Global Knowledge ®

Expert Reference Series of White Papers

Web Intelligence to the Rescue

Web Intelligence to the Rescue

Jim Brogden, BOCP-BOE, BOCP-MS, BOCP-CR

Introduction

The life of a business intelligence (BI) developer can be very rewarding, but it can also be challenging to say the least. Every client and every project is different and they always present new and unexpected twists. Being skilled in various technologies is a major benefit. But even if you're an absolute champion in a handful of specific tools and an MVP in a couple of others, being versatile enough to pick up the client's tool of choice and run with it will make all of the difference in the world.

If you're lucky, your client or the company that you work for will have SAP BusinessObjects in place. With this powerful business intelligence platform, you'll be able to create refined analytical documents that can be used throughout your company. It's versatile enough to handle almost any set of requirements and easy to share with your users through a browser-based portal—the BI Launch Pad. Since its biggest strength is with its ability to query multiple databases to create data-rich custom reports, developers have many ways to bring business data to life for analysts to uncover trends and important results.

Even though some users and data consumers generally view reporting as simple and straightforward, most BI team members will tell you otherwise. The reports are the "tip of the iceberg" in a data warehouse. The majority of the work often occurs behind the scenes. Almost all reports are usually always preceded by extracting data from source systems, transforming it into a workable format, and loading it into tables. Reports also include the development of calculations and formulas needed to comply with the business logic that most effectively communicates results to users. And almost all of these steps are never seen or known to report consumers. From a user's perspective, the data warehouse is, simply, the reports and dashboards.

That's why every effort must be made to produce perfect reports. If mistakes are found, your users will begin to lose faith in the entire business intelligence and data warehouse implementation. No pressure. Just be perfect. We'll talk more about this later. But for now, remember these three steps and you'll be on your way to ensuring accuracy and success: validate, validate, and validate.

So how do we as report developers overcome ridiculously challenging requirements? Simple. Push all difficult tasks to the extract, transform, and load (ETL) team to handle. I'm only kidding, of course. We'll be able to handle our share of the workload in our SAP BusinessObjects universes and reports. In all seriousness though, complex data issues should be handled in ETL and at the table level. For additional modifications to data, case logic and other transformations can be applied at the Universe level.

And for more unique and ad-hoc needs, apply IF-ELSE logic to variables and formulas in reports. For the most powerful reports, take advantage of every layer available to you to create an extensive collection of objects with precise definitions.

From Design to Delivery

In this white paper, I'll describe how a report developer can overcome numerous reporting requirements to produce useful documents that meet and exceed expectations. Even with knowledge gaps in both business data and business intelligence skills, SAP BusinessObjects is intuitive enough for intermediate and beginner developers to create valued reports.

2

Believe it or not, sometimes a project manager (or if you're a consultant, someone from the sales team) will make seemingly impossible promises to the client. This usually includes the amount of time it will take to complete a project or the numbers of developers needed to complete the work. Lastly, the functionality in a dashboard or information in the final report deliverables can also feel like an insurmountable task. While developers will always prefer to "under promise" so they can "over deliver", project managers usually have a good handle on what can actually be achieved, even though they're often accused of over promising.

Below is a small list of report requirements assigned to a developer. It's common in the real world to get handed lists much lengthier than the sample set used in this paper. Precise reporting requirements are helpful because they serve as clear blueprints for report developers. The drawback is that they often lack creativity and don't allow developers to take advantage of all the features in the chosen reporting or dashboard tool.

Let's talk about how SAP BusinessObjects and Web Intelligence can be used to "rescue" a report developer from a challenging set of business requirements that might feel impossible to achieve otherwise. I'll follow up each requirement with recommendations for completing each objective from an SAP BusinessObjects perspective.

Report Requirements

1. Query the data warehouse to get sales numbers (revenue, units sold, and cost per unit) for two years. Display the data for several different dimensional categories, including:

   a. Year
   b. Quarter
   c. Month
   d. Date
   e. Type
   f. Category
   g. Description
   h. Region
   i. State
   j. City

Recommendation: Begin by setting up a connection to the database that you'll be connecting to. Then create a new Universe. Universe design must be completely error-free, so take your time here.

