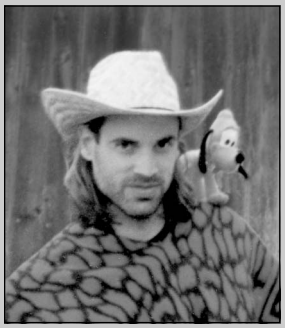
Get Double Productivity from the Classroom: Master the Art of "Speed-Learning" Popular Courses Such As BW, APO, CRM, and More!

Kurt Goldsmith



Kurt Goldsmith of ICM America specializes in identifying the simple and unambiguous relationships among complex systems. Applications of this approach within the SAP world include R/3 integration design and mystery resolution, report development (BW, CO-PA, Report Writer/Painter, ABAP), teaching, and project management.

(complete bio appears on page 64)

Whether taking a training class or reading a book on most SAP software application products, what at first glance might seem to be new and complex is oftentimes based on something familiar.

Newer SAP products such as BW data warehouse, APO, and CRM are no exception. I've sat in on both the BW205 and BW210 classes twice — once in the US and once in Australia — and witnessed eager and willing students develop feelings of information overload as more and more material was presented. As a result, the overall learning productivity experienced in the classroom for them was about half of what a small percentage of "know-the-game" students enjoyed — students who entered the classroom with a specific mission. Students who have mastered the art form of learning software-related topics in a hurry.

What's interesting about using the BW course as an example for my article is the following: Whether or not you have experience with data warehouse fundamentals, and whether or not you have experience with SAP software customization, you can *easily* learn enough about complex applications such as BW to (1) understand it, and (2) configure it. The secret is to not get forced into trying to do two things — take in vast amounts of BW-related information *and* think up a proper context for interpreting those terms — at the same moment! Nobody can do that. You'd become frustrated and probably stop trying to learn. Or, at best, your mind would latch onto the first task it thinks it *can* do, which in the classroom tends to be the "accomplishment" of the end-of-chapter, connect-the-dots, online exercises. Hmmm. Good for your ego. Not so good for your learning curve.

While the pages that follow use the mySAP BW module to illustrate the speed-learning approach that I have used the past six years, both for myself and for my clients' project team members, the techniques we will review apply to *any* SAP software.

The goal is not to eliminate the need for formal training. The goal is to double the hands-on learning productivity you get in front of a computer that has the application's software loaded, whether that's in a public class such as BW210, or back in the safety of your office cubicle. Instead of feeling "information overload" when meeting a topic's terminology for the first time, you're more likely to feel inquisitive and creative. Bold and dashing. You'll feel like a movie star! Oh, I think I got carried away there. Sorry about that.

For those who are not interested in learning BW customization, but are gazing fondly at the prospect of learning one of the many other mySAP solutions, the approach illustrated in this article is highly useful for you, as well. As you will see, the "trick" to helping yourself absorb large amounts of specific "how-to" information on abstract material such as software applications is to *not* begin your adventure with an "open mind." Instead, you should:

1. Start with a mission to become able to write out in your own words a summary of what the new application's work steps are.
2. Think up (identify) something already familiar to you, which has the exact same implementation work steps (context) as those in step 1.
3. Either during class or while reviewing the course content after the class, ask yourself three questions (which I will disclose later in the article) about some of the major new terms (vocabulary words) that you encountered.
4. Relax … and have a cocktail.
5. Renew your subscription to the *SAP Professional Journal*.

As we'll see, there is no requirement that either the summary or the answers to the three questions that you come up with are the 100 percent "correct" definitions. Instead, the mere quest to generate these answers represents an all-important first step toward mentally connecting what's identical between the "new" information and something else you are already extremely comfortable with. And that first step is all you need to double your learning productivity, as it yields the most interesting side-effect of turning your mind into an information sponge! Skip this first step, however, and instead simply greet new terminology one term after the other, and your mind more likely will turn into a soft bowl of mush.

The truth of the matter is that anybody can gain usable skill sets in SAP software applications, such as the BW data warehouse topic.[1] Anybody. Enjoy!

Agenda:

Section 1 —
Context, Context, Context! Summarize BW's Implementation Work Steps

Section 2 —
Identify the "New" with Something "Old" (Microsoft Excel Exercise)

Section 3 —
"What Is the Purpose of This Term?"

Section 4 —
"What Is the 'Create' Transaction for This Term Doing … and *Not* Doing?"

Section 5 —
"How Does the Term Interact with Other Terms?"

[1] By "topic," I mean the BW terminology and transactions that a person uses to design, develop, and implement BW-powered data analysis. What I do not "prep" anyone to learn in this article are, in my mind, the logically separate subjects of the strategic use of data warehousing technology in an organization, and the effective management of deployment issues such as access (security) and growth (scaling) of the data warehouse contents. For professional support in those areas, please contact ICM America's senior partner for BW projects, Julian Pasley-Smith, at 610-578-4808.

Section 1 — Context, Context, Context! Summarize BW's Implementation Work Steps

The entire success of speed-learning a new SAP software product depends on three things: context, context, and context. Your goal is to help your mind make the connection between what you're studying right now, and something that *already* makes sense to you. In other words, we'll make an attempt to define the "new" in terms of the "old."

The trick to creating a mental connection between old and new begins with a summary — using your own words — of what you think the topic's implementation steps are.

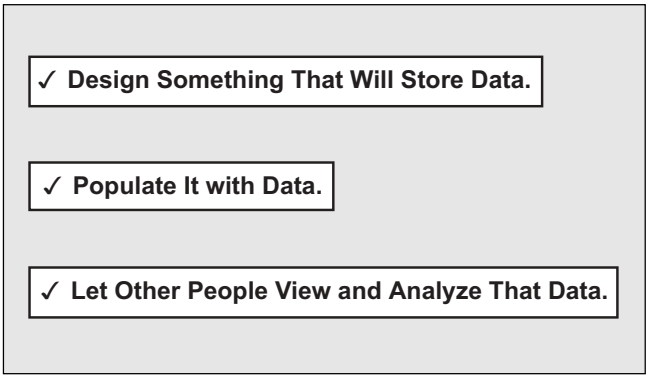
Will you be able to form a useful summary on your first day of class, or after your first hour reading SAP's online help? No, probably not. You'll have to wait until you've completed the class and/or reading. That's okay. Knowing that you need to summarize at the *end* gives your mind a focus *during* the class and/or reading! That's good. That's what we want … an active mind. And, as a result, instead of responding to your SAP instructor's "Any questions?" comments each morning with a blank stare, you may find yourself with a good deal more curiosity. I can almost guarantee it. In fact, I can boldly predict just what kinds of questions you'll be asking yourself!

✓ What are we actually doing, in non-technical language?

✓ What else can I think of that is similar to this?

As an example of this "summarization" step, I've listed in **Figure 1** my own version of a summary of the exercises and terminology presented in SAP's BW205 and BW210 classes, using the most unambiguous and technology-free work step descriptions that I could think of. It's better to form your summary statement with fewer words than with more, with unambiguous language rather than buzzwords, and with technology-independent vocabulary rather than "techie" terms. However, do not become overly fixated on doing this step "correctly." Getting the summary 100 percent "accurate" and 100 percent unambiguous is less important than using descriptions that you're personally comfortable with.

