



Five Levels of Embedded BI: From Static Reports to Analytic Applications

Leveraging a BI platform with built-in multi-tenancy provides better tenant security and provisioning and increased branding opportunities. In addition, it creates one single, easy-to-manage system that supports multiple customers.

INTRODUCTION

The expanding role of data promises smarter operational applications that manage and automate better processes. These new, intelligent, analytic applications are transforming how other applications can use information to drive improved business performance and competitive advantage.

Most organizations derive value from their data through reports, dashboards, and analytic visualizations using stand-alone business intelligence (BI) and data warehouse tools. Yet only a small percentage of people actually use this style of BI due to user interface complexity, lack of current information, and data inaccuracy issues.

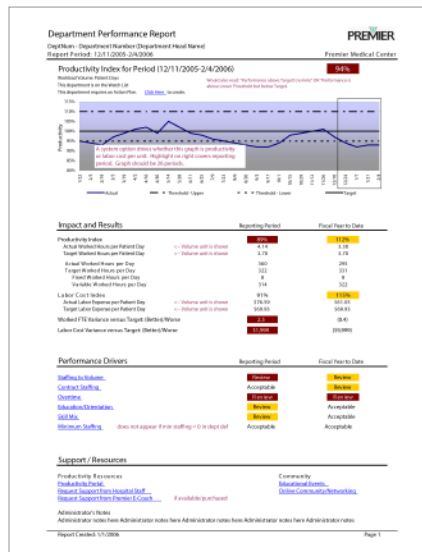
Operational applications already in use offer an opportunity to evolve from using static operational data exclusively to including interactive analytics that foster more powerful decision-making.

Using a fictitious business, Acme Fresh Delivery Foods, and use cases involving its IMS2 inventory management system, this ebook steps through five levels of increasing engagement and user value that can be achieved by embedding BI in to any application:

- Level 1: Static reporting using an embedded reporting library
- Level 2: Managed reporting with simple interactivity, scheduling, security, and distribution using a reporting server
- Level 3: Highly interactive reports and dashboards using a reporting server
- Level 4: Self-service ad hoc reports using a BI server
- Level 5: Self-service data exploration against a data mart using a BI server

The IMS2 inventory management system organizes product information, inventory units, location data, movement history, and bill of material information. It is used by a variety of personnel, including inventory clerks, warehouse managers, retail agents, shop floor managers, and executives.

LEVEL 1 STATIC REPORTING



Example:

Joel is an inventory clerk at Acme Fresh Delivery Foods retail store in Phoenix. He wants to know how many batteries his distributor provided last month and how many are planned for the rest of the year, presented by week in a table and bar chart. He wants to run the report against the latest, most up-to-the-minute data. The application administrator, Kelly, implemented the report as a menu option in the IMS2 application; it is output in PDF format.

End Users Just Want a Report

When users want to view data from their operational application, it's typically provided through a static report. Some reports are formatted and ready to print,

while others are available to download as a Microsoft Excel worksheet. The reports provide a static view in time, typically coming from the live operational database within the application.

Application Developers Want an Easy Architectural Solution

Application developers have several options when determining how best to provide a static report for their operational application. They can build their own tool, embed an open source reporting engine, or buy a commercial solution. They can look at several architectural options to seamlessly enable reporting without disrupting the architecture of the application. Limiting the impact on the application architecture usually results in a limited reporting solution for end users.

Considerations

An embedded reporting library allows a user to either run a report on demand or request the application run the report in the background and store it. The library's pre-built reports are designed by a developer who defines the report layout and export formats (PDF, XLS, HTML). Each report must be designed to prevent significant performance hits to the application. The reporting library is often built inside the application in the same programming language. Additional code is typically written to manage access, security, scheduling, and report storage.

Embedded BI Components to Look For

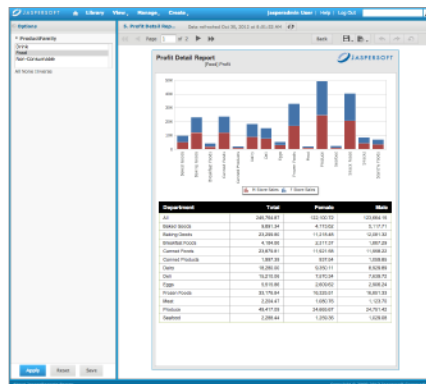
- Reporting library for services such as report compilation, layout, and export format
- Desktop-based report designer for report creation

