

SAPtipsJournal

Unlocking the Value of your SAP® Investment

Undocumented Features...Timesaving Solutions...Best Practices

Journal | Training | Consulting | Document Library | Tip of the Month | Software Solutions | SAPtips Community

COVER STORY

- 8 ♦ Logistics/MM**
Mastering Material Masters Part I - Material Types: Make Them Work For You, by Jocelyn Hayes, SAPtips Director of Training and Consulting. Are you mastering Material Masters—or are they dominating you? Jocelyn Hayes gives you a sneak peek at a new class that will help you get a grip on this tough-to-tackle subject.

HR

- 15 ♦ What Is Time Evaluation and Why Do We Need It?**, by Satish Badgi, SAP HR Consultant. What is time? In SAP, it means different things for different employees. Satish Badgi shows you how to keep it all straight.

- 20 ♦ Creating Your Own Infotypes in Organizational Management – A Guide for SAP® HR Consultants and Developers**, by Rehan Zaidi, Siemens Pakistan. Create useful OM Infotypes with this handy guide from Rehan Zaidi.

FINANCIALS

- 25 ♦ FI/CO**
Unlock Payment Processing Benefits Using SAP®'s Lockbox Capabilities: Part I, by Anurag Barua, Independent SAP Consultant. Lockbox Payment Processing - Get started today and save money on the money you make!

BW

- 31 ♦ Understanding DataStore Objects (DSOs) and Their Assorted Flavors in BI NetWeaver™ 7.0**, by Anurag Barua, Independent SAP Consultant. There are some key differences between DSOs and their predecessor—ODS. Anurag Barua points out the changes, plus what to expect in this current release.

SOA

- 37 ♦ Building the SOA City – Part III: The Magic of Asynchronous Processing**, by Axel Angeli and Lynton Grice. We multi-task, so why shouldn't our systems do the same? Axel Angeli and Lynton Grice demonstrate how this is possible in SOA.

BASIS

- 44 ♦ NetWeaver 7.0 Web Application Server: Defined**, by Eric Walter, SAP Consultant. Get straight on the terms and architecture of NetWeaver 7.0 Web Application Server. Eric Walter provides the details.

ABAP

- 47 ♦ Getting By with What You've Got: How RFC Can Be Your Friend**, by Roger Myers. Sometimes the answer to a problem is right under your nose. Roger Myers shows how Remote Function Calls – an oft-forgotten technology – can help with an emerging issue in SAP.



Cheryl A. Cave, Managing Editor

From the Managing Editor

The arrival of fall brings cool nights, falling leaves, and the Thanksgiving holiday. Here in the U.S., we tend to believe it is our exclusive holiday; yet many cultures the world over celebrate and give thanks for a bountiful year's harvest. At SAPtips, we are grateful for our dedicated subscribers and consulting/training clients, and our world-class contributors.

From the Editor continued on page 2

DEPARTMENTS

- 4 ♦ SAPtips Training and Consulting**
6 ♦ Software Solutions
7 ♦ User News from ASUG
69 ♦ SAPtips Associate Editors



Publisher: Andy Klee
 Director of Publications: Laura Donovan
 Managing Editor: Cheryl A. Cave
 Editors: Emil Marx
 Colleen Low Larkin
 Graphic Designer: Mark Hunter

Klee Associates, Inc.

Table of Contents continued on page 2

53 ♦ SAP Mail Function Module Cookbook: A Guide for Developers and Consultants, by Rehan Zaidi, Siemens Pakistan. Trigger Email notifications automatically in SAP—Rehan Zaidi demonstrates how!

65 ♦ To ASP or Not to ASP?, by CJ Rhoads, EMT Associates. Rent or own? Before you consider going with an Application Service Provider (ASP), see what CJ has to say!

CIO Corner

60 ♦ Turning ERP on Its Head: Can SAP eSOA Change Companies' Approaches to ERP?, by Jon Reed, JonERP.com. How can eSOA work for you? Let Jon Reed explain.

SAPtips ONLINE

Be sure to visit the SAPtips Document Library to download these new white papers.

ON LOGISTICS / SALES AND DISTRIBUTION

How to Determine Whether to Forward or Backward Schedule **By Thomas Woelfel**

This paper is an excerpt from the SAPtips Mastery Level Workshop book for Sales and Distribution. We sincerely hope that the information included here will help you take the riddle out of forward/backward schedule determination.

ON ABAP

Creating Search Helps for Fields via ABAP Programming **By Rehan Zaidi, Siemens Pakistan**

Here's yet something else to be thankful for: Search Helps within various ABAP applications. Rehan Zaidi defines and lists the advantages of using this functionality to help make your program executions and data entries easier for you and your users. So prime your finger above the F4 key and get ready to search for the answers you need.

From the Editor continued from page 1

Opening our October/November issue is a cover story that will be welcomed by those of you who are ready to master Material Masters. Jocelyn Hayes, SAPtips Director of Training and Consulting, begins this series with a focus on Material Types (page 8). We are excited that Material Masters – An Overview, will be our next SAPtips University Mastery Level Workbook. We'll be announcing more on that next year, so keep an eye out.

We're addressing one of today's hot topics – lockbox processing with SAP – in an article by Anurag Barua (page 25). As is true of all SAPtips contributors, Anurag brings impressive credentials (including 15 years experience in SAP) to the table. In this first installment of a two-part article, Anurag presents an overview of lockbox processing and some configuration steps. Follow Anurag's sage advice to reduce payment float and expedite dollars to the G/L, and who knows...your boss may be motivated to thank you with a nice holiday bonus. Anurag also contributes to this issue with helping us understand DataStore Objects! We believe you'll also be thankful for our continued investigation of SOA with a third installment of Axel Angeli and Lynton Grice's series on Building the SOA City, on page 37. This time around, Axel and Lynton describe the magic of asynchronous processing along the SOA highway. If you need a macro (rather than micro) view of SOA, you'll want to read Jon Reed's observations of the eSOA trends on page 6. You'll find technical advice from Roger Myers as he explains how RFC can be your friend (page 47) and from Eric Walter as he helps you get a grip on WebAS (page 44). As usual, there's a feast of tips, facts, and shortcuts at the SAPtips table.

From the Editor continued on page 3

From the Editor continued from page 2

As we write this, our fall SAPtips University workshops in Dallas are in progress. We are pleased that the attendance on these classes has grown significantly since we first launched the University a couple years back. We'll be in Dallas again in the spring for another round of Mastery Level Workshops. Meanwhile, we can provide onsite training and/or consulting services to help you get the most out of your SAP investment. Contact Jocelyn Hayes at 877-832-2594 x 122.

Whether you're augmenting, optimizing, or commiserating...at SAPtips, we're thankful to be here for you through all your ups and downs.

Sincerely,

Cheryl A. Cave,

Managing Editor, SAPtips

Cheryl.Cave@SAPtips.com



Want a quicker way to access new articles and white papers on the SAPtips Document Library main page?

One of our loyal readers made a suggestion, and we liked it. Hope you will, too!

We now have more choices when you access our Document Library page. No more navigating to the desired category and searching for the desired article. Now you can link directly to the new articles right from the main page.

The August/September 2007 issue of SAPtips is available.

[Table of Contents](#)
[Download entire issue](#)
[Preview Previous Issues](#)

Individual Articles:

- COVER STORY - HR - [Payroll Go-Live in SAP: Planning and Managing the Cutover](#)
- LOGISTICS AND PRICING - [Lafite or Express-o? Interface SAP Shipping with Express Delivery Services](#)
- FINANCIALS - [SAP's New General Ledger in ECC 6.0: Document Splitting - Part III](#)
- FINANCIALS - [Sea Cap P01: Exploring the Tidal Wave of Power in CKAPP01](#)
- BW - [Rock Those Enterprise Reports with BEx Report Designer](#)
- SOA - [Building the SOA City - Part II: Programs to Business Processes](#)
- BASIS - [Just the Fax, Ma'am: Simple Front-End Faxing from SAP](#)
- ABAP, JAVA and J2EE - [The "FOR ALL ENTRIES" Construct: A Guide for Developers and Consultants](#)
- CIO CORNER - [Midsize Companies and the "Human Side" of SAP Implementations](#)
- CIO CORNER - [Optimizing Financial Investment and Technology Use for Business Health](#)

Plus...New White Papers Now Available!

Every issue of SAPtips comes with additional white papers, too lengthy for the Journal. It's like getting a bonus issue, free, with every edition of SAPtips!

- ABAP, JAVA and J2EE - [REQ Programming Primer: A Guide for SAP Consultants and Developers](#)
- CIO CORNER - [Writing a Business Case for SAP Products and Upgrades](#)
- HR - [Mastering PNP and PCH Macros: A Guide For SAP HR Consultants and Users](#)



*They didn't
build it to
capsize at
the first sign
of bad weather.*

When you've got something strong, built to explore and conquer new worlds—wouldn't you want to use it for all it's worth?

SAPtips can help you get the most value from your investment in SAP® software.

We offer:

- **Mastery-Level Consulting Services** www.SAPtips.com/Consulting.asp
- **Custom Training** www.SAPtips.com/CustomWorkshops.asp
- **SAPtips University** www.SAPtips.com/Workshops.asp

**Contact Jocelyn Hayes at Jocelyn.Hayes@ERPtips.com,
1.877.832.2594, ext. 122.**



**Discover how far you can go
with your SAP system!**





We're flying high over our great turn-out for the November session of SAPtips University. Start making your plans now for April 2008, destination: DALLAS

We'll be offering:

- **Reporting and Modeling in BW**
- **Basis and Security**
- **Sales and Distribution**

And introducing: SAPtips Material Management Overview

We also have several other classes in the planning stages – we'll announce those as soon as we can!

Just go to www.SAPtips.com/WorkshopSchedule.asp for more details, course descriptions, and how to register!

All SAPtips workshops are presented in a group-live environment. Click on the course links above to view detailed course descriptions, including objectives, prerequisites, who should attend, and CPE credit and continuing education credit information.

Here's what one person who attended our public workshop in Dallas, Texas, had to say about the value of SAPtips training:

"The course and instructor were very beneficial in our deployment of SAP; others should indeed take this class."

– Brian J. Pullin, Sr. Network Engineer, PCF, Inc.

What could SAPtips do for your next project? Call or email Jocelyn Hayes today to find out!

**Jocelyn.Hayes@ERPtips.com
1.877.832.2594 ext 122**

Klee Associates, Inc. is registered with the National Association of State Boards of Accountancy (NASBA) as a sponsor of continuing professional education on the National Registry of CPE Sponsors. State boards of accountancy have final authority on the acceptance of individual courses for CPE credit. Complaints regarding registered sponsors may be addressed to the National Registry of CPE Sponsors, 150 Fourth Avenue North, Suite 700, Nashville, TN, 37219-2417. Web site: www.nasba.org.



Each of our SAP® complementary software products helps you extract even more value from your investment in SAP software. Visit www.SAPtips.com/SoftwareSolutions.asp to learn more.

Or contact Gabriel Ramos at Gabriel.Ramos@ERPtips.com or Toll-Free at 1.877.832.2594 ext 170 for a free demonstration or to speak with one of our software partners.



RFgen – Data Collection Solutions

An RF bar-coding solution that's low cost and easy to implement? With RFgen software, you can be up and running as quickly as 30 days from now! And, with a greater ROI than other RF solutions. RFgen's integration with SAP has been certified by SAP. RFgen is the only data collection software solution to offer a full production free trial download that can be up and running in less than 1 hour.

www.SAPtips.com/DataCollection.asp

High Availability

iTera High Availability – For the Information You Can't Afford to Lose

When disaster strikes, where will your business-critical information be? Be it a hardware failure or the wrath of a Category 5 hurricane, High Availability for iSeries offers you a reliable failover solution that allows you to continue on with business. Isn't it time you looked into iTera—an affordable solution for the information you can't afford to lose?

www.SAPtips.com/HighAvailability.asp



ACOM Solutions:

We can do this the hard way, or the EZ way!

ACOM helps you manage your documents and payments electronically, so that the paper jungles can be banished forever! Create, capture, distribute, manage, or store documents easily with EZeDocs and EZContentManager. EZPayManager allows you to manage payments seamlessly, while EZConnect extends your capabilities to EDIXML translations. Put them altogether, and you've got an easy-to-use, secure Integrated Document Management solution that lets you say good-bye to fax machines, tedious bookkeeping, and yes, that shin-buising, finger pinching, over-stuffed nightmare that is your filing cabinet.

EZeDocs: www.SAPtips.com/Document-Management.asp

EZContent Manager: www.SAPtips.com/Document-Management.asp

EZPayManager: www.SAPtips.com/payment-software.asp

EZConnect: www.SAPtips.com/edi-software.asp



Here are the upcoming events through the end of 2007:

November 2007

Start Date	Event	Event Type
November 20, 2007	RESCHEDULED: Update on Tank Scheduling in APO Webcast	Education/SIGs
November 27, 2007	Planning with APO (SNP, PP/DS) and xLPO Webcast	Education/SIGs
November 28, 2007	ASUG New York City Metro Area Chapter Meeting	Chapters
November 28, 2007	RESCHEDULED: ASUG Presents: SAP & TechniData Roadmap to Address REACH Compliance	Education/SIGs
November 29, 2007	ASUG Minnesota Chapter Meeting	Chapters
November 30, 2007	ASUG Wisconsin - Chapter and the UWM Center for Technology Innovation Workshop	Chapters

December 2007

Start Date	Event	Event Type
December 4, 2007	ASUG Connecticut Chapter Meeting	Chapters
December 5, 2007	ASUG Michigan Chapter Meeting	Chapters
December 5, 2007	SAP ERP Upgrade Symposium	Education/SIGs
December 5, 2007	CRM Customer Value Networking Summit	Education/SIGs
December 6, 2007	ASUG Pennsylvania - Philadelphia Chapter Meeting	Chapters
December 13, 2007	ASUG Missouri - St. Louis Chapter Meeting	Chapters



Mastering Material Masters Part I - Material Types: Make Them Work For You

By Jocelyn Hayes, SAPtips Director of Training and Consulting

***Editor's Note:** Mastering Material Masters—hard to say, even harder to do. Jocelyn Hayes has been tackling this issue for the last eleven years, and even she admits, there's a lot to learn – way too much for one article! In this first article in a three-part series, Jocelyn focuses on Material Type. Read up as she delves into the configuration and setup of this very important component in Material Masters. And if you want even more, keep an eye out for our announcement of our SAPtips Mastery Level Workshop - Materials Management Overview.*

Introduction

Back when I first started SAP® consulting eleven years ago, one of my first SAP projects had me leading the Material Master conversion. At the time, I thought, this should be easy – right? WRONG! Material Masters are the “Masters of Logistics” and if you set them up incorrectly, you will lead your company down a path of inefficiency and inconsistency. Many articles have been written about best practices in Material Masters and here I will try to focus on just one component of configuring and setting up Material Masters – the material type – the base of the base of logistics. Configuration of the material type determines what data will be required and the purpose of the material when you create the Material Master.

This article is the first installment of a three-part series on Mastering Material Masters. Later installments will explain how to configure the screens to streamline Material Master entry for the end user, and a white paper detailing Best Practices of Material Master deployment and maintenance.

Overview of Material Types

When you create a new material, you have to select the industry sector and material type. Let's forget about the industry sector for now – I have yet to see a client put that field to good use. The material type, on the other hand, drives the entire use and usefulness of the Material Master. It determines the purpose of the material – is it configurable, is it a raw material? It determines whether the master number is assigned internally or externally (check out my upcoming paper for some guidelines on the infamous “internal vs. external”

assignment question). The material type determines the fields you see on the various user views (tabs) and the order of those views. The material type also determines the procurement type (in-house or external) and whether changes in material value are updated to the Financial Accounting module.

Material masters are the “Masters of Logistics” and if you set them up incorrectly, they'll lead your company down a path of inefficiency and inconsistency.

Many standard material types exist in the standard SAP system. I would highly suggest you select the one that fits the material with which you are working and plan to copy it to your own custom material type. The example in this article will be to copy from a KMAT (configurable material) to a ZMAT.

Copy Standard Material Type to Custom Material Type

If you have users who create only a single material type, you can further simplify Material Master entry by assigning that material type to a special Create transaction. To do so, follow the path in the IMG:

Logistics – General > Material Master > Basic Settings > Material Types > Assign Material Types to Special “Create” Transactions

Note: The menu path (see Figure 1) in SAP remains the same (the material type description you see in the IMG does not change the material type you see in the SAP menu). For this, you will have to request your SAP Administrator to either change the menu or create a Z transaction for your use, which you can also have them add to the menu.

Field Selection

I remember the first time I went through and looked at all the fields in the Material Master – it was overwhelming.

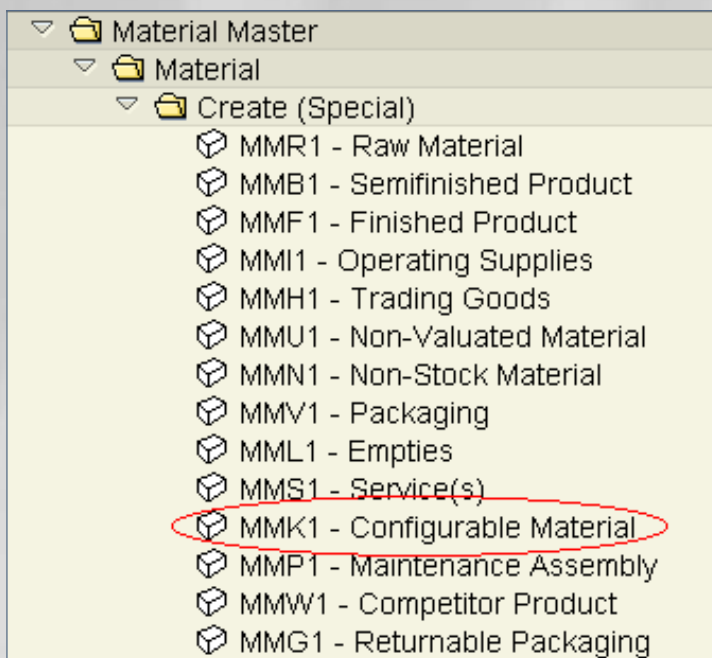


Figure 1: SAP Material Master – Create Special Menu

These days, my first goal when I arrive at a client is to see how many fields in the standard Material Master I can eliminate. Now, SAP configuration is a bit daunting in this area and the IMG Help is, let's just say, "light".

I err on the side of lots of custom configuration with the end goal of making the screens more user friendly and less error prone. Let's face it, if you were a new SAP user and had to look at twelve user views in the Material Master, but only had to populate ten actual fields, you would immediately question the intelligence of the system – it seems quite wasteful to the end user.

In the steps that follow, we'll discuss Assigning Fields to Field Selection Groups, Maintaining Field Selection for Data Screens, Defining Industry Sectors for Industry –Sector-Specific Field Selection, Defining Plant-Specific Field Selection and Plant-Specific Screen Select, and Defining Lock Relevant Fields.