Figure 1 Summary of the Work Steps Presented in BW205 and BW210



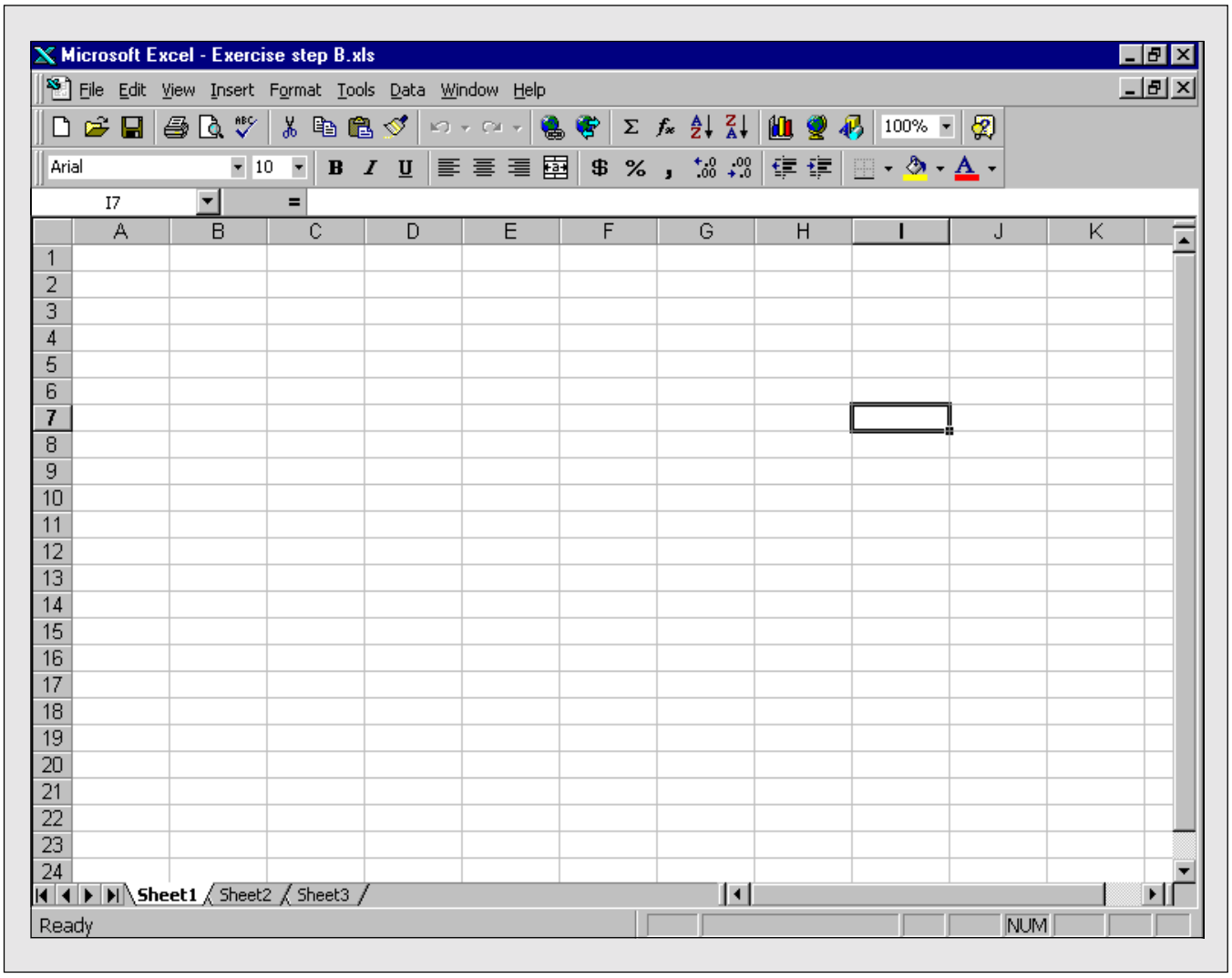
If in reading my summary you fear that I have actually over-simplified the topic, and that you have never before personally done work in those three areas (i.e., because they are actually too generic), then relax. In a few seconds, I'm going to make sure that *everyone* who reads this article can claim to have a hands-on comfort level with performing each of the three listed tasks.

After that, we can turn our attention during the remainder of the article to walking through an example of answering our three specific questions, in Sections 3, 4, and 5.

That example will aim at linking the BW terminology related to the first item in my summary statement — the "Design Something That Will Store Data" work step — to something "old." In this case, that something "old" will be the Microsoft Excel exercise I'm about to have you do.

And the payoff? Read on and you'll find out!

Figure 2 A Blank Excel Spreadsheet

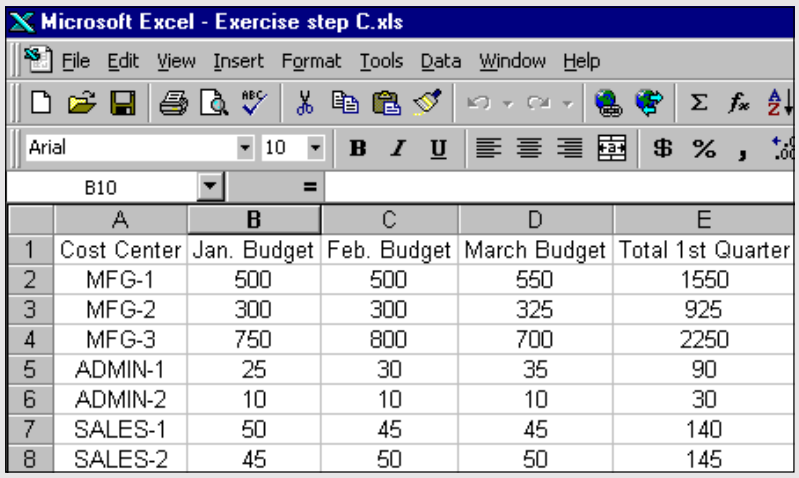


Section 2 — Identify the "New" with Something "Old" (Microsoft Excel Exercise)

The point of the following exercise is to prove conclusively that — even if you don't think so — you already know everything you need to know to quickly learn the BW application "how-to's" for each of the three summary work tasks I identified in Figure 1. These were designing a storage thing, populating it with your data, and letting other people view that data. Again, our mission is to link in our minds that which is new to that with which we're already comfortable. Therefore, I want to prove that everyone reading this article is already comfortable with each of the three work tasks I elected to form my summary statement with. Here are the exercise steps:

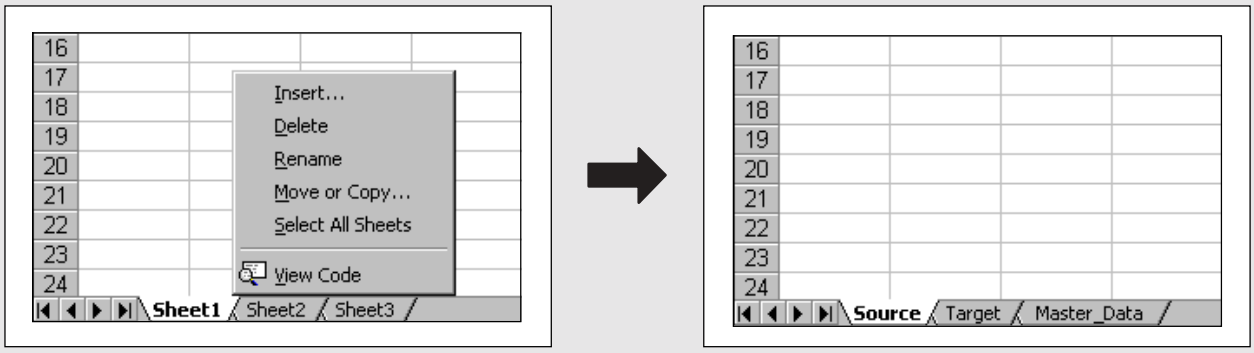
A. Assuming that your computer is on, launch the program Microsoft Excel. (If you don't know how, ask anybody in your company's accounting department for help.)

