



7 Reporting Essentials for ISVs

Reporting Requirements for ISVs

Overview

All applications have some basic functionality in common, whether they are created by an IT department for internal use or are purchased from an Independent Software Vendor (ISV). Applications always have some way to collect data and make it available, to both aid in business decision making and report on the data collected.

Reporting requirements for ISVs are similar to reporting requirements for IT built applications, but with additional complicating factors. Usually an ISV will want the 3rd party tools that they include in their product to be invisible to the customer. They want the customer to feel the ISV is providing the complete solution, and they do not want to depend on 3rd party installation, licensing, support, documentation and training.

Background Information

From the 1970s early report writers, to the current advanced Business Intelligence (BI) platforms, reporting issues have remained consistent. There has been a persistent need for a way to enable non-programmers to write their own self-service reports. The traditional waterfall model to gather requirements, design, code, test and publish reports written in procedural programming languages does not work well for dynamic reporting requirements. There needs to be a means of providing the information that users need in a timely and efficient manner, so that decisions can be made — which is, of course, the primary function of the application. Unfortunately, report writing is often seen as an activity deserving of little IT and ISV resources. Reports are often written after the main application is complete, and are frequently poorly designed and difficult to maintain. Often they only support one format such as HTML or PDF, and do not provide for any self-service use. These reports tend to be a constant drain on engineering, as adding extra features and reports commonly take weeks, or even months. By the time the additions are finished the reports are often obsolete, and more changes are requested.

This problem is compounded for ISVs, as they receive requests for changes from their customers. ISV's must either prepare estimates and charge for changes, or include report changes as part of their support contract. In any case, the engineers find they are spending much of their time doing custom work for existing customers. This generates lower return when compared to selling new software, which provides a noticeably higher profit margin.

For ISVs it is even more important to choose the best BI tool for their customers than it is for an IT department, which has only a single set of users. Every time an ISV acquires a new customer the number of requests for changes to existing and new reports increases, so the reporting work load grows alongside the ISVs success. Further, new customers want reports to

follow their style guidelines and corporate branding, a stipulation IT departments do not need to be as concerned about.

Rapid Report Development

ISVs must provide reporting options to customers that are easy to write and create with little training or effort. They also need to be able to quickly create reports if the end customer can not build it with self service. Each customer has their own requirements for export format, such as RIA AJAX, HTML, PDF, Excel, Printed, Word, XML, etc. Each customer also has their own requirements for page numbering, fonts, colors, images, logos, copyright notices, etc. The ISVs application must be easily customizable, so each customer feels they are getting a solution designed just for them, and at the cost of an off the shelf product.

Seamless Integration

The reporting tool should be able to easily be private labeled to fully integrate with the ISV application. The customer should never be aware that reporting is not an integrated part of the application. The user interface should be fully incorporated with the application, as should all fonts, colors, messages, help, and even the installation directory structure. No distinct 3rd party product should be visibly installed on the customer's systems.

Installation and Licensing

The BI tool needs to be seamlessly installed and licensed as part of the ISVs application, rather than with a separate installation and licensing step. Adding a 3rd party license step for several different tools, such as a DBMS and BI tools, reduces the value of the ISVs application as well as making it more difficult to support. The ISV needs to make sure that the entire application, including all 3rd party tools, looks and feels like a custom application and is managed with a single software license. With some commercial tools it is very difficult to enforce named user licenses, which puts the ISV at risk. Other licenses, such as GPL open source licenses, put undo limitations on selling and distributing the software.

Cross Platform Support

An ISV should also provide cross platform support. The BI tool must run on all potential platforms customers may want to use. These platforms can range from mainframes such as IBM or Unisys, mid-size computers such as AS-400s, Sun and HP Unix systems and a wide variety of Linux and Windows based computers. The customer may even want to run it on Macintosh servers. Only BI tools written entirely in Java, and therefore able to run in all application server environments, can provide this cross-platform support.

Java-based BI tools can achieve cross-platform support without requiring the customer to purchase additional non-standard hardware just for reporting, or call for the ISV to build separate install packages for each platform. Both of these options increase the cost of the ISV's solution and make it much more difficult to sell and support.

