipulating spreadsheets was time-consuming and error-prone, and generally not the best use of resources. The new self-service data environment is saving everyone across the company time and giving executives quick access to critical operational data. As the COO, Jean-Paul Saenz, aptly said, "Haskell BI is amazing!"



This sample dashboard illustrates the type of report and visualizations that Haskell employees now have at their fingertips, giving business users the real-time data needed to make competitive decisions. The screenshot shown here contains sample data only, and does not represent actual Haskell information.