Figure 3 Texts and Numbers to Type into Your Blank Spreadsheet



	A	B	C	D	E
1	Cost Center	Jan. Budget	Feb. Budget	March Budget	Total 1st Quarter
2	MFG-1	500	500	550	1550
3	MFG-2	300	300	325	925
4	MFG-3	750	800	700	2250
5	ADMIN-1	25	30	35	90
6	ADMIN-2	10	10	10	30
7	SALES-1	50	45	45	140
8	SALES-2	45	50	50	145

Figure 4 What the Tab Labels Should Look Like at the End of Exercise Step "D"

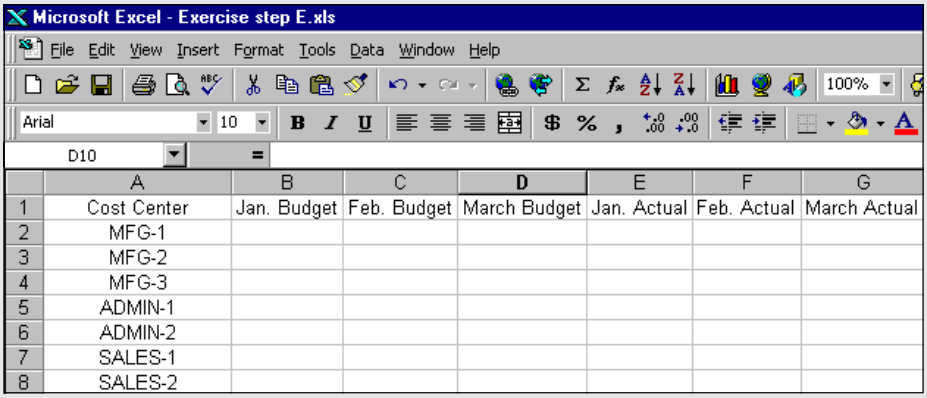


B. You should now be staring at a blank electronic spreadsheet that looks something like **Figure 2**.

C. Now, enter the texts and numbers shown in **Figure 3** into your blank spreadsheet.

D. Careful, now! We're about to get really fancy. Toward the bottom of the Excel screen are small, rectangle-shaped tabs. At the moment, the labels of those tabs should read simply: "Sheet 1," "Sheet 2," and "Sheet 3." If you position your cursor over any given tab and then press once on the *right* (not the left) button of your mouse, you'll see an option called "Rename." Use this option to change the label of "Sheet 1" to "Source," and of "Sheet 2" to "Target," and of "Sheet 3" to "Master_Data." When you're done, your spreadsheet should look similar to what's shown in **Figure 4**.

Figure 5 What the "Target" Tab's Spreadsheet Should Look Like at the End of Exercise Step "E"

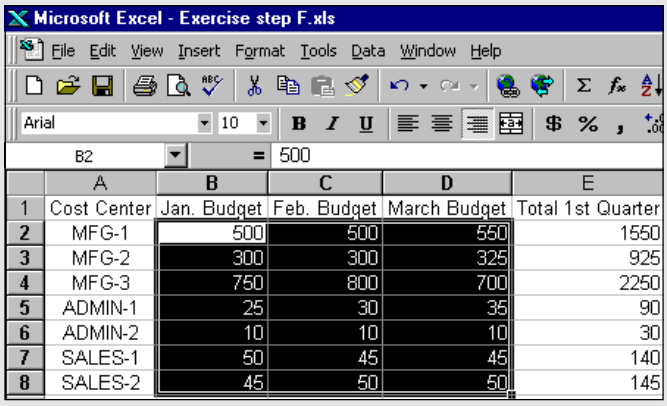


	A	B	C	D	E	F	G
1	Cost Center	Jan. Budget	Feb. Budget	March Budget	Jan. Actual	Feb. Actual	March Actual
2	MFG-1						
3	MFG-2						
4	MFG-3						
5	ADMIN-1						
6	ADMIN-2						
7	SALES-1						
8	SALES-2						

Figure 6 What the "Master_Data" Tab's Spreadsheet Should Look Like at the End of Exercise Step "F"

	A	B	C	D	E
1	Cost Center	Long-Text Description	Name of Manager	Name of Hierarchy Group	Cost Center Location
2	MFG-1				
3	MFG-2				
4	MFG-3				
5	ADMIN-1				
6	ADMIN-2				
7	SALES-1				
8	SALES-2				

Figure 7 What "Black Reverse" Highlighting Looks Like, to Indicate That You Have Successfully Selected a Specific Subset of Spreadsheet Cell Values to Copy



	A	B	C	D	E
1	Cost Center	Jan. Budget	Feb. Budget	March Budget	Total 1st Quarter
2	MFG-1	500	500	550	1550
3	MFG-2	300	300	325	925
4	MFG-3	750	800	700	2250
5	ADMIN-1	25	30	35	90
6	ADMIN-2	10	10	10	30
7	SALES-1	50	45	45	140
8	SALES-2	45	50	50	145

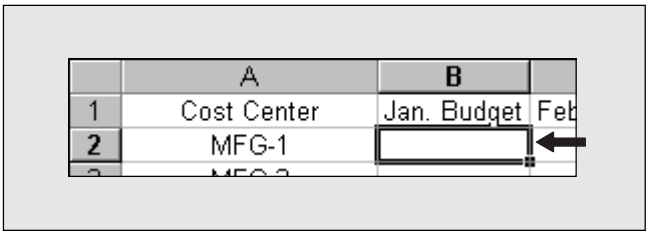
E. Mouse-click once directly on the "Target" tab, and then type text into that tab's spreadsheet as shown in **Figure 5**.

F. We'll do a similar task with the "Master_Data" tab. Mouse-click once directly on it, and then type text into that spreadsheet as shown in **Figure 6**.

G. Now, mouse-click once directly on the "Source" tab. What we're going to do is ask Excel to copy only the numbers in that spreadsheet's "B," "C," and "D" columns, without copying any of the text. You can do this by clicking one time in cell "B2," holding down the left button on your mouse, and then moving the mouse cursor over the area you want to copy. That area should show up in "black reverse," as can be seen in **Figure 7**. At that point, you can remove your finger from the left mouse button. Finally, to copy the highlighted area's contents to your computer's short-term memory, you can select "Edit" → "Copy" from the menu bar. Or, you can also simply simultaneously press the "Ctrl" and "C" keys on your keyboard. Note: When you give this command, Excel responds by flashing a dotted-line border around the cells that it has copied for you.