Limitations

- **Static point-in-time information:** Embedded reports typically present a limited timeframe of information based on a lack of historical data stored within the application's database. This means reports can't show trends. Additionally, the report is usually static and doesn't allow users to drill from summary data to underlying details to gain further insight.
- **Unfulfilled Report Requests:** Each embedded report is designed by an application developer, which means assumptions are made about how and what to present to the end user. New report requests can only be addressed by the developer, which means custom report requests go unfulfilled or are delivered according to the developer's availability.
- **Impact to Application Performance:** If the application does not provide report scheduling and a report repository, the same report query could be run repeatedly by different users at the same time, impacting the application's performance. Additionally, the compilation and layout of each report takes computational resources.
- **Developer Effort:** With each new report, the developer must consider the performance impact on the operational application and any security implications. As end users request new views to the application data, the developer must balance new report requests against new feature enhancements.

LEVEL 2

MANAGED INTERACTIVE REPORTS



Example:

Janet is a manager for Acme Fresh Delivery Foods' western regional stores. She wants to know how many total shipments were delivered last month per distributor, how many items are in stock, and how many will be shipped next month for each store in her region. She then wants each report to run nightly and be sent via email to each store manager. Paul, the manager of the Santa Fe store, has requested an inventory aging field for his store's report. Kelly, still responsible for maintaining and extending

IMS2, knows the current level of embedding isn't up to the task.

Managed Application Reporting for Better Business Performance

The success of an operational application often leads to new business requirements and new challenges for the group managing the application. Managed reports help drive business performance with information sharing and pre-defined key performance indicators (KPIs).

Kelly knows she can't support Janet's and Paul's requirements with an out-of-the-box Level 1 reporting solution. She would need to build a scheduler and report distribution service as a customization. Additionally, Kelly needs to build a handful of new custom reports, either by extracting the data from the application and building reports with another tool, or by requesting an enhancement of the IMS2 vendor. As is often the case, she is also responsible for other tasks, such as maintaining other applications, which results in delays to the fulfillment of Paul's and Janet's requests.

Considerations

Automated delivery of pre-built and custom reports is an expected feature in most enterprise operational applications. Application end-user engagement and efficiency increases when the data needed is easily available and used for day-to-day planning and decision-making. If the application gets in the way of progress, the possibility of new solutions or competitors rises. Level 2 reporting requirements can be met by complementing the reporting library with a report scheduler, reporting repository, report distribution service, role-based security, and a report designer for new report requests.

The application can provide these services natively with a fair amount of development effort, or the services can be provided by an integrated, packaged BI solution. With these services enabled, the embedded BI solution can provide more reports to more users.

Embedded BI Components to Look For

- BI server for data security, metadata layer, dashboard framework, and reporting services (scheduling, distribution, and organization)
- Desktop-based report designer for highly complex reports
- Customizable UI framework for seamless branding and integration with the operational application

Limitations

A reporting server can improve information delivery and engagement within a company through scheduled, interactive reports. It can also improve operational application performance by offloading report compilation to a separate reporting server. But some limitations restrict the application's ability to address the changing dynamics of a business:

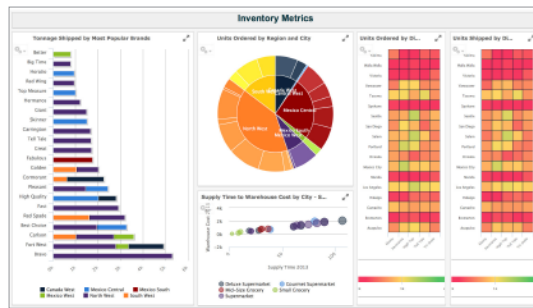
- **Simplistic Data Security:** A simple report object model such as a Level 2 reporting solution does not provide query (SQL) level security. A reporting repository does not inspect report queries, so the report developer or administrator must build the security attributes within the report and assign these attributes to the report object manually.
- **Limited Availability of Custom Reports:** New custom reports require the expertise of a professional report developer due to the complexity of the underlying data source, security models, and report formatting requirements. Most organizations don't have professional report developers, so the flow of new reports to business users can be slow.
- **No Dashboard Reports:** Once the basic reporting requirements are met, the application's users soon request more complex features, such as executive views. This type of reporting is enabled through dashboards, which provide at-a-glance summaries of critical performance indicators. Most dashboards support the ability to drill down from a summary view into underlying details for further inspection.

