

# **Enterprise Reporting Best Practices in an SAP Environment**

---

---

Author: Neil Raden Hired Brains, Inc. August 2004

Contributors: Alison MacDonald, Dan Kearnan, Melissa Rollins

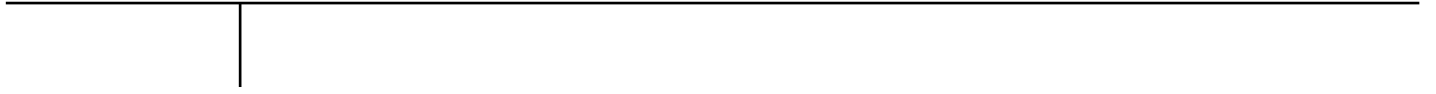
Audience: This paper is intended for information technology and business-line executives who wish to understand how best to implement enterprise reporting strategies in an SAP environment.

---

# Contents

---

<b>Author and Contributors</b> .....	<b>.ii</b>
<b>Table of Contents</b> .....	<b>.iii</b>
<b>Executive Summary</b> .....	<b>.1</b>
<b>The Reporting Landscape</b> .....	<b>.3</b>
Operational versus Analytical Reporting .....	.3
Requirements for Reporting Solutions .....	.6
Timing .....	.6
Pertinent Data .....	.7
Metadata .....	.7
Latency .....	.8
Logical versus Physical Design .....	.8
<b>REPORTING FROM SAP R/3</b> .....	<b>.10</b>
Generating Reports Against the Tables .....	.10
Extracting Data into a Data Warehouse .....	.10
Using BAPI Interfaces Supplied by SAP .....	.11
Writing Procedural Code in ABAP .....	.11
Native Driver for SAP R/3 .....	.11
<b>BUSINESS OBJECTS REPORTING SOLUTION FOR SAP R/3</b> .....	<b>.13</b>
<b>BUSINESS OBJECTS REPORTING SOLUTION FOR SAP BW</b> .....	<b>.15</b>
SAP BW as the Reporting Platform .....	.15
Business Objects Reporting for SAP BW .....	.15
<b>CONCLUSION</b> .....	<b>.17</b>



---

## EXECUTIVE SUMMARY

---

The functional reach of SAP's suite of software solutions is vast. No other software company offers support for such a comprehensive range of business processes, information flows and analytics. However, all of that capability comes with a price – complexity. As a result, reporting on and drawing insight from the data captured and managed within SAP is challenging. Many customers of SAP's flagship products, R/3 and mySAP, find the process of designing and developing new reports time-consuming and expensive. The internal data structures of SAP are designed for performing their direct functions, not to facilitate reporting. Devising reports from SAP tables directly requires knowledge of three separate domains, all of which are very sophisticated: the underlying reference model of SAP, which is vast, and spans many different modules; the ability to develop code in ABAP; and the functional knowledge of the business processes themselves.

Business Objects' enterprise reporting solutions solve these problems in two ways – first by providing a native driver to SAP's OpenSQL that bypasses the ABAP programming language as well as offering exceptional operational report-writing capabilities. And secondly by delivering analytical reporting capabilities through advanced Business Intelligence tools that utilize a semantic interpretation layer allowing business analysts without knowledge of the SAP model access to the data.

Reporting using SAP's Business Information Warehouse (BW) poses some of its own challenges. The supplied data presentation tool, Bex Analyzer, is spreadsheet oriented and best suited to cross-tab and OLAP-type reporting against the “cubes,” or queries of the cubes, which are relational tables modeled as star schemas. These dimensionally oriented models are very useful for management and analytical reporting, but in practice, many SAP clients use BW as the vehicle for operational reporting, which is not always well suited to a multidimensional approach. In fact, SAP promotes BW as the vehicle for all reporting in SAP, pointing to the ODS<sup>1</sup> as the appropriate vehicle in BW for operational reporting, a strategy that has some very clear drawbacks. SAP's strategy to use a version of Business Objects' reporting technologies within BW points to their own awareness of limited operational reporting capabilities within the core BW solution. BW is squarely in the middle of SAP's rapidly evolving composite applications strategy, including SCM, PLM, CRM and SRM<sup>2</sup>, so it is certain that BW will continue to grow in size, complexity and centrality to the entire SAP offering. However if operational reporting requirements top your must have list then you need to investigate complementary third party solutions.

A common drawback in a vendor-supplied data warehouse solution is the need for integration of data from other sources, such as internal applications from other vendors and external data from data and service providers, customers, suppliers and other critical participants in the operation of the organization. Creating an integrated data warehouse facility for SAP has been a work in progress, but admirable progress has been made. Nevertheless, organizations typically have large investments in separate data warehouses and data marts and the duplication of that effort may

<sup>1</sup> ODS is the Operational Data Store, a term that has many meanings. In its original concept, the ODS was merely a collection of very recent transactions, not integrated and cleansed like the rest of the data in the data warehouse, and provided the capability for the data warehouse environment to produce some rudimentary operational reporting. In BW, the ODS is more of a staging area, used as a precursor to the construction of the cubes. Because of its non-dimensional schema, it is more amenable to flat operational reporting.