Step 1: Assign Fields to Field Selection Groups

This is really SAP data that you should not attempt to change. However, if you have created your own custom fields that you want to appear in the Material Master

The screenshot shows the 'New Entries: Details of Added Entries' screen. The screen has a title bar with the text 'New Entries: Details of Added Entries'. Below the title bar is a toolbar with icons for creating, saving, and deleting. The main area contains a form with the following fields:

- Field name: A text field with a yellow background.
- Field attributes (industry and retail): A section with three checkboxes: 'Propose field cont.', 'Maint. status', and 'ALE field group'.
- Field attributes (retail only): A section with three checkboxes: 'Restrict matl cat.', 'Copy field content', and 'Incl. initial values'.

Figure 2: New Field Selection Group Entry

– this is where you would assign them to a field selection group. To add one, click New Entries, and complete the data in Figure 2. Use Field Section Group numbers 111-120 as these are the customer-designated values for Field Selection Groups (SAP has reserved the others).

You can also group all the fields that you wish to hide together into one field selection group to make them easier to hide. I'll explain this further next step.

SAP assigned long texts to a dummy field (i.e., LTEXT_BEST is for Purchase Order Text) so that you can define the entire screen as hidden, displayed, mandatory, or optional.

Step 2: Maintain Field Selection for Data Screens


In the next section of this article (Material Types), I will discuss the configuration for the material type – see the discussion in that section under “General Tab” to see where you will be assigning the Field Reference to the Material Type. The Field Reference determines which fields are hidden, displayed, mandatory, or optional when you create the Material Master.

The Field Reference can be for a material type, an industry sector, or a transaction code. When creating a Material Master, all of these Field References are taken into account when determining whether a field is hidden, displayed, mandatory, or optional. Table 1 shows the convention for Field Selection control.

For example, if you are entering a Material Master for Industry Sector M, which has in its Field Reference to Hide Field X, and material type KMAT, which has in its Field Reference to Require Field X, then Field X will actually be hidden because Hide trumps Required.

Priority	Function	Character
1.	Hide	-
2.	Display	*
3.	Required	+
4.	Optional	.

Table 1: Field Selection Control

To create the Field Reference, it is wise to copy from an existing Field Reference rather than starting from scratch. The selection of the Field Reference from which you copy is up to you, but I suggest you use the Field Reference that is the default from the material type from which you copied to create your custom material type. In my example, I copied material type KMAT, which carries with it Field Reference KMAT. I copied material type KMAT to ZMAT and I will copy Field Reference KMAT to ZMAT as well. To copy a Field Reference, select the Field Reference as shown in Figure 3 and click the copy  icon.

You will now be on the “Change View ‘Field Selection for Data Screens’: Detail” screen (see Figure 4), from which you can define which Field Selection Groups are hidden, optional, mandatory, or displayed.


Field selection (Field selection group 1)					
Field ref.		Hide	Display	Reqd entry	Opt. entry
KMAT		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

Figure 3: Selecting Field Reference to Copy
















Change View "Field Selection for Data Screens": Details						
 New Entries     						
Field reference:	ZMAT					
Sel. group		Hide	Display	Reqd entry	Opt. entry	
1		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	
2		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	
3		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	
4		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	
5		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	
6		<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	

Figure 4: Change View ‘Field Selection for Data Screens’: Detail Screen

You can click the Where Used icon  to see which Material Types use the Field Selection – see Figure 5 for an example of the Where-Used screen. Note that for the screen shot in Figure 5, I had already assigned the Field Selection ZMAT to the Material Type ZMAT.

You can also drill into the detail of each Field Selection Group (click ) to see which fields are assigned to each Field Selection Group. Figure 6 displays this screen.

Step 3: Define Industry Sectors for Industry – Sector-Specific Field Selection

As discussed in Step 2, in addition to the material type driving the behavior of a field when creating a Material Master, the Industry Sector also drives this behavior. If you want to create additional Industry Sectors, you can do so in this step. You can either reference an existing Field Reference or you can go back to Step 2 and copy a Field Reference to a new one for your own use. To create one, simply select New Entries and complete the screen as shown in Figure 7, specifying the Industry Sector, Description, and the Field Reference.

Step 4: Define Plant-Specific Field Selection and Plant-Specific Screen Select

Another driving value of determining field behavior is the plant. The plant can be assigned to a Field Selection in this step. To add a new plant, click New Entries and enter data as shown in Figure 8.

You can also restrict the maintenance status in this step. The maintenance status is simply the views, tabs, or user departments available for entry on the Material Master – see Figure 9.

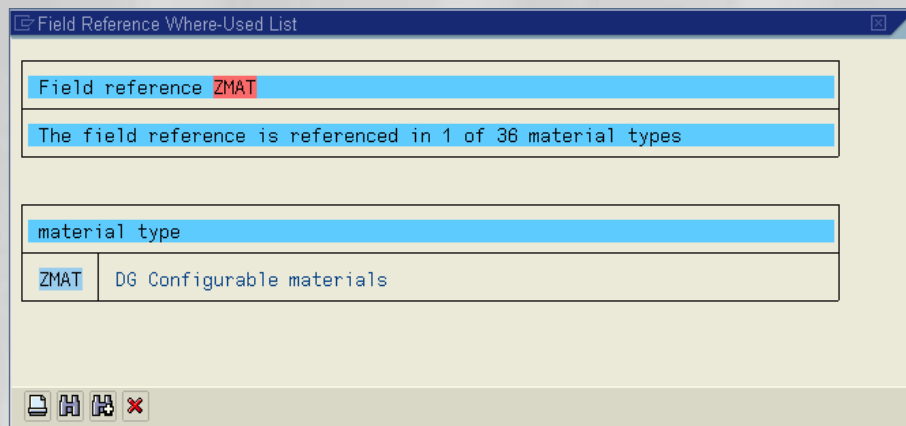


Figure 5: Field Reference Where-Used List

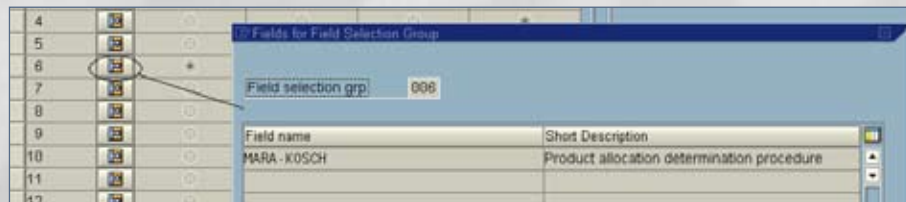


Figure 6: Detail of Field Selection Group – Displays Fields Included in Field Selection Group



Figure 7: Create Industry Sector

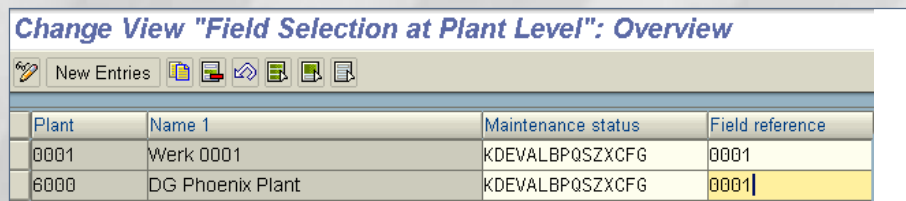


Figure 8: Assign Plant to Field Reference

Another driving value of determining field behavior is the plant.

S	Status description
A	Work scheduling
B	Accounting
C	Classification
D	MRP
E	Purchasing
F	Production resources/tools
G	Costing
K	Basic data
L	Storage
P	Forecasting
Q	Quality management
S	Warehouse management
V	Sales
X	Plant stocks
Z	Storage location stocks

Figure 9: Maintenance Statuses

Step 5: Define Lock Relevant Fields

You can define a field as Lock Relevant if you want the value to be entered into them once and never changed. Keep in mind that if you select this, and you have a scheduled change for a material, you will lose the scheduled change.

To mark a field to be locked after initial entry, simply find the field in the list and select the “lock relevant” indicator.

Material Types

In this section, I will illustrate how to configure material types. First, review the material types that exist in the standard SAP system and find the one that suits the needs of the materials that are going to be created. Select the material type that you want to copy and click the Copy icon – see Figure 10.

Once you have clicked the Copy icon, you will be taken to the Create Material Type screen. Enter a name for your new material type – I have entered ZMAT in this example.

General Data Tab

Figure 11 displays the fields in the General Data Section of the Material type creation screen. Complete this screen as shown in Table 2.

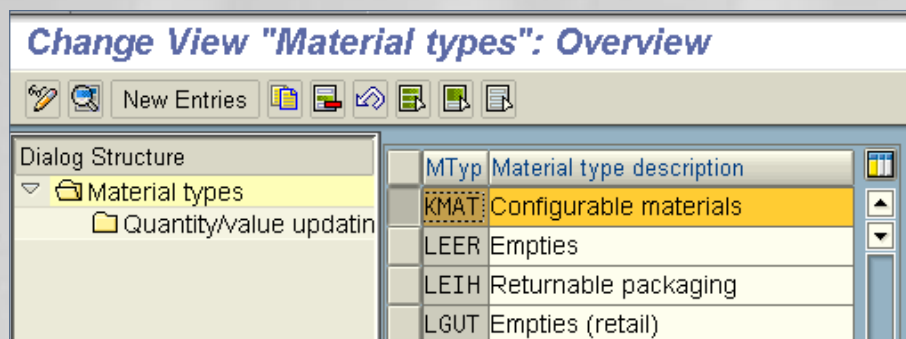


Figure 10: Copy Material Type

General data	
Field reference	ZMAT
SRef: material type	ROH
Authorization group	ZMAT
<input checked="" type="checkbox"/> External no. assignment w/o check	
X-plant mati status	<input type="checkbox"/>
Item category group	0002
<input type="checkbox"/> With Qty Structure	
<input type="checkbox"/> Initial Status	

Figure 11: Create Material Type: General Data

Field Name	Description
Field Reference	The field reference was created in the previous section of this article – in our example, the field reference is also ZMAT. The field reference determines which fields are mandatory, optional, hidden, and display only.
Sref: material type	Screen Reference Material Type. The screen reference determines which screens appear and the order of the screens, when creating a Material Master record.
Authorization Group	Authorization Group is required to limit certain users from being able to create certain material types. The first check SAP performs when a user attempts to create a new material is to determine if they have access to the ‘Create Material’ transaction. The second check is to verify they have access to create that material type. You can leave this blank if you do not want to restrict or maintain the material types to be created by specified users.
X-plant mati status	Cross-Plant Material Status can restrict a material to be used for specific purposes or actions. Another configuration step exists in the IMG to configure the behavior of the selectable statuses. For example, a material type may not be able to be procured, so the configuration of the material status would indicate a warning or error message to a user who is trying to procure that material type.
Item category group	Item Category group drives the processing of the material type in the sales order. The combination of item category group and order type determines the item category defined for a Sales Order line item.
With Qty Structure	If a material is costed with a Quantity Structure, turn on this indicator to improve system performance. You can override this checkbox in individual Material Masters.
Initial Status	Set the Initial Status indicator if this is a batch material and the initial batch of this material should be set to ‘restricted’.
External no. assignment w/o check	Check this if you want to allow the material master to be numbered externally (assigned by the user) without SAP checking to make sure the number is within a valid range.

Table 2: General Tab Field Completion

Special Material Types Tab

On the Special Material Types tab (Figure 12), you can mark if the Material is:

- Configurable
- For a process (product produces co-products)
- Pipeline mandatory material
- Manufacturer part

Internal/external Purchase Orders Tab

Figure 13 shows the Create Material Type: Internal/External POs tab. This area sets whether internal or external purchase orders will be allowed.

- Select 0 if they are not to be allowed
- Select 1 if they are allowed, but SAP will issue a warning message
- Select 2 if they are allowed without warning

Classification Tab

If there is a standard material classification that you use for your materials of this material type, you can enter it in the Create Material Type: Classification tab (see Figure 14) to save your users from selecting it each time they create a new material.

Valuation Tab

Figure 15 shows the Create Material Type: Valuation screen. This section determines what the default price control will be for your material type. You can select either standard price or moving value, or leave it blank. A valuation class assigns the material to a group of G/L accounts that are updated when a goods

Figure 12: Create Material Type: Special Material Types

Figure 13: Create Material Type: Internal/External POs

Figure 14: Create Material Type: Classification

movement occurs. The Account Category Reference is a group of valuation classes (which again, is a group of G/L accounts) that SAP will allow you to enter when maintaining the Accounting View of the Material Master record. If your material type does not utilize the Accounting View, this is not relevant.

Quantity/Value Updating Tab

You will also need to determine the Valuation Area for the material type.

A valuation area is either a plant or company code. Most SAP customers set their plants as their valuation areas. It is simply the level at which you value your

Figure 15: Create Material Type: Valuation

Figure 16: Create Material Type: Quantity/Value Updating

stock – either by quantity or value. You have the choice to display the quantity and value of your stock within the Material Master record. If you need to be able to see this information in the Material Master, you can select it here; but keep in mind that it does add a load to your system to track it at this level (see Figure 16).

User Departments Tab

Figure 17 shows the User Departments setting for the material type. In this grid, you select which views (also called user departments, represented by tabs when you create a Material Master) you will need to maintain for this material type

Conclusion

In this article, we covered how to define the behavior of fields in the Material Master – meaning whether they were mandatory, optional, hidden, or display only. We covered how to configure new material types. These steps represent some of the more difficult areas of configuration for the Material Master. Keep an eye out for the next installment where I will demonstrate how to configure the screens to streamline Material Master entry for the end user, and a white paper detailing Best Practices of Material Master Deployment and Maintenance.

Figure 17: Create Material Type: User Departments

Jocelyn Hayes, *Director of Consulting and Training for SAPtips*, has over ten years of SAP experience. She has enjoyed a successful consulting career for a Big 4 firm as well as independently. Jocelyn's focus primarily has involved the Logistics modules of SAP, and has also developed ESA Roadmaps using NetWeaver™ tools, including Visual Composer, XI, Enterprise Portal, and BI. She also led a CRM Business Process Reengineering project. Contact Jocelyn at Jocelyn.Hayes@ERPtips.com.



What Is Time Evaluation and Why Do We Need It?

By Satish Badgi, SAP® HR Consultant

***Editor's Note:** It's one of the most complex issues in HR: Time Management. Since some employees' pay is time-based, while other employees are salary-only; Time Evaluation isn't a one-size-fits-all proposition. Fortunately, SAP offers the functionality to calculate time accurately—as long as you understand the different options and when you'll need to use them. Satish Badgi helps de-mystify the various scenarios of Time Evaluation and when you'll want to employ each option.*

Introduction

The Time Management module in SAP HCM is probably one of the most complex, yet also most powerful, tools available. It's loaded with a lot of flexible functionality, which adds to its complexity. However, it's often the case that implementation teams and users get caught with different scenarios related to time management, then struggle with the scope of this module. In this article, we will learn about some of these confusing scenarios, especially Time Evaluation, and discuss the overall approach SAP takes in its Time Management functionality.

As a reference point, the discussion in this article is generally applicable to all SAP versions from 4.6C through to ECC 6.0.

Scenarios of Time Management

Before we jump to the topic of Time Evaluation, let's cover some basics in terms of background and definitions we'll be using. In the U.S., the Fair Labor Standards Act (FLSA) governs the definition of Exempt and Non-Exempt employees, which drives the need for overtime calculation:

Exempt Employees – Employees in exempt positions do not receive overtime, regardless of the number of hours they work. Typically salaried employees are part of this category. (Although, in some cases, you will come across salaried non-exempt too).

Non-Exempt Employees – They are paid overtime when they work more than 40 hours in a work-week. Typically, hourly employees are part of this category.

The Time Management module in SAP HCM is probably one of the most complex, yet also most powerful tools available.

So what does the exempt or non-exempt employee status have to do with Time Evaluation? For non-exempt employees, this functionality kicks in for the overtime calculations. The calculations could be automated or manual, depending upon the number of employees, complexity, size of the organization, and similar factors. Let's check the SAP definitions around this topic.

Table 1 lists the different SAP definitions which will then be used to track the exempt/non-exempt employee population as necessary. The categorization of exempt/non-exempt is normally driven through HR structures in most implementations.

Scenario	Applicable Situations
Positive Time Tracking	<ul style="list-style-type: none"> • Attendances and breaks are tracked • Typically for Non-Exempt/Hourly employees
Negative Time Tracking	<ul style="list-style-type: none"> • Only the absences are tracked • Typically for Exempt/Salaried employees
With Time Evaluation	<ul style="list-style-type: none"> • Automated system based calculations of overtime, shift differential, and other time-based payments • Complex rules for calculations or collective agreements with unions • Time Evaluation is applicable to both salaried and hourly employees. In the case of salaried employees, Time Evaluation may be used for vacation accruals
Without Time Evaluation	<ul style="list-style-type: none"> • Paid absence and attendances can be managed through data entry instead of time clocks or automated calculations • This scenario is applicable to both exempt and non-exempt employees – if the data is managed manually

Table 1: Potential Cutover Impacts for Payroll

The FLSA, as well as related SAP definitions noted in Table 1, decide the primary need for using Time Evaluation in SAP. In addition, each implementation can have its own situation where Time Evaluation may or may not be used. To better understand this, I have presented an overview of two of the implementation scenarios here. We have chosen two ends of the Time Evaluation spectrum:

- With time clocks and Time Evaluation
- Without time clocks and no Time Evaluation

We'll start with a comparison as shown in Table 2.

To better understand each of the two scenarios, we will list the steps that broadly describe the business process as well as SAP functionality.

With Time Clocks and with Time Evaluation

This is probably the most complex scenario in the time management area. The broad steps shown in Figure 1 are as follows:

- **Step 1**– Employees swipe cards or enter their entries/exits in the time clock system. Typically the time clock systems can go in two different ways:
 - Time clock system evaluates time.
 - Time clock system sends raw data to SAP, and SAP evaluates time.
- **Step 2**– Run an interface to send the time clock data to SAP. You will notice that depending upon the two streams just listed, you'll need to send the appropriate data to the SAP HR module. **Example:** In the case of evaluated data, you may send overtimes that are already calculated in time clock systems. In the case of time clock systems where data is not evaluated, you may send Clock Start/End time pairs.