H. You now have copied the numeric values of 21 spreadsheet cells from the "Source" spreadsheet.

Figure 8 Where the Cursor Should Be During Exercise Step "H"



Your next step is to paste those values into an appropriate place in the "Target" spreadsheet. Do this by first clicking once directly on the "Target" tab. Then, click once directly in cell B2, as shown in **Figure 8**. Finally, choose "Edit" → "Paste" from the menu bar. Or, simply simultaneously press the "Ctrl" and "V" keys on your keyboard.

I. At this point, your "Target" spreadsheet should look quite similar to the one shown in **Figure 9**.

Figure 9 What Your "Target" Spreadsheet Should Look Like at the End of Exercise Step "H"

	A	B	C	D	E	F	G
1	Cost Center	Jan. Budget	Feb. Budget	March Budget	Jan. Actual	Feb. Actual	March Actual
2	MFG-1	500	500	550			
3	MFG-2	300	300	325			
4	MFG-3	750	800	700			
5	ADMIN-1	25	30	35			
6	ADMIN-2	10	10	10			
7	SALES-1	50	45	45			
8	SALES-2	45	50	50			
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							

Source | Target | Master_Data

If it doesn't, repeat the exercise from the beginning, because you've taken a wrong turn somewhere. If it does, save your entire Excel workbook to any folder you like, just as long as that folder is located on your company's network drive rather than on your computer's local hard drive (i.e., other people will have a tough time viewing your spreadsheet's data if the file is stored only where you alone can access it!). You can name the file "Kurt_Told_Me_To.xls", if you like.

Finished? Wow — you did it!

> **? *Question***
>
> *What is it that you just did? Would you believe me if I told you that you did everything that you need to do to design, develop, and implement BW-powered data analysis? You did. I promise. To prove it, let's review the summary points listed earlier in Figure 1 against the work steps we did in the Microsoft Excel exercise.*

Summary Point: Design Something That Will Store Data

You accomplished this in small steps during the exercise, via work steps "A," "C," "D," "E," "F," and "I." But, the important thing is that you accomplished it.

The "A" work step was to merely launch Microsoft Excel. This particular software application has built-in functionality that allows anybody to easily store data (i.e., just type something in, then click on the "Save" button). Sweet! But, that alone was not enough. Was it okay to store your data into any combination of cells, in any sequence whatsoever, in any "Sheet" of any "Workbook"? No, it wasn't. Makes it tough to find and use the data later on if you take that approach. That's why you needed to also identify three unique storage areas, as a way for keeping source data separated from master data, and both of those separated from target data (step "D").

You also had to identify a set of column names in each of the three spreadsheets (steps "C," "E," and "F"), in order to give meaning to the raw numbers (data) that were to be stored there. And, of course, at some point, you had to save your design work (step "I"). Was it easy to do? Hope so. Because, as you'll see, we'll be doing these *exact same work steps* when we "Design Something That Will Store Data" in BW. Only the terminology and the menu paths will change!

Summary Point: Populate It with Data

Your "Target" spreadsheet that you worked so hard to design wasn't all that useful to anybody until after work steps "G" (copy the budget data from the "Source" spreadsheet) and "H" (place that copied data into the "Budget vs. Actual" section of the "Target" spreadsheet) were completed. Although neither the "Actual" number values of the "Target" spreadsheet nor the "Master Data" auxiliary information about the budgeted Cost Centers were there at the time we saved our "Copy" → "Paste" work, we can certainly imagine that data being brought in at a later time, by some other person. In fact, this is what we'll see in BW.

Once we design our BW "things that store data," many different sources can be set up to populate it with transaction data or with master data. The updates from each of the different sources can occur at the same time. Or, they can occur at different times.

The truly important realization to make here, however, is that extremely specific instructions were performed during work steps "G" and "H" regarding what data to take, what data *not* to take, where to take it from, and where to "paste" it. These exact same kinds of specific instructions

need to be formed *anytime* data is copied from one place to another, regardless of the software program being used. There's nothing "new" or "complex" about that. If you can do it in Excel, you can do it in BW. I promise.

Summary Point: Let Other People View and Analyze That Data

Although work step "I" (save your file to the network drive) seemed so easy that even the 10-year old neighbor kid down the street could have done it, you were unwittingly accomplishing *two* important tasks in just a single step! Yes, you put the data onto the network. But, there's more.

You see, you had your choice to save that file as something *other than* an Excel file. Via the last four digits of the file name (.XLS, .TXT, .WK1, etc.), you controlled the type of "frontend" report tool that other people will later view that data with! This, along with other options for saving the file that I could have had you do (such as freezing the data to prevent edits/changes, or saving it as a PivotTable) greatly impacts other people's ability to do three things *with* your data:

- Decide for themselves which of the saved data **rows** they are going to view.
- Change which **columns** are visible in the report.
- Apply their own **sorting/subtotaling** preferences to the data.

We will similarly choose from among these *exact same* three "viewing impact" options when defining our reports for our BW saved data. Only the terminology and the menu paths that we use to design the reports will be different.

Again, there's nothing "new" or "complex" to learn here. If it makes sense to you in one instance (e.g., Excel), it will make sense to you in a second instance (e.g., BW).

Section 3 — "What Is the Purpose of This Term?"

In Sections 3, 4, and 5, we will walk through an example of answering three questions in response to new BW terminology, as a way to take advantage of the "sponge" effect that the summary statement generates for our minds.[2]

These three specific questions, when combined with that summary statement, can allow us to take in quite a lot of "how-to" detail, while keeping us safe from the side-effects of information overload.

We begin the demonstration of this technique by asking the first of the three questions — "What is the purpose of this (BW) term?" — as we look at each of the following terms:

- PSA table
- ODS table
- InfoCube Fact table
- InfoCube Dimension table
- InfoObject Master Data table
- InfoObject (not the same thing as an InfoObject Master Data table!)

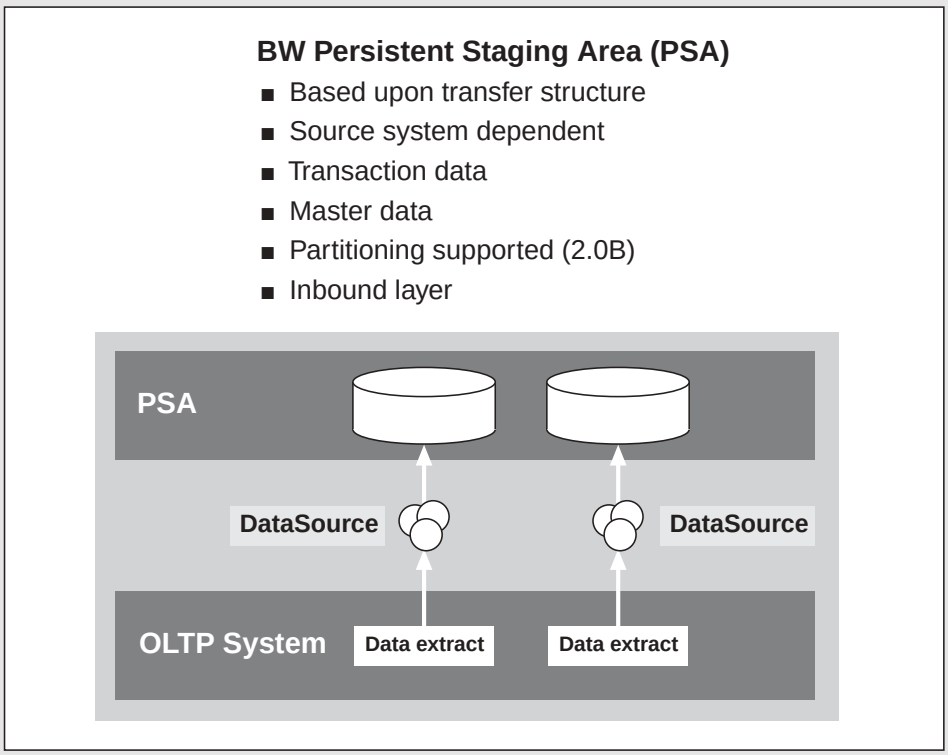
Hey — now how's that for a set of intuitive-sounding vocabulary words! (Take note of the implied sarcasm.)