<sup>2</sup> SCM – Supply Chain Management; PLM – Product Lifecycle Management; CRM – Customer Relationship Management; SRM – Supplier Relationship Management

---

exceed the value in the migrating to SAP BW. This is especially true where there is a substantial amount of non-SAP data involved. In those cases, organizations are more likely to opt for reporting solutions that can access and process from multiple sources in a single report, including SAP R/3 and SAP BW.

A robust reporting tool must be able to handle on-the-fly integration where necessary.

In today's environment, all IT initiatives are scrutinized carefully and various techniques are employed to insure that they are founded on a solid set of assumptions and facts. Return of Investment (ROI) and Total Cost of Ownership (TCO) are commonly applied. Because time is money, being able to deploy a solution quickly is a key determinant of adoption, and a reporting solution that is dependent on the successful construction of a large and potentially expensive data integration effort (data warehouse, for example) is more time consuming than one that can deliver immediate value and be deployed quickly. The point is not that integration is unnecessary, because for many organizations it is essential, but that reporting solutions that can stand-alone and add value out-of-the-box are important elements of an overall solution. Business Objects' own cost-effective, pre-packaged integration solutions are a very suitable and less complex option than BW if your specific business or information requirements demand integration efforts.

Part of the value engineering principle of SAP is using a single architecture for a broad range of applications instead of integrating offerings from various vendors in a "best-of-breed" fashion. This potentially leads to improved ROI and TCO as a result of smooth interoperation as well as the relaxed requirement for building integration points. A reporting tool that can operate effectively without additional structures and can work through a tightly coupled set of drivers directly with the source system, instead of an additional set of mapping tables (metadata) and transformed data (data warehouse) follows the same logic.

Business Objects offers a complete set of business intelligence solutions for SAP, including features that are unique in the industry. These include reporting solutions for R/3 directly, reporting tools bundled with SAP BW and the full range of Business Objects tools for data integration, ad hoc query and analysis, and performance management to be used with or without a data warehouse. This paper will examine the capabilities of these tools, with a highlighted focus on Business Objects' operational reporting solutions and evaluate their utility for customers of SAP.

---

# THE REPORTING LANDSCAPE

---

Reporting is a wide and varied subject. Simple models or taxonomies, like operational versus analytical, or enterprise versus departmental versus personal, fail to capture the detail and nuance of the variety and texture of reporting in organizations today. Even the nature of a report itself has morphed in the last few years from a textual or graphical presentation on paper or a monitor to unattended, event-driven alerts through email or wireless devices and even “reports” that are requested and consumed by other systems in real-time. Because one tool cannot easily address all needs, however, it is convenient to divide the reporting landscape into functional areas in order to match solutions to requirements.

## Operational versus Analytical Reporting

The most common division is operational versus analytical. It is impossible to define these crisply but there are obvious differences that can be charted. The table that follows lists a number of attributes and describes their nature for each type of report, but this is the general case, there are many exceptions:

*Figure 1. Operational Versus Analytical Reporting*

	Operational	Analytical
Forms	Yes	No
Level of Detail (in report)	Detailed	Summary
Timeframe	Current	Trended
Staged data	No	Heavily
Layout	Banded, listing	Cross-tab, grid/matrix, graphical, dashboard
Format	Formal, banded, pixel-perfect	Intuitive, navigable, grid
Interaction	Drill down	Parameter, ad hoc, OLAP
Delivery	Scheduled, pushed	Pulled, event-driven
Audience	Large	Small

Operational reports range from forms, such as payroll rosters, bills of material, even paychecks to highly and precisely formatted reports such as regulatory submissions. Operational reports meet the requirements of the broadest and most numerous groups of users within an organization, all the way from the information consumer to the executive who needs quick snapshots of information.

The level of detail is usually very low in analytic reports, often a listing of one or more atomic records. Analytical reports almost always deal with data at a summarized level, though the lowest level of detail does come into play in two important ways. First, summarized analytical data almost always provokes questions about causality: “Why is this number what it is?” Good analytical systems provide the means to drill down into lower levels of detail. Second, though an analytical report may present summarized data, the logic behind the report very often needs the finest grained data available to arrive at its result. This is a common misconception, that analytical tools and reporting solutions only need to deal with aggregated derived data. It is precisely the

---

reason that spreadsheets have only a limited role to play in enterprise analytical reporting solutions. The attributes of atomic transactions, that disappear when rolled up to summary levels, often form the basis for constraining, grouping and evaluating data.