End User Self-Service

As well as creating pre-defined reports which can easily be customized by the customer, the ISV should provide self-service reporting. Self-service reporting enables customers to easily create new reports, and to change existing reports based on styles and templates that match the application. Without needing to learn SQL or how to best access the data, the customer needs to be able to easily create new reports using normal tools such as a browser interface and wizards that they already know how to use.

Customized for each use?

Each end user should also be able to create their own private versions of reports with the grouping, sorting and filtering they need for their job without intervention from the ISV or the IT department. Also, each user should be able to set up their personal profile so they see reports in the language, styles, formats and other options based on their user login.

Flexible Software Development Kit (SDK)

The ISV also may provide an API so that the customer can add additional capability to the application. The BI tool also needs to provide an SDK so that the entire feature set of the application is available to the customer. There are many different IDEs the customer may want to use. The SDK provided by the BI tool vendor should be able to work in any of these IDE environments: such as Oracle JDeveloper, Sun/Oracle NetBeans, IntelliJ IDEA, IBM WebSphere Studio, Embarcadero JBuilder and Eclipse. Many BI tools support only a single IDE, such as a plug-in to Eclipse, which may not be the IDE that the customer's engineers know and is the corporate standard.

JReport Solution

Now that we know what ISVs need, how does JReport address these requirements?

JReport is a 100% Java commercial BI tool which is composed of 3 primary products: JReport Designer, JReport Server Live, and JDashboard.

Rapid Report Development

JReport Designer is a Java Swing based WYSIWYG design tool. It allows an ISV to create a catalog to support reusable data sources, queries, business views, report cubes, formulas and security policies. The objects in the catalog support any number of report set templates, each of which can contain any number of reports and sub-reports. JReport is the only product which allows you to place any number of reports into a single template file, which makes managing the reports much easier for the ISV. By adding the resources to the catalog and not directly to the report templates it is very easy to target the catalog and report templates to the end customer's physical database URLs. This means styles and logos can be adjusted without making any changes to the report templates themselves. This makes it very easy to provide reports which look customized to the customer, but which are actually identical for every ISV customer; with the exception of the targeted external property files, images and style sheets.

Seamless Integration

Most ISVs want the BI tool user interface (UI) to seamlessly merge with the application's UI. Some of the open source tools claim that you need to use an open source product to do this. What they don't say, however, is that if you actually do download and modify the source code, they will not provide support. Furthermore, the open source code is not organized in a way that allows for easy style and interface changes, because all of the source code is available. Commercial tools, on the other hand, do not provide the source code and so can not normally be seamlessly integrated. These tools are generally going to look different from the ISV's application and the users will need to learn a whole new set of commands to get the BI tool to work.

JReport sets a balance between these two extremes. The JReport UI is 100% JSP, JavaScript and property files so the entire UI can be easily modified by the ISV to be a seamless interface. The end user will not be aware when he leaves the application and enters the JReport UI. In addition, doing these modifications are not only supported but highly encouraged. Our training classes demonstrate how end users can easily create these alterations.

Installation and Licensing

Most BI tools require that the product be installed on the customer's computer with full installation, and the BI vendor's license control and visibility need to be menu items and/or directories that the customer must use. JReport can be distributed by the ISV as a single WAR file with no installation and no licensing other than that of the ISV application. This is one of the reasons we find companies like IBM with their own BI product still embed JReport as their embedded BI tool because their own tools can not be seamlessly embedded, installed and licensed as part of their ISV application.

100% Pure Java Cross Platform

JReport is 100% Pure Java and runs on any application server or as a standalone BI service. Most BI tool vendors now provide a Java interface, but to run the full server and get all the features for security, performance and clustering, you need to use an OS specific installation for a limited number of platforms. This may limit the ISV's options when selling the software, and they may lose sales to vendors that are 100% platform agnostic. Don't let the 3rd party tools dictate what platforms you can support.

Once JReport is part of your application, installed, licensed, and seamlessly integrated on all platforms, you must address the self-service aspect, so the customer does not need to go back to the ISV for every slight modification. Without self-service, customers must either pay to make changes or take valuable ISV resources to make changes for free.