With Time Clocks and with Time Evaluation	Without Time Clocks and No Time Evaluation
Clock in and clock out time pairs need to be gathered	We do not need to worry about actual time in/out
Breaks need to be monitored and also found out from clock in/out	We do not need to monitor the breaks
After collecting the in/out pairs, we need to come out with buckets for day/week as needed by rules. Example – If an employee works more than eight hours in a day in first shift, he/she is entitled to overtime pay	If we need to pay overtime in this scenario, it should be manually calculated and fed through an Infotype such as 2010 or 0015
The system needs to generate time wage types. Example – Wage Type 1001 for Overtime 1.5 or Wage Type 1002 for Overtime 2.0	Users will manually select the overtime wage types and enter the data
Time schema and rules need to be customized and configured to handle time pairs, time wage types, and bucket management	The Payroll schema will handle the payments and you can use an option to use schema UT00 (or its copy) to manage some custom rules
With any changes to time clock data, you will need to run Time Evaluation again or with retroactivity	All changes are contained within Infotypes and payroll processing

Table 2: Comparison of Options for Time Evaluation

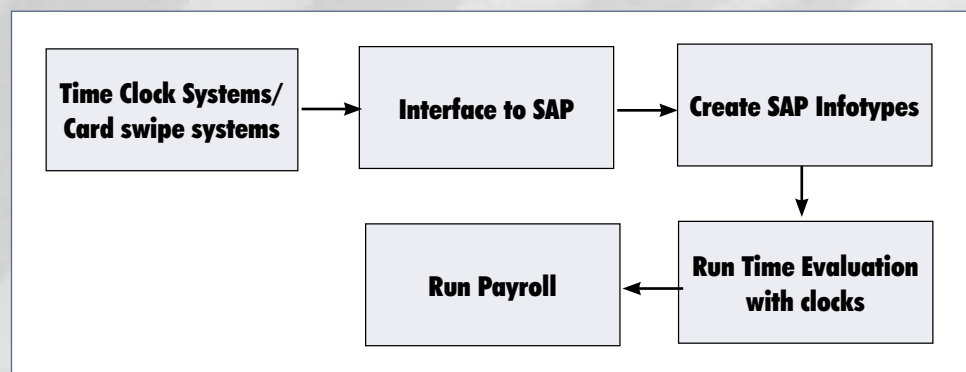


Figure 1: With Clocks and with Time Evaluation

- **Step 3**– In the case where the Time Clock system sends data which is not evaluated, we will need to run Time Evaluation in SAP.
- **Step 4**– After Time Evaluation, either time results will be sent to an external payroll system or, if you are running SAP Payroll, then the payroll function will read and begin processing the time clusters.

Without Time Clocks and With Time Evaluation

This “mid-point” scenario is typically used to drive vacation accruals or similar time calculations that are not dependent on actual time-clocking. The picture in this case will look as shown in Figure 2.

• **Step 1**– Capture the time data (absences and attendance) from CATS or any other time system to SAP Infotypes. In this scenario, you will be capturing the hours and not the clock in/out pairs.

• **Step 2**– Run Time Evaluation using the time data. You will need to configure the schema and rules. (*Note:* SAP standard time schema for Time Evaluation without clock times is TM04.)

• **Step 3**– Run payroll using the time clusters.

Without Time Clocks and Without Time Evaluation

This is probably the simplest scenario in the time management area. Within the earlier complex scenario and this simple scenario, we will cover the spectrum of different combinations. The broad steps in Figure 3 are:

- **Step 1**– In the absence of a time-clocking system, the time data comes into SAP via tools such as:
 - Cross Application Time Sheet (CATS). You can then use delivered SAP programs to transfer the CATS data to the time Infotypes.
 - Direct data entry in SAP Infotypes such as 2001, 2002, and 2010.

• **Step 2**– Run payroll (probably with some custom rules in payroll schema related to time calculations).

Technical Details

We will start our discussion with the Infotype 0007 configuration. In SAP, the options for Time Evaluation are managed at employee level.

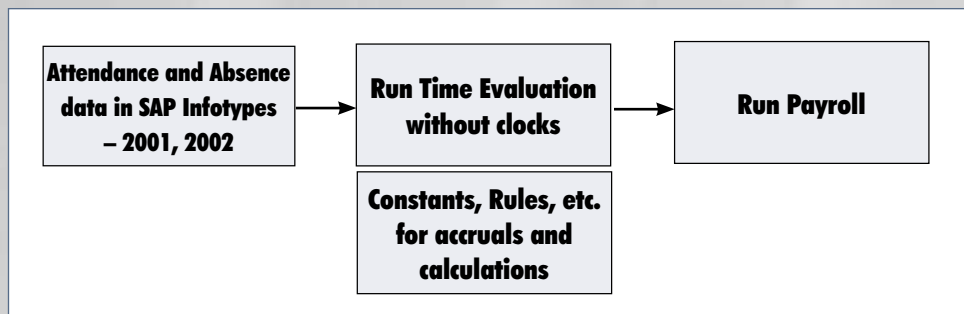


Figure 2: Without Clocks and with Time Evaluation

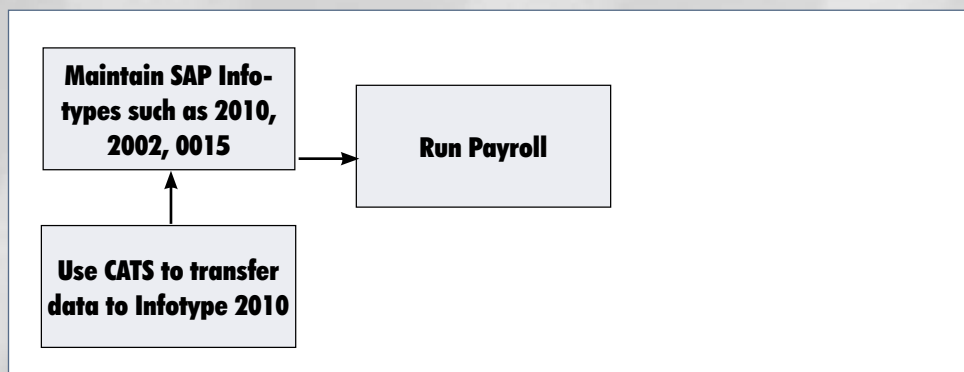


Figure 3: Without Clocks and Without Time Evaluation

Overview of Configuration – A very brief overview of the configuration is presented here:

1. The decision on Time Evaluation starts at the Employee Master Data level. Infotype 0007 is shown in Figure 4. Options 1 and 9 are respectively used for Time Evaluation with clock or without clock (planned time only). While creating employee master data (through hiring or data conversion), the appropriate choice of Time Evaluation status needs to be done in Infotype 0007.

Work schedule rule		
Work schedule rule	PFLEX	7.5 Hour Flex Shift
Time Mgmt status	0 - No time evaluation	
Working week	0 - No time evaluation	
<input type="checkbox"/> Part-time employee	1 - Time evaluation of actual times 2 - PDC time evaluation 7 - Time evaluation without payroll integration 8 - External services 9 - Time evaluation of planned times	
Working time		
Employment percent		
Daily working hours	7.50	
Weekly working hours	37.50	
Monthly working hrs	162.50	
Annual working hours	1950.00	
Weekly workdays	5.00	

Figure 4: Infotype 0007 for Employee

2. After we decided the employee's time evaluation status in an earlier step, we need to work on time types and processing types. Figure 5 shows the IMG path for Time Evaluation (to configure the time types).
3. Using the menu path from Figure 5, we get to time types. What are time types? The time types are the smallest units used to cumulate the main time balances for balance formation. Figure 6 shows a sample time type for attendance. In the figure, notice that this time type is used to form a day's balance. The PS Grouping (Personnel Sub area grouping) is used in Time Evaluation configuration as in the case of work schedules, absence, or attendance configuration in the Time Management module. SAP has delivered time types, and you can define new time types if required.
4. Decide the attendance/absence types from Infotypes 2001 and 2002, which will require balance formation and associated time wage types generation.

5. You can group these absence/attendance types by linking them with appropriate values in the Processing Type/Time Type class.

Figure 7 shows the IMG step to carry this out. As far as the employee is concerned – we always take an attendance or absence view; when our configuration is concerned, we need to take the Time Types and

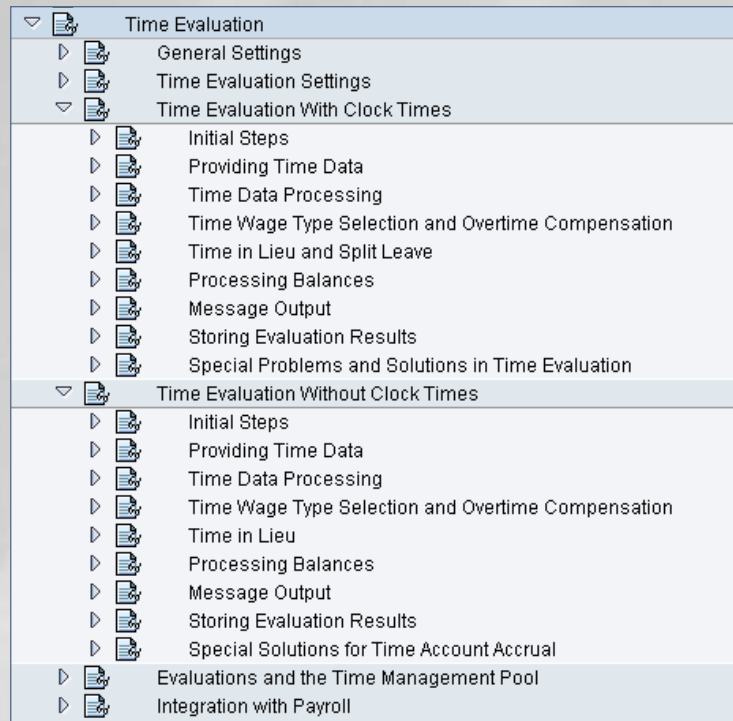


Figure 5: IMG for Time Evaluation


Figure 6: Sample Time Type

Processing Types view. Therefore by linking these two views, the configuration helps you to bring the time clocking world together with the normal business processes of attendances and absences.

6. Manage time schemas
- SAP has delivered schemas TM00 (with clocks) and TM04 (without clocks) for managing the time rules and processing. Similar to payroll schemas and rules, you will need to manage the customization. The functions and operations are specific to time management. We will address the time schemas and rules in a separate article in the future.

Conclusion

Time Evaluation is no doubt a complex topic. However, with the right analysis, you will be able to decide whether Time Evaluation is necessary for your implementation. You will also notice that Time Evaluation itself can be relatively simple or very complex depending upon usage of time clocks in the implementation. Depending upon which scenario of Time Evaluation you will be using in the project, the implementation resources and efforts requirements will change substantially.

Satish Badgi, SAP HR Expert. Satish is a senior consultant with specialization in SAP HR and Payroll. He has over ten years of experience implementing and supporting SAP systems across multiple countries in a variety of industries. In his recent engagements, Satish has been involved with implementing SAP payroll systems in complex environments and integrating them with Finance, Costing, and Funds Management. You may contact Satish at SAPtips.Authors@ERPTips.com. Be sure to mention the author's name and/or the article title. 

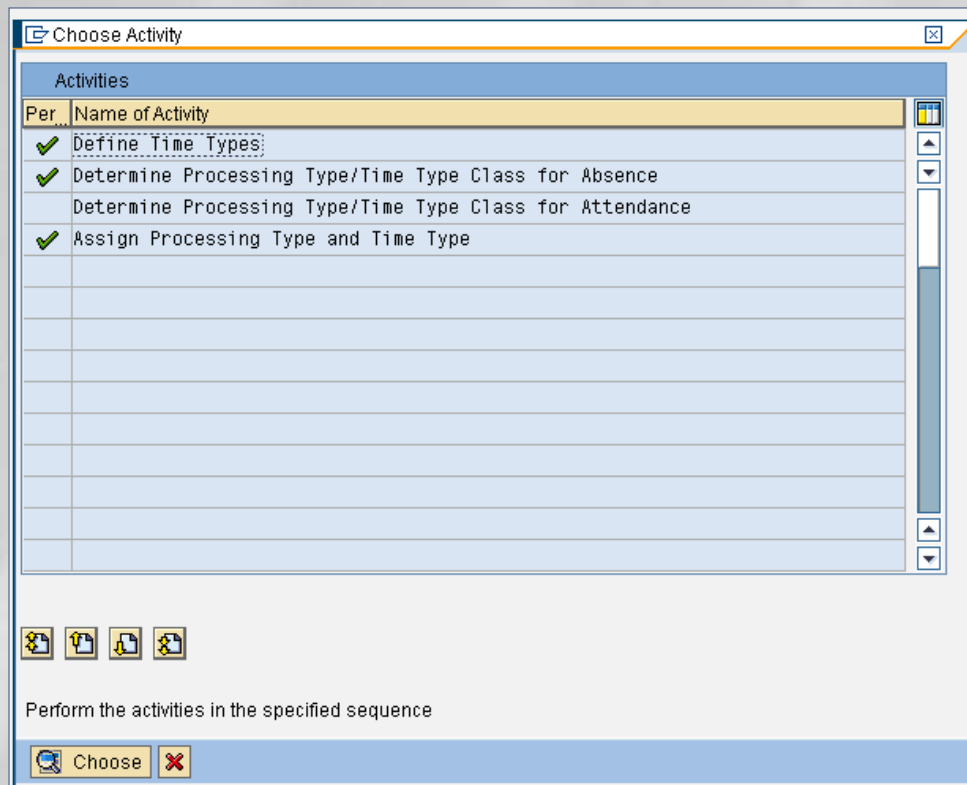


Figure 7: Assigning Processing Types

Creating Your Own Infotypes in Organizational Management A Guide for SAP HR Consultants and Developers

By Rehan Zaidi, Siemens Pakistan

Editor's Note: *There's a vast world of Infotype creation out there—and Rehan Zaidi is eager to help you explore it! If you're an SAP HR consultant or developer, you'll appreciate this handy guide to creating Infotypes in the Organization Management module. Rehan discusses the tables, structures and types involved, as well as the steps to take to create an OM Infotype.*

The SAP HCM module provides functionality in all areas for customizing standard applications/screens according to your customers' requirements; the Organizational Management (OM) module is no exception. Like with the Personnel Administration sub-module, you may also define customer-specific Infotypes for Organizational Management. You may generate complex Infotypes quickly and easily without changing the standard SAP code. These advantages make having knowledge of the Infotype creation pre-requisites and process steps is an absolute must for HR programmers and consultants.

The aim of this article is to help you understand the steps required in creating OM Infotypes. Here are some of the questions that this article will address:

- What are the various tables and structures involved when working with OM Infotypes?
- What are the minimal steps required to create a customer OM Infotype?
- What are the two types of Organizational Management Infotypes?

I will start with an overview of OM Infotypes, their types, and technical architecture. Then, I will cover, in detail, the steps required to create the customer field and table Infotypes. Finally I will discuss an example taken from a practical company scenario in which a customer OM Infotype was required. As always, I'll highlight any tips learned from my experience and observation.

This paper is primarily intended for HR developers and functional consultants, so I will assume that the

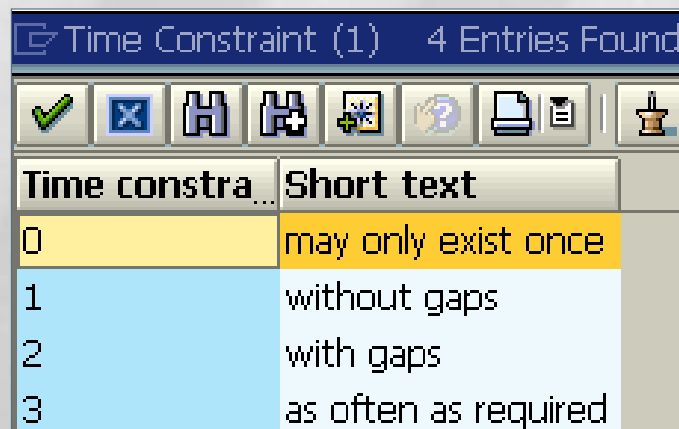
reader is familiar with basic HR concepts and HR programming in an SAP environment. For more information, refer to the SAP documentation on <http://help.sap.com>. All the examples and screenshots have been taken from Release 4.7.

**You may generate complex
Infotypes quickly and easily
without changing the standard
SAP code.**

OM Infotypes: An Overview

As with the Personnel Administration sub-module, there are also Infotypes within Organizational Management (OM). There are a few commonalities between Infotypes of these two areas:

- The OM Infotypes also have a start date and an end date.
- There are time constraints also for the OM Infotypes. However, the time constraint codes and their meanings are different from those in Personnel Administration. The time constraints applicable for OM Infotypes are shown in Figure 1.



Time constraint	Short text
0	may only exist once
1	without gaps
2	with gaps
3	as often as required

Figure 1: OM Time Constraints

The commonly used organizational objects and their codes are shown in Figure 2.

Organizational Object	Code
Organizational unit	O
Position	S
Job	C
Personnel Number	P

Figure 2: Organizational Objects and Relevant Codes

Your users may create a variety of versions of organizational plans. However, only one version has a status of Active, and this version is integrated with the Personnel Administration sub-module. Each Organizational Management object's basic information is stored in Infotype 1000, the corresponding table of which is HRP1000. The organizational links are created in relationships Infotype 1001, using the appropriate relationships and relationship codes.

For each OM Infotype NNNN, there is a corresponding structure HRINNNN and a table HRPNNNN defined in the database, and a module pool having the name format MPNNNN00. The screen 2000 contained in this module pool is used for maintaining and displaying the Infotype data.

There are two OM Infotype categories: Field and Table. Field Infotypes are comprised of a set of fields that are filled by the user. There is no tabular data involved. On the other hand, the table Infotypes let you enter tabular data within the Infotype entry screen. This tabular information is stored in an HRTNNNN table within the database. For example, for table Infotype 9111, the corresponding table that contains the Infotype's tabular content is HRT9111. An example of a tabular Infotype is Cost Planning, as shown in Figure 3.

The relevant tables that store Infotype information are T777T(Infotype text), T778T(Infotypes), and T777I(Infotype attributes). Each Infotype must be assigned to an object type. For example, you may have a loan limit Infotype for

positions, and so on. Table T777I is used for carrying out this assignment.

Minimal Steps Required in Creating a New OM Infotype

Setting up a new customer Infotype requires a few mandatory steps. Let's go through them individually:

1) Creating necessary structures and specifying the Infotype-specific fields

Before generating your Infotype program and screens, you'll first need to define a structure in the ABAP dictionary (transaction SE11) and specify the Infotype-specific fields. For field Infotypes, the structure HRINNNN must be defined. On the other hand, the structure PTNNNN must be created in case of a table Infotype. For example, if you are creating a table Infotype 9011, you must first define structure PT9011.

You may include character, amount, or currency fields within the HRINNNN or PTNNNN structure. If your Infotype structure contains amount fields, you also need to include a currency field within your structure. This currency field must then be linked to the respective amount field. This is done by making entries in the Currency/Quantity Fields tab for the amount field row. Moreover, for the currency field, the table TCUR must be assigned as the check table using the Entry/Help tab. You may also link an Infotype field with a standard or custom check table. The values of the table may be accessed using a search help assigned to the relevant Infotype field.

Tip: In order to check whether a field has been assigned correctly, simply double-click on the field name of the

Position	ELS						
Planning Status	Active						
Validity	21.10.2007		to		31.12.9999		Change Information
Salary surveys	Normal case						
Cost Planning 01 S 00000571 1							
Wage element	Dir.eval.	Amount	Currency	T	Time unit	Percentage	
	<input type="checkbox"/>						
	<input type="checkbox"/>						
	<input type="checkbox"/>						
	<input type="checkbox"/>						

Figure 3: An Example of a Table Infotype

SE11 transaction. A pop-up box appears showing the search help name.

2) Generating the Infotype Screens and Database Table

The next step involves generating the Infotype screens and tables. The transaction used for generating these is PPCI (meaning Personnel Planning Infotype Copier). Call transaction PPCI. The screen shown in Figure 4 will appear.