Operational reports, by definition, are primarily concerned with the near-term timeframe. This is actually a driver in the move to data warehouses, because operational systems, with their limited historical perspective, rarely make provision for either storing or, even more importantly, keeping consistent, historical data. Analytical reporting may be just as focused on the present, but there is always the added need to understand trends. This is primarily the reason analytical reports rely on staged data, such as a data warehouse and operational reports do not. Once again, these distinctions are for the general case.

Forms are generally constructed as a series of fields like any other sort of markup process, and operational listing layouts are generally banded with indents, control breaks, color-coding, special fonts, sub-totals and a host of other document formatting techniques. Analytical reports are often cross-tabulations, showing the numerical values of certain measures, or metrics, by one or more categories over one or more other categories, such as current month and year-to-date, this quarter versus same quarter last fiscal year, of same store sales of products by region.

Both kinds of reports can use the same type of formats, but in general, operational reports are either listings or very carefully crafted documents, while analytical reports are constructed to be informative and navigable. Where the two types of reports diverge most notably is in their interaction with users. Operational reports are generally just presented, while analytical reports are presented with a wide range of interaction potential. Reports can be specified, but the actual variables, or parameters used at runtime can be prompted (though this is also common with operational reports); ad hoc reporting allows users to actually create a report on-the-fly and OLAP provides a wide range of interactive navigation, drilling down into detail, rearranging the rotation of the display, changing the filters and an almost limitless range of interactions.

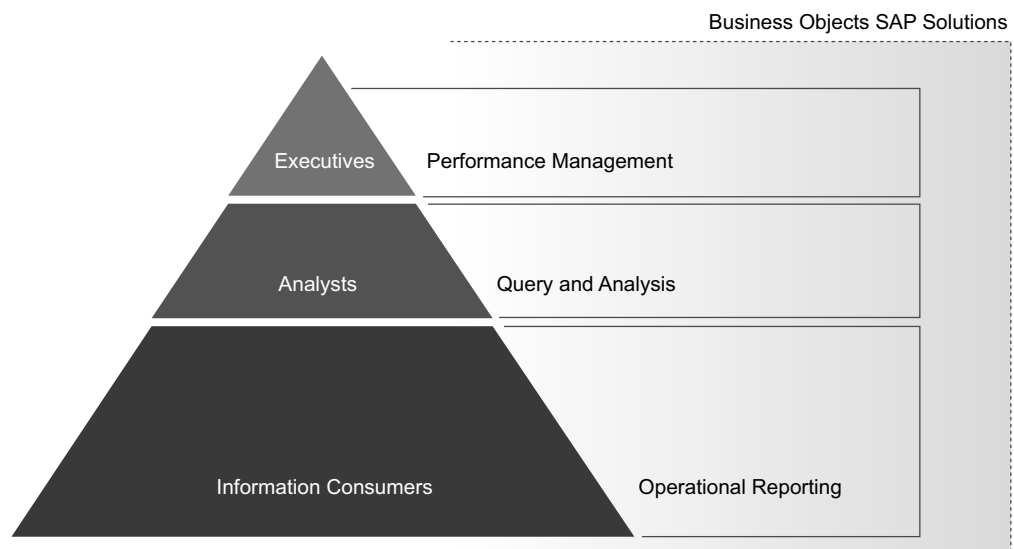
Delivery of reports can range from the interactive request and modification of an OLAP tool, to a “pushed” static report to email, a wireless device or a portal. While the content of the latter type, pushed reports, can range from strictly operational to the most sophisticated analytical, what determines the how it is presented is the purpose to which it is applied. Regardless of whether a report is a trial balance from SAP FI pushed to a portal or a complex cash flow projection for a new product, the result is the same – it is a static report. So for the purposes of delivery, the origin of the report is not the issue, the delivery and use of it determine what requirements it places on a reporting solution.

In general, the audience for operational reports is much larger than the analytical base for the simple reason that more people in an organization do things rather than analyze things. Operational reporting is focused on informing people about the current situation, about documenting things and about performing required disclosures, such as statutory or regulatory reporting. Analytical reporting is typically aimed at the smaller audience that evaluates information for strategic planning, medium-term planning, and for insights into why things occur and what to do about them. This may not always be the case, but if you consider pricing lists pushed to the sales force as opposed to pricing models in marketing, the reasoning holds.

There is no question that operational and analytical reporting are very different aspects of enterprise reporting, despite their similarities. Operational reports support the existing work processes, the status quo. With the exception of a few strikingly original people, most people at work view their role in a manner that is in consonance with their surroundings. Translation – in organizations, most people rarely question their patterns and routines. As the cognitive scientist Edwin Hutchens observed, this underlying knowledge is “...often transparent to those who use it. Once learned, it becomes what one sees with, but seldom what one sees.”<sup>3</sup> So one way of differentiating the operational from the analytical is to take a cognitive look at how and why people use the reports.