JReport Web Self-Service

JReport supports self-service, by allowing the report designer to create a meta-data layer on a query for an end user view of the available data for a report. Report Cubes can be created to match report templates and provide a resource view for the end user to modify the report at run time. Based on user profiles, advanced users can hide fields, remove fields, drag new fields onto the report, change the width and format of fields, change groups, change filters, change sorting and change styles. In other words, almost everything on the report can be modified at run time if they are given permission, so most small changes can be made by the customer. Even individual users can modify a report template and keep a copy in their own "My Reports" area so they have their private view of the report, without changing it for anyone else.

Report Cubes can also be used so that end users can create their own ad hoc reports directly from the report cubes defined in the catalog for an application. Without needing to know anything about the underlying data source, be it SQL, stored procedures, views, imported SQL statements from other applications, XML files, Java Objects or one of many other types of data sources, the user can simply create a new report component. These components include banded reports, tables, crosstabs or charts, and the

user can simply drag and drop data fields, formulas and summaries. Properties are available to modify to change the presentation to as much detail as desired with page numbers, report date and time, images and parameters.

JReport User Profiles

JReport allows the administrator to set up default profiles for each user or group of users so without modifying the report templates, the user will see the reports based on the CSS styles, colors, fonts, export type and other properties specifically for that user. This can save the ISV a lot of money and time compared to manually changing templates which then have to be maintained separately for each customer. The JReport administrator can also set up the system so based on the JReport user id or group a different user and password and even DBMS driver and URL are passed to the DBMS. Compared to other systems where connection information is stored in the template, this makes it much easier to maintain templates for 100s of customers without modifying the template itself.

JReport Software Development Kit

JReport also provides a complete SDK. Everything that can be done using JReport Designer and JReport Server Live can be done by making API calls directly from the application. In addition, the SDK is IDE agnostic, the ISV and their customer can use any IDE that they use internally. This is much more flexible than tools which require you to use an Eclipse plug-in; however, if you prefer to use an Eclipse plug-in for development and template design, it is also available as a free upgrade to JReport Designer.

Case Study

Passkey is a leading online group reservation system provider. Reporting is vital to their business. Not only do their customers rely on reports to perform critical business functions around managing group reservations, but Passkey uses reports internally to analyze each of their accounts.

Initially, Passkey implemented an in-house solution which suited their needs for a time. As their business grew, their home-grown solution became difficult to maintain. The increasing demand for new reports and features meant a longer time to production. There was also an escalating need for new reports and new ways to present them.

Passkey started an evaluation of several BI tools which could potentially solve their problems. Upon completion of the evaluation phase, Passkey chose JReport as the solution to help them fulfill their customers' needs. Besides being feature rich and user friendly, JReport's simple integration and superior security model put it ahead of its competitors.

With JReport, Passkey is, for the first time, able to rapidly deploy new report widgets, drill down capabilities and other BI in immediate response to customer demand/needs. As the economy shifts different measures become more important, and Passkey will be able to deploy those measures while they are still relevant instead of months down the road. The following is an example of one of their dashboards.



More Information

[Schedule a Free Evaluation to Review Your Reporting Needs as an ISV](#)
[Request a Live Demo to See How You Can Incorporate JReport into Your Solution](#)

[Test Drive JReport](#)

Contact Us

Tel:	+1 240-477-1000
Fax:	+1 240-465-0355
Website:	<u>http://www.jinfonet.com</u>
General Info:	<u>info@jinfonet.com</u>
Sales:	<u>sales@jinfonet.com</u>
Support:	<u>support@jinfonet.com</u>
Marketing:	<u>marketing@jinfonet.com</u>

About Jinfonet

Jinfonet Software is a company committed to delivering flexible, timely, and actionable information to all users across an enterprise via advanced visualization. Headquartered in Rockville, Maryland in the heart of the I-270 Technology Corridor, and equipped with a team of more than 160 expert Java developers, Jinfonet is the provider of the leading embedded Java reporting solution. Founded in 1998, and experiencing year-on-year growth since, Jinfonet is currently in its 11th release cycle of JReport.

About JReport

The JReport Product Suite is comprised of JReport Designer, JReport Server Live, and JDashboard. This comprehensive reporting software is the leading embedded reporting solution offering intuitive reporting from directly within existing applications. Featuring 100% Java architecture, The JReport Product Suite reaches millions of end users worldwide on a daily basis via the more than 25,000 downloads completed since the product's inception.