The SAP BusinessObjects Universe is where you'll add the tables needed for the reports, make the joins, and create the objects to be used in the report. The objects in SAP BusinessObjects are aliases for fields in tables (or views). This is where you'll be able to apply extended business logic such as adding Case statements. There is tremendous potential in this layer to create highly sophisticated reports.

2. Create a series of reports by querying data from three different database contexts and include results from all three on the same page.

Recommendation: A single Universe or multiple universes can be used. Either way, at least one new query will need to be created for each context. Contexts are like mini-schemas that have been set up by a designer. Objects added to queries can only come from their context. A single document in Web Intelligence can contain multiple queries with each coming from a different context as long as each query is made up of objects from a single context. See the screenshot below to get a visual perspective. Data from different queries can be joined in the report by using the Merge Dimension feature.

3. The queries will not be simple. They'll include nested constraints and require unions.

Recommendation: Nested constraints is a fancy way of saying filters grouped by *AND* or *OR*. An example of this type of filtering scenario is:

- City = Dallas and Category = Baseball
          OR
- State = Alabama and Category = Football

The screenshot below shows the Query Filters panel with the objects set up as described above.
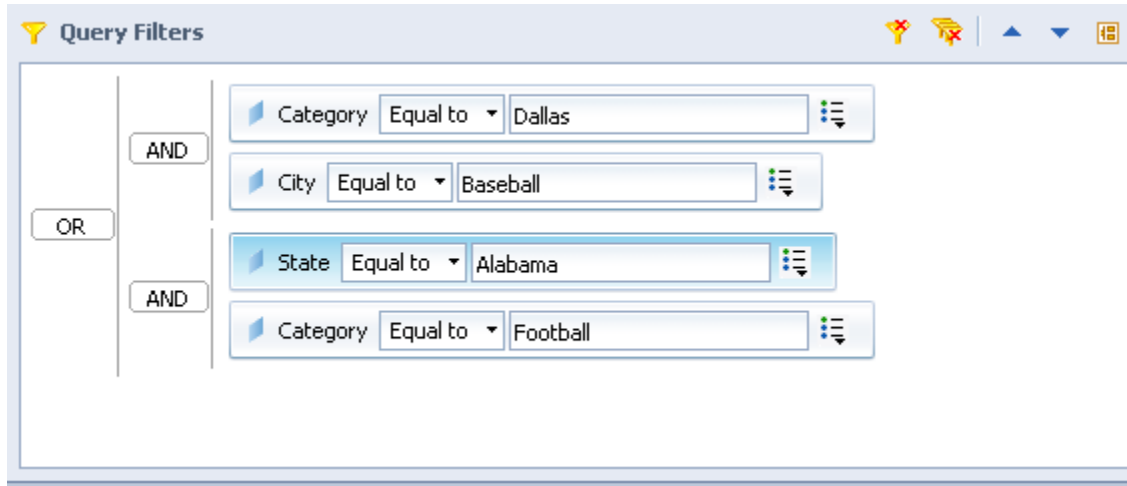
Nesting of predefined conditions or user-defined filters is done easily and graphically without having to write any SQL. This wizard-driven method of writing SQL removes the vast majority of technical complexity from writing queries.

Union queries can also be added in Web Intelligence with just a few clicks. Just add a "combined query" as pictured in the screenshot below to add a union to an existing query.

After the union query has been added, make any necessary changes to the second part of the union and click Run Query. The union, or combined query, will be represented in Web Intelligence as pictured below.
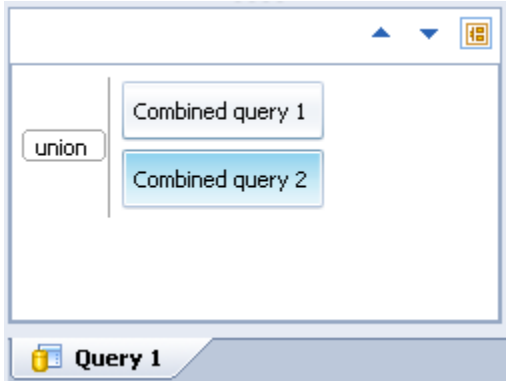
4. Create a series of reports to be used for various purposes by different groups of users. This will include summary reports for high-level aggregate data and detailed reports for deeper analysis.