Though there are some clean divisions between operational and analytical reporting, the two areas share some fundamental requirements, such as how to access and understand the data, how to provide different constituencies authoring tools that fit their levels of skill and how to present and distribute the results. Because of differences in requirements, usage patterns and orientation, it is clear that a single approach to reporting for the enterprise will leave one or the other areas wanting. In order to address reporting effectively, a solution needs to address all of these requirements with separate approaches, but with a single unifying architecture to prevent gaps and disconnects over time. The Business Objects approach does offer the unifying architecture and has solutions to meet the needs of various audiences within an organization.

Figure 2. Business Objects - Meeting the Needs of All Users



<sup>3</sup>Edwin Hutchins, Culture and Inference. Cognitive Science Series, 2. Cambridge: Harvard University Press, 1980.

---

## Requirements for Reporting Solutions

The fundamental aspects a reporting solution that are common across all types of reporting are:

**Architecture:** How does the tool interact with authors/designers and requesters; where does the tool assemble the data and format it; how is the report distributed; how does the engine request data from sources; how does the tool upgrade gracefully with the other systems in a cooperative way?

**Engine:** The mechanism by which the tool requests data, merges it, interprets it, formats it and performs other needed manipulations such as calculations

**Authoring:** The part of the system that allows authors to design new report objects, modify existing ones and often, delete unneeded ones, while maintaining the integrity of the overall set of reports and report objects

**Data access:** The methods by which the reporting engine accesses and consumes data, including query generation; development of native drivers to application sources; adhering to standards as they emerge; direct connect to application layers of third-party tools or APIs; own metadata scheme where necessary

**Distribution:** Managing the output of the system for destinations, subscriptions, caching, aging, security and availability; interfacing cleaning with third-party distribution channels such as portals

**Development environment:** No multi-user development system is complete without a development environment that automates tasks like version control, change management, impact analysis, dependency analysis, profiling, testing and release control

While the basic components of a reporting solution should be common, the type of reporting they are meant to address causes different tools to optimize their procedures differently. For example, a tool primarily used to design forms that will be produced in paper and filled out by hand is primarily a mark-up tool and will not possess data access capabilities or very sophisticated distribution tools. On the other hand, a tool designed to produce summarized management information highly formatted for paper and electronic distribution would likely require advanced capabilities in all of the categories above.

### Timing

Timing and latency are also an issue. Many organizations today use a data warehouse or data marts (or both) to provide the bulk of their managerial, informational and analytical reporting. Some use them for operational reporting, too, but this is an area that is a little controversial. The original concept for a data warehouse was to house time series data, meaning, data that represented a series of time-stamped states or observations, such as the end of a quarter or monthly. Today it is very common for that periodicity to be a single day and, in many cases, sub-daily or the actual transactions themselves. Regardless of the size of the time slice, a data warehouse process is still an extract-and-load process, meaning, the data is never exactly real-

---

time. And the more complicated and convoluted the data warehouse structure, the greater the latency. For reporting that requires instantaneous information, a data warehouse is not an acceptable alternative<sup>4</sup>.

### **Pertinent Data**

If one is going to make the investment in mapping source data to a data warehouse, integrating it and transforming it and maintaining it over a period of months or years, then it stands to reason that that data will be used, viewed and massaged many times. Otherwise, there is little economic justification for the effort. Following that line of reasoning, in a world where things are characterized as constantly changing, it is very likely that the data needed today to answer questions may not be needed tomorrow. Data warehouse maintenance costs are significant and in many cases, reports are opportunistic. These situations argue for a solution without an intermediate data repository like a data warehouse. A variant of this scenario is a report that is derived from a data warehouse, a standard report, but with the addition of some new and perhaps temporary elements that are not part of the data warehouse structure.

A data warehouse offers some advantages, though, that have to be compensated for otherwise, such as locating and understanding the data at its source and presenting an understandable model to the report author. A reporting solution has to either have its own data mapping and transformation capabilities or have the ability to tap into the source systems through their own metadata.

### **Metadata**

A reporting system has to provide a mechanism for the designer to access data from other systems. There are many different ways to do this. In the simplest case, the tool uses a standard request language, such as SQL, to retrieve records from a relational database. In order to make sense of the physical data, either for a programmer to view like reference material, or for a data access tool to read in order to construct a query, the semantics of the data need to be exposed through the use of what is called metadata.

SAP R/3 contains a vast amount of metadata, information that describes the contents of the persistent data store as well as all of the other information needed for the operation and control of the various systems and subsystems. For example, the name of a General Ledger account used in financial statements in English in a division is metadata; the balance of that account at the end of month is data.

A broad discussion of the nature and practice of metadata is beyond the scope of this paper, but understanding the role metadata plays in reporting, especially SAP reporting, is important. In order to understand the contents of databases or other types of data files, report designers need a map to describe the meaning of the structures.