"A business intelligence dashboard is a data visualization tool that monitors critical business processes and activities using metrics of business performance that trigger alerts when potential problems arise. They analyze the root cause of problems by exploring relevant and timely information from multiple perspectives and at various levels of detail. They also manage people and processes to improve decisions, optimize performance and steer the organization in the right direction."

— Performance Dashboards: Measuring, Monitoring, and Managing Your Business, Wayne Eckerson

LEVEL 3

HIGHLY INTERACTIVE REPORTS AND DASHBOARDS



Example:

Steve is the head of inventory operations for all Acme Fresh Delivery Foods stores. He wants to see up-to-the minute metrics for key inventory and point-of-sale (POS) performance indicators, presented in a single easy-to-read dashboard. He wants his reports to be interactive and provide drill down to detailed data,

filters, and easy-to-see flags associated with metric outliers. These metrics from IMS2 are combined with another POS application to provide a central business performance dashboard. Kelly realizes once again that the current solution isn't going to be adequate for this new level of service. She will have to take it to the next level to really wow Steve.

ADDRESSING NEW USER PROFILES WITH APPLICATION DASHBOARDS

While the need for reports mentioned in the first two phases provides detailed application information for daily tactical decisions by line managers and shop floor users, it may not suit an executive or line of business manager. Executives usually don't use operational applications daily, but do rely on weekly, daily, or hourly snapshots of business performance. This view of information is usually presented in an easy to consume interactive dashboard.

A performance dashboard measures short- and long-term trends with quick access to underlying details to help managers react tactically or strategically to their business needs. Dashboards can also feed from several different application sources to present a holistic view of the business.

Considerations

Level 1 and 2 reporting can't effectively present an interactive dashboard. Dashboards are a collection of small reports, called dashlets, that are assembled on a single canvas, often with interactive controls that allow the user to change the view of data by time, location, or other parameter. The framework to control these integrated dashlets requires an orchestration layer typically managed by a metadata layer within the reporting server environment. To improve utility and draw decision-makers to the dashboard, the overall layout and design requires compelling elements like interactive charts and drill-down-enabled dashlets so users can quickly explore and better understand the business activity underlying the dashboard's information.

Embedded BI Components to Look For

- BI server for data security, metadata layer, dashboard framework, and reporting services (scheduling, distribution, organization)
- Desktop based report designer for highly complex reports or dashlets
- Customizable UI framework for seamless branding and integration with the application

Limitations

Level 3 embedded BI allows new user profiles to use the data stored within the IMS2 application. Dashboards can drive new strategies, improved decisions, and better planning. However, Level 3 does not alleviate the continuing requests for custom reports by other types of users. Success at this level often surfaces these new requirements:

- **Lack of Custom Reports:** New custom reports require the expertise of a professional report developer due to the complexity of the underlying data source, security models, and report formatting requirements. Most organizations have few to no professional report developers, a situation that can impede response to business users' requests for new reports. The ideal solution is a report design tool that less technical users can use to build their own reports without relying on IT or skilled report developers.
- **Lack of Data Exploration and Analysis:** Dashboards help visualize complex processes in easy-to-understand terms. However, a compelling dashboard inspires user curiosity and the desire to learn more about their data and why a metric is under- or over-performing. The answers to these questions are often beyond the scope of the dashboard and its underlying detailed reports. Supporting these deeper, spur-of-the-moment questions requires exposing the data through an interface that end users can interact with. Data exploration often requires queries comparing various products, locations, and times. Users want to look at the data in different dimensions to find trends or outliers.

LEVEL 4

SELF-SERVICE REPORTING FOR OPERATIONAL APPLICATIONS

QUESTIONS TO CONSIDER:

- 1 Do your customers have multiple stakeholders with sophisticated report requests?
- 2 Do these individuals want to create their own reports?
- 3 Do these individuals have enough knowledge and desire to build their own reports without relying too heavily on IT or other technical resources?

Product Name	Store Type	Sales
Product A	Store 1	\$5,121
Product B	Store 1	\$4,547
Product C	Store 1	\$4,089
Product D	Store 1	\$3,531
Product E	Store 1	\$2,973
Product F	Store 1	\$2,415
Product G	Store 1	\$1,857
Product H	Store 1	\$1,299
Product I	Store 1	\$741
Product J	Store 1	\$283
Product A	Store 2	\$5,121
Product B	Store 2	\$4,547
Product C	Store 2	\$4,089
Product D	Store 2	\$3,531
Product E	Store 2	\$2,973
Product F	Store 2	\$2,415
Product G	Store 2	\$1,857
Product H	Store 2	\$1,299
Product I	Store 2	\$741
Product J	Store 2	\$283

Example:

Paul is an inventory planner for Acme Fresh Delivery Foods' corporate office. He wants to build his own custom reports for his wholesale distributors. These distributors change every couple of months so the reports change frequently. Some reports are driven by product line, while others are based on specific store

metrics. The current static reports provided by the IMS2 system don't provide the flexibility to generate these reports, forcing Paul to request new reports from Kelly every month or so. Kelly is getting similar requests for custom reports from other employees and can't keep up with demand.