Recommendation: Don't try to do everything in a single report. I like to think of reports as a "reporting solution" rather than a few individual unrelated reports. A reporting solution will include a small series or collection of related reports that have a consistent appearance. Always use the same size headings, fonts, cell heights, column header colors, column header border colors, and layout style. Users will get really annoyed and maybe even confused if each report has a different style. You'll also be able to connect the reports with document links and prompted filters in report hyperlinks.

For the summary reports, you'll be able to blur the lines between traditional "rows and columns" reports and dashboards. In other words, display the data in both data tables and visually with charts. If displaying data in charts, be sure to emphasize what's important. Give the important measurements the most prominent position on the screen. For less significant metrics, make the size of the charts smaller.

5. Allow users to drill into the data to view results by various dimensions to pinpoint issues and get a better understanding of the data.

Recommendation: Setup hierarchies in the Universe. This will allow for a custom collection of objects to work as cascading filters in any report that uses the objects. While in the report, enable simple report filtering, then add the objects in logical order. The purpose of this type of filtering is to filter the data in all subsequent filters when a selection is made.

To add hierarchies to a Universe, click the Hierarchies icon located in the standard toolbar as pictured below.

When the Hierarchies Editor loads, choose from the objects available to be part of a new custom hierarchy. You can move the objects up or down the list to define their order in the hierarchy. See the screenshot below to get an idea of how these custom hierarchies are set up.
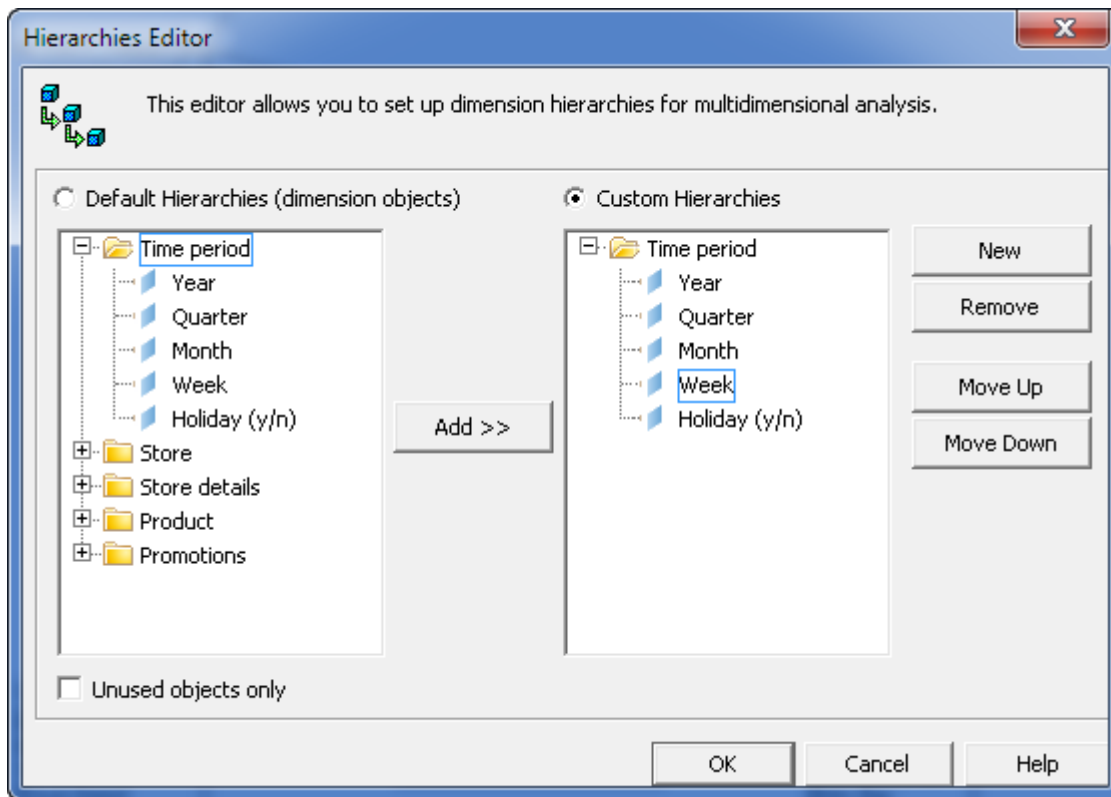
For example: Let's say that it's May 2015 and you have the following filters added to a report: [Year] and [Month]. If you select 2015 from the [Year] filter, you should only see five months in the [Month] dropdown. It wouldn't make sense to select December 2015 when we're only in May. With hierarchies set up for the two objects at the universe level, when 2015 is selected, only January through May is displayed in the month filter. If 2014 is selected then all 12 months will appear in the month filter.

