

A Better Calculation Engine

How the Formula One e.Spreadsheet Engine Combines Java Projects
With Business Logic and Calculations in Excel Files



A report from ReportingEngines, a division of Actuate Corporation
<http://www.ReportingEngines.com>

A Better Calculation Engine

How the Formula One e.Spreadsheet Engine Combines Java Projects With Business Logic and Calculations in Excel Files

The Situation

Many businesses drive their daily operations based on suggestions and data from internal business analysts, financial specialists, or scientific experts. And in most cases, these people build complex calculations and business rules in Excel spreadsheets to help determine and deliver this important information.

The Challenge

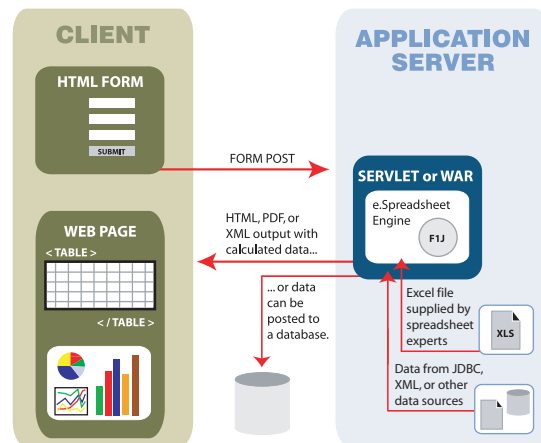
Many developers are quick to realize the importance of using a company's spreadsheet model to drive the business logic in their applications. However, most of these same developers, especially those who work in J2EE environments, struggle with automating or incorporating the traditionally manual manipulation of spreadsheets in their projects because of Excel's desktop-bound application architecture. In fact, most Java developers avoid the issue altogether and translate the Excel functions and formulas provided by their spreadsheet experts into Java code.

The Formula One e.Spreadsheet Engine Solution

The Formula One e.Spreadsheet Engine provides the proper Pure Java toolset for effectively integrating spreadsheet models into J2EE projects and applications. The e.Spreadsheet Engine provides a robust API that includes the ability to 1) read in Excel spreadsheets, 2) update data values and formulas at the cell level, 3) force recalculations, and 4) export data as Excel workbooks, individual values from cells, XML, or to a database.

Embedded in a J2EE project, the e.Spreadsheet Engine is able to fully automate the process of accessing, updating, calculating, and extracting data from a company's Excel-based spreadsheet model. As a result, applications that utilize the e.Spreadsheet Engine as an automated calculation engine reap numerous benefits including:

1. Business rules and calculations remain in Excel. Spreadsheet experts continue to modify and update formulas and functions in a familiar Excel environment and don't have to learn a new rules engine environment.
2. No programming time or resources are



The Formula One e.Spreadsheet Engine uses Excel files in Java environments to calculate, format, and export data.

required when spreadsheet models change. New Excel models simply replace outdated Excel models.

3. Application downtime is reduced or, in some cases, eliminated. The posting of updated Excel models can be done on the fly.
4. Mistakes in calculations are drastically reduced. Excel formulas and functions remain untouched in the Excel file created by the spreadsheet expert and are no longer translated to Java code.
5. Important elements of the application remain with their rightful owners. Java developers own the Java code; spreadsheet experts own the business logic and calculations.

Case Study:

City of Chicago Department of Environment

Following is a question-and-answer session with David Eslinger (University of Illinois at Chicago, Energy Resources Center) that appeared in the April 2004 issue of *Intelligent Enterprise*. Eslinger served as project manager for the Residential Energy Assessment Program (REAP), an application for the City of Chicago Department of Environment that utilizes the Formula One e.Spreadsheet Engine as a calculation engine.

Continued on Next Page >>>

A Better Calculation Engine (continued)

Q: Please describe your application. What information is gleaned, how often, and what actions does it support?

A: Sponsored by the City of Chicago's Department of Environment, the Residential Energy Assessment Program (REAP) was designed to provide personalized information to Chicago residents concerned about rising home heating costs. Our web application allows homeowners to describe the characteristics of their home including size, number of windows, insulation thickness in the walls, etc. Once user input is submitted, the application generates a custom report detailing specific ideas for increasing their home's energy efficiency. Potential dollar savings are also calculated based on the user's historical utility bills.

User input is collected within a Java-based environment and stored in a SQL database. The accompanying engineering (or business) logic was originally created in Excel by energy engineers at the University of Illinois at Chicago. Rather than translate that logic into a Java-based engine and traditional reporting tool, we used embedded BI to integrate the existing calculation engine and create reports dynamically within the Java environment. The reporting software, Formula One e.Spreadsheet Engine from ReportingEngines, allowed us to instantaneously transfer charts as well as data into reports in both HTML and PDF formats.

Q: Jim Gallo wrote in *Intelligent Enterprise* last summer in an article called "Preserve the Data Warehouse" that, "BI capabilities embedded in transactional systems degrade confidence." Here's an excerpt: "*Why did data warehousing become such a viable concept in the first place? Remember when data and systems were in complete disarray and we survived through Herculean efforts and a web of extracts? Remember when the business users sat in a room arguing over whose numbers were right? Read any book on data warehousing, and you'll learn more than you ever wanted to know about why the concept of the DW is important. Data warehousing was born to eradicate these problems. BI was born as a way to add true value to the DW.*" But clearly, in your experience, going with embedded BI was a better solution than a data warehouse solution. What are the factors that make embedded BI a better solution than a data warehouse for a given problem? Does the issue of "multiple versions of the truth" crop up in your application?

Residential Energy Assessment Program (REAP)

- **URL:** <http://erc.supergloo.net>
- **Sponsor:** Antonia Ornelas, City of Chicago Department of Environment
- **Project Manager:** David Eslinger, University of Illinois at Chicago, Energy Resources Center (www.erc.uic.edu)
- **Website and Marketing Developer:** Eric Gautschi, Agentse, Inc. (www.agentse.com)
- **Application Developer:** Todd McGrath, Supergloo, Inc. (www.supergloo.net)

A: User input data is stored for future use by both the end user and the city administrators. However, the critical component is the business logic that translates this data into a functional report for the user. The results of the engineering calculations generated are not stored but are instead regenerated for every report. We are not coordinating multiple datasets or other applications that would create data conflicts or "multiple versions of the truth" as Jim Gallo writes. During development, we did have to ensure that the Excel engine properly updated calculations on multiple worksheets in order to reflect current user inputs in charts and data cells.

Q: Forrester analyst Philip Russom was quoted in *InfoWorld* as saying, "*Before embedded reporting you had two technology stacks, but embedded reporting puts the report right there in the user interface of the application. There's an obvious productivity boost, and it also keeps the tactical decision-making very closely tied with the use of the operational app.*" Do you relate to what he said? Do you have examples of productivity boosts or any other benefits from using embedded reporting?

A: Embedded reporting allowed our engineering team to easily transfer information to the Web application developers. We saved significant development time by avoiding translating a multi-sheet Excel file into Java or JavaScript. Furthermore, changes in the energy engineering calculations, to reflect rising natural gas prices or new building insulation materials, are made to the Excel file by the application owners. The application requires no maintenance by the Web developers. This was critical for the City of Chicago given a limited project budget.

From the user's side, embedded reporting provides instant feedback in two universal reporting formats, HTML and PDF.



Web: <http://www.reportingengines.com> • **Email:** sales@reportingengines.com
Sales: 888-884-8665 • **Outside of North America:** 913-851-2200

FORMULA ONE
Enterprise Reporting Toolsets for Java Projects & Portals