



Case Study

Using Sherlock to Identify Calculation Engine Changes During Upgrade

St. Louis – March 25, 2013

If your organization has maintained SAP BusinessObjects platforms for any length of time and been through a migration, I'm going to go out on a limb and assume that like me, your team furiously reads the release notes in detail and then proceed to wonder what reports are going to break based on a calculation engine change. It happens. It may even seem like it happens a lot lately. Bugs happen, and SAP's product teams do their best to squash them along the way. This is a story about one such calculation engine change, the uncertainty it caused, and how we tackled it using Sherlock to head it off larger setbacks.

We at EV Technologies were working with a customer that was running nice and safely on SAP BusinessObjects XI 3.1 SP2. It was a nice run. It was a stable run. But, as that customer grew in their use of SAP BusinessObjects, specifically, Web Intelligence (Webi), they began to experience some documented instabilities that had been addressed in SP4. We were engaged to help with that patch to both test and implement it.

In the first pass, candidate content was identified for testing per our guidelines for testing. In side by side comparisons of reports, out of 100+ reports tested, only three reports didn't tie. On closer examination, there was an impact to those three reports in which the Webi calculation engine changed its behavior when "ForEach/ForAll" and "Where" operators were used in a formula for variable. This posed a great big question: Of the 15,000 reports in this system, how many were impacted by this change? There was really no way to uncover the scope of the impact without help from our Sherlock Inspector Suite. To us, it didn't matter if it was 15,000 reports or 500,000 reports- Sherlock was up to the task.

The change was subtle, but significant to the calculation engine. This (crazy) example:

```
=([GP (L8)].[Amount] Where([GP (L8)].[Account Group]="COGS – SAP")) +  
([$ Sales (L8)] ForAll ([GP (L8)].[SAP Profit Center]) Where([GP (L8)].[SAP Profit Center] =  
"19003"))*0.681
```

Changed to this:

```
=([GP (L8)].[Amount] Where ([GP (L8)].[Account Group]="COGS – SAP")) +  
([$ Sales (L8)]) Where ([GP (L8)].[SAP Profit Center] = "19003")*0.681  
ForAll ([GP (L8)].[SAP Profit Center])
```

In our analysis, 539 reports were uncovered that met the criteria that caused the test to fail. It was easy to quickly prune the list down to something manageable. 44 reports were in inboxes. 206 reports were in user's favorites. 21 other reports were instances. We quickly pruned the list and gave the organization plenty of notice on impacted reports that would need remediation and could let owners know on a case by case notice what action needed to be taken all the way down to the formula or variable that was going to go crazy in their report. We could easily include more information to make even more informed decisions like the last modified date of the report, or even, the last day it was viewed or refreshed, to infer relevance to the business.

While it can be asserted there is a cost of change, there is likely a bigger cost in not knowing or being able to take any action whatsoever in remediation in this environment. Sherlock is a sturdy mechanism for gaining better insights to managing your SAP BusinessObjects environment, both day-to-day, and more strategically as it matures.

About EV Technologies

A predominantly Web Intelligence shop. EV Technologies is an SAP Partner and an SAP Authorized Education Partner based in the United States and Australia. Customers contract EV Technologies for strategic business intelligence architecture, managed services, and application development needs in the SAP Analytics, Database, and Technology ecosystems. Sherlock 3, the flagship solution, is available for the SAP BusinessObjects BI Platform, SAP Edge solutions, and SAP Crystal solutions.

For more information, please visit: evtechnologies.com