NOTE: Before publishing reports to production and making them accessible to users, be sure to thoroughly validate the data. Results should be authenticated, certified, and approved before putting them into the hands of your users to trust and make business decisions.

Most formal BI implementations will require results to be validated by the quality assurance (QA) team before they're approved for production. In self-service environments, QA testing should be performed on the universes. Testing universes generally includes verifying that the correct tables and views are being used; checking join conditions; ensuring contextual integrity, object definitions, predefined condition syntax, class structure; and validation of the business logic applied to objects.

Universes can also be protected from potential SQL mistakes, such as preventing Cartesian products from being produced by objects from mixed contexts. Additionally, maximum row counts can be set in the Universe along with maximum query execution times. These objects are found in the Controls tab in the Universe Parameters window pictured below.
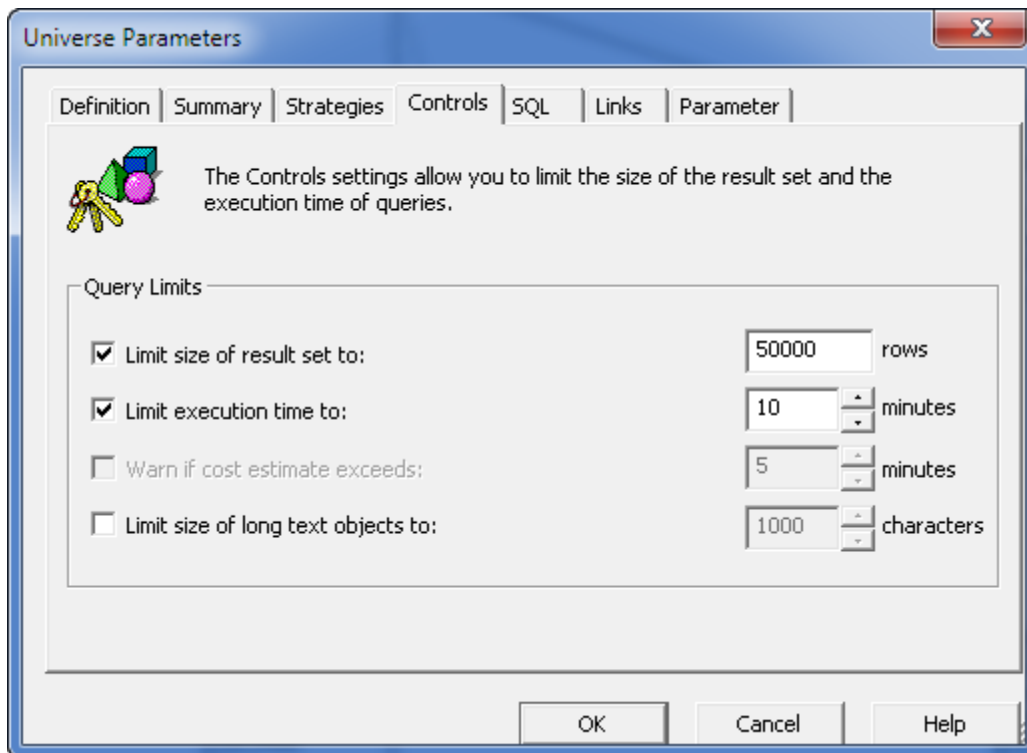
Predefined conditions can be set up to automatically apply to any query that uses an object from the current object class or Universe. This setting can be seen when editing the properties of a predefined condition as pictured below.
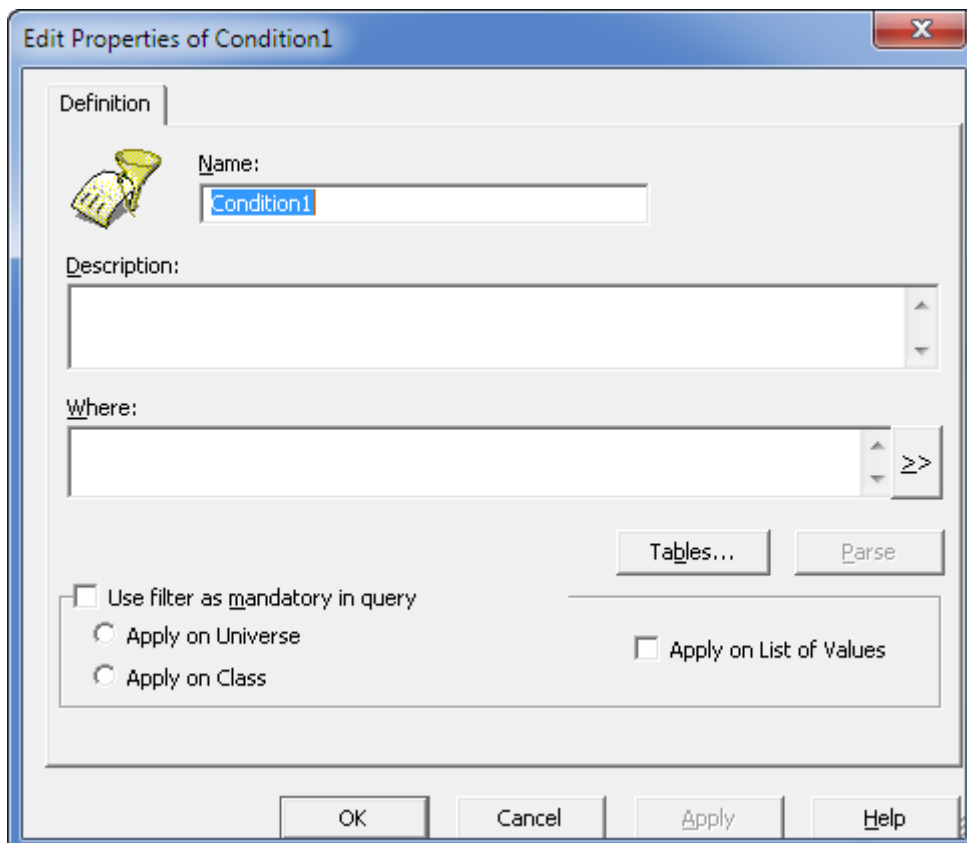
TIP: A well-designed business intelligence reporting solution is a major asset to an organization, and data integrity must be taken seriously in order to maintain user confidence.

6. Share the results with business users while maintaining a single version of the report.
   a. Reports must be published for viewing on-demand.

Recommendation: If at all possible, you should have at least two SAP BusinessObjects environments—one for testing and developing, and the second as your production environment. You should never "develop in production"—that's why you have a dev system. After your QA team has validated the accuracy of data in your test/dev/QA system, the next step is to migrate the reports and universes to production.

Once your reports are in production, users should be allowed to view and refresh them on-demand. Use these report versions for scheduling instances of the report.

   b. Maintain the format and the look-and-feel when the data size changes.

Recommendation: To ensure that your data tables and charts maintain a safe distance between each other and to accommodate potential growth, you'll want to enable the Relative Positioning feature. This option works exactly how it sounds: it maintains the amount of space between charts and tables when filters are applied and when the number of rows in a report increases.

These settings are pictured below in the Format Chart settings window. To access these settings, right-click on a report element (chart or data table) and then select the Format editor.