Enter the number of the Infotype and the Infotype Name in the fields provided. Select a suitable Infotype category. The two values permissible are field Infotype and table Infotype.

There are two other checkbox indicators: Language-dependent Infotype and Country-specific Infotype. Check the relevant indicator, if applicable to your requirement. The Country-specific check box must be checked for Infotypes that must be displayed if the country-specific settings are chosen.

Click the Create button to generate the database table HRPNNNN of the relevant Infotype, along with the module pool MPNNNN00. The structure PNNNN, the screens 1000, 2000 and 3000 and the relevant PAI and PBO modules are also generated. The entries in table T777T (Infotype text) and T778T (Infotypes) are also created by the Infotype copier transaction. However, entries in table T777I must be created manually (we will see this in the next section).

Note: Make sure that the HRINNNN or the PTNNNN structure exists when you are creating field and table Infotypes respectively. Otherwise, an error dialog is generated, as shown in Figure 5.

Make sure that the HRINNN or the PTNNNN structure exists when you are creating field and table Infotypes respectively.

Figure 4: The Main Screen of Transaction PPCI

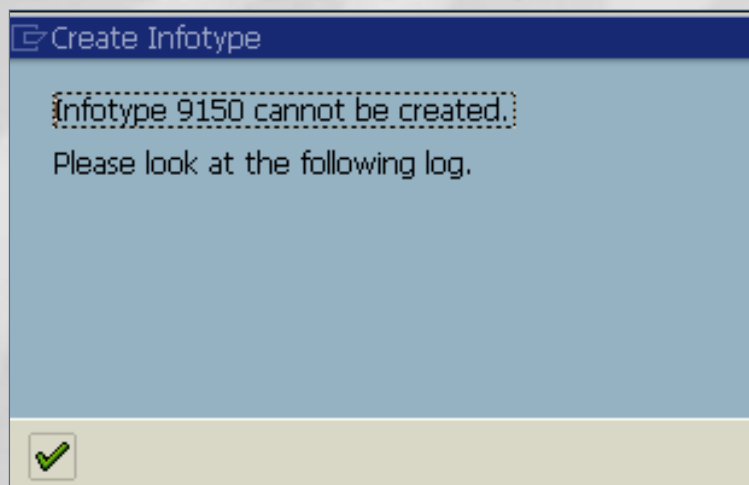


Figure 5: Error Dialog Displayed on Transaction PPCI

3) Assigning the Newly-Created Infotype to Object Types

The Infotype must then be assigned to the relevant object type(s). The steps required are as follows :

- Follow the IMG path Personnel Management → Organizational Management → Basic Settings → Data Model Enhancement → Infotype Maintenance → Maintain Infotypes. Alternately, call transaction SM30. Enter the table name; i.e., T777I, in the field provided, and click the Maintain button. This leads you to the screen shown in Figure 6.
- Select the row of your newly created Infotype and double click the Infotypes per Object type node in the left pane. The display of the screen changes as shown in Figure 4.

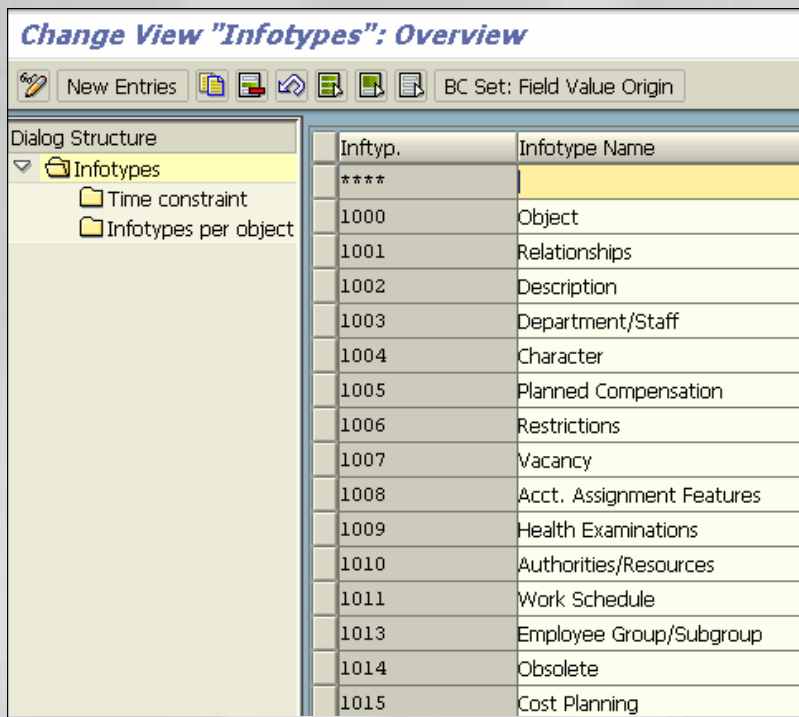


Figure 6: Maintenance View of Table T777I

- Click the New Entries button and enter a new row defining the linkage between the object type (for example S - Position) and our newly-created Infotype, as shown in Figure 7.

Note: The No Maintenance checkbox on the Infotypes per Object Type view is used to control the maintenance of Infotypes. Checking the indicator does not allow the user to maintain the Infotype via standard maintenance transactions.

You'll first need to define the field's variable in the code of the generated MPNNNN00 program. Then, using the screen painter, these fields must be drawn on the screen. Then, place the code for the relevant PBO module generated by transaction PPCL. Your Infotype will contain a PBO module INIT_9010. Insert the relevant code within this module. This module is called each time the screen is displayed.

Ob	Object type text	IT	Infotype Name	Alt.screen	No mainte.
S	Position	1007	Vacancy		<input type="checkbox"/>

Figure 7: The Infotypes per Object Type View

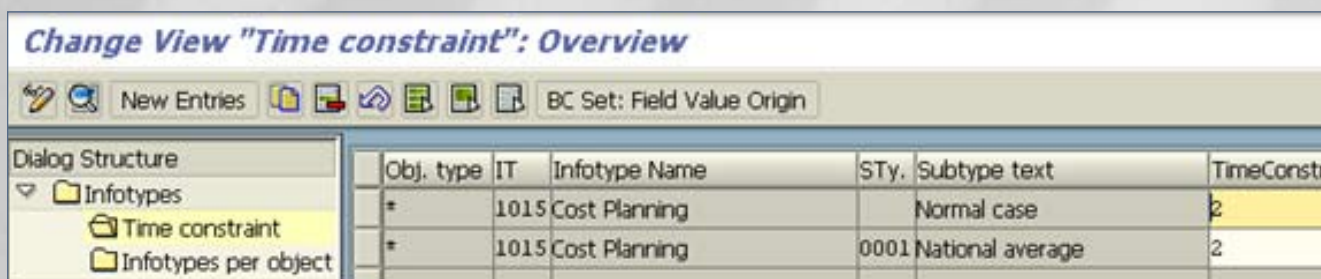


Figure 8: Specifying the Time Constraint for Your Infotype

4) Setting the Time Constraint

Next, you need to specify the time constraint applicable to your Infotype. Double-click on the Time Constraint node in the left pane of the screen. The screen changes in the right pane as shown in Figure 8.

Click the New Entries button. Now, you'll need to enter the suitable record defining the time constraint pertinent to your Infotype in the fields provided. Enter suitable values in the Object type, the Infotype number, and the time constraint fields, then click the Save button.

5) Populating the Screen Fields of Infotype

The next step is to create and populate any screen fields pertinent to your Infotype. For example, when the user enters a loan type on the screen, the screen field loan text must be automatically populated with the appropriate loan text.

. INCLUDE	PT9401	
ZHR_PERNR	ZHR_EMPLNO	NUMC
ZHR_BEGDA	BEGDA	DATS
ZHR_ENDDA	ENDDA	DATS
ZHR_GROUP	ZHR_GROUP	CHAR

Figure 9: Fields in Infotype 9401

Putting It All Together

In this section, I will use the concepts mentioned in this paper to show how they may be used to fulfill a simple requirement. Let's consider the following company scenario: Say your company user wants to store all historical data pertinent to holders of a particular position. The Infotype should store the start and end dates and the employee numbers of the personnel who held a particular position.

Since more than one record of employees is involved, this requirement will be fulfilled by a table Infotype. We give this Infotype a number 9401. The fields of the Infotype are first defined in structure PT9401 as shown in Figure 9.

Next, the transaction PPC1 is used for generating the Infotype module pool. The relevant transaction is used for creating the entry in the Infotype attributes table T777I. The entry for assigning the Infotype to the Position is shown in Figure 10.

Ob	Object type text	IT	Infotype Name	Alt.screen
S	Position	9401	position holder	

Figure 10: Assigning Infotype 9401 to Object Type Position

Then, a suitable Time Constraint is created in the maintenance view of table T777I. Finally, after all the settings, our Infotype appears in transaction PO13, as shown in Figure 11.

Conclusion

In this paper, I covered the rudiments of OM Infotypes, as well as their technical architecture and the two types that SAP HCM supports. I also discussed, in detail, the steps needed to create even the simplest Infotypes. I hope that this paper provided you with valuable insights and will help you in creating customer-specific (field and table) Infotypes in the least possible time.

Rehan Zaidi, Senior SAP Consultant, Siemens Pakistan. Rehan has been involved in both ABAP development and functional configuration for SAP HR implementations at multinational and local companies, and also has experience with SAP Workflow. He has contributed articles to the SAP Professional Journal, the HR Expert newsletter, and to the TechRepublic Website. He is currently working on his first book, specifically designed for SAP HR Users and Managers, as well as a guide for ABAP/Workflow Consultants titled "201 Interview Questions on Workflow". Rehan is the founder of the web site www.siteofSAP.com. You may contact the author at SAPtips.Authors@ERPtips.com. Be sure to mention the author's name and/or the article title.

Figure 11: Our Position Holder Infotype Appearing in Transaction PO13

Unlock Payment Processing Benefits Using SAP®'s Lockbox Capabilities: Part I

By Anurag Barua, Independent SAP Consultant

***Editor's Note:** Anurag Barua has a lock on how lockbox processing functionality works in SAP. The good news: He's generously providing us with all the key concepts in this two-part series. In Part I, Anurag introduces the key concepts behind SAP's lockbox processing, and then provides the configuration to get you started.*

Introduction

Not too long ago – actually seven years ago – the U.S. Democratic Presidential candidate and recent recipient of the Nobel Prize for Peace, Al Gore, made a major contribution to the public lexicon – the word “lockbox”. Of course, the context of his frequent use of the word was not in the context of SAP, but in his (unsuccessful) proposal to put Social Security funds in a “lockbox”. Now, jumping to the world of SAP, the fact that standard SAP supports Lockbox processing should give you comfort and (pun intended), security. If your organization has implemented SAP Financials (FI) using R/3 or ERP 2004 (ECC 5.0) or ERP 2005 (ECC 6.0), chances are that you have been exposed to Lockbox functionality in one of the following modules:

- Accounts Receivable (AR)
- Cash Management
- Bank Accounting
- The new Financial Supply Chain Management (FSCM)

The fact that standard SAP supports Lockbox processing should give you comfort and security.

In the first piece of this two-part series, I am going to introduce you to the key concepts behind lockbox processing in SAP and all the configuration steps that you'll need to carry out to prepare your SAP system for lockbox processing. In the second part of this series, I will familiarize you with the activities around importing the lockbox file(s) and the subsequent processing of these in SAP. I will also shed light on some of the technical components in the lockbox realm.

What Is Lockbox?

Companies, in an effort to reduce check processing time and increase liquidity, increasingly prefer to let banks handle incoming payments. So, customers send their payments to “lockbox” accounts, instead of sending them directly to the payee. These lockbox accounts are set up by the payee company and the individual customers are informed about these accounts and thus, where they need to send their payments. Upon receiving payments from customers, the bank credits the customer. This information is then collected and transmitted to the payee company in a file (in a fixed format, i.e., BAI or BAI2 – more on this later) at pre-agreed intervals for further reconciliation in the payee's accounting system of record. If that system of record happens to be SAP, assuming you have configured things correctly, SAP takes over and carries out the relevant G/L postings.

A Sample Lockbox Process

In a typical lockbox scenario, such as the one shown in Figure 1, once customers have signed up with a bank to avail themselves of lockbox services, they send their payments along with relevant remittance information to a designated (physical) address. At this point, it would be pertinent to explain who is the company and who is the customer. The “customer” owes the “company” money in return for consuming the latter's goods and/or services. The company decides to take advantage of lockbox services provided by both its payment collection bank(s) and SAP. It (the company) sets up lockbox account(s) at the bank and informs its customers to send payments to these lockbox accounts instead of sending them directly. Once this setup with the bank is completed, the bank (that houses this lockbox) then processes this payment information, consolidates them and puts them in a (flat) lockbox file in either BAI or BAI2 format.

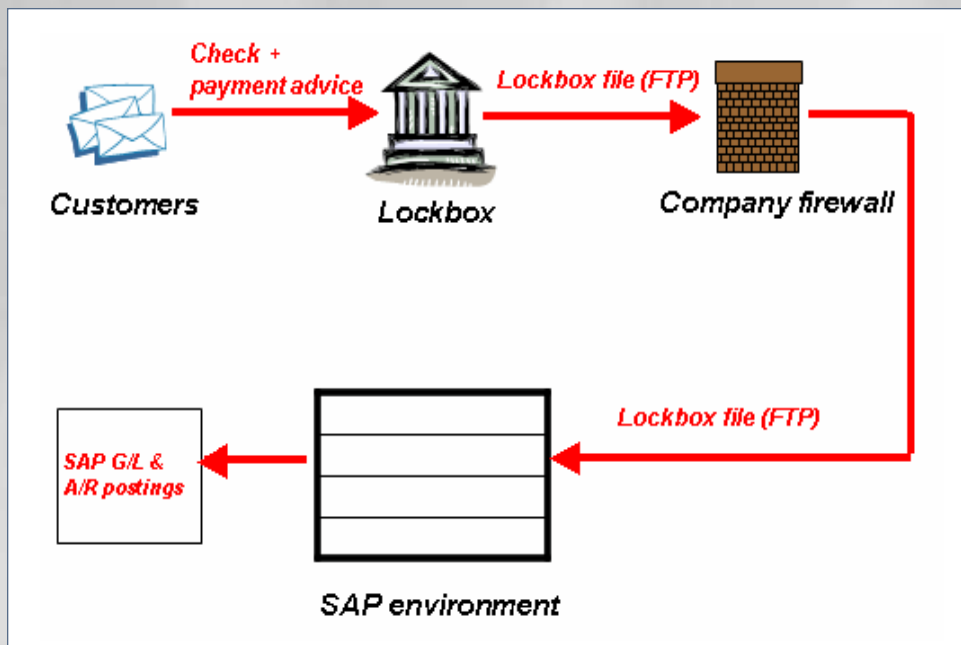


Figure 1: Anatomy of a Typical Lockbox Environment

On the accounting side of things, the bank credits the company's account. This file is then sent (usually FTP'd) to the company at regular, pre-determined intervals. This is usually once or twice a day. Once this file is processed inside our SAP system, the relevant G/L and A/R postings take place.

What Are BAI and BAI2?

BAI stands for Banking Administration Institute and has become synonymous with the file format designed by this organization in 1971 for lockbox services. This format has evolved over the years and in 1987, another format, the BAI2 was introduced. SAP supports both these formats.

A lockbox file in BAI format contains a check amount for several invoices and additionally contains invoice numbers and the customer Magnetic Ink Character Recognition (MICR).

A lockbox file in BAI2 format contains the check total line items that are separated out into respective invoice numbers and their corresponding invoice amounts. Additionally, it also provides the customer MICR. Because of this level of granularity, BAI2 provides a greater possibility of matching invoices with payments, with faster clearing and less manual work, when com-

```

100YPCCDESTINYPCCORIGIN9812151201
2YPCCDESTINYPCCORIGIN
50010010012345981215YPCCDESTINYPCCORIGIN
60010020001000000011000390556677889 000912301
4001003601719000075 00001000000000000000
4001004602719000076 00001000000000000000
4001005603719000077 00001000000000000000
4001006604719000078 00001000000000000000
4001007605719000079 00001000000000000000
4001008606719000080 00001000000000000000
4001009607719000081 00001000000000000000
4001010608719000082 00001000000000000000
4001011609719000083 00001000000000000000
4001012610919000084 00001000000000000000
700100500123459812150100001000000
80010060012345981215001000010000009 1000000 1000000
9000010
  
```

Figure 2: A Sample Lockbox File

pared to processing lockbox files in BAI format. In technical terms, both these formats conform to their respective data structures. A data structure for a particular format contains several record types. Each record type is intended to hold information of a particular type, and a file that's in a BAI2 format must conform to the data structure and record types associated with it. (A complete discussion on record types is outside the scope of this article).

Figure 2 displays the contents of a very simple lockbox file. In this file, you will see various pieces of information such as customer MICR numbers, invoice num-

bers, check numbers, etc. As I mentioned earlier, the individual records correspond to the various record types. One of the key pieces of information that stands out is the sequence of invoice numbers from 19000075 through 19000084 that are being paid. The gross amount in the invoices exactly matches the amount of payment (\$10,000). Obviously, you will not be able to make a whole lot of sense by eyeballing a lockbox file. Actually, this is not necessary. It is the SAP system that will process this file.

Note: In my experience, banks do not strictly adhere to these two formats and may send files in their own custom formats. These “custom” formats are often slight variations to the BAI and BAI2 standard formats. Of course, this will not be done without your knowledge; if you are implementing lockbox in your SAP environment, you and your bank need to agree on the exact format. Files sent in such custom formats will require extra work on the SAP side to map the changes to the record types of the standard formats. This could be a change to one or more SAP structures and/or programmatic changes.

In my experience, banks do not strictly adhere to these two formats and may send files in their own custom formats.