<sup>4</sup>Real-time data warehouses are a recent phenomenon, but not widely deployed. The tools that extract, transform and load data into data warehouses (ETL) have recently added real-time trickle feed capabilities, but acceptance and implementation of this capability is still very limited.

---

## Latency

Another attribute of a data warehouse is its integrated nature, meaning, data in the warehouse is sourced from many different systems and transformed and conformed into a single model. Reporting applications that do not require integrated data or, for that matter, require the reporting elements to be identical to those of the source system, cannot use the data resources of a data warehouse, they must read directly from the original system<sup>5</sup>. This is another problem for many operational systems, including ERP systems like SAP, that are constrained by physical resources (hardware, memory, etc.) and the added load of reporting can have a detrimental effect on their performance, which is critical. One solution is to confine direct operational reporting against the source system to those reports that are critical and the least intrusive.

## Logical versus Physical Design

When designing a complex system that uses a relational database, architects start with a logical design, one that captures the relationships and elements of the model without detailing the actual physical implementation. At the logical level, designs are driven by requirements and standards and the logical model, once accepted, should only change due to changing or emerging requirements. This provides continuity over time and allows for a common set of assumptions to be employed by all participants. However, for a myriad of reasons, the actual implementation of the design at the physical level is always very different and changes are more rapid and often, more radical. For example, a single logical design, applied to two different relational database systems, would take advantage of the innovations (and idiosyncrasies) of each. Releases of database software (or other software or hardware or standards) that are used in the application stack are staggered and don't conform to the software vendor's release schedule.

In addition, a logical model is mostly concerned with the conceptual aspects of the application and its design may be inefficient. For example, not all SAP R/3 data is actually stored in relational tables. Instead, some elements are calculated at runtime. In other cases, multiple table are combined into "pool" or "cluster" tables," which cannot be accessed by SQL. These tables compress several logical tables into one physical table. As a result, there is no direct correspondence between the SAP R/3 logical tables described in the SAP R/3 data dictionary, and the tables or views stored in the database

In many installations, customers add in their own customizing tables, which are invisible to tools that don't use the SAP application layer. Without visibility into these tables, a reporting solution would be unable to avail itself of critical pieces of information.

Another example is security. Creating and maintaining security for thousands of people, even at a role-based level, is a daunting effort. Imagine having to rework the entire arrangement every time there was even a small change in the physical data model, due to versioning or even just internal reference data changes to mirror changes in the organizational structure. By building the security on the logical model and allowing the application to manage the logical-to-physical translations, the task is manageable. By using the application layer, Business Objects leverages and honors the

<sup>5</sup> It is possible in principal to replicate the data from source systems into a data warehouse, but in practice, data is transformed as it enters the data warehouse. At the very least, keys are changed in order to integrate data from different sources with different key structures. If records are mapped one-to-one and key values are maintained (in non-key fields), operational reports could be generated, but at some measure of effort and expense that may not be justified.

---

security in R/3 rather than needing to replicate it. The same is true of reporting. Instead of addressing physical objects in the database directly, as some SAP reporting solutions do, Business Objects calls standard objects in the application layer (a combination of logical model and metadata), which provides a level of simplicity and continuity.

Discussions about integration, new features, enhancements, etc. typically take place at the logical level. Interaction with external systems is preferred at the logical level, too, because it isolates the conceptual framework of the application (which is more stable over time) from the nuts and bolts, which is more variable.

For all of these reasons, the ideal arrangement is that interaction between the reporting solution and the SAP modules occurs at a logical level and that internal SAP functions handle the translation to and from the physical (or virtual) layers. The persistent information that facilitates this process is called the metadata and the programs that manage the translation are part of the application layer.

---

## REPORTING FROM SAP R/3

---

(The discussion in this chapter is confined to operational reporting from R/3, not analytical reporting or reporting from SAP BW)

The SAP R/3 architecture is multi-tiered and composed of a number of components (this is vastly simplified):

- ▶ Database server: A relational database that stores and manages all of the persistent data in an SAP R/3 system.
- ▶ Application server: All of the programs that comprise that SAP functionality
- ▶ Presentation server: The programs that run on the clients' workstations that handle interaction with the application sever (this is the classic client-server arrangement; in mySAP, most clients are thin clients, but the separation between database, application and presentation services is the same)

Currently, there are a variety of methods for developing reports from R/3:

1. Generating reports directly against the tables
2. Extracting data from the tables into a data warehouse or mart
3. Using the BAPI interfaces supplied by SAP
4. Writing procedural code in ABAP
5. Using a native driver that provides an interface to the data through SAP's application layer

The first four options may have drawbacks. Only the fifth option, using a native driver, can provide the ease of use, flexibility and sustainability needed to deploy and maintain enterprise reporting for SAP. To illustrate this more fully, a detailed description of each approach is presented here.

