August/September 2005 Volume III Issue 4

SAPtips Journal

Creating a Well-Developed Master Data Management Solution in BW

By Arthur "Pat" Pesa, arthurpesa, inc

Editor Note: Pat Pesa delivers another one-two-three punch with his discussion of developing, refining, and managing a solid master data structure with SAP®'s BW product. He illustrates a well-thought out, cross-functional approach to craft lean, flexible master data that will support your R/3 solution across a range of modules. Pat describes a solution that utilizes the master data's unique characteristics in terms of time dependency, multiple attributes, and unique updating mechanisms to optimize performance and ensure data viability. Loading and working with R/3 is only part of the equation. The real challenge comes into play when loading data derived from a system outside the SAP framework. Special consideration must be given to capture the data and manipulate its attributes to align with the SAP master data structure. Pat offers up proven concepts and screen-shot examples to help you fine tune your master data machinery.

Introduction

As cross-functional integration becomes a primary focus in environments that have had BW up and running for a number of years, it becomes very important to pay attention to detail when designing the master data components. In this article we will focus on master data management in BW, and techniques that can be used to more effectively manage your master data when approaching cross-functional solutions. To clarify a couple points before going further, let's briefly discuss what we mean by cross-functional. In SAP® (and therefore BW) each functional area is linked together by components of workflow, user exits, and master data. This is to say that in R/3, SD (Sales and Delivery) and FI (Finance) are linked together using master data that has attribute links to FI and other functional areas. After we bring SD and FI data into BW, we also have the same master data that provides the links between these two functional areas. We have invoice numbers, sales documents, and other attributes of Sales that provide the correct links to FI. What this means in terms of master data in BW is that we have to ensure we do not degrade these links or attributes, or we lose data integrity. Over the course of several years, I have written many articles whose primary focus has been data integrity. It almost seems that aside from building more and more functionality into our BW environments, we spend most of the time ensuring that the data we provide to the end-user community has been maintained and stored under strict and regulated conditions, thus ensuring said data integrity.

This article will apply to all BW systems from 1.2B through NetWeaver™,and has roots and context in SAP R/3, and therefore all data environments. Don't get me wrong, this won't be a rambling article; on the contrary each and every example, pointer, and technique will be from actual BW examples. I would just like each and every reader to position this article in a more general data management context; for purposes of this article we just happen to be reading it with an eye towards BW.

To summarize what we will be discussing throughout this article, we will begin with master data management concepts in both SAP R/3 and BW. We will then focus on BW techniques to create and maintain valid and verified master data that works with all your functional data implementations.

Once you have developed a solid master data model, there are usually no real issues when extracting, loading, and reporting.

Master Data Management

In both R/3 and BW, we have strict rules that govern master data and its subsequent configuration and management. To begin with, master data has qualities that normal transactional data lacks. Time dependency, multiple attributes, and unique updating mechanisms are all qualities that make master data more time consuming to manage in design and development. Once you have developed a solid master data model, there are usually no real issues when extracting, loading, and reporting on the master data itself.

Master data in BW has several functions:

First, it provides identifying attributes of transactional data context.

Second, these identifying attributes are commonly used to tie different functional areas together. Consider the master data element Work Breakdown Structure. This commonly used master data element has many attributes, one of which is Investment Number. This attribute of Work Breakdown Structure, Investment Number, will tie a project number back to its initial and subsequent investment opportunities. Using this example, you can conceive of the myriad uses of master data in this fashion.

Third, the one key to using master data efficiently in this manner is time dependency.