Configuration for Lockbox Processing in SAP

You’ll need to configure lockbox functionality in SAP to make sure that it is ready to accept the incoming lockbox files as well as able to support the post-processing steps. The good news is that lockbox configuration in SAP consists of a few key steps. Please note that all these configuration activities are available within the

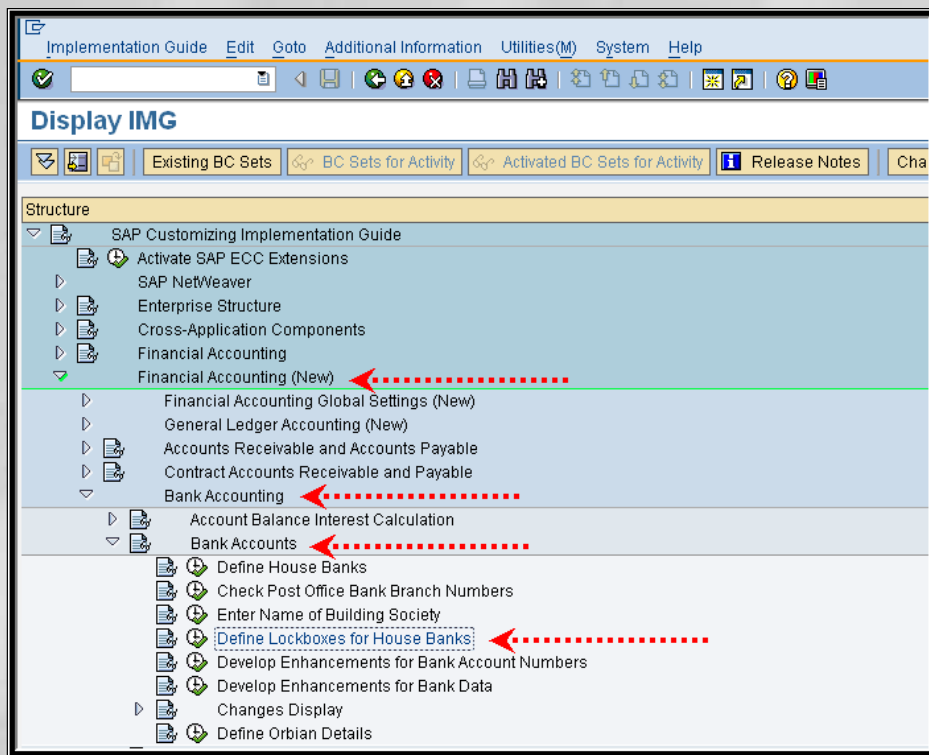


Figure 3: Define Lockboxes for House Banks in the IMG

Bank Accounting section of Financial Accounting. Let’s look at each of these steps in more detail:

- 1. Creating Lockbox Master Data.** You need to configure basic information about your lockbox, such as lockbox number for your house bank (which you need to have configured in advance) and the company code. To do so, you can either run transaction OB10 or, in an ERP2005 system (running ECC 6.0), carry out the navigation in the IMG by following the sequence of red arrows as shown in Figure 3.
- 2. Define Control Parameters.** In this step, you’ll identify the format(s) you will be using for receiving the files (BAI or BAI2) and their properties. This key configuration activity is accessible either by running transaction OBA1 or by following the sequence of red arrows as shown in Figure 4.

Let’s assume that your organization will be handling files in a BAI2 format. When you execute the “Define Control Parameters” transaction or transaction OBA1, you will be taken to a screen that displays existing procedures and formats. The good news is that the system is pre-configured for control parameters for both BAI2 and BAI formats. Let’s double-click on the default entry for BAI2 that has a procedure name of “Lockbox”. This is shown in Figure 5.

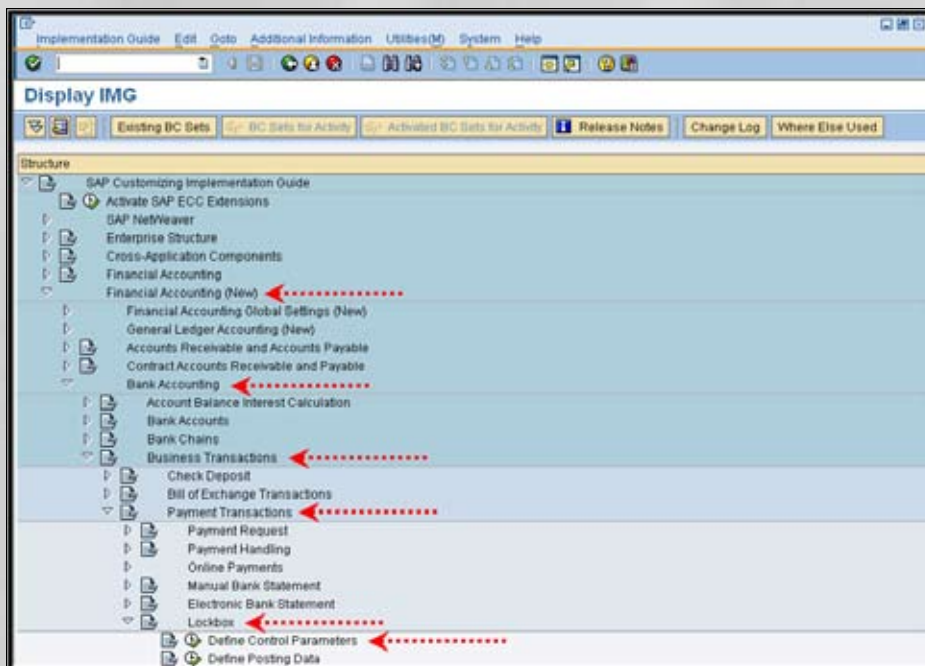


Figure 4: Define Control Parameters in the IMG for Lockbox Processing

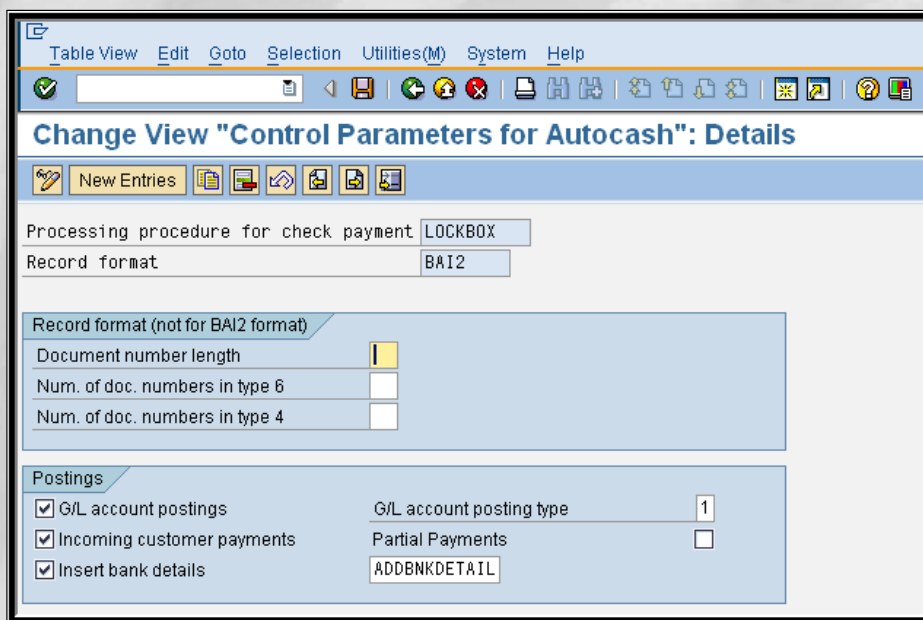


Figure 5: Maintaining Control Parameters for Lockbox Procedure

You are taken to the maintenance screen for control parameters. As you will notice, all the relevant information is defaulted in and you can modify these parameters as you desire. Please note that if lockbox processing is already operational in your system, you'll need to exercise caution when making changes to the defaults. Since the "Record format" section is not relevant for the BAI2 format, you do not need to

enter anything here. If you were setting up a procedure for the BAI record format, you would have to enter values in this section.

The next section is "Postings". There is important control information that you'll need to specify, regardless of the format that you intend to use. Table 1 highlights each field and its purpose.

Posting Fields	Purpose
G/L Account Postings	When this box is checked, it means that when the lockbox file is processed, relevant credit and debit postings take place in the specified G/L accounts. (You specify the G/L accounts in the configuration activity that I will discuss next). A credit is posted to the cash amount and a corresponding amount is debited from your bank account. This box is checked by default.
Incoming Customer Payments	When this box is checked, it means that incoming payments are posted to customer accounts. This is usually a customer preference. This box is checked by default.
Insert Bank Details	If you want the MICR number relating to new bank details to be automatically updated in your Customer Master record, you will want this box to be checked. Furthermore, by checking this box, you are indicating that you want this to happen via a batch input session. This box is checked by default. The default name for this batch input session is "ADDBNKDETAIL".
G/L Account Posting Type	Postings to the relevant G/L accounts are controlled by the Posting type ("1", "2", or "3") chosen in this option. The default is "1"; keep this setting as default if you want each check to generate a single posting to the relevant bank account.
Partial Payments	When this box is checked, the system posts partial payments when the payment amount is less than the invoiced amount. This box is unchecked by default.

Table 1: Posting Fields

3. Define Posting Data. In this configuration activity, you'll specify the source of the lockbox and destination of the transmitted files. You'll also enter the lockbox bank data and the relevant posting parameters. This activity is accessible via transaction code OBAX or by following the sequence of red arrows as shown in Figure 6.

Next, you'll be taken to a screen where you'll see a couple of SAP-delivered posting templates. You can copy one of these and rename it or you can create a new entry by clicking on the "New Entries" button; the latter option takes you to a maintenance screen similar to the one shown in Figure 7.

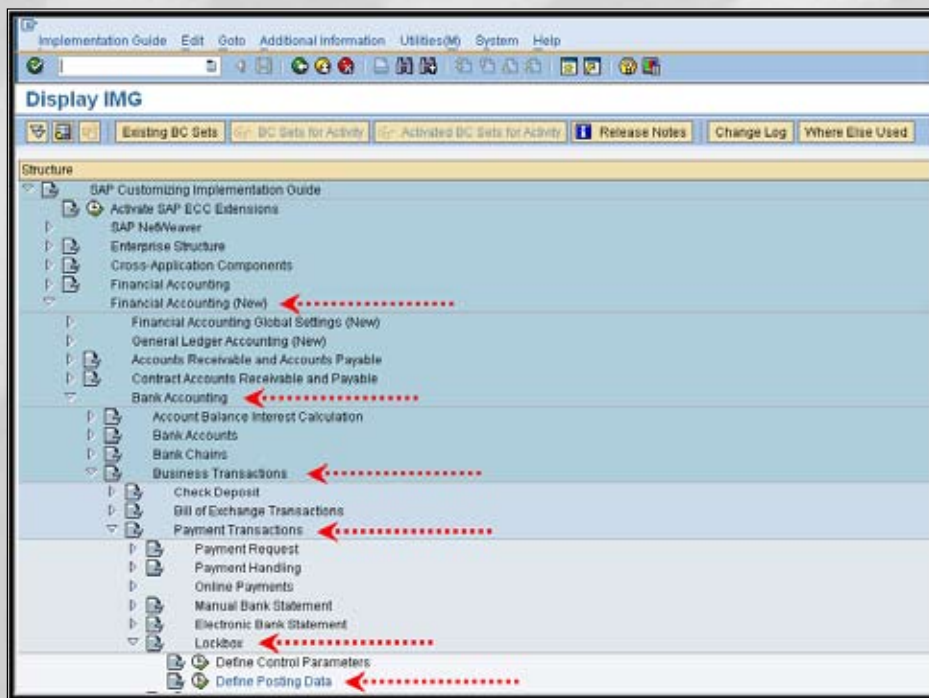


Figure 6: IMG Entry for Maintaining Lockbox Posting Data

Change View "Posting Data For Autocash With Lockbox": Details	
Destination	0026801867
Origin	0053000219
Lockbox bank data	
Company Code	1000
House Bank	ST001
Account ID	DEPCH
Bank (G/L) acct	100680
Bank clear.acct(A/R)	105200
Posting parameters	
Bank pstng doc. type	SA
Cust pstng doc. type	Z1
Pstng key: debit G/L	40
Pstng key: credit G/L	50
Post key: credit cust	15
Post ky: D cust	

Figure 7: Configuration of Lockbox Posting Data

As you can see, you'll need to enter the company code, then select the house bank and the account ID within the house bank. You'll have to enter two pieces of account information, one for the G/L account, which is the clearing bank account for the lockbox. In the second account information field, i.e., "Bank clear. Acct(A/R)" field, enter the G/L account for the Accounts Receivable clearing account. In the "Posting Parameters" section, specify the document type for the bank and customer postings, and the posting keys for the debit and credit postings.

Conclusion

In the first part of this two-part series, I introduced you to the basic concepts of lockbox, lockbox processing in SAP, and the configuration steps involved in setting up lockbox functionality in your SAP system. Now that I have let this genie out of the (lock) box, I will take you deep into the actual processing of the lockbox files, post-processing activities in SAP, and other valuable technical information, in the second part.

Anurag Barua is an Independent SAP Consultant. He has 15 years of experience in conceiving, designing, managing, and implementing complex software solutions, including nearly 9 years of SAP experience. He has been associated with several SAP implementations in various capacities. Anurag's core SAP competencies include FI/CO, Logistics, BW/BI, NetWeaver™, ABAP, SOX compliance, reporting, and Project Management. Anurag is a frequent speaker at various SAP conferences and contributes to several publications. He has a B.S. in Computer Science and an MBA in Finance. You may contact the author at SAPtips.Authors@ERPtips.com. Be sure to mention the author's name and/or the article title.



Understanding DataStore Objects (DSOs) and Their Assorted Flavors in BI NetWeaver™ 7.0

By Anurag Barua, Independent Consultant

Editor's Note: What does DSO do that ODS doesn't? After you say that five times fast, read up on this new feature found in NetWeaver BI 7.0. Anurag Barua compares DataStore Objects to its older self—the ODS Object, then goes deeper into the types of DSOs you'll encounter, when you'll want to use them, and touches on the configuration.

If you have upgraded or are considering upgrading to NetWeaver BI 7.0 (formerly known as NetWeaver 2004s BI), you've probably encountered several new features that SAP has added. Some of these new features are for improvements and some are innovations. For more details on the new features overall, please check out Arthur "Pat" Pesa's article, "To Upgrade or Not to Upgrade: What NetWeaver BI Brings to the Table" that appeared in the October/November 2005 issue of SAPtips Journal. In this article, we're specifically focusing on DataStore Objects (or DSO for short, fka ODS Objects). This is an area that combines improvements to the existing functionality (in the BW 3.5) with some innovations. I will introduce you to the concept of DSOs, point out the differences between their predecessors (ODS Objects) and the new functionality, and take a deep dive into the various kinds of DSOs, discussing the configuration of each kind wherever applicable.

First, let's start with the basic navigation. You can get to DSO Maintenance either via the Data Warehousing Workbench (fka Administrator Workbench) by using the old RSA1 transaction and then navigating to the

desired DSO (as in previous releases). Or, you can get there by running transaction RSA11 (which takes you to the InfoProvider tree) and then navigating to your desired DSO, as in earlier releases. This takes you to DSO Maintenance when you double-click on the name of the desired DSO. Figure 1 displays a DSO Maintenance screen.

The ABC's of DSOs and ODS Objects

In BW 3.5 and earlier releases, there were basically two kinds of ODS: The standard ODS object and the transactional ODS. The purpose of the standard ODS Object is to store the data from the source system or the PSA (Persistent Staging Area) for cleansing and consolidation in order for business logic to be applied on the fields (via update rules) prior to transmitting the data to the InfoCubes. A transactional ODS object, on the other hand, is a single table (as opposed to a set of three tables as per the standard ODS) and is a replica of the data that are written to it by an application. It provides brisk access to data for other consumers; certain appli-

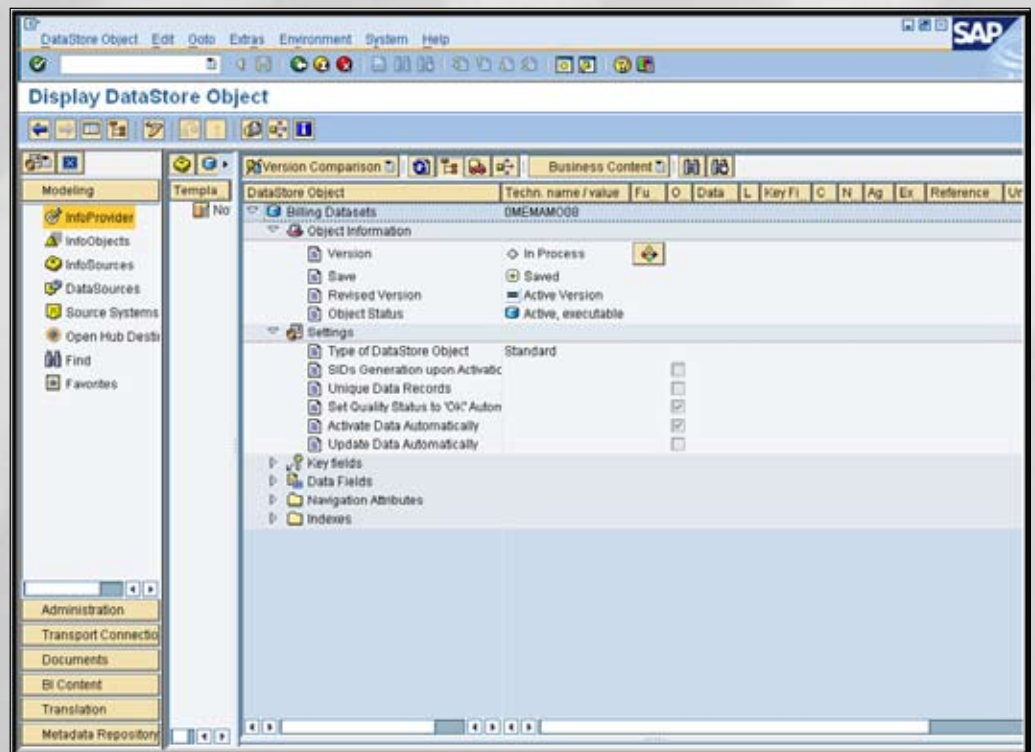


Figure 1: DSO Maintenance in NetWeaver BI 7.0

cations, like SEM (Strategic Enterprise Management), extensively use transactional ODS. Generally speaking, DSOs are needed for the same purposes as are their predecessors. Of course, as we will learn, DSOs have both additional and enhanced functionality over their predecessors.

NetWeaver BI 7.0 has been augmented with a third type of ODS (or DSO). So from the standpoint of someone who is making a first foray into BI or who doesn't much care about the evolution of the DSO, there are three types of DSOs that are available in the latest release:

1) Standard DSO –

A standard DSO is the same as the standard ODS Object from the previous BW 3.X releases. Nothing has changed from the earlier releases. In NetWeaver BI 7.0, it still consists of three tables:

- a. Change Log Table
- b. Activation Queue
- c. Active Data Table.

Keep in mind that this is more than just a name change. In terms of activating data, it is a lot faster and more efficient in this latest release.

2) Direct DSO –

This type corresponds to the transactional ODS Object from BW 3.X releases. Very little has changed between the predecessor and its successor. As the name suggests, you write data directly to the DSO, circumventing any transfer/update rules, and into a single table that stores data in the same format as the source from which it is pushed. Furthermore, you do not use any internal BI mechanism such as InfoPackage or a DTP to populate it. Data are written directly by the source system to a direct DSO. Just as transactional ODS Objects were widely used by Strategic Enterprise Management (SEM), in NetWeaver BI 7.0, direct DSOs are used widely in SEM.

3) Write-optimized DSO –

This is a brand new type of DSO containing just one table – the active data table. As the name suggests, this type of DSO is optimized for writing data in a speedy manner. One of the reasons that this is possible is that you don't need to activate the data (as you would in a standard DSO, which is often a very time-consuming process) – data are active by default. To further expedite the writing of data into such a DSO, SIDs (Surrogate IDs) are not created for the values of characteristics. Please note the trade-off that occurs here. When you run a BEx query off such a DSO, it runs slower than its standard counterpart – this is the price you pay for the significant performance gain you realize during the loading of data. This type of DSO is frequently used as a staging area for data that have been subjected to some common set of complicated transformations. The data that flows out to the data targets can then have more target-specific (and simpler) transformations and rules applied.