8
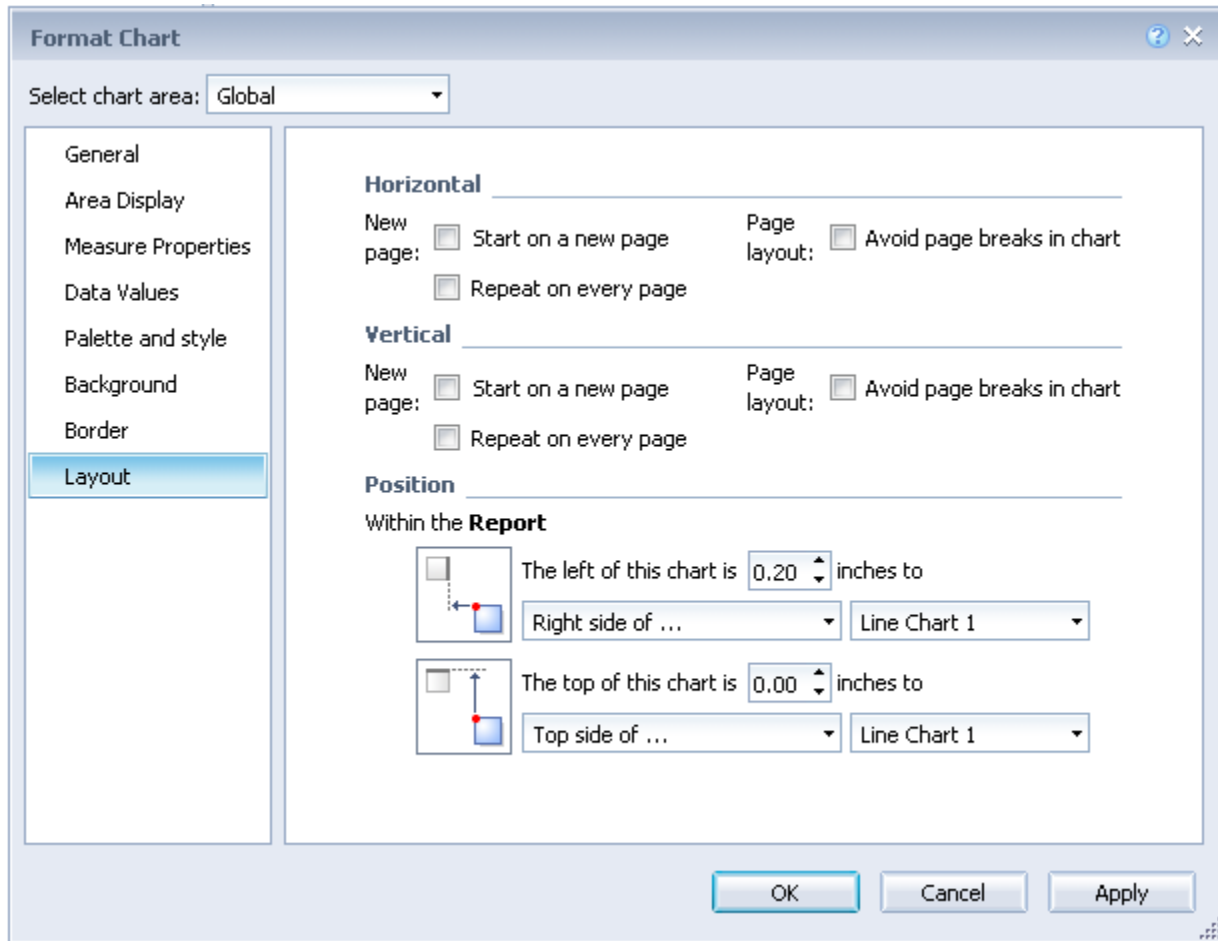
    c.   Schedule the main summary report to run on the following schedules:
        i.   on the first day of every week—analysts
       ii.   on the last day of every month—directors and managers

Recommendation: Scheduling is a valuable feature in Web Intelligence and allows for reports to be emailed to a specified list of recipients at a specified time. Files can be sent to users as either XLSX, CSV, TXT, or PDF for viewing without having to log in to the portal.