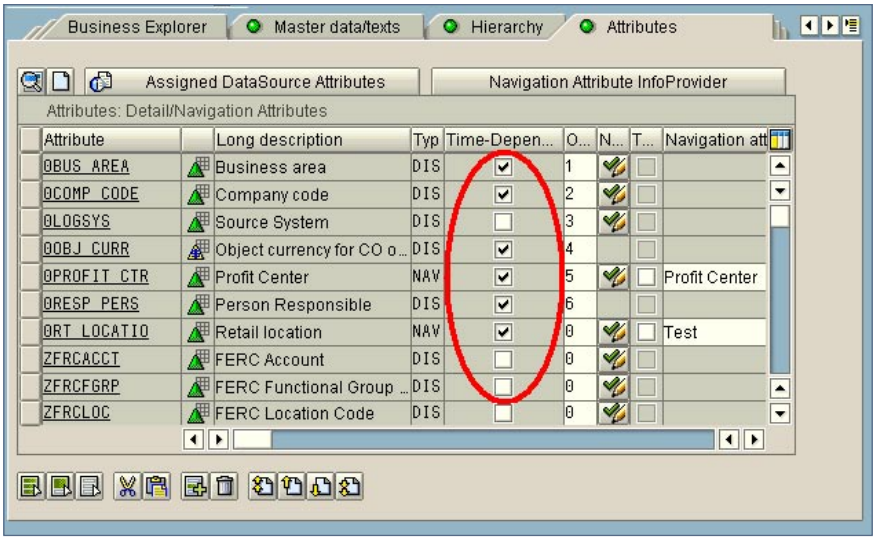

Figure 1: Time Dependent Configuration for Master Data InfoObject 0COSTCENTER

It becomes critical to understand how BW stores and maintains master data. Since there are time-dependent attributes, it would only be reasonable to expect time-independent attributes as well.

And, last of all, the one elusive concept, and one that is common to all data environments, is the law of diminishing attributes. I know I have mentioned this in prior articles, most notable the white paper titled "A Functional Approach to Enterprise Data Management in Business Warehouse", published in April 2004. What is meant by the words "law" and "diminishing" is simple. As a rule of thumb (and law), all data attributes are subject to decreased or diminished efficacy over time. Simply stated, data relationships change like everything else in the universe. One account number, referenced last month, may have changed when an asset was re-assigned or purchased. Capturing these time dependent changes is often a primary focus. Fortunately for BW, SAP R/3 already manages time dependency effectively, therefore assigning validity periods to a time-dependent attribute. In essence, all we have to do is make sure to check the time-dependent check boxes for attributes that are time-dependent, within the InfoObject configuration screen. Please refer to Figure 1 for an illustration of time dependency configuration for attributes of 0COSTCENTER.

In practice though, it is a bit more complex than a check box, especially if you have used this attribute within an update routine. It then becomes critical to understand how BW stores and maintains master data. Since there are time-dependent attributes, it would only be reasonable to expect time-independent attributes as well. As a result of the complexity surrounding effective technical data management at the database level, attribute versions are stored as follows in several tables.

- M-Table – Master data view
- P-Table – Time-independent attributes
- Q-Table – Time-dependent attributes
- Y-Table – SID Table for time-dependent attributes

Please refer to Figure 2 for a description of how these tables are displayed and related for a given master data InfoObject. Remember

that these options will only be active when your InfoObject is configured "With master data", which you can also see in this illustration. The "With master data" check box tells BW that you will be loading data into this particular InfoObject and therefore it needs to set up the technical data storage options.

Each of these tables is identical, with the exception of the "From Date" and "To Date" elements in the Q-Table. These values are updated directly from R/3 values, as specified during the configuration of specific master data in R/3. They serve to allow BW to manage the active and deleted versions of time-dependent attributes. When master data is loaded for the first time, a record is added to both the "P" and "Q" tables. This record is then flagged as active and is available in the reporting environment. If a subsequent master data load delivers the same record with a changed attribute value, the prior active version of the record is flagged for deletion, and the new record is inserted as modified. Thus, when an Attribute Change Run is executed, both records are managed according to the change flag. In this scenario, the initial record is now marked for deletion (and is therefore deleted), and the subsequent record with the changed attribute is modified (and therefore flagged as active) during the change run. Please refer to Figure 3 for an illustration of an attribute change run log.