Considerations

When Acme Fresh Delivery Foods knowledge workers want their own custom reports that are not part of the packaged operational application, there are two options:

Option 1:

Provide Paul or Kelly direct database access to the operational application schema, allowing them to pull a CSV dump of the data to their local machine, or give them a report design tool for more advanced requirements.

The downside to this option are the training costs and resources required to teach non-technical users how to use powerful, full-featured report design tools to build each report manually, along with the associated lag time. There is added risk to application performance due to the uncontrolled queries that this approach introduces to the environment. For larger implementations, there is added security risk in providing general access to the database, as well as additional training costs for the larger group.

CONSIDERATION FOR PRESENTING YOUR DATA

Operational data to be used for reporting can be very messy. Often times it is spread around many tables with very complex joins that are only understood by developers. If the aim is to provide self-service reporting capabilities, a metadata layer should be used to hide the complexity of the underlying data and present a clear representation for the non-technical end users.

Option 2:

Provide Paul an easy-to-use report design environment for his ad hoc reporting and analysis needs. Kelly can define an easy-to-understand semantic layer that sits on top of the application database. This layer allows non-technical users to understand the column names and the data while providing a security access model for the underlying database. The metadata can be designed against the operational database or a specialized data mart.

With these elements, and a graphical drag-and-drop report designer, knowledge workers like Paul can now build their own reports on demand, without requesting support from Kelly. In addition, an in-memory query cache can further reduce the impact to the application database, enabling custom reports that don't impact application performance. When integrating a self-service BI environment with another application, there's a sizeable risk of losing the consistent look and feel of the original application. Select BI tools able to change their UI to blend seamlessly with your application.

Limitations

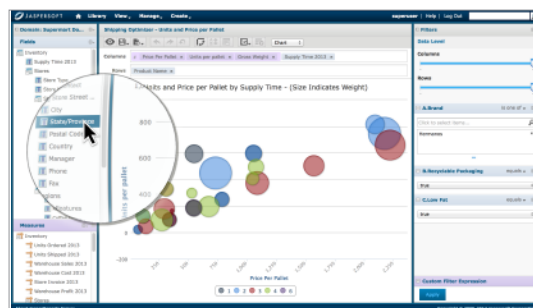
- Consistency of Application Look and Feel: The look and feel of an application is important for software vendors and end users. Application developers who integrate an existing off-the-shelf BI platform should look for tools that allow easy customization that provides a consistent look and feel.
- By definition, self-service reporting tools should be easy to use for a non-technical user like Paul. They should require little training to build a self-service report, the reality is that many of the BI platforms offered today do not meet this requirement.
- If the application database contains lots of data, the amount of data stored within an in-memory engine is limited to the amount of memory available on the computer hosting the BI server. If the application database is large, it's often better to configure a specialized analytic application database that leverages columnar storage and data compression.

Embedded BI Components to Look For

- BI server for data security, metadata layer, web-based easy-to-use report designer, dashboard framework, reporting services (scheduling, distribution, organization)
- Desktop-based report designer for highly formatted reports
- Customizable UI framework for seamless branding and integration with the operational application

LEVEL 5

SELF-SERVICE DATA EXPLORATION FOR DEEPER INSIGHT



Example:

Susan is a product line manager for Acme Fresh Delivery Foods' headquarters. She wants to understand why margin costs are dropping for deliveries from the northwest stores compared to those in the Southwest. She would like to explore a variety of data and dimensions including wholesale unit costs, retail price, inventory aging, shipping costs, and product promotion data.

She plans to compare products by store and date (month and year), including pulling in data from Acme Fresh Delivery Foods' product promotions system. She'll be looking for trends over time, possibly caused by seasonal store patterns and additional expenses on the supply side. To address this new need, Kelly considers these options:

- Develop custom reports that collect the data Susan needs
- Give Susan data exploration tools that let her intuitively find the data she needs

Considerations

One way Kelly can get answers to Susan's questions is to create several more custom reports. However, it's time consuming to manually develop, execute, and review each report. It's also inefficient to access the data this way, as it can degrade the performance of the operational application itself.