The screenshot below shows the options available for recurrence times when scheduling a report.
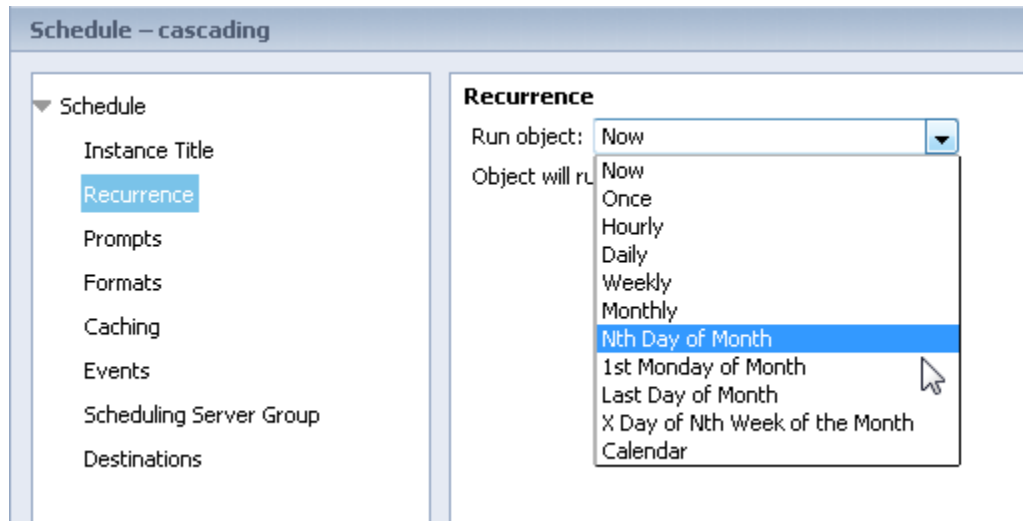
7. Make the on-demand version of the reports interactive and designed for printing.

Recommendation: To make reports interactive, they should contain some form of dynamic filtering and/or drilling. Since we covered drilling in step 5 above, let's discuss other interactive options. The first is Element Linking. This feature encourages user interaction by allowing a selected bar value in a chart to be passed to other report elements on the same canvas. In other words, a chart can be used to drill into the details contained in other elements. This form of contextual filtering provides an excellent method of diving into relevant details.
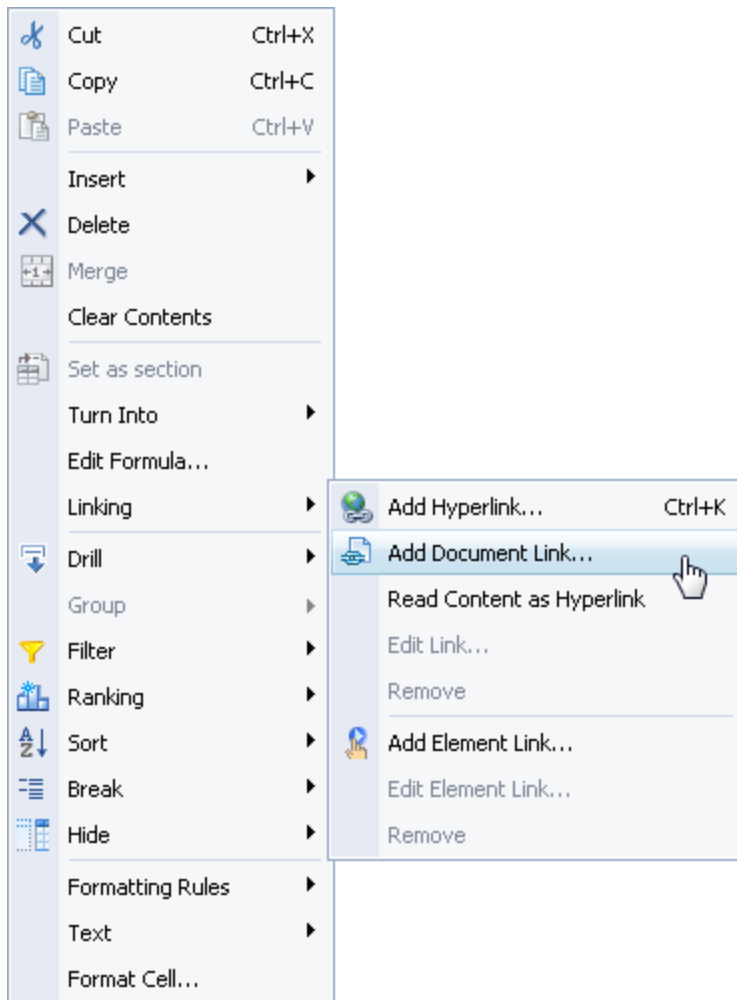
To design reports for printing, change the default page layout from Quick view to Page Layout.