This is relatively simple to understand when talking about R/3 and BW. However, when introducing master data from other systems into BW, it is imperative that a solution be developed to manage time dependency, or you will surely have data quality and integrity issues. One mechanism that I have used is to build corresponding logic in the non-R/3 source system that derives validity periods. This is

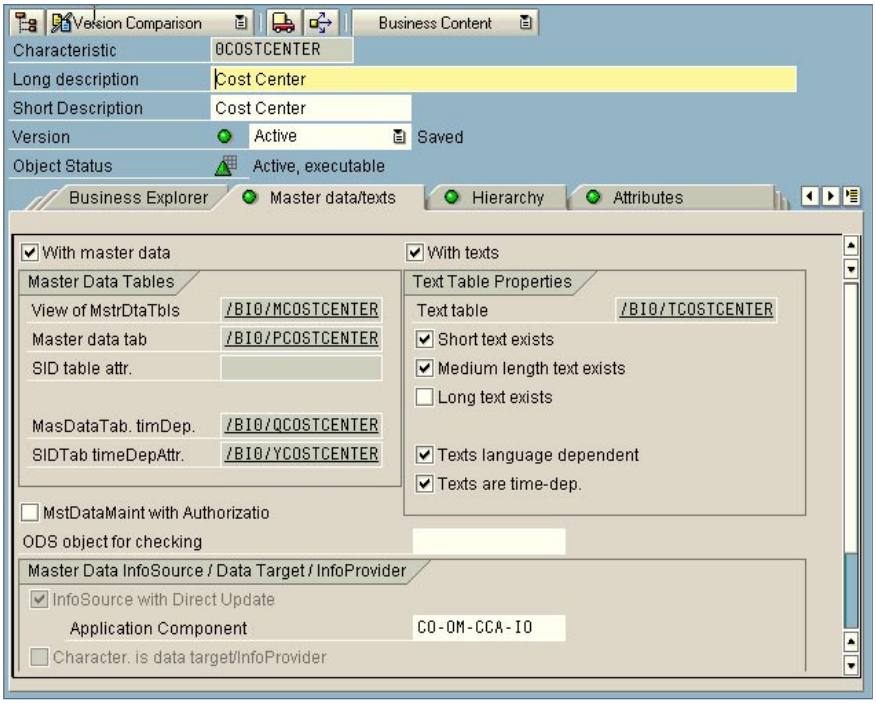

Figure 2: Master Data Technical Storage Table Structures

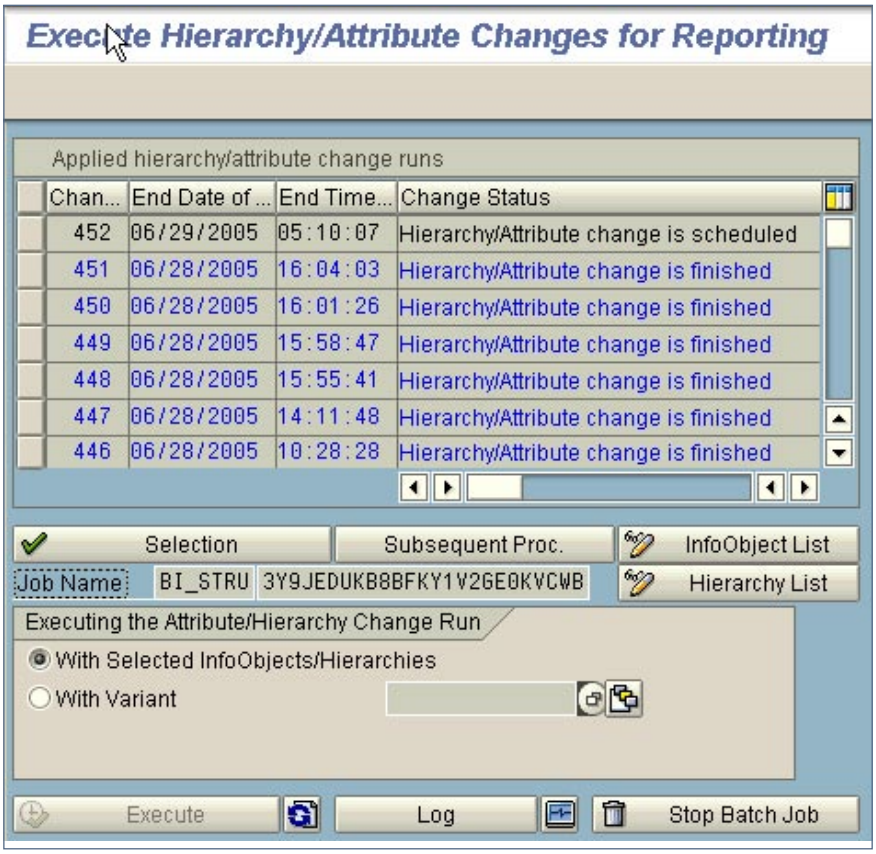

Figure 3: Attribute and Hierarchy Change Run

not always possible, but should be vetted against other methods, another of which is to ascribe an effective date when the master data is loaded into BW. Then, during subsequent loads, as each record comes across, it is checked against the existing data record. When changes are recognized, you programmatically assign a change flag. Then, when the attribute change run is executed, it can act on those flags. This mechanism is particularly effective when you can't determine validity periods in the non-R/3 source system, but know the attributes can change at will. Short of procedural and process related options, this may be all you can do to keep master data attributes assigned correctly over time. One note on this is that a deep understanding of master data management, in so far as change flags, date ranges, and attribute change runs, is needed in order to deliver this type of model.

Loading Master Data in BW

