

Designing Fast Crystal Reports

Planning Your Reports

Are you getting the data from the place that makes the most sense?

For example, if designing a job report with current totals, it will be much more efficient to use the fields from the Job master file than from the Job transaction file. Your needs will drive this. If you sometimes need to back date a report and sometimes just need the most current data, it may be worth having two reports – one for back dating (transaction file) and one for speed (master file).

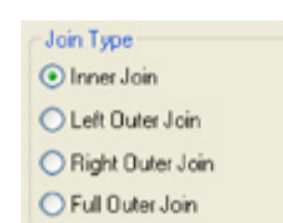
Avoid unnecessary processing**Avoid linking through unnecessary tables**

If you are designing a vendor tax report and don't need to include invoice detail, then avoid including those tables you don't need. The AP Tax Distribution table has the Vendor field on it, and that is all that is required to get the Vendor Name.

The fast approach in this case is to go from MASTER_APM_TAX_DISTRIBUTION –MASTER_APM_VENDOR.

Inefficient*More efficient***Reduce the amount of data processed with the select expert and parameterize if possible**

Filters, Conditions, Select Expert, etc. reduce the volume of data processed. This is one of the most effective tools for speeding up reports.

**Use inner joins wherever it makes sense (know the database)**

If you are designing a report where you only need jobs with transactions, then use an Equal Join for the two tables. This will filter the report to only give you job information for jobs with transactions. A Left Outer Join, the typical default, will give you a report with all jobs – transactions or not – and it will take longer.

Let Crystal Reports generate summaries where possible (Summaries, Running Totals)

If you are designing a job totals report based on transactions, it may be best to simply summarize the totals vs. creating formulas to derive the same numbers. Formulas require more processing by the report and will always take longer.

Be smart about formulas

Become familiar with formulas options that perform the same task. If you have a formula with multiple nested IF statements, it may make sense to use a SWITCH statement instead. Again, the report will run faster because it has fewer individual commands it needs to process.

Link tables by indexed fields

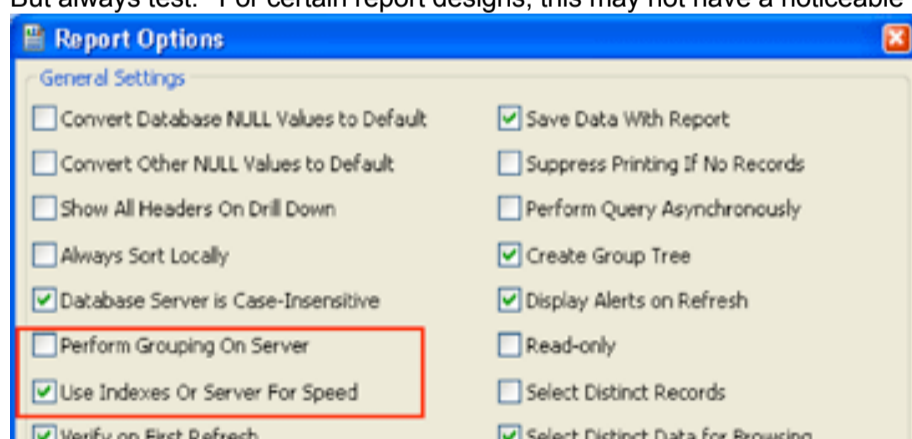
Databases are optimized for this. The report will process the data faster because you've joined the various tables together by indices.

**Use Indexes or Server for Speed when possible (report option)**

Two report settings that will help with performance include:

- Use Indexes or server for speed
- Perform grouping on server

But always test. For certain report designs, this may not have a noticeable impact

**Use an On Demand Subreport to mimic a drill down**

For example, if you have a report in which you want to show totals on the main page, with an option to drill down to a table with a million records of data, an On Demand Subreport will be very efficient. The main page will process only the content that is displayed. If it had typical subreports it would need to process all of those as well before any information is displayed.

However, with an On Demand Subreport, none of the drill down data is calculated until you actually ask for it. So, if you have project managers waiting a long time for a report that includes a mix of totals and detail and all they need is the totals this time, consider changing it to use an On Demand Subreport.

Use a Subreport if you have a gazillion Joins

Having many joins will slow your report speed. In this case, it is probably faster to have a typical subreport do all the work as the report is initially processed. The report will gather your data more efficiently.

UDFs (User Defined Functions)

These are a special type of formula for lack of a better term. Basically, a UDF will make one trip to calculate whereas a standard formula may make 1000's or even 100,000's of trips querying the database and performing calculations. It's much more efficient to use a UDF when possible.

Where possible, reduce the number of times you call a UDF. If, for instance, you use tsCustomDescription() to retrieve a custom description, it may make more sense to only call the function one time in the report header and store the value in a global (or shared) variable so you can use it elsewhere in the report. Since the custom description doesn't change, you only need to ask for it one time.

Note: If you have additional tips or suggestions, please forward them to erin-todd.hansen@sage.com.



800-858-7095

www.sagecre.com
 DOWNLOAD
[Printer Friendly Article](#)
Suggestions and Feedback

We welcome suggestions and feedback regarding *Your Sage Timberline Office Update! for Construction and Real Estate*. Please submit your comments, article ideas, compelling statistics, and success stories to vantagepoint.na@sage.com. If you have a customer support question, please call 800-551-8307.