DSO Properties and Configuration

In Figure 1, I showed you a screenshot of DSO maintenance. Now let's look at how you would go about selecting your desired DSO type. On the maintenance screen of a particular DSO, click on the Change icon for the "Type of DataStore Object" option in Settings. This will display a pop-up window – here you will need to select one of the three available options for DSO types, as shown in Figure 2.

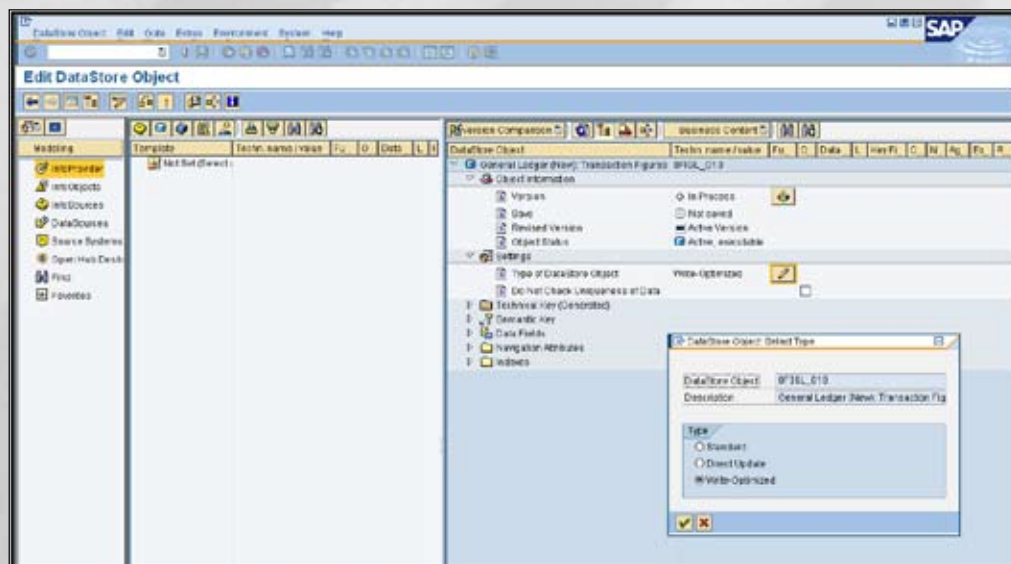


Figure 2: Selecting the Desired Type for a DSO

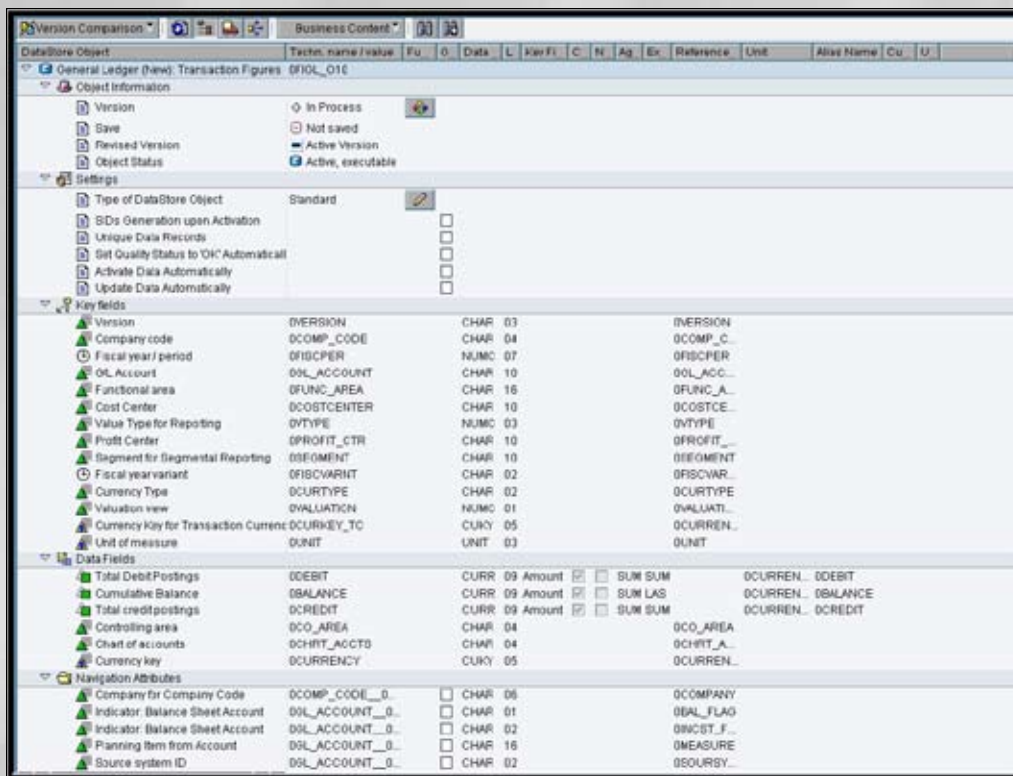


Figure 3: Setting a DSO to Standard Type

Configuring a Standard DSO

You select the desired DSO type in the pop-up window. An important fact to keep in mind is that you cannot change the type of a DSO if it contains data. You will notice some interesting effects caused by selecting each of the three types of DSOs. When you select the “Standard” option, you will see the Maintenance screen as shown in Figure 3. (For purposes of this discussion, I will be using the General Ledger (New): Transaction Figures DSO with the technical name of 0FIGL_O10).

If you peer at this screen for a while, you will notice that a standard DSO has several configuration settings. Some of these have been inherited from the previous release. You should make a certain setting only after you have assessed the need and the impact. Let’s discuss each of the options in the “Settings” node:

- **SIDs Generation upon Activation:** In an attempt to improve data load performance for standard DSOs, SAP® now provides the option to not have SIDs generated when data are activated. Keeping this box unchecked seems counter-intuitive because if a speedier data load into a DSO is what you are looking for, you should be choosing the write-optimized DSO type and not the standard DSO. But if you must

keep it unchecked, please note that while you can run queries in BEx off this DSO, they will run slower in the absence of SIDs.

- **Unique Data Records:** This is inherited from previous releases. You should flag this option when you are sure that the data being loaded into the DSO are all unique, meaning that the values of their key fields are not identical. When this is the case, setting the flag instructs the system to directly insert the data instead of wasting time checking for non-unique keys.
- **Set Quality Status to “OK” Automatically:** Another carry-over from the past, you should keep this checkbox flagged unless you want an additional quality check of the data and then a manual change of status to OK or green. I prefer to keep it checked. If you prefer to do a manual check (which might be the case if your enterprise has such an operational procedure), do not flag the checkbox. Remember that data is not available in the Active Data table (i.e., not activated) if the status is not changed either manually or automatically to OK.
- **Activate Data Automatically:** If the quality status is “OK” or green, checking this box means that data

that are loaded into a DSO are activated automatically without the need for manual intervention. When data are activated, data are populated in the Active Data and Change Log tables.

- **Update Data Automatically:** Your DSO is often the source, or at least the intermediary, between the source and other data targets (such as another DSO or InfoCube). Checking this box means that the data targets that are connected to this DSO will be updated automatically. This step will not be carried out successfully if the prerequisite (i.e., the previous step of activating data in the DSO) activity aborts or terminates.

Configuring a Direct Update DSO

A direct DSO, as explained earlier, is a pretty plain-vanilla entity. Thus, when you set your DSO to “Direct Update”, you will find no configuration options. This

is because you cannot do the kind of configuration you would do with a standard DSO. In other words, a direct update DSO fulfills limited needs and there is not a whole lot of variety that goes into fulfilling them. To reiterate, a direct update DSO saves the data in one table (Active Data table) in the same format that it came in, and no transformations can be applied to data either coming in or going out. Figure 4 displays a screenshot of DSO Maintenance as set to “Direct Update”.

Configuring a Write-Optimized DSO

When you set your DSO to “Write-Optimized”, the Maintenance screen changes accordingly. You’ll notice some changes and terms not encountered in previous releases such as semantic keys, data transfer process (DTP), etc. Figure 5 displays a screenshot of DSO Maintenance as set to “Write-Optimized”.

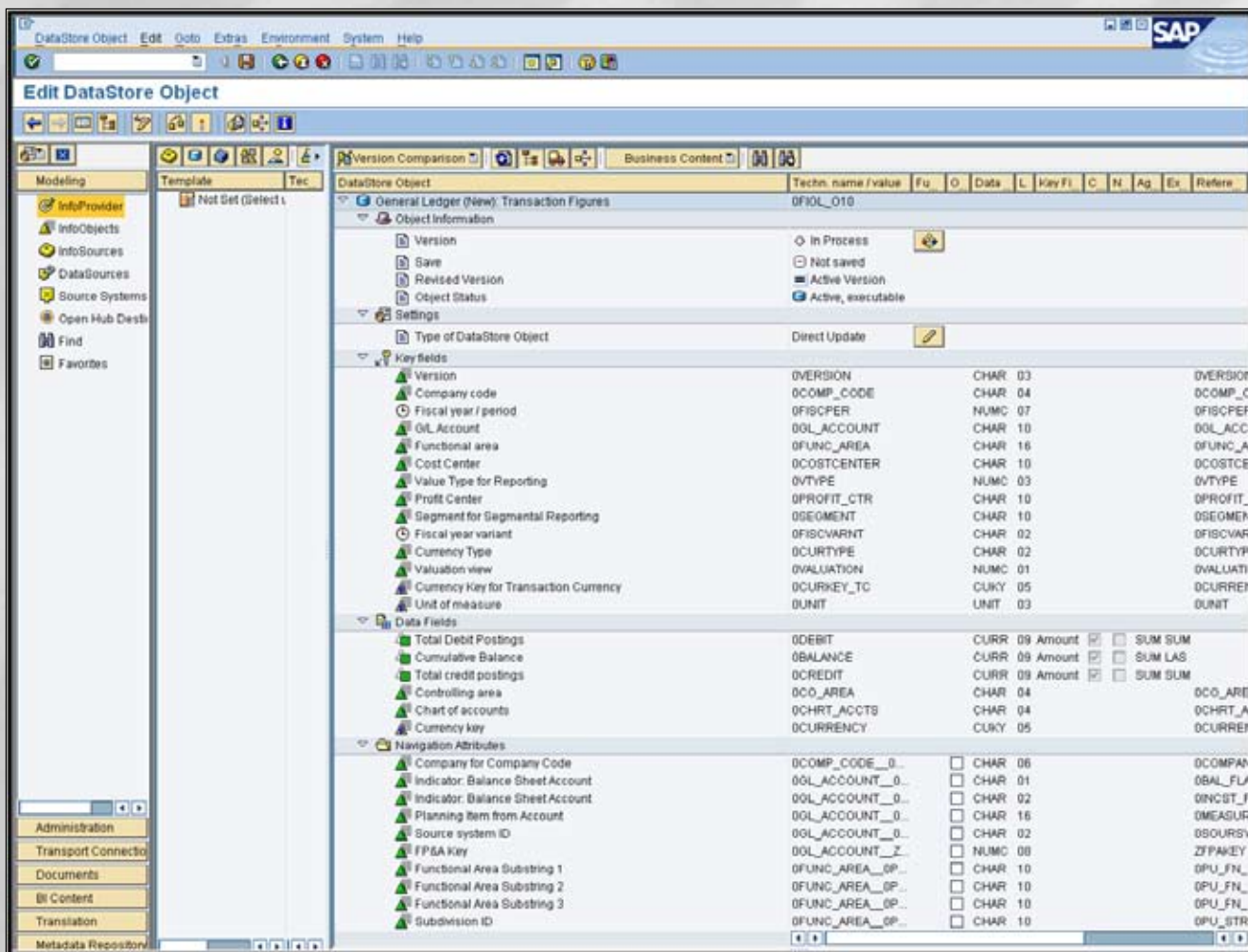


Figure 4: Maintenance Screen for Direct Update DSO

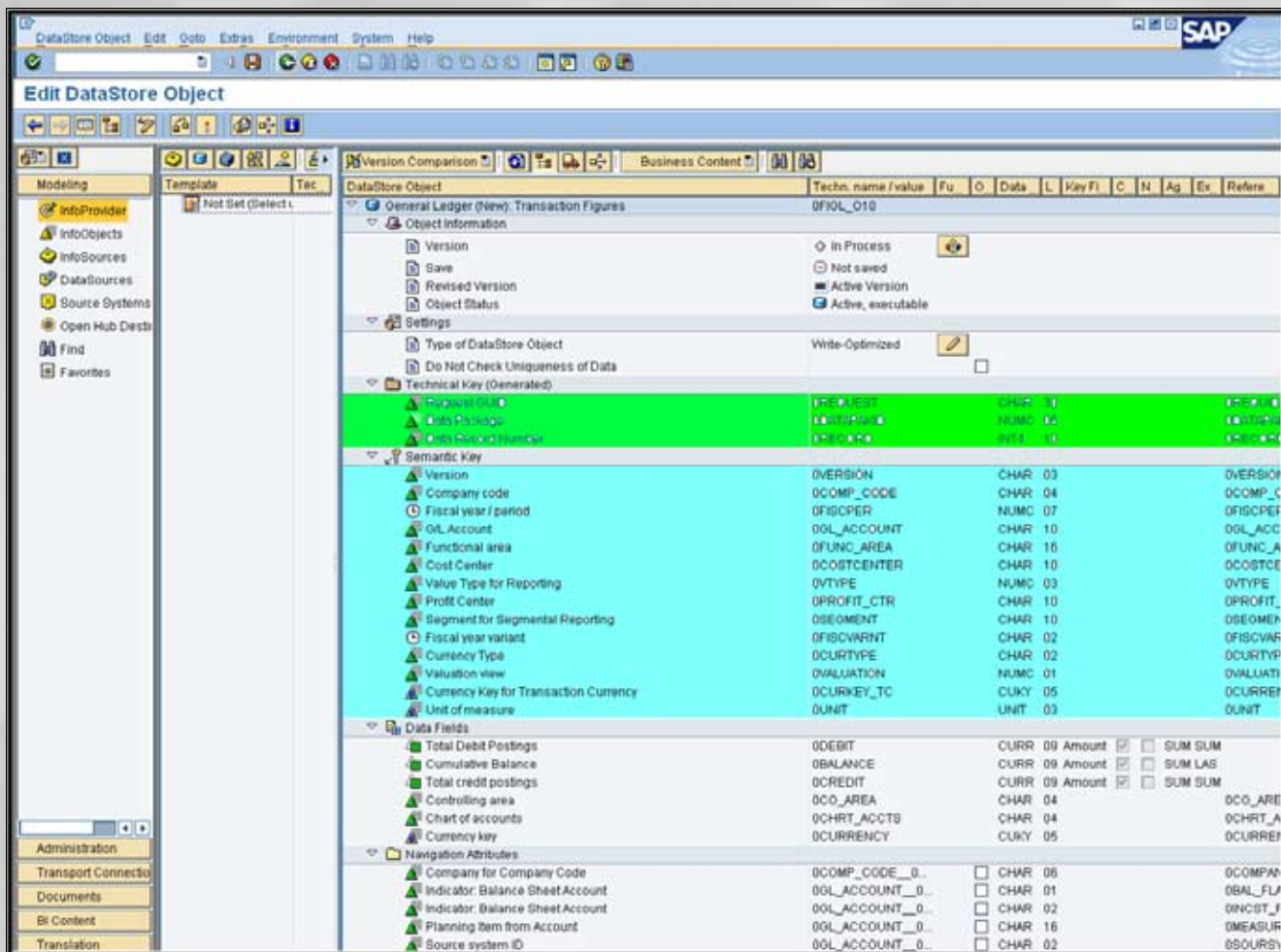


Figure 5: Maintenance of a Write-Optimized DSO

Before we look at the single configuration setting under “Settings”, let’s take a look at the two nodes where expanded portions are highlighted in green – for the technical key portion and blue – for the semantic key portion as per Figure 5. In a write-optimized DSO, there is a new (generated) composite key called Technical Key. Every write-optimized DSO consists of three fields in the technical key: a request number, a data package number, and a data record number. The key fields that you find in standard and direct update DSOs are now available under a new node called Semantic Key. In such a DSO, the standard key fields (that are now grouped under Semantic Key) do not play the role of keys in identifying each record. It is the technical key that plays that role and identifies the uniqueness of each record. If you compare the fields under “Key Fields” in Figure 3 with the fields under “Semantic Key” in Figure 5, you will find them identical. The “Data Fields” node remains unchanged.

The configuration in the “Settings” for this type of DSO consists of a single item, “Do Not Check Uniqueness of Data”. When you check this flag, and thus indicate that you do not want to check for uniqueness of data, records with the same technical key (i.e., the same combination of Request ID, data package, and data record number) can potentially exist in your DSO. If you do not check this box and thereby indicate that you do want to check for uniqueness of data, a unique index with the name “KEY” is created for the Active Data table of the DSO. To verify this, let’s look at the Active Data table for this DSO. The technical name of this table is /BI0/AFI_GL_O1000. Go to transaction SE11, type or paste the technical name, and then click “Display”. Now click the “Indexes” push-button; you will see the newly created index called “KEY”, as displayed in Figure 6. (**Note:** The Indexes push-button is highlighted in green for easy identification).