In earlier versions of BW (prior to 3.0), we only had direct update of master data. This meant the master data InfoObjects themselves were directly linked to their InfoSources. This did not allow for any manipulation in an update routine. It did limit what one could do with master data in BW, during loading. To get around this limitation, SAP introduced what is known as Flexible Master Data Update. This allows the development team to configure a master data InfoObject as a data target under InfoProviders, thus allowing them to create update rules and subsequent logic to manipulate data as it is loaded into data targets. Refer to Figure 4, Master Data InfoObject 0ASSET, configured as a data target.

You can also specify multiple data targets for a single InfoSource, update ODS targets, and preserve data context throughout the physical data layers (i.e., PSA, transfer rules, update rules, and finally data targets).

Another interesting and effective aspect of flexible master data updates is that transaction data can be used, and updated, into master data. This seems contrary to what we may understand master data to be, but consider the fact that the transactional sources have master data elements, some of which can be used to create master data entries, or may be used for update logic. In most circumstances, you will not use transactional data as a source of master data, but to have it available will certainly make life easier in complex situations.

Whether or not an InfoSource is flexible is determined during configuration. If you want to use flexible updating, you must configure the InfoSource as such. Please refer to Figure 5, Flexible Master Data Update Configuration.

Design Considerations

In order to realize the benefits of well-maintained and efficient master data, you have to design your overall solution to enable growth, integration, and consistency. For master