### Generating Reports Against the Tables

The physical database schema of R/3 is comprised of over 10,000 tables, many with 200 or more columns and the meaning of the contents are often determined through many levels of indirection or abstraction. In addition, much useful information is not stored in the tables at all; it may be generated at run time. Furthermore, SAP R/3 does not implement in the physical schema all of the objects described in the logical model. Instead, cluster and pool tables are used which further frustrate the direct read access approach. In short, simply looking at the name of a table and its attributes is insufficient for determining its contents. This problem is multiplied by the need to join tables to gather different attributes. The tables change with version releases. Customization is applied at almost every site, typically through the use of customizing tables, whose semantics are encoded in the application metadata; just looking at the physical tables is rarely adequate.

### Extracting Data into a Data Warehouse

Building a data warehouse and developing the extraction routines to pull data from R/3 tables is subject to all of the drawbacks of the first option, in addition to the complexity introduced with a data warehouse environment, such as scaling, tuning, security, access and cost, to name a few. By

---

physically moving the data to another environment, the built-in security of SAP R/3 is lost as well. An alternative is to develop generic data warehouse models and extractors once and reuse them across many SAP customer sites, etc. This is clearly an approach that can be considered by software vendors and resellers, but would not find efficiencies in a single organization. SAP developed its own version of this approach, the Business Information Warehouse (BW).

For organizations not interested in the SAP BW option, Business Objects has developed its own data mart solution for SAP customers. This solution is not discussed in detail within this paper.

### **Using BAPI Interfaces Supplied by SAP**

In order to provide a more “open” environment for their clients, SAP developed a set of programmatic interfaces called Business Application Program Interface, or BAPI. Developing a program with BAPI is a complicated process, not only to conceive and execute, but also to maintain over time. Most report developers are not trained to be software engineers and even if they can develop a program, they may lack the background and supervision to develop it according to the standards of the organization.

BAPI's are not metadata, they are program calls to the SAP application and the embedded routines within SAP access the data and metadata as needed, but the metadata itself is not exposed to the BAPI programmer. Instead, the developer has to have a comprehensive understanding of the set of BAPI's (there are over 1000 of them covering the different modules), a background that is not accrued casually. In addition, BAPI's were not developed by SAP with reporting in mind, specifically, but rather, to provide a way for integrating SAP with non-SAP programs and for building customizations to the SAP portfolio. With the release of SAP's NetWeaver technology, which SAP will favor as its integration technology, the BAPI set will most likely not be improved over time. Business Objects does enjoy certified BAPI access to SAP data for its solutions for customers who require it.

### **Writing Procedural Code in ABAP**

Producing reports with ABAP is extremely hard to develop and maintain because of the limited capabilities of OpenSQL, which causes the programmer to develop complex routines. It also requires a fairly extensive knowledge of both the library of BAPI's and the much larger set of RFC's<sup>6</sup>. For standard reports this is not that much of a drawback because they can be developed once and released into production. In most organizations, though, the most critical reports are typically the ones most recently requested. This puts enormous pressure on the programmers, who are almost always in short supply due to the level of skill and specialization needed.

### **Native Driver for SAP R/3**

Rather than communicating with an application program (R/3) remotely through a prescribed set of procedure calls (BAPI or RFC), an alternative that can only be effected with the cooperation of the vendor is a native driver that acts as an adjunct to the application itself and performs the bulk

<sup>6</sup>An RFC in SAP is a Remote Function Call, similar to an RPC, remote procedure call. In the SAP system, there are over 10,000 RFC's and effective ABAP programmers must be familiar with a significant number of them.

---

of the report processing. By employing software tools that understand the application layer of R/3, this solution can provide abstraction to insulate the report developer from the complexity of the raw database tables, insure compatibility across version upgrades and bind tightly with services, such as security, that are already provided by the application program instead of having to replicate and attempt to synchronize them.

Of all the alternatives, the native driver is the easiest and fastest solution for ramping up operational reporting from R/3. It has the additional benefit of relieving IT from the chore of developing reports and it can integrate non-R/3 data easily. The next section describes the Business Objects reporting solution for R/3 in detail.

---

# BUSINESS OBJECTS REPORTING SOLUTION FOR SAP R/3

---

The Business Objects reporting solution for SAP R/3 is unique.

It does not require building and maintaining of a data warehouse or other additional structure, requires no additional metadata and it can access the full range of R/3 data as well as integrate data from other systems into a report.

Many reporting tools for SAP R/3 either query or extract data directly from the transparent physical tables of the relational database layer and bypass the application layer. This approach complicates maintenance, as there is no guarantee that the physical model will not change. The smallest change to the physical schema requires careful examination of all routines to insure that the impact is handled, typically requiring a tour through the development and test environments before reaching production, a process that can take weeks or even months for changes large or small.