Different options can help solve this problem. A pre-built view into the underlying data that is structured for the sole purpose of analysis lets data analysts quickly inspect large datasets and perform complex queries over periods of time, which is not easily done with a transactional database.

Business analytics through data exploration can be a source of competitive advantage for a business, and including it into operational applications like IMS2 can mark a strategic move for Acme Fresh Delivery Foods. While self-service reports allow users to create their own tabular reports, data exploration allows aggregations across multiple dimensions (for example, average sales by product, region, and time period). The visualizations created are usually charts or crosstabs and allow easy comparative and time-based analysis (for example, drill-down, slice-and-dice, pivot, and filter).

Embedded BI Components to Look For

- Data integration tools for extracting, transforming, and loading data from the operational application database to a data warehouse or data mart
- Data warehouse or data mart for efficient processing of data for analytic purposes
- Either an in-memory engine, push-down aggregations, or a traditional OLAP engine, for performance and analytic processing
- BI server for data security, metadata layer, data visualization, web-based easy-to-use report designer, dashboard framework, and reporting services
- Desktop-based report designer for highly formatted reports
- Customizable UI framework for seamless branding and integration with the operational application

Limitations

While providing greater access to data presents a host of benefits to organizations, including insight into new revenue categories, improvements in operational processes, competitive advantages, and more, the cost to the application developer and the application administrator can be high. A few specific challenges for an embedded analytics solution are:

- **Architectural Complexity:** To properly deliver responsive and powerful advanced analytics, the environment must include additional services, including a database for analytic queries, data integration software, and metadata for aggregation definitions.
- **Application Maintenance:** With added software comes added maintenance for administrators. A common request by data analysts is to provide a new view into the data warehouse that requires a new data integration job and new query definitions.

CONCLUSION

There are a number of options available when electing to enhance an application with Business Intelligence. It's important to understand the benefits and user requirements associated with each level of capability. Acme Fresh Delivery Foods progressed through all five levels of embedded BI. Recognizing the value of the data within the IMS2 application, Kelly can now give Steve, Janet, Paul, and Susan the data and tools they need to make decisions when and how they need to. Each level of BI brought the Acme Fresh Delivery Foods team closer to realizing the value of their data and delivering a clearer understanding of their business. Embedded BI can make your applications and customers more knowledgeable—and it can give you the opportunity to deliver greater value and end-user engagement for operational applications.

Once you determine how to proceed, consider whether to build your own BI framework or buy a packaged solution. Consider these questions:

- 1 Do you have the time and resources to build and maintain an embedded BI solution for your application?
- 2 Do you have the expertise in BI to address emerging trends such as mobile and social capabilities?
- 3 Do you want your embedded BI to be delivered in a seamless unified user interface?
- 4 Will your application be delivered as a SaaS and/or as on premise?

LEARN MORE

TIBCO Jaspersoft® empowers millions of people every day to make faster decisions by bringing them timely, actionable data inside their apps and business processes. Jaspersoft is an embeddable, cost-effective reporting and analytics platform that allows anyone to quickly get the answers they need. It scales architecturally and economically to reach everyone. Thanks to a community that is hundreds of thousands strong, Jaspersoft commercial open source software has been downloaded millions of times and used to create the intelligence inside hundreds of thousands of apps and business processes. For more information, visit <http://www.jaspersoft.com> and <http://community.jaspersoft.com>.



Global Headquarters
3307 Hillview Avenue
Palo Alto, CA 94304
+1 650-846-1000 TEL
+1 800-420-8450
+1 650-846-1005 FAX
www.tibco.com

TIBCO Software empowers executives, developers, and business users with Fast Data solutions that make the right data available in real time for faster answers, better decisions, and smarter action. Over the past 15 years, thousands of businesses across the globe have relied on TIBCO technology to integrate their applications and ecosystems, analyze their data, and create real-time solutions. Learn how TIBCO turns data—big or small—into differentiation at www.tibco.com.

©2015, TIBCO Software Inc. All rights reserved. TIBCO, the TIBCO logo, TIBCO Software, and Jaspersoft are trademarks or registered trademarks of TIBCO Software Inc. or its subsidiaries in the United States and/or other countries. All other product and company names and marks in this document are the property of their respective owners and mentioned for identification purposes only.
 09/21/15