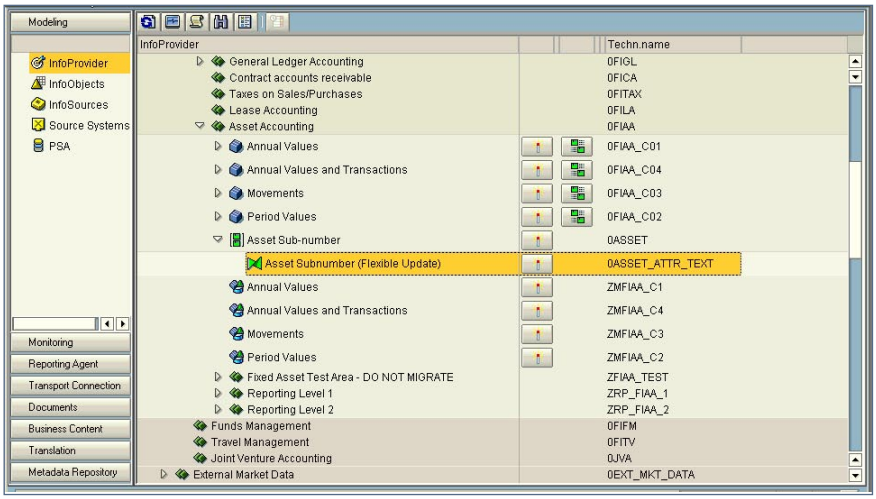

Figure 4: Master Data Configured as Data Target with Update Rules

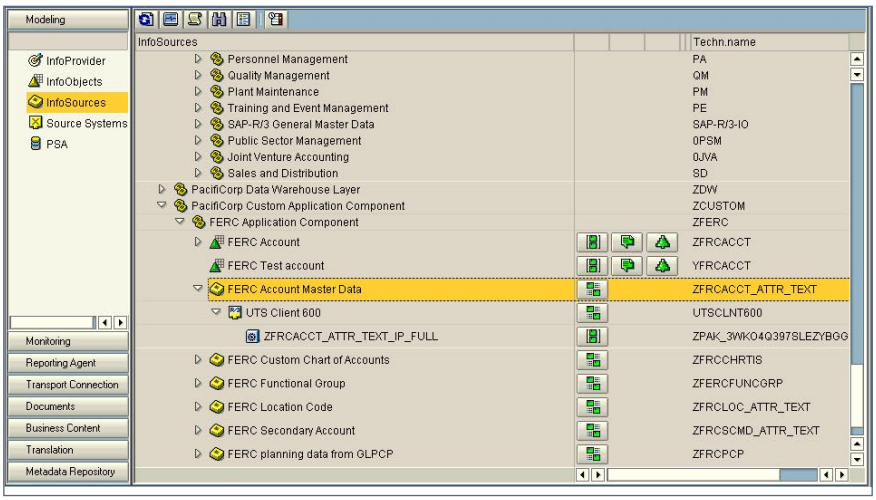

Figure 5: Flexible Master Data Update Configuration

data, this means you need to create a master build sheet, or design sheet, that captures all the initially used master data from the beginning. The main goal is to make sure you have defined each InfoObject of master data, and are able to reuse (and extend, if necessary) each master data element, as additional development is created in BW. As your BW environment grows, you will notice that the common links throughout the environment will be master data. Making sure to reuse master data consistently and throughout will ensure clean integration points between functional areas. Also you will need to provide well-thought-out and maintained definitions for each master data element. This includes standardization across functional areas. This is especially important when considering data from another non-R/3 system. It is very unlikely that non-R/3 data, such as customer and vendor, or asset, etc., will be configured and defined as though it were found in R/3. In these cases where you will be dealing with non-R/3 data, you will not only need to standardize the naming conventions for master data, but you will need to also approach data conversion. Converting master data coming from non-R/3 systems to look and feel like R/3 data will go a long way towards ease of maintenance and thorough integration. While these concepts are well understood, they are not always practiced. This design work will go farther than any development effort, and, more importantly, it serves as great documentation.

Conclusion

It is always worth writing about data management strategies. This not only allows the readership to perhaps get a second perspective on some of the issues currently being overcome, but allows me to re-think and further my own development of strategies and techniques. To a consultant, these strategies and techniques become the increasingly growing toolbox, something of value that makes one worth the expense associated with consulting. The development of these tools is not trivial and often takes years and years of real project work. The incidental reading of a colleague's strategies that define a particular toolset is perhaps the easiest and cheapest way to add to one's own toolbox, so to speak. That is why I chose this topic and have chosen similar ones in the past.

The development of these tools is not trivial and often takes years and years of real project work.

In this article, as I stated in the introduction, we covered data management with an eye towards BW. While all the examples are BW-specific, the concepts are broad-reaching in terms of data management within an information architecture. These same techniques could be said to apply to all data. However, for this article we used specific examples that have been used and proven. To reiterate the value of an efficient and well-developed master data management solution would be an understatement. Master data in BW, and R/3 for that matter, is the basis for all transaction data found within the system. It not only identifies working attributes of transactional data, but also has a role in deciding how data is processed and transformed as it is loaded into BW. It is used to link functional areas together when necessary. It also adds to the formatting of reports and queries.

The possible uses for master data are limited only by your creative development and design. However, to gain the benefits of master data in BW, you must first develop and maintain a thorough BW master data management solution.

Arthur "Pat" Pesa, *arthurpesa, inc.* Pat is a senior independent consultant specializing in SAP Business Warehouse and Application Link Enabling. He has over nine years of SAP R/3 experience, specializing in ABAP/BAPI development, Internet based order-processing systems, and Demand Management integrations with third-party Demand Planning systems. Pat has spent the last three years as a Business Warehouse consultant, specializing in technical leadership roles and solution design and development. Pat's email address is Arthur.Pesa@SAPtips.com.