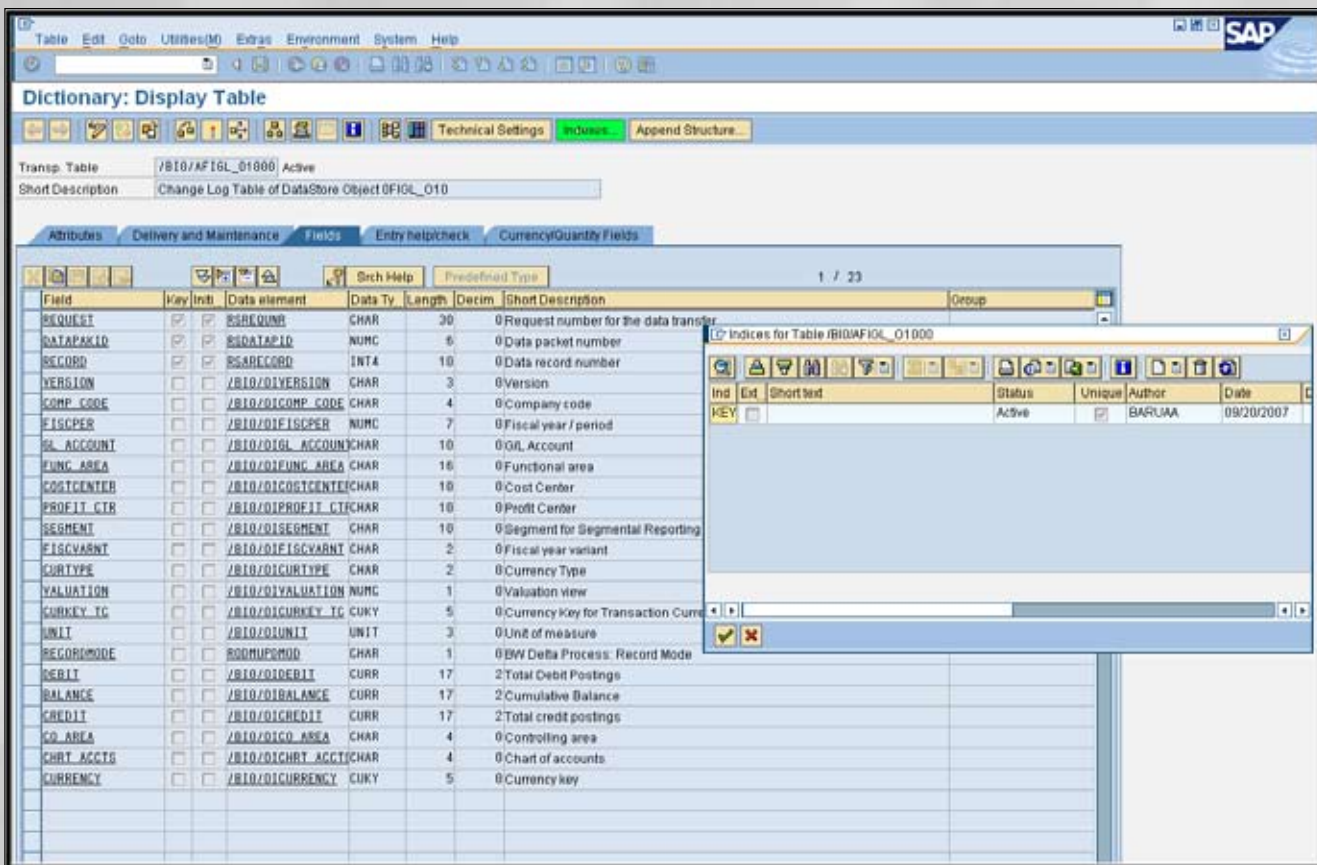


Figure 6: Index on the Active Data Table of DSO 0FIGL_010

Conclusion

In this article, I provided you with an overview of DSOs in BI NetWeaver 7.0 and described the three different flavors: standard, direct updated, and write-optimized. I also walked you through their properties and configuration. I hope you have a better understanding of DSOs and that you will be better able to select the right type of DSO to meet your specific business needs.

Anurag Barua is an Independent Consultant. He has 15 years of experience in conceiving, designing, managing, and implementing complex software solutions, including nearly 9 years of SAP experience. He has been associated with several SAP implementations in various capacities. Anurag's core SAP competencies include FI/CO, Logistics, BW/BI, NetWeaver, ABAP, SOX compliance, reporting, and Project Management. Anurag is a frequent speaker at various SAP conferences and contributes to several publications. He has a B.S. in Computer Science and an MBA in Finance. You may contact the author at SAPtips.Authors@ERPtips.com. Be sure to mention the author's name and/or the article title.

Building the SOA City – Part III: The Magic of Asynchronous Processing

By Axel Angeli and Lynton Grice

Editor's Note: Which would you rather do? Place an order and wait, and wait, and wait until it's available? Or place your order, and then go about your business until your order is ready? Yep, we thought so, too. Whether it's an order for a drink or a job on our computers, we're not inclined to sit and wait for the final product—we like to make use of our time more productively. Axel Angeli and Lynton Grice return to this issue to show you how asynchronous processing found in SOA can provide a much needed sense of "place your order and return when it's ready" that so many technology users crave.

Programmers fancy bars with service. You enter the bar and order a beer. During the three minutes when the beer is drawn, you can go away to do some other important things, like visiting the rest rooms or answering another 12 emails. When you get back, the beer is waiting for you. Just imagine if there were many people wanting beer: it wouldn't be efficient for the bartender to take the order, prepare the beer, and then serve the next one. It's better if the bartender asks all thirsty folks for their orders and then starts preparing all orders in parallel.

What a difference this is compared to some fast food restaurants. You have to line up properly and wait for your turn to give your order. Then you have to wait until the meal is prepared (compiled) and put on your tray. This often takes longer than it does to get a freshly drawn beer and your time is completely wasted. You cannot leave the line and return later. The system is not prepared for it.

We have an analogous situation in our SOA City. The pattern used by the bartender is an asynchronous process. The client submits the request to the server. Then it is up to the client whether to wait or do some other activities in the mean time. The fast food restaurant, however, is a de facto synchronous process, requiring the client to wait until the request is fully processed.

This example should make it clear that the asynchronous process is by far superior (in efficiency) to the synchronous process. The asynchronous process does not exclude the synchronous behavior at all. With an

asynchronous framework, you can always emulate synchronous behavior; just stand-by and wait for an answer. A synchronous framework, however, requires the requester to wait for a reply, whether it wants it or not. Hence, an asynchronous process leaves it as an option to the client as to what communication strategy – synchronous or asynchronous – is eventually applied.

Despite the name, SOA is definitely more than just architecture. SOA is a new paradigm, another way of thinking, a new way to view application programming. SOA transforms IT landscapes from autocratic rulings into democratic governance, where market-oriented, Darwinistic principles and strategies come into play.

**Despite the name, SOA is
definitely more than just
architecture.**

SOA is a collaboration infrastructure that allows for and encourages collaboration by making it easier to reuse common components over the Web, instead of building isolated silo solutions.

But there is one important issue to consider when you work with many mutually independent and autonomous processes running on a commonly shared infrastructure. You need mechanisms that allow for proper ad hoc communication between the running instances. We need a well-designed architecture that guarantees a reliable transport and supervision of events. This has a dramatic consequence on the understanding of any development in an SOA context. Programs need to assume proper responsibility to permanently alert the underlying infrastructure when it reaches an important step. In addition, the same programs need to provide "callback agents" that allow other programs to request additional information related to the event. This is called an "Event Driven Architecture" and one corollary of a proper SOA.

Note: An event-driven architecture (EDA) is a mandatory component of a full SOA.

In general, we can say that an event-driven architecture is the technical soul of an SOA and the core component of an Enterprise Service Bus, which stands for the technical incarnation of a SOA.

SOA Has Already Been Here for Many Years: Under the Lid

Unfortunately the knowledge of the “event-driven” architectural model for integrating large enterprise solutions is not widespread among the people who implement ERP systems. For people who come from an electronics or microprocessor development arena, these concepts are much more common. We can even say that all the thrilling new concepts of SOA are mostly quite old and long-established ideas.

EDA is the oldest architectural concept in concurrent IT. In the beginning, computers were all concurrent processors; you had one big computer that served many jobs in parallel. All computers of the 1960’s and 1970’s were purely EDA. Only the PC revolution kicked us back into the Neanderthal era of computers and made the new generation believe that sequential processing is the common ground. This has made EDA now appear as something new ... which it isn’t.

EDA is the oldest architectural concept in concurrent IT.

Those who are familiar with microprocessors or operating system architecture are also very familiar with event architectures. Every microprocessor, every RISK processor, every slightly complex silicon chip is an EDA; it would not work otherwise. When billions of transistors need to work in harmony, they cannot be sequenced in a serial processing scheme.

So, only the highest level of software design is still at such a low evolutionary level that it can work in a serial, synchronous fashion. While this is true for the application design level, it already looks quite different when we look directly at the ABAP engine. The ABAP engine is full featured, highly asynchronous EDA. For instance, when you save a sales order, the data is NOT written directly to the database, but the request is submitted to the Update queue and an event is raised

to wake up the queue. This underlying concept of EDA is the true reason ABAP is highly robust and enjoys unlimited scalability.

The NetWeaver™ Java stack, however, is quite different. It is a simple execution engine with very limited support for asynchronous processing of programs. A “misbehaving” program has the potential to bring down the whole runtime engine.

SOA has numerous advantages, like agile development, reusability, and standard-based development. But to achieve true “loose coupling” between components, you will need to turn your attention to EDA to realize the true benefits of asynchronous communication.

SOA – “Spaghetti Oriented Architecture”

Concurrent and asynchronous processing are the core principles of a stable and reliable concurrent processing architecture. Let’s consider an analogy. It’s 6:30 am, your alarm goes off to wake you up. Your car is running on empty, so it tells you that it needs more fuel by displaying a warning message. Your cell phone rings and a text message arrives telling you that money has been transferred into your bank account...the list is endless. Computers have become pervasive. Events and alarms are all around us.

It would be laborious and inefficient to check all this information in a traditional serial “one-after-the-other” manner. Most of these systems do not change their physical state very often. That would mean that most of the status checks are done in vain, and contribute more to system overhead than successful system operations. The solution to all this is designing the data collectors so that they “listen” in parallel to many devices. In fact, the listeners will hibernate until an alarm event wakes them up. This introduces concurrency into our scenarios.

Concurrency is getting a lot of attention these days. The real world is based on concurrent (or asynchronous) actions. The concurrent nature is also found in the governance of an SOA. If an SOA project is not correctly implemented and designed, it can most certainly disrupt the business. Numerous business areas should be involved from the onset to ensure that services are designed with reuse in mind. Without getting everyone involved, you may build the wrong services or design them the wrong way, and therefore you won’t be able to compose new business applications or processes with them. In fact, if the services are not tested or are managed incorrectly, you may put key business processes at risk, and the business could become more fragile.

You've probably heard this before; SOA will make your business more agile. But what does this really mean? It means getting all stakeholders around a table and deciding on how to design solutions so that it "may" be possible to use them again in the future. This, in itself, is a big undertaking, and once again requires a paradigm shift in mindset and a solid governance policy from the top. An effective SOA requires governance, not in the sense of imperative ruling, but in finding a new way of democratic collaboration between all IT users and service providers in the whole organization.

But getting everybody involved does not mean that everybody needs to be present all the time. A proper strategy would rather allow everybody to develop at their own pace and occasionally alert the teams to what their current status is. From time to time, a meeting would be held to synchronize the parallel processes.

Most of us have experienced the "weekly meeting". The team meets and it's always the same old story: endless discussions, deviations from the topic, yawning, wasted time. This begs the question, "If meetings are so important, why does everybody call them inefficient?" The reason is simple: Most of the discussed topics go into some peer to peer discussion. The administrator discusses some issues that are not of interest to the developer; the developer wants to know about master data from the sales team and bores the remaining team, etc.

Wouldn't it be more productive if only the sub teams that are really concerned about the topic at hand sat together and discussed the solutions? Only the results would be signaled to the rest, allowing them to react in exceptional cases and otherwise use their time in a more useful way. We'll go back to the bar analogy: You place your order and return when what's important to you is on the table.

Same goes for programming. It is better to delegate the individual work steps into small and parallel tasks, and then do a roundtrip to collect the results. In ERP development, like SAP®, this is seldom the case. Let's see how SAP would process a simple sales order item as shown in Figure 1.



Figure 1: Sales Order Processing in SAP

This means that you can calculate the price only after you determined the availability, although they might not be dependent on each other. In a service-oriented architecture, we would here identify this as a candidate for concurrent processing. We would submit as shown in Figure 2.

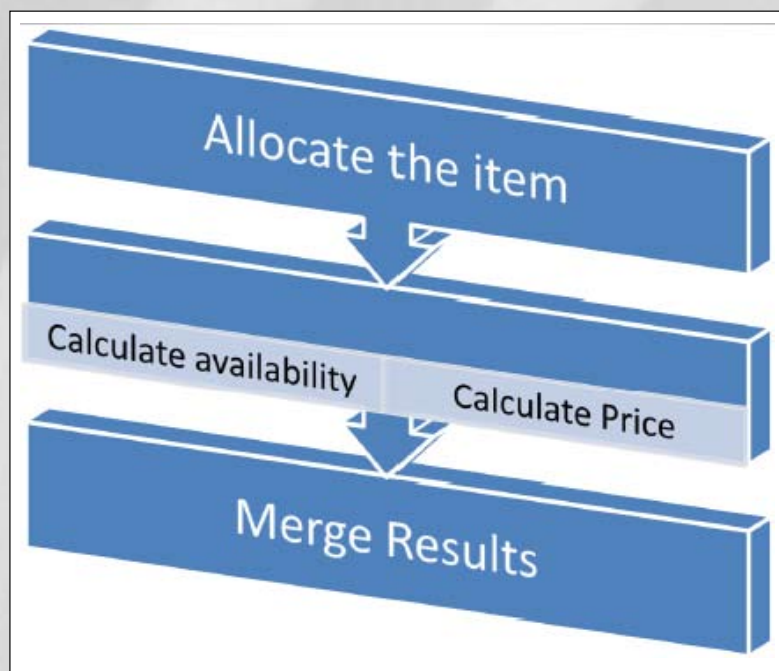


Figure 2: Concurrent Sales Order Processing in SOA

It is now the indispensable duty of the underlying SOA to cater for mechanisms that make the synchronization of the parallel tasks completely transparent to the developer. And here we come to the value of a programming language. Besides the fact that an SOA language must be more or less a script language to allow for “on the fly” changes to the code, the SOA language must be aware of concurrency. Unfortunately most programming languages implement this important feature half-heartedly. If they do, they typically implement the concurrent core as an API library that can be imported into the code.

Not even in the core competence sectors of SOA – the Web development - where concurrency and asynchronous processing are rather the norm than the exception, do we find helpful concepts. Yes, it is very easy to build Web services. Just open up a .NET IDE and you’ll have a handful done in one afternoon. But without proper support for concurrency, you’ll end up with a different kind of SOA – “Spaghetti Oriented Architecture”. (An interesting, historic side note—one that “younger” readers might not be aware of: This discussion about concurrency and its importance already took place in the late 1970s and had been forgotten about until now.)

And now another example: Let’s say your code makes an HTTP call to some service in the World Wide Web, but that service is currently down. What happens? Well after a lengthy timeout, an exception will probably appear (like a time-out message). Then the calling program will have to handle the exception after a long time spent waiting in vain. Normally the request works fine when we refresh the page. For a programmer, things are much more difficult. First of all, making an HTTP request is awkward in most programming languages. And saying something like “try making an HTTP call to that service, but if you don’t get any response within four seconds, go and do something else and try again later” is for the most part not a standard solution.

Although having such robustness and flexibility is the desired way to implement the solution, the modern programming languages still neither actively support nor encourage such a programming style. There are programming languages that do so, but they are still living in some niches. One very good example is the programming language Erlang. Both direct TCP support and concurrency are part of the language kernel. This language has been designed to work in embedded data communication devices like

mobile phones or RFID scanners and is an example of how a modern, reliable programming language needs to be designed. The cunning thing about Erlang is that it is based on asynchronous processes which lead to inherent runtime stability. Unfortunately, the great idea of Erlang is ridiculed by obviously unreadable language syntax. Erlang is very different from “the norm”. Erlang has been designed to be a highly robust and reliable language, so it is doubtful that there would ever be any big changes to the language that could introduce incompatibilities like Python has recently done. And there’s the trade-off: a stable, reliable, and backwards compatible language is simply not compatible with rapid development.

The example shown in Figure 3 illustrates how Erlang would handle a “timeout” scenario when sending some data to a socket.

As you can see, we simply added a timeout to the receive statement. This sets the maximum time that the process will wait to receive a message. So the code below says: “Try sending data to the socket, but after 4 seconds, issue a timeout...”

ABAP has some of this, too. It has no intrinsic support of HTTP, everything is handled through libraries, and class CL_HTTP_CLIENT is the entry point to do this. The concurrency support is, however, part of the language. You may submit requests as RFC calls in a separate task while specifying a call back routine. The intrinsic WAIT statement, as shown in Figure 4, also allows hibernation of the program until a certain Boolean condition is met. It allows the specification of a

```
socket_client (Str) ->
{ok, Socket} = gen_tcp:connect("localhost", 4321, [binary, {packet, 2}]),
gen_tcp:send(Socket, term_to_binary(Str)),
receive
    {tcp,Socket,Bin} ->
        Val = binary_to_term(Bin),
        io:format("Client result ~p~n", [Val]),
        gen_tcp:close(Socket)
after 4000 ->
    io:format("Connection timed out~n", [])
    exit(timeout)
end.
```

Figure 3: “Erlang” Time-out Message

```
Wait until sy-zeit = '120000'.
```

Figure 4: Wait Statement


```

GL-FUNCNAME = 'ZLOGOS_BPEL_ASYNC_SUBPROCESS'.
concatenate 'BPEL' FUNCNAME into GL-TASK_NAME.
call function GL-FUNCNAME
  starting new task GL-TASK_NAME
  destination GL-RFCDEST
  performing RFC_INFO on end of task
  exporting
    GUID = GUID

```

Figure 5: RFC Function Call

```

receive results from function GL-FUNCNAME
  importing
    MESSAGE_OUT = GL-MESSAGE_OUT.
  * exceptions
  * SYSTEM_FAILURE = 1 MESSAGE GL-MESS
  * COMMUNICATION_FAILURE = 2 MESSAGE GL-MESS.

```

Figure 6: Receive Results from RFC

Boolean condition that evaluates to true or false. As long as the condition is false, the calling program hibernates.

So if you were to make a statement like the one shown in Figure 4, the program would halt execution until noon and then continue. This statement unfolds its true power when it is used with some asynchronous RFC processing. It allows you to submit the function as a separate task and later pick up the result using the RECEIVE statement (see Figure 5).

When the routine has finished, it will call the form RFC_INFO. There we can fetch the results shown in Figure 6.

In the full ABAP example shown in Figure 7, we show a small program that submits the same RFC function call several times in a background task. Then the program waits until all called RFC routines report back a successful execution.

```

function ZLOGOS_BPEL_ASYNC_ARBITER.
*****
*****Local Interface:
***** IMPORTING
***** VALUE(GUID) TYPE BAPIGUID OPTIONAL
***** VALUE(MESSAGE_IN) TYPE STRING DEFAULT 'Hello World'
***** VALUE(QNAME) TYPE TRFCQOUT-QNAME OPTIONAL
***** EXPORTING
***** VALUE(MESSAGE_OUT) TYPE STRING
***** VALUE(START_TIME) TYPE BAPIBP_TIMESTAMP
***** VALUE(END_TIME) TYPE BAPIBP_TIMESTAMP
***** VALUE(RUNTIME) TYPE I
***** VALUE(TID) TYPE ARFCTID
***** VALUE(FNUM) TYPE QRETSTATE-QRNUM
*****

  data: begin of L,
    T0 type I,
    T1 like L-T0,
    end of L.

  get time.
  concatenate SY-DATUM SY-UZEIT into START_TIME.
  get run time field L-T0.

  GL-TASK_FINISHED = ABAP_FALSE.
  GL-FUNCNAME = 'ZLOGOS_BPEL_ASYNC_SUBPROCESS'.
  concatenate 'BPEL' FUNCNAME into GL-TASK_NAME.
  *

  add 1 to GL-SND_JOBS.
  *

  call function GL-FUNCNAME
    starting new task GL-TASK_NAME
    destination GL-RFCDEST
    performing RFC_INFO on end of task
    exporting
      GUID = GUID
      MESSAGE_IN = MESSAGE_IN.

  wait until GL-RCV_JOBS >= GL-SND_JOBS.
  MESSAGE_OUT = GL-MESSAGE_OUT.

  get run time field L-T1.
  RUNTIME = L-T1 - L-T0.
  get time.
  concatenate SY-DATUM SY-UZEIT into END_TIME.

endfunction.