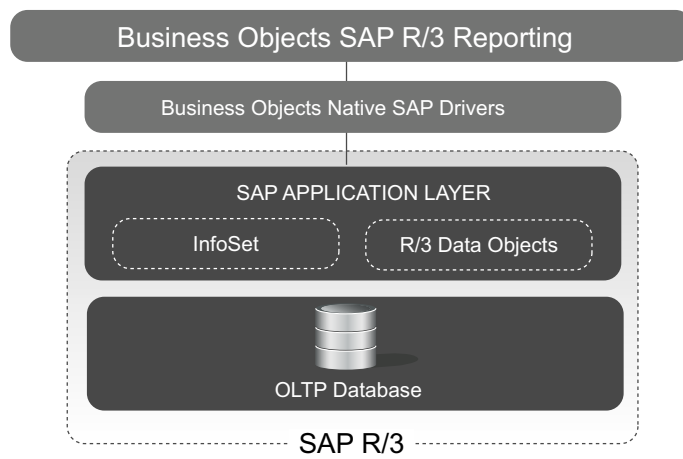
Reporting directly against the database also limits functionality because not all of the data in R/3 is in persistent tables or other structures that are visible. In addition, a substantial amount of setup time is required to investigate, understand and map any customizations a client may have made, eliminating any out-of-the box advantage. Reports routinely fail, as changes upstream are not reliably communicated downstream. In extreme cases, reports perform without it, but return erroneous results that are accepted as correct.

One alternative is to use the SAP-supplied application interface, BAPI, but for purely operational reports, this solution is both too costly and too time-consuming because of the level of effort and skill it place on the report author.

The Business Objects solution for R/3 is a native driver that attaches directly to the SAP application layer. It bypasses the BAPI, actually loads on the R/3 server as additional modules in reserved namespaces and tablespaces and appears as a background job. In fact, because it moves the formatting of the reports to its own server, it is actually less taxing on the R/3 server than ABAP itself. The Business Objects solution relies on the stable basis layer of R/3, which remains constant across versions, databases and operating systems. It has been consistently maintained and improved since 1998 and provides out-of-the-box functionality with minimal start-up time and because it eliminates data extract and movement and costly translations through multiple levels of transformation, it performs with minimal impact on the other operating components of the SAP system.

Business Objects customers report drastically reduced cycle times for the creation of reports over other methods, especially the most common approach, ABAP/4 code using BAPI and native RFC's.

Figure 3. Business Objects Reporting Solution for SAP R/3



---

# **BUSINESS OBJECTS REPORTING SOLUTION FOR SAP BW**

---

Part of the Business Objects reporting solution for SAP BW, Enterprise Reporting, is bundled with SAP BW. A more powerful version is available as an upgrade option. In addition, there are optional sets of functionality available for query and analysis, data integration and performance management, all designated as “Powered by Netweaver.”

## **SAP BW as the Reporting Platform**

SAP positions BW as the reporting foundation for all of the SAP components, but this may not be the best solution in every case. To the extent that purely operational reporting is required, such as invoices, bills of lading, explanation of benefits or even just up-to-the-instant inventory listings or mailing lists for other processes, both the latency and the underlying data model in BW may not be appropriate. Organizations that deploy SAP components, but run in a “best of breed” environment with software tools and packages from other sources, may find that an SAP-designed data warehouse is not the best solution. Even for those organizations that do rely predominately on SAP tools, integration of external data from value-added data suppliers can complicate the process, leading to a custom-built data warehouse solution instead of BW.