8. Set up linking between the reports. Make the summary report be the source and create links to several of the detail reports. Make the links "smart links."

Recommendation: Document linking is the procedure used for connecting reports for on-the-fly analysis of related information in different reports. The purpose of this feature is to view more information about a specific value. That's where the smart links come into play. A smart link is a hyperlink to another report that passes values in the link to prompted query filters in a target document.

To set up the target document, change certain filters to work as prompted filters and give the filter object a unique and identifiable filter name. These names will be used by source reports when passing values to the target reports.

To add a link to a report, right-click on the object or cell to add the link to, then select Add Document Link. This menu of options is pictured below.

✂	Cut	Ctrl+X	
📋	Copy	Ctrl+C	
📋	Paste	Ctrl+V	
	Insert	▶	
✕	Delete		
⊞	Merge		
	Clear Contents		
⊞	Set as section		
	Turn Into	▶	
	Edit Formula...		
	Linking	▶	🌐 Add Hyperlink... Ctrl+K
⤓	Drill	▶	📄 Add Document Link...
	Group	▶	Read Content as Hyperlink
▼	Filter	▶	Edit Link...
⊞	Ranking	▶	Remove
A↓Z	Sort	▶	🔗 Add Element Link...
⊟	Break	▶	Edit Element Link...
⊞	Hide	▶	Remove
	Formatting Rules	▶	
	Text	▶	
	Format Cell...		

9. How will a developer ever be able to meet all of these requirements?

Recommendation: Web Intelligence to the rescue! By using SAP BusinessObjects as your selected BI reporting platform and Web Intelligence as your reporting tool of choice, you'll be able to create an entire reporting solution, not just a few random reports thrown together.

The connectivity to the semantic Universe layer provides a stable path directly to your data warehouse. By applying contexts, derived tables, and custom-written object definitions, report logic is transformed into objects that are used by users.

For reporting developers can provide data consumers with a wide variety of features. From adding drilling and filtering, to presenting data in dashboard/report hybrids, Web Intelligence is an ideal data analysis tool. You'll be able to query databases from Microsoft SQL Server to SAP HANA, and then create nuggets of data gold for your users.

Conclusion

Reporting is so much more than just pages of data in rows and columns. We just scratched the surface on what SAP BusinessObjects and Web Intelligence can deliver. When it comes to accessing large amounts of data and then creating customized and sophisticated analytical reports, Web Intelligence is a wise choice.

11

Learn More

Learn more about how you can improve productivity, enhance efficiency, and sharpen your competitive edge through training.

SAP BusinessObjects Universe Designer XI 3.0/3.1:Universe Design (BOU310v91)

SAP BusinessObjects XI 3.0: Advanced Universe Design (BOU320v91)

SAP BusinessObjects Web Intelligence Report Designer (BOW310v15)

SAP BusinessObjects Web Intelligence Advanced Report Design (BOW320v15)

SAP BusinessObjects Web Intelligence: Advanced Report Design (BOW320v096)

Visit **www.globalknowledge.com** or call **1-800-COURSES (1-800-268-7737)** to speak with a Global Knowledge training advisor.

About the Author

Jim is a senior data warehouse developer on the enterprise reporting team for Intercontinental Exchange (ICE) at their headquarters in Atlanta, Georgia. He's the lead author of all three official SAP BusinessObjects Web Intelligence textbooks and has more than 12 years of business intelligence development experience.