```

Figure 7: Parallel Processing in ABAP

Figure 7 – continued on next page

```
*&-----*
*&   Form rfc_info
*&-----*
form RFC_INFO using NAME.
  data: TASK_WA like line of GL-TASK_LIST.
*
  TASK_WA-NAME = NAME.
  GL-RCV_JOBS = GL-RCV_JOBS + 1.
*
  GL-TASK_FINISHED = ABAP_TRUE.
*
* receive results from function 'ZLOGOS_BPEL_ASYNC_SUBPROCESS'
receive results from function GL-FUNCNAME
  importing
    MESSAGE_OUT = GL-MESSAGE_OUT.
* exceptions
*   SYSTEM_FAILURE      = 1 MESSAGE GL-MESS
*   COMMUNICATION_FAILURE = 2 MESSAGE GL-MESS.
if SY-SUBRC = 0.
  TASK_WA-DEST = GL-INFO-RFCDEST.
else.
  TASK_WA-DEST = GL-MESS.
endif.
append TASK_WA to GL-TASK_LIST.
endform.          "rfc_info
```

Figure 7: Parallel Processing in ABA – continued from previous page

In Figure 8, we see a sample function module to represent a time consuming sub-process, while Figure 9 illustrates the global variable declarations for the examples shown in Figures 7 and 8.

```
function ZLOGOS_BPEL_ASYNC_SUBPROCESS.
*"-----*
***Local Interface:
** IMPORTING
**   VALUE(GUID) TYPE BAPIGUID
**   VALUE(MESSAGE_IN) TYPE STRING
** EXPORTING
**   VALUE(MESSAGE_OUT) TYPE STRING
**-----*
concatenate 'Success: ' MESSAGE_IN into MESSAGE_OUT.

wait up to 5 seconds.

endfunction.
```

Figure 8: A Sample Function Module to Represent a Time Consuming Sub Process

```
function-pool ZLOGOS_BPEL.          "MESSAGE-ID ..

type-pools: ABAP.

types: begin of TASK_TYPE,
       NAME type STRING,
       DEST type STRING,
       end of TASK_TYPE.
*
data: begin of GL,
      RFCDEST type RFCDEST,
      TASK_FINISHED type BOOLEAN,
      SND_JOBS type I,
      RCV_JOBS type I,
      EXC_FLAG type I,
      INFO type RFCSI,
      MESS type C length 80,
      INDX type C length 4,
      TASK_NAME type C length 8,
      ..FUNCNAME type funcname,
      TASK_LIST type standard table of TASK_TYPE,
      MESSAGE_OUT type STRING,
      end of GL.
*
*
load-of-program.
GL-RFCDEST = 'NONE'.
```

Figure 9: Global Variable Declarations (for the examples shown in Figures 7 and 8)

Another example of how “callbacks” are done can be shown easily using Python.

The code shown in Figure 10 says the following:

1. Try connecting to “<http://www.cnn.com>” in an asynchronous fashion and execute a callback method when the results are available.
2. When the HTML response is received executes the CALLBACK method called “printResult”. This will print out the HTML from “<http://www.cnn.com>” to the console.
3. If an error occurs during processing then execute the CALLBACK method called “errorHandler”, passing in the error and printing it to the console.
4. If after five seconds of trying to connect to “<http://www.cnn.com>” there is still no response, then issue a timeout. This is set up in the code below by using the “callLater” method and issuing a timeout in seconds.

```
from twisted.web.client import getPage
from twisted.internet import reactor

def errorHandler(error):
    print "An error has occurred: <%s>" % str(error)
    reactor.stop()

def printResult(result):
    print result
    reactor.stop()

def timeout():
    print "Timeout has occurred..."
    reactor.stop()

deferred = getPage('http://www.cnn.com/')
deferred.addCallback(printResult)
deferred.addErrback(errorHandler)

reactor.callLater(5, timeout)

reactor.run()
```

Figure 10: Python Call Backs

Conclusion

Concurrency is a Core Principle of SOA

SOA, by its very nature, is an architecture that copes with highly distributed applications. With “old school” designed applications, you can still attain reasonable performance from a bad design. However, with SOA, the distributed architecture becomes much more apparent. A rock solid stable design becomes ever so important and is achieved by providing mechanisms that allow asynchronous mechanisms on a very basic communication layer as well as intrinsic support for concurrency as a prerequisite for a successful architecture shift. Unfortunately, the market-dominating solutions are still far away from delivering useful assistance for concurrency and “assume” the responsibility for asynchronous and reliable quality to communication services.

Axel Angeli, logosworld.com. *Axel is a senior SAP and EAI advisor and principal of logosworld.com, a German-based enterprise specializing in coaching SAP and EAI project teams and advising IT management on implementation issues. Axel has been in the IT business since 1984, and throughout his career, he has always worked with cutting edge technologies. Axel's SAP experience stems from the good old R/2 days, and he is an expert on SAP's NetWeaver technology and any kind of ABAP development. A speaker of several languages, Axel specializes in coaching and leading large multinational teams on complex projects with heterogeneous platforms and international rollouts. Known for his intensive and successful trouble-shooting experience, Axel has been nicknamed by his colleagues as the “Red Adair” of SAP projects. He is the author of the best-selling tutorial “The SAP R/3 Guide to EDI, IDocs, ALE and Interfaces.” You may contact the author at SAPtips.Authors@ERPtips.com. Be sure to mention the author's name and/or the article title.*

Lynton Grice, logosworld.com, *is an SAP integration specialist contracting in Johannesburg, South Africa. He has a passion for programming and has a particular interest in the “business process and SOA space” incorporating things like workflow, BPEL, SAP XI, etc. Lynton is a champion of the Blue Elephant League of SOA Excellence. You may contact the author at SAPtips.Authors@ERPtips.com. Be sure to mention the author's name and/or the article title.*

¹ Python is a dynamic object-oriented programming language that is a true winner when it comes to agile SOA development. (<http://www.python.org/>)

NetWeaver™ 7.0 Web Application Server: Defined

By Eric Walter, SAP® Consultant

***Editor's Note:** Are you confused by the ever-changing terms and constantly evolving architecture of SAP's NetWeaver 7.0 Web Application Server? Then let's let Eric Walter help us sort it all out. Eric provides an overall look at the components involved, then discusses the kernel architecture integration for each.*

Many years ago, in the R/3 systems, SAP architecture was defined in a much simpler manner than it is in today's SAP Web Application Server. In addition, SAP tends to frequently change its terminology and jargon, thus adding yet another level of complexity (and confusion) to the evolving technical architecture. It can become overwhelming to keep up with the pace of SAP's dynamic technology and marketing terms.

In this article, we're going to try and alleviate some of this confusion. We will define and explain much of the current NetWeaver 7.0 Suite (formerly termed NetWeaver 2004s) and lay out the Business Suite applications at a high level, as well as at a technical level. In the technical level section, I will discuss the kernel architecture as it integrates with the data of SAP applications.

Background

During SAP's formative years, R/3 was one isolated system containing "function modules" that integrated all enterprise activities with one common relational database. These business entities included modules such as Human Resources (HR), Materials Management (MM), Production Planning (PP), Plant Maintenance (PM), Finance and Controlling (FI-CO), Sales And Distribution (SD), and Accounts Receivable (AR). These functional areas have dependencies on each other and maximize business productivity by ensuring that details needed from other enterprise members are met.

As time passed, however, and the information technology world evolved to embrace open systems and communication methods as standards, the need to expand the R/3 environment spawned new SAP systems with additional functionality and integration. In R/3, ABAP was the object-oriented language that ran inside of the SAP database and presented the business logic that ran the enterprise. This worked with the Basis kernel to deliver presentation through the SAP GUI client that was installed on an end-user's computer. While ABAP is

still the backbone of the SAP logic, the need for outside communication and integration led the SAP architecture to expand and include a "J2EE Engine". Now, this Java 2 Enterprise Edition addition to the SAP ABAP kernel has morphed into SAP's Web Application Server technology, which is commonly known as WebAS. This is the NetWeaver architecture that is common among each of the SAP business suite applications. This WebAS infrastructure integrates the new technology standards, allowing for communication between systems using Web services and Service Oriented Architecture (SOA). Today these standards not only allow integration between SAP applications, they allow for communication and automation between business partners as well!

It can become overwhelming to keep up with the pace of SAP's dynamic technology and marketing terms.

Overview

We'll now take an overall look at the components included in the SAP NetWeaver Business Suite and define the technical architecture of NetWeaver 7.0.

Let's say you are about to implement SAP. First, you would receive a very large box of DVDs in the mail with a bill of materials for about 60 to 70 DVD labels. These labels would include titles such as SRM Master, ERP 2005 Master, NetWeaver Kernel 700 (with a variety of operating systems), installation exports, upgrade exports, SAP Business Suite 2005 Master, and Database Installation Master. It's easy to be confused as to what needs to be done with all this media, so let's try to get a better handle on it by looking at the big picture.

The Business Suite

First of all, it is important to understand that NetWeaver is made up of different systems that perform different business functions with separate databases (unless you are using SAP's Multiple Component One Database-MCOD architecture). Figure 1 provides an example of the NetWeaver components.

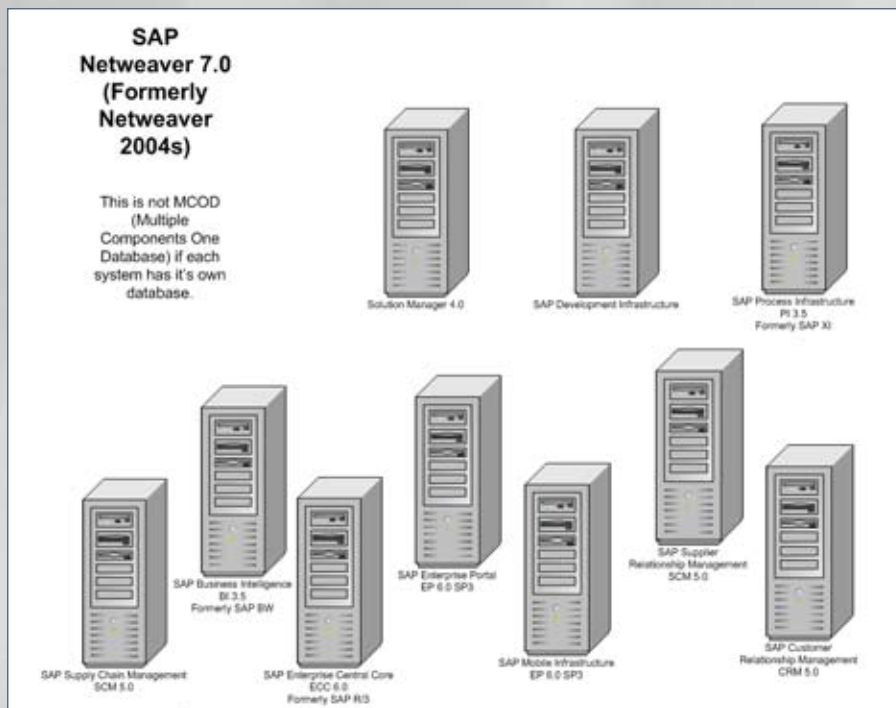


Figure 1: SAP NetWeaver Business Suite

As you can see, the standard NetWeaver 7.0 Business Suite includes many systems that perform different activities. Each one of these systems is a separate SAP instance. However, the technical architecture of each one of these systems uses the same kernel architecture, which at the time of this writing, is kernel level 700. The data that is imported into each of these systems databases during installation is what makes them different. As mentioned earlier, NetWeaver also includes a J2EE engine that allows for Web front-end presentation of the GUI and Portal through a Web browser. It also allows for information exchange using its information integration broker, Process Infrastructure (PI – which was formerly Exchange Infrastructure – XI).

ECC 6.0 and Solution Manager

If you have worked with earlier releases of R/3, you would upgrade your R/3 system to what is now called Enterprise Central Core 6.0 (ECC 6.0). Another name for this ECC 6.0 system is ERP 2005 (as you can see, this terminology and versioning can become confusing).

The R/3 functionality of ECC 6.0 (or ERP 2005) remains in the system along with upgraded enhancements. The kernel for this release will still be level 700.

SAP has a pre-requisite that an instance of Solution Manager be installed on your system before you can install or upgrade to ECC 6.0. When you install or upgrade any NetWeaver application, one of the first steps in the procedure will prompt you for a Solution Manager Key. This enforces the Solution Manager prerequisite.

Once you have installed Solution Manager 4.0 and ECC 6.0, you will notice that the filesystem structures appear identical to one another. You will also notice that the kernel levels are the same. Figures 2 and 3 show the output for the kernel release for menu

option system-> status for an ECC 6.0 system, and a Solution Manager 4.0 system, respectively.

After seeing this, it should become clear that the difference between these NetWeaver products comes strictly from the data imported into these systems, and not the general architecture. This is true with the rest of the NetWeaver Suite applications (i.e., CRM, SRM, BI).

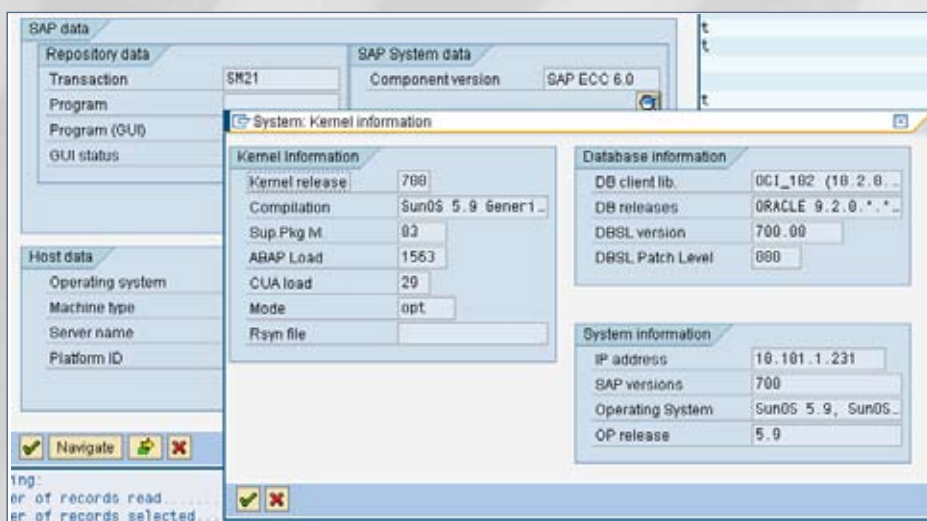


Figure 2: SystemStatus Menu Option Showing the Kernel Level 700 of an ECC 6.0 System

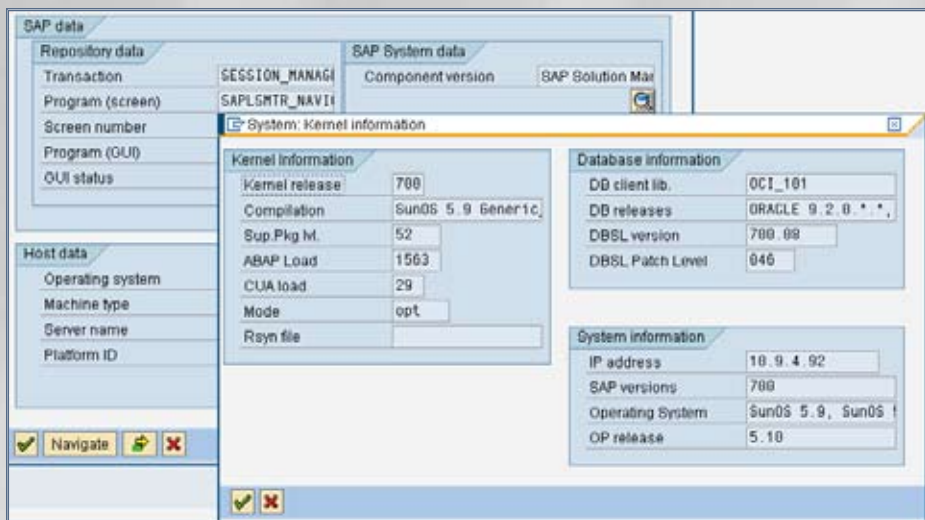


Figure 3: System Status Menu Option Showing the Kernel Level 700 of a Solution Manager 4.0 System

J2EE Engine

Another difference between NetWeaver and R/3 architecture is the J2EE engine. The ABAP engine is similar to the original architecture, where the executables/kernel live in the common directory `/usr/sap/<SID>/DVEBMGS##/exe`. This is not much of a change from very early releases of R/3. The new addition is the J2EE Engine. It also lives in a standard SAP filesystem structure and is located in:

`/usr/sap/<SID>/DVEBMGS##/j2ee`.

The files located in the J2EE directory are used to launch a Java Virtual Machine (JVM) that is similar to other J2EE Java Virtual Machines such as IBM's Websphere, BEA's WebLogic, or Jarkarta's Tomcat. One enhancement of SAP's J2EE engine is the ability to integrate directly with the ABAP workbench in SAP. This means that ABAP and Java now have the ability to integrate directly with one another. The ABAP Workbench has the ability to integrate with NetWeaver's developer interface called NetWeaver Development Infrastructure (NWDI). This integration between Java and ABAP is developed using a technology called WebDynPro. The power of this technology allows for calling an SAP Remote Function Module, enterprise service model, a Web service using the standard SOAP protocol, and/or using Enterprise Java Beans (which are also a standard J2EE). This Web DynPro component technology may be used from within the ABAP Workbench or from the Java side (NWDI). These technologies may also use the SOAP protocol as mentioned above. This enables your system (using the Process or Exchange Infrastructure – PI/XI) to integrate with systems in your company's

landscape, but now you have the ability to use standard markup languages to integrate your systems with systems running in your business partner's landscape. This may integrate with non-SAP systems as well.

Conclusion

As you can see, while much of the original R/3 functionality still resides in the NetWeaver ECC6.0 (ERP2005) application, the newer platform extends much further than R/3. The "old" SAP technologies used 3-tier architecture as a client, application, and database tiers. Now, NetWeaver has taken the Nth tier concept to a new level. Not only does the technology integrate all of your NetWeaver applications and provide Web functionality, it also allows your suppliers, customers, and end users to be connected. This greatly enhances all of your business processes.

Eric Walter is a senior consultant specializing in SAP Basis technologies along with J2EE and Java integration with SAP. He has eight years of experience with SAP R/3 Basis and security, UNIX administration, Oracle database administration, and five years with Java technologies, integration servers, and portal environments. In this time, he has led and supported numerous public and private implementations of these technologies along with developing and implementing architecture at all levels, from programming interfaces to hardware improvements and upgrades. You may contact the author at SAPtips.Authors@ERPTips.com. Be sure to mention the author's name and/or the article title.



Getting By with What You've Got: How RFC Can Be Your Friend

By Roger Myers, eBusiness Specialist, Brenntag Canada Inc.

***Editor's Note:** Sometimes the answer is right in front of you. Say you have new functionality you need to use; the SAP® platform you need to use it in doesn't support it—yet another system in your environment does. So who ya gonna call? RFC! Roger Myers has found that Remote Function Calls—a seemingly blast from the past—are a simple solution to a new problem that