Foreword

Application development tools and technology have come a long way since the late 1970s, when I took my first job out of college in Hewlett-Packard Company's IT (Information Technology) department. Of course, IT was not the term we used to refer to the discipline back then; our preferred acronym was EDP (Electronic Data Processing).

And maybe that difference between simply "processing" data and delivering "information" was reflected in our development tools. We worked on TTY terminals connected to 16-bit mini-computers over 2400 baud lines. We used simple line editors to make changes to our COBOL programs, and we kept our application data in non-relational hierarchical databases. Debugging was COBOL WRITE statements, and source code control was keeping full copies of every version on tape or in separate directories.

Reports for our applications were typically afterthoughts, and they were done by hand in the same technology we used to develop the base application, i.e., COBOL. We designed them—when we did design—by laying them out in pencil on the report design pads that IBM had developed for RPG and COBOL programmers. Because we created them without much forethought, and because junior programmers like me often got the assignment of coding them, our users often found them inadequate, and the cost of making changes to accommodate their true requirements was high.

But while today's application developer may scratch his or her head in wonder at the primitive tools and technologies we employed in building our base applications in the late 1970s, he or she may not find my description of our approach to report development so very unfamiliar.

JSP = COBOL and Banded Report Writers = Report Design Pads

The majority of Java developers still hand-code reports for their applications using JavaServer Page (JSP) technology. This is analogous to our approach of hand-coding them in COBOL and has all the same downsides: high development cost, low user satisfaction, and inflexible, high-cost maintenance.

A minority of Java developers do use tools to develop reports; however, almost all of these tools—be they commercial or open source—are what's known as

"banded report writers," and they support a design metaphor that has essentially evolved from the old IBM report pads. Each section in the report writer—header, detail, footer—corresponds to a section in the report with the detail sections repeating as needed to accommodate rows from the data source.

Because they were created before the advent of the internet, banded report writers are not intuitive to web application developers, who are most comfortable with the web page-oriented design metaphor that one finds in modern graphical web development tools. In addition, web concepts—such as tables, graphical object containment and inheritance, cascading style sheets (CSS), and scripting in web-oriented languages like Java and JavaScript—are not supported.

Enter BIRT

The Eclipse Foundation's Business Intelligence and Reporting Tools (BIRT) project aims to take report development into the age of the internet. Based on industry-leading Eclipse IDE and Rich Client Platform (RCP) technology, BIRT was built from the ground up for web applications.

As Senior Vice President of Engineering for Actuate Corporation, I'm proud of the leading role my company has played in the project. We've leveraged our 10+ years of experience in the reporting and business intelligence space and put to work a significant number of full-time developers (or "committers," in Eclipse Foundation parlance) on the development of the platform. And while BIRT is a relatively young project that is only in its second major release, I think this investment is apparent in the rich and robust technology that the project provides. BIRT is an extensible, full-featured reporting platform that is ready for use in and integration with production applications.

We are seeing evidence of significant adoption already. A number of ISVs, including some very big names, have or will integrate BIRT technology into products. And BIRT is not just for Java programmers. Zend Technologies, the leader in PHP technology, is integrating BIRT technology into their Eclipse-based IDE and their Zend Platform so that PHP programmers will be able to design and run BIRT reports seamlessly in their PHP applications. Likewise, enterprise IT developers and system integrators have embraced BIRT and are already integrating it into important business applications.

All of these constituents—ISVs, IT, and SI developers—contribute to the Eclipse Foundation BIRT community, which is a vibrant one. BIRT often leads all other Eclipse projects in liveliness on the Eclipse project dashboard and is one of the most searched-for terms on the Eclipse web site. (Liveliness is a metric that measures project activity based on bug fixing, project e-mails, and newsgroup postings.) Feedback from the community has helped to drive project priorities, give direction on feature implementation, uncover defects, and once in a while, deliver some "attaboys" to the project team. Here are just a few comments posted by developers in the Eclipse BIRT newsgroup:

> *"I had installed BIRT the other day just to check it out and barely went through the introductory tutorial. Today I was able to drag and drop my way to replacing a*

broken report (600 lines of somebody else's perl) and all I can really say is it was almost too easy."

"I gave BIRT 2.0 M3 a test run and wanted to comment on it. First of all, excellent tool, you did a wonderful job. I did a report on an existing db table and it took me almost no time to get it up and running."

"We are an ASP (application service provider) and loving the BIRT project! Great work guys!"

"Have upgraded to 2.0 M2 and must say—it rocks! . . . Thanks for all the great work."

"We love BIRT."

I hope that you will leverage the information in this book to become a successful member of the BIRT community as well. And, in the off chance that you are standing in a bookstore aisle, having picked up this book with no idea what BIRT is all about, may I suggest that you rush home—after buying the book, of course—and download the software from the Eclipse BIRT web site:

`http://www.eclipse.org/birt`

Take it from me—it's the best way to prevent yourself from being lumped into the same category as 1970s COBOL programmers!

Mark Coggins
Senior Vice President of Engineering, Actuate Corporation