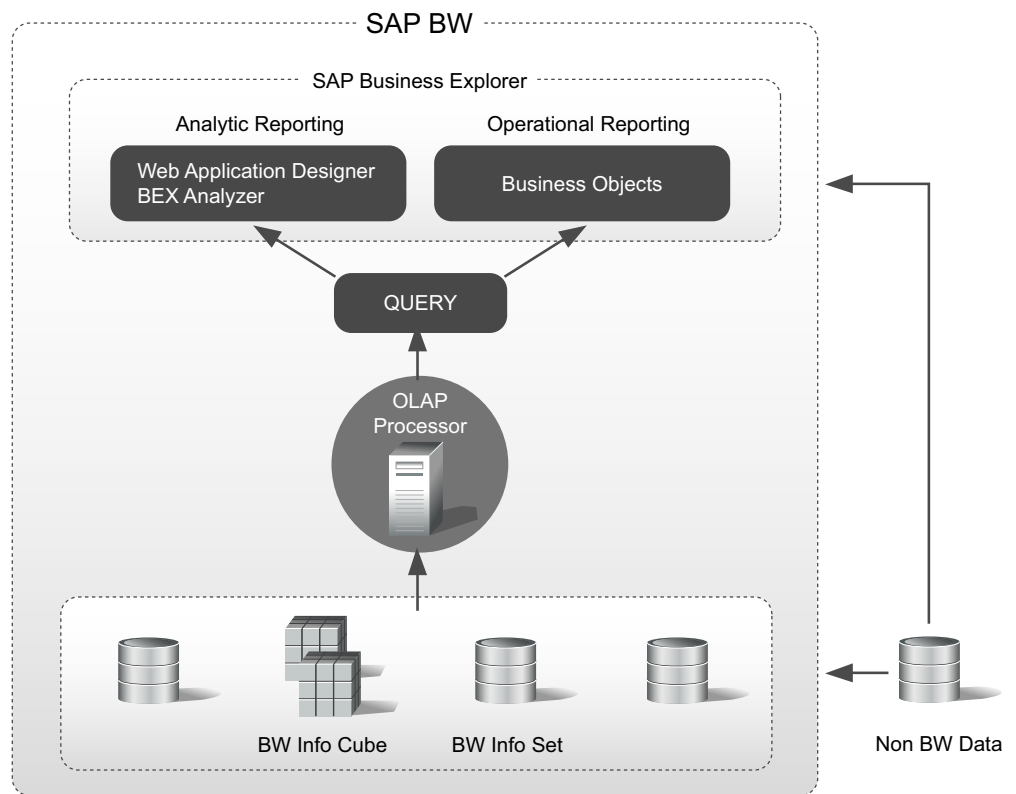
What is clear is that SAP will continue to expand both the functionality of BW as well as the breadth of data mapped to it from its expanding roster of solutions, so these distinctions may not hold up over time. As with any careful assessment, getting the facts as they stand and reasonable assessment of the future is key to making the right decision.

## **Business Objects Reporting for SAP BW**

The Business Objects solution for BW reporting is similar to the R/3 solution in terms of its access to the application layer for completeness, consistency and sustainability. Because volumes in BW can be huge (there are reported BW installations in the 10’s of terabytes now), the high scalability of the Business Objects architecture meets a crucial requirement.

Business Objects for SAP also allows for creation of reports and analyses pulling data from BW as well as R/3 and any other source, merged into a single report.

Figure 4. Business Objects Reporting Solution for SAP BW



---

## CONCLUSION

---

Reporting is no longer divided cleanly by operational and analytical categories, and the overlap between them is growing larger. Employing a set of tools from a single vendor to provide a comprehensive reporting solution is a smart move. Because the characteristics of the different uses and sources of information are so varied, a single interface or even application may not be possible at this time, but selecting for the greatest level of reuse and synergy across the various reporting, analysis and integration toolsets is well-advised. Those vendors that can address the entire range of requirements with a single architecture (and preferably single or at least interoperable layer of metadata) present the least risk of ROI drain or even outright failure.

SAP is a particularly challenging environment because of its broad reach of functionality, its after-the-fact approach to reporting historically and its lineup of products and modules stretching both horizontally and vertically through the enterprise application stack. No vendor can release an “SAP solution” overnight – it takes years of learning and understanding to gain an understanding of the nuance.

Today there is to a massive amount of data that cannot be directly accessed in R/3, an increasingly sophisticated set of information objects in BW that are similarly not transparent, a rapidly evolving set of hybrid operational/analytical applications, the xAPP's, that rely on BW for access to information. All of these issues point to the need for an integrated rationalized approach to building and sustaining reporting environments. Business Objects addresses all of these areas with its Business Intelligence platform, Business Objects XI.

**Americas**

Business Objects Americas  
3030 Orchard Parkway  
San Jose, California 95134  
USA  
Tel: +1 408 953 6000  
+1 800 877 2340

**Asia-Pacific**

Business Objects Asia Pacific Pte Ltd  
350 Orchard Road  
#20-04/06 Shaw House  
238868  
Singapore  
Tel: +65 6887 4228

**Europe, Middle East, Africa**

Business Objects SA  
157-159 rue Anatole France  
92309 Levallois-Perret Cedex  
France  
Tel: +33 1 41 25 21 21

**Japan**

Business Objects Japan K.K.  
Head Office  
Yebisu Garden Place Tower 28F  
4-20-3 Ebisu, Shibuya-ku  
Tokyo 150-6028  
Tel: +81 3 5447 3900

For a complete listing of our sales offices, please visit our web site.

► [www.businessobjects.com](http://www.businessobjects.com)

Business Objects owns the following U.S. patents, which may cover products that are offered and licensed by Business Objects: 5,555,403; 6,247,008 B1; 6,578,027 B2; 6,490,593; and 6,289,352. Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Enterprise, Crystal Analysis, WebIntelligence, RapidMarts, and BusinessQuery are trademarks or registered trademarks of Business Objects SA or its affiliated companies in the United States and other countries. All other names mentioned herein may be trademarks of their respective owners. Copyright © 2005 Business Objects. All rights reserved.