

Crystal Reports – An Installation Case Study

One of our customers recently decided to move to source a new reporting tool to deliver a range of standard reports to general users (approx 130) and also provide an easy to use ad-hoc reporting capability to its power users (approx 5). Our previous experience was with Oracle Discoverer, but the power users found this unwieldy and too complex. We looked at the current solutions and decided that Crystal Reports, being the de-facto standard, was the safest and easiest solution to justify.

We decided to give the power users Crystal clients on their local machines and provide the standard users their pre-packaged reports through the Java Reporting Component (JRC). That was when the fun started.

Environment

Our customer is running an Oracle 9i Database with an Oracle Forms 6i based application. We had already developed a number of reporting solutions for them, including:

- Oracle Reports 6i (cumbersome, dated and overly complex)
- Custom HTML and RTF generators – using database procedures to create the reports in a text format (either HTML or RTF), using Oracle Forms6i to write this text to temporary files and then kick off the required application (usually MS Word).

Setup of Java Reporting Component

Documents detailing the architecture set-up are available at the Business Objects website. We are using a Windows2000 Server running Tomcat. We connect to the database with the JNDI/JDBC drivers. There were no major issues with the basic setup, although it did take about 4 days.

Pre-packaged Reports

Many of the pre-packaged reports were quite complex, requiring joins across tens of tables and containing many sub-reports. These reports change continuously depending on the date and the user, so running it, saving it as a PDF and distributing it to users was not an option. We found that the only way to get the reports to run efficiently was to code them using SQL Commands, where possible, and database procedures where the conditions were too complex for pure SQL. Developing the reports and running them through the Crystal Standard Client worked fine. But we ran into issues when we published the reports to the web:

Issues

Parameters in the JRC:

Text parameters are treated differently in SQL Commands when run thru the JRC. Basically, Crystal adds in extra apostrophes around the parameters. e.g.

Assuming SQL Command is:

```
SELECT employee_name FROM employees WHERE employee_name LIKE '{?inEmp}'||'%'
```

When you move this report to the web, and feed in a text parameter from your JSP, you'll get an error like "SQL Command not ended properly". It turned out that the JRC was adding extra apostrophes around the value. So, to make this work, you need to change the SQL Command to:

```
SELECT employee_name FROM employees WHERE employee_name LIKE {?inEmp}||'%'
```

But, when testing the report in the Client, you have to put apostrophes into the value to get the report to run.

The issue continues with sub-reports – but is much harder to deal with. If the parameter for the sub-report is the same as one of the parameters coming into the main report, its not too bad – just deal with it as above. But, if the parameter is a data field from the main report, then you can't actually test the sub-report in the Crystal client. i.e. Create the report, test it in the client, move it to the web, test it (you'll get an error from the apostrophes), re-open the report, remove the apostrophes that you had in the sub-reports SQL Command, and then save it – do not try to run it in the Crystal Client, as it will fail to run and won't let you save it.

Handling SQL Commands

We found our version of Crystal Reports (10) kept crashing whenever we tried to make changes to the SQL Commands and/or the parameters in a SQL Command report. You can fix this by downloading the latest service packs from Business Objects. Its not clear which ones need to be installed and in what order, we just grabbed the lot and prayed.

JNDI Object Not found

If you get this error, you need to increase the number of connections allowed in your JDBC configuration

Performance in the JRC

We had major performance issues with the JRC. Using the standard code available on the web to run reports, you'll find the report gets re-run every time you navigate to a new page. We were unable to find any solutions to this, but designed our own system for caching the report which got rid of this problem. Until we did this, some of the other functions such as 'Go To Last Page', 'Export to PDF' didn't work.

Future Changes

Now we have our caching system, it should be a fairly small step to automatically save off copies of reports that we know won't change e.g. October's Organisation Performance Stats run in November. This will reduce the load on our main production database, while not requiring the set up of a scheduling system, special jobs, etc.