Are you ready for me to tell you what a "PSA" is within BW?

---

[2] Due to space limitations, the BW-specific terminology I use as examples will only be that relating to the "Design Something That Will Store Data" item from my BW summary statement (refer back to Figure 1, if necessary). Please realize that we can use the same technique on all the terminology in the BW application, however.

Figure 10 BW210 Course Book Explanation of the BW Term "PSA"



- The data in the Persistent Staging Area (PSA) is based upon the transfer structure and is not transformed via transfer rules. The data is all system dependent. Data in the PSA can be transaction data as well as master data.
- The overall picture is that the PSA is actually the inbound layer of the BW.
- The PSA supports partitioning to segregate data according to time. That contributes considerably to the increase of the performance and has advantages when removing data from the PSA.
- The PSA is a transparent table for storing the detailed requests in the format of the transfer structure. The PSA is defined according to DataSource and source system and has the key fields of request number, packet number and record number. The request number, however, is not visible except in the change log.
- The update of the PSA and the InfoCube can run sequentially or in parallel.

Figure 10 shows the official explanation, which is taken verbatim from a recent BW210 course book.

Although it might not seem like it, the course book's definition in Figure 10 of the term "PSA" is actually quite good. It *packs* information. That particular trait is not the reason why the explanation does not help you to mentally connect the term to something you're already comfortable with. Instead, the reason that what's shown in Figure 10 feels somewhat irrelevant is that it *lacks* context that you, the student, have already learned in the past. And it has defined the term in question ("PSA") using *other* BW-specific terms (e.g., "transfer structure,"

"DataSource")! This combination … to a new student … is a sure recipe for information *overload*. And, for a slowdown in your learning process.

Luckily, we *have* some context that we've already learned in the past! We have our summary statement (in Figure 1). And, we have our Excel exercise. *Now* we can speed-learn, by asking and answering our first, pointed question against each of the BW terms related to "designing things that store data."

My examples are below, based on notes I took in my December '00 SAP-Australia BW210 class, but remember, our personal descriptions regarding the "What is the purpose of this term?" question need not be 100 percent accurate! That's not our goal at this stage of the learning curve. All we want in the beginning is merely to kick-start our minds' natural process of curiosity as to why these BW terms exist, how we work with the BW system to create them, and how each term interacts with the other. Just the same, take note on how much detailed information I was able to include in *my* answers that *you* are able to understand. And a lot of you probably haven't even had the benefit yet of visually logging onto a BW system!

✓ **PSA tables** — The acronym stands for "Permanent Storage Area." The purpose? These are "things that store data." This is where you will strategically want to keep exact copies of any form of data that you import into the data warehouse. End-user reporting is almost never fulfilled using the data that you'll store in these PSA things, for two reasons. For one, you might want to visually validate for yourself the data values that have been copied into the BW system before you put that data someplace else in BW where it can get viewed in somebody's report. Thus, to do that, you might find it useful to have some kind of a holding area (such as PSA tables), to which only you and a few other people have access. A second reason for not using PSA-stored data for end-user reports is because BW can receive copies of data from many types of sources. And, as a result, data when it first arrives into BW might have definition consistency conflicts with other BW-copied data. A common example of this consistency conflict deals with your company's product numbers. In one of your source systems, raw materials might be numbered "R-#####", while in another source system the exact same raw materials might be numbered "R#####" (no hyphen), or even just "#####" (no "R").

Important: For data stored in these PSA table things, we do not fix "problems" such as these "consistency issues" problems! Why not? At some point, we're going to move our PSA-stored data into the BW storage things that *do* feed our end-user reports. If we edit the data when it is in the PSA table, instead of merely editing *what we send* over to our ODS, InfoCube, and/or InfoObject Master Data tables, then we have a big problem for our report readers. We would have *no way* to prove (reconcile) where the data in our BW reports *originally* came from! Therefore, in these PSA storage things, we want *exact* copies of what we brought over from the various source systems. No "cleaning" or "fixing" allowed!

✓ **ODS tables** — The acronym stands for "Operational Data Stor(ag)e." The purpose? These particular "things that store data" are 100 percent optional in BW, and you might end up deciding that you don't need them. If you do decide to use them, you absolutely will want to apply data conversion logic that "cleans up" the data that's stored in the PSA, at the time when you transfer that information into the ODS tables you've designed. You do not want to actually store any "consistency issues" type conflicts here, because you may want to fulfill certain kinds of end-user reporting requirements from this ODS-stored data.

Typically, the question of whether to meet a report requirement via ODS-stored data vs. the other main option for BW-powered analysis (InfoCube-stored data) comes down to response speed for your report user. Simple requirements, such as "Show me a current list of all the partially delivered 'Open' sales orders," can be met with fast response times from ODS-stored data, and can include "drill-down" into individual sales order documents. Such "drill-down"

requirements can also be met via InfoCube-stored data. But, if you elect to store individual documents (such as each sales order) in an InfoCube, the sheer volume of that kind of stored detail might easily interfere with the InfoCube's main purpose. Let's look at that purpose next.

✓ **InfoCube Fact and Dimension tables** — The purpose? Namely, to allow very fast cause-and-effect analysis/research on historical transaction data. The term "cause-and-effect analysis" implies that the report reader will interact with the report quite heavily. He or she will not merely run the report, and then perhaps scroll up and down the pages to view what's shown. Instead, the reader will use functionality within the software tool (in this case, that tool is BW) to try to answer questions related to "Why?" For example: "Why is the company P+L for June only $100,000 when last month it was $200,000?" Or, another example: "Why is there an upward trend this year in the amount of employee 'sick leave' in our manufacturing departments, but not in our other departments?" Or, another example: "Why hasn't the on-time delivery percentage of our lingerie product line shown improvement, even though we implemented that new barcode scanner equipment in the shipping area four months ago?" (End of examples.) Regarding terminology, the terms "Fact table" and "Dimension table" refer to the two different kinds of information that our inquisitive report analysts will need when researching the stored data for insight into these kinds of "Why?" questions.

When we design our **Fact table**, we will decide on which specific area of the business (such as Human Resources' Employee Training Effectiveness, Cost Center Spending Efficiency, Sales Force Coordination Efficiency with Manufacturing Staff, etc.) this table is meant to offer insight into. Then, we will further decide on which specific measurements and/or statistics relating to that topic we will compile (e.g., dollar values, headcount, "Touch Time to Cycle Time" ratio, etc.).

When we design the **Dimension tables** that will link to our Fact table, we instead focus on organizing "Who/What/Where/When" kinds of information, such as Cost Center #, Employee #, Customer #, Plant Site #, Text Descriptions, Fiscal Period, Transaction Date, Ship-To Country, etc. into logically related groups. For example, if we wanted to include with our InfoCube measurements and statistics the fields called "Employee Name," "Employee Start Date," and "Employee Department," we would have the option to group each of the three fields into its own, separate "thing that stores data" (i.e., a Dimension table), and to give each of the three Dimension tables any name we liked (e.g., Names dimension, Starts dimension, and Dept. dimension). Or, we could decide to group all three of those fields into a single Dimension table (e.g., the Employee dimension). The choice is yours to make. But you can only make and attach 16 of these Dimension table things to your InfoCube's Fact table. And using a Dimension table is not an option. You must do it. The only question for you to answer is how many you will use — three, eight, twelve?

Design constraints: While there's not necessarily a technical limit to fear on the number of different Who/What/Where/When columns we can include in any single Dimension table we choose to create and link to our Fact table, there *are* some practical limits! For one thing, just because you have the technical ability to include a column in one of the InfoCube's Dimension tables does not mean that there's any logical way to acquire that information from one of the available source systems. So, that column would merely be filled up with blanks, which would be useless to our report readers. Secondly, it's important to keep in mind what happens when you include many different columns in one "storage" thing. Even in an Excel spreadsheet, for example, putting in a large number of different columns means that your spreadsheet's data will not be easy to work with, because the spreadsheet will increase dramatically in *both* width *and* depth (i.e., each unique combination of Who, What, Where, and When results in a new spreadsheet row of data!). This same "data growth" issue needs to be given our attention when we design InfoCube storage things, as well. We will want to limit the number of different columns we

choose to include in each Dimension table, to avoid a design that allows a huge number of unique "rows" of data to be created in that table, as that would lead to slow reporting performance.

Finally, quite a lot of Who/What/Where/When information can be *left out* of the InfoCube's Dimension tables, but still made available to our report readers. This is accomplished merely by choosing to link one column of any of the InfoCube Dimension tables (such as a column for holding "Cost Center" numbers) to a Master Data table that stores information about the values that are going to be stored in that column. Examples of this information would be the name of the cost center manager, the geographical location of the cost center, and the category of that cost center (e.g., Admin vs. Mfg vs. Sales). Overall report speed is a bit slower when we link like that than if we had simply elected to put all of our desired data in one of our InfoCube's available Dimension tables. But it does avoid the "data growth" problem. Thus, linking to a Master Data table can be a useful option. Let's look at Master Data tables next.

✓ **InfoObject Master Data table** — The purpose? We can design these Master Data table things to store potentially insightful "cause-and-effect" information about some of the Who/What/Where/When values stored in the InfoCube's Dimension tables. This can really help our report readers in their quest to find answers to their "Why?" questions, *while avoiding* the "data growth" issue that comes from including too many columns in the InfoCube. As an example of linking an InfoCube's Dimension column to a Master Data table, suppose that one row of data stored in an InfoCube was this: "$500 sale of one pair of Acme leather shoes to Kurt G. on June 15th, 2000." Our report user might find that particular "When" detail of June 15th, 2000 useful for his or her analysis task. But "June 15th" really could not have *caused* the $500 sale. That's where master data comes in handy. In an InfoObject Master Data table related to the "When" category that we've decided to call "Transaction Date," we can store details such as what kind of advertising we ran in the newspaper on June 15th, and what size the moon was at 12:01 that morning. Hey — maybe every time the moon is full, people react by running to the store to buy expensive shoes!

Okay, probably not. But, you get the idea. In the reports that we design to pull data from an InfoCube set of tables, we'll have the option of allowing our report readers to call into the report some centrally stored "master data" (i.e., possible cause-and-effect variables) related to some of the InfoCube's Who/What/Where/When columns. It is up to you, as the designer, to decide which subjects (materials, customers, dates, etc.) to store master data for. And, it is also your choice as to which types of information you want to include in each Master Data table. Great. Just one "heads-up" warning to look out for: The BW system actually gives you the option of up to three different kinds of Master Data tables that you can design! There is one to store "attributes" such as the name of the cost center manager, or the day/month/year of that cost center's most recent cost center manager change. There is one to store "hierarchies" such as a cost center grouping schema of Acme Corporation / North-South-East-West / North Admin-North Mfg-North Sales-North Maintenance / (etc.). And, there is one to store language-dependent texts, to allow you to have available to your report readers the text description of the cost center not just in English, but also in French, Spanish, Japanese, and so forth.

✓ **InfoObject** — The purpose? In two words: column header. In the same way that we won't have a very useful Excel spreadsheet unless we type in some column headers (as a way to give meaning to the raw numbers/values stored in the spreadsheet's rows), we also need to do this for our BW "things that store data." Although creating a column header in a BW "PSA," "ODS," "InfoCube," or "InfoObject Master Data table" is not as simple as merely typing in our desired texts, it is simple. In BW, these column headings are called "InfoObjects",[3] and the BW product comes pre-delivered with quite a large

[3] Although, in actual fact, the role played by the InfoObject is more than merely "column headings," that simple definition is good enough for our purposes right now.

number of these all ready for you to simply activate and use. Some are descriptive (like "Cost Center"), and some deal with measurements (like "Total Headcount" or "Total Budget"). And, of course, you are also allowed to create as many additional "InfoObjects" as you feel you need, if the ones delivered by SAP don't completely fill your requirements.

The biggest challenge, actually, is to avoid getting the term for column headers ("InfoObject"), mixed up with the term for a thing that stores master data information ("InfoObject Master Data table"). Not only are the two terms similar-sounding, but both also share just a single "Create" transaction. Let's use that particular customization transaction to demonstrate forming an answer to the second of our three "fast-learning" questions, right now.

Section 4 — "What Is the 'Create' Transaction for This Term Doing … and Not Doing?"

In Section 3, we saw how making an effort to summarize in our own words an entire topic (such as the BW module) allowed us to both ask and answer the question "What's its purpose?" to each of that topic's "new" terms. The amount of detailed information we were able to fit into our answers — while also avoiding information overload — seemed like a lot.

But, of course, once we feel comfortable that we know *what* a term such as an "InfoObject Master Data table" is, we won't necessarily feel comfortable logging on to a BW system and creating one. Aside from trying to figure out the correct menu path to use, there is an even bigger challenge. This challenge deals with the fact that many customization transactions in the SAP world are not particularly intuitive to use! Once again, this is where our context-related preparation work pays off for us, in our quest to understand the "new," by mentally linking it to the "old." Please say out loud the following statement:

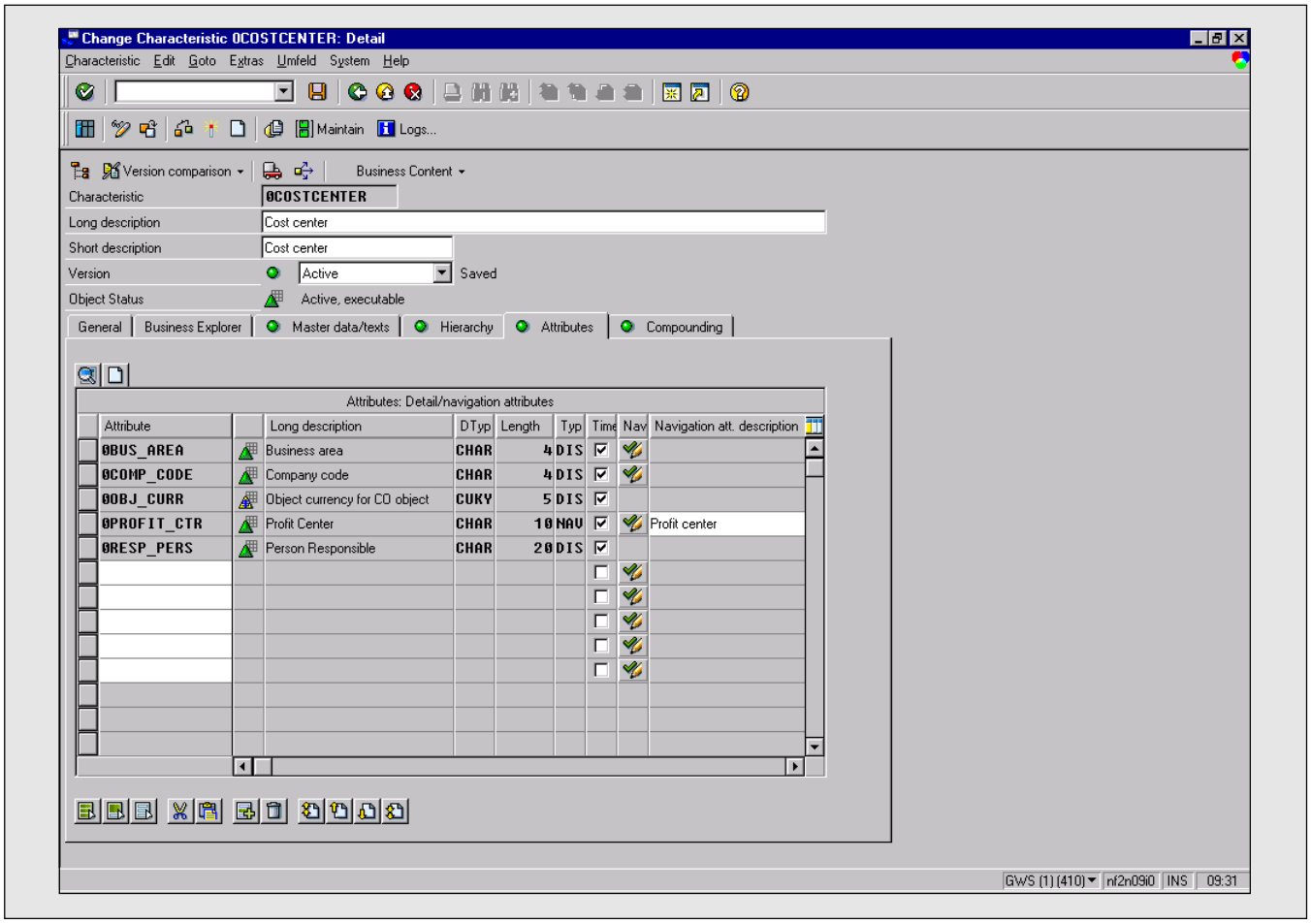
> *"Compared to Excel, your logic needs to flip inside out when working with BW. Instead of opening a spreadsheet to store master data and then typing in your desired column headers, you're going to be opening a column header and then typing in what you want the rest of the master data spreadsheet to look like!"*

For example, let's say you wanted to have a column heading called "Cost Center" available for some of the "things that store data" that you planned to create, and that this item did not already exist as an InfoObject in BW. No troubles. You'd simply create it yourself. You'd choose one of the several available ways to access BW's "Create an InfoObject" transaction, give a generic name to your InfoObject (such as "Cost Center"), and then save your work. Super! You'd now have yourself a column heading that you can use later, when designing the structure of a thing to store your transaction data or master data.

Now what? Well, how about designing that master data storage structure? Perhaps you'd like to design one that looks just like the spreadsheet shown in Figure 6, to hold auxiliary information about that "Cost Center" column heading you just created. You might be tempted to look for a BW transaction called "Create an InfoObject Master Data table." Forget it! It doesn't exist. Instead, you'll have to perform this step from *within* either the "Create an InfoObject" transaction, or the "Change an InfoObject" transaction. **Figure 11** shows how the column header and its Master Data table structure exist on the same BW customization screen.

Once you actually figure that out, your job is easy. All you do is type the names of other existing InfoObjects (i.e., these will be the column headings for your Cost Center Master Data table) into the appropriate place (the "Attributes" section) of the screen. At that point, you merely click on a few Yes/No checkboxes, click "Save," and BW automatically generates the master data storage things (tables) for you.

Figure 11 Designing the Structure of a "Column Header's" Master Data Table from the Same Screen You Use to Create That "Column Header"



Q: What is the "Create" transaction doing?

A: It's used both for creating the "column header" and then, at your option, up to three different kinds of Master Data tables (Attributes, Hierarchies, Texts) that can store auxiliary information related to that column header's subject.

Q: What is the "Create" transaction *not* doing?

A: It's not behaving in a similar way to how we create column headers and master data spreadsheets in Excel. In fact, the logic is exactly backwards. Instead of opening a thing that will store our data (i.e., a spreadsheet) and then typing in the names of the master data column headers we want, we open the main column header (e.g., Cost Center) and then type in what we want the storage thing to look like (i.e., the other column headers of the spreadsheet).

Section 5 — "How Does the Term Interact with Other Terms?"

In Section 4, we once again saw how making the learning preparation effort of forming a summary statement allowed us to take in a lot of information, by asking and answering a specific probing question. The particular question was on what the "Create" transaction for the term "InfoObject Master Data table" was and was not doing. In this final section, we demonstrate asking and answering the last question related to our speed-learning technique: "How does the term interact with other terms?" This question tends to also give us a lot of usable information, mostly because all SAP products feature a high degree of integration among the different pieces.

As a refresher, let's list once again the BW terms that we met in this article:

- PSA table
- ODS table
- InfoCube Fact table
- InfoCube Dimension table
- InfoObject Master Data table
- InfoObject

Of course, our answers back in Section 3 had such good depth that we *already* have useful insight regarding interaction. For example, we know that the PSA tables are used to store exact copies of the various kinds of data we bring into our BW system, and will not be used to meet end-user report requirements. We know that these PSA table things can act as feeder systems for the storage places that *will* be used to meet reporting requirements — the ODS, InfoCube Fact/Dimension, and InfoObject Master Data tables.

Similarly, we also already saw how the Fact table, Dimension tables, and Master Data tables can all link together, in order to produce *both* a happy InfoCube report reader *and* a stress-free database administrator (i.e., lots of Who/What/Where/When details for the reader, without generating lots of unique data rows to manage for the DBA — review Section 3, if necessary). All of this "interaction" detail from before is useful to us. If we had not identified it in our earlier work, we would definitely be asking about those relationships here, in this final section.

But rather than examine even more details about interaction among the terms, I want to use the rest of this section to once again stress that there really isn't a "100 percent correct" way to attempt to answer the three questions of the speed-learning technique. You, personally, might easily view this final question in a slightly different way than I have. There's nothing wrong with that. As an example, I've listed some additional "term interaction" ideas. All will help you to achieve the goal of assisting your mind's natural ability to cross-index the new to the old (i.e., as a way to absorb tremendous amounts of detailed information). In the end, your own personal experiences and history is what should determine your approach and your choice of responses.

✓ **Interaction Between the Term and … Itself?**

- Example: Can one set of InfoCube Fact and Dimension tables interact with a second set of InfoCube Fact and Dimension tables?
- Description of the interaction: It is possible to do this. When would you want to? Suppose that you've designed two different InfoCubes, "A" and "B." You designed "A" to track statistics and Who/What/Where/When details about your company's Purchasing patterns. You designed "B" to track statistics and Who/What/Where/When details about your company's Manufacturing patterns. Both "A" and "B," by themselves, support insightful research and reporting. But because your company has many materials that the Production Planners can choose either to make or to purchase, one of your report readers eventually asks you if it's possible to have a report that supports analysis of *both* Purchasing and Manufacturing, together. At that point, you have two options in BW: (1) you can design a completely new set of InfoCube Fact and Dimension tables, and then use that to store the needed data; or (2) you might be able to create a link between "A" and "B" for some reports, while leaving each one alone for the existing reports. No new data would need to be stored because you're not actually creating a new "storage" thing. The term in BW for this kind of linking is called "Creating a Multi-Cube."

✓ **Differences in Interaction Between One Term and One Event, If That Event Has Two Processing Options?**

- Example event: Populating an InfoObject's Master Data table.
- Processing options: You can either (1) copy the source data from an R/3 system into the BW system, or (2) copy the source data from somebody's Excel spreadsheet into the BW system.

- Do any interaction differences exist between the InfoObject Master Data table and the "Populating" event when processing option 1 is used vs. processing option 2?

- Description of the interaction difference: The answer is yes. There is an interaction difference. This difference can be easily understood if we think back to our Excel exercise, when we did our "Cut" → "Paste" work steps (steps "G" and "H"). In order to accomplish this correctly, we had to give Excel very specific instructions on what data to copy, what data not to copy, where to get the data from, and where to paste the data into. Although this "Cut" → "Paste" action using BW is not as simple as manually selecting a group of spreadsheet cells and then pressing "Ctrl" + "C" followed by "Ctrl" + "V" on our keyboards, it still requires merely those same specific four instructions.

  The interaction difference? In the case of processing option 2, the instruction regarding where to get the data is very simple for us to give to BW. We only need to type into an appropriate data entry field the exact directory path where the spreadsheet is currently located. For example, if the spreadsheet "Kurt_Told_Me_To.xls" was on our own computer's hard drive, we might type in a path such as "C:\Cost Accounting\MyFile\Kurt_Told_Me_To.xls". Simple. On the other hand, if we need to instruct the BW system to look for the source data in an R/3 system, our work becomes more involved. This is because R/3 systems don't have directory paths. Instead, they have identifiers known as "IP Addresses" and "Logical System Names."

  Furthermore, once we successfully instruct the BW system on how to find the correct R/3 system, we *still* need to give an instruction as to which table(s) to look for once it gets to that R/3 system! (That is, when copying data from a spreadsheet, only that one spreadsheet exists, but when copying data from an R/3 system, literally thousands of tables exist inside that system.) Therefore, we're going to need to learn about a BW thing called a "DataSource." This is a thing that resides in the R/3 system, and contains instructions about the specific table(s) to search in to find specific kinds of R/3-stored data.

  We'll also keep of a simplified copy of this "DataSource" table identifier in our BW system, because it is from within BW that we configure the "where to get the data from" instruction. Thus, although somewhat confusing, we'll see diagrams in our BW210 class that show that a "DataSource" thing exists in two places — in our BW systems and in our R/3 systems. Just remind yourself why it's in the diagram, and you will feel less confused. It's there because we need some way to both *give* the "where to get the data" instruction (i.e., in BW) and *execute* the instruction (i.e., in R/3).

Summary

This article began with the following premise:

> *You already know everything you need to know in order to effortlessly learn the mechanics and functionality of a mySAP module such as SAP BW. You just don't realize that you already know!*

The speed-learning technique demonstrated in this article is based strictly on the human need during learning to mentally connect "new" terminology and concepts to "old" terminology and concepts. The goal is not to *eliminate* the need for formal training, but rather to double the learning productivity that we experience *from* our formal training.

In Section 1, we formed a summary in our own words on what we think the topic is about. In this article, the example topic was SAP's data warehousing application, BW. And, my example summary reduced the topic of BW down to designing things that store data, populating those things with data, and allowing other people to view that stored data.

In Section 2, we verified that designing things that store data, populating those things with data, and

allowing others to view that stored data, are generic tasks that most of us are already quite comfortable with. We completed an exercise using Microsoft Excel to establish the fact that those tasks truly are common, and are not some brand-new thing that is specifically unique to the BW software.

In Sections 3, 4, and 5, we walked through examples of asking and answering three specific questions that are designed to take advantage of the "sponge effect" we get in our brains from the technique's first step, from Section 1. Those three questions are "What is the purpose of this term?", "What is the 'Create' transaction for this term doing and not doing?", and "How does this term interact with other terms?" What we saw is that the trick to generating extremely detailed knowledge via those questions has a lot more to do with using words in our responses that make sense to us, and a lot less to do with trying to be 100 percent technically accurate with those answers. This is because complete accuracy is less important during the initial stages of the learning curve than mentally linking "new" to "old" is.

Anybody can quickly learn advanced topics, such as the how's and why's of enterprise-wide software applications. Anybody.

If you would like more information about the techniques that I've written about, you may e-mail me at the address shown. If you would like onsite help with speed-learning SAP topics for either your end users or your project configuration "super users," please contact Beth Dietrich, ICM America's Team Training supervisor, at 610-578-48043 or BDietrich@ICM.de.

Kurt Goldsmith specializes in identifying the simple and unambiguous relationships among complex systems. Targeted relationships include the exposure of unavoidable formulas, hidden tradeoffs, and buried context. Complex systems include organizations, software applications, and organizations that hope to use software applications to increase their long-term profitability. Applications of this approach include SAP R/3 mystery resolutions (all modules), report development (BW, CO-PA, Report Writer/ Painter, and ABAP), teaching, constraints diagnostics (bottleneck measurements, decision-making modeling), and project management.

When not wearing his SAP consulting hat, Kurt can be found either at the local vegetarian grocery store, in the bubble bath, or in his home city of Austin, Texas. A 1995 graduate of the U.T. Austin MBA program, Kurt gained initial SAP project work through Thom Morgan and Warren Norris of SAP America's Dallas office, multi-national project work through John Wade and Bill Arrigo at IBM in New York, and eventually multi-module and multi-application integration design work through Winni Hesel and Julian Pasley-Smith at the Malvern, Pennsylvania, offices of ICM America LLC. You can contact Kurt with questions or comments about this article at KGoldsmith@ICM.de.

About ICM America LLC

International Consulting Munich (ICM) has offices globally, and specializes in the full suite of R/3 and SAP-related product implementations. For information on and assistance with SAP's e-commerce options — including "Business-to-Business" and the "mySAP Workplace" — please contact Managing Partner Winni Hesel at the ICM America offices (610-578-4807). For information on and assistance with SAP's backend products — including the Business Information Warehouse, APO, and Customer Relationship Management applications — please contact Managing Partner Julian Pasley-Smith at the ICM America offices (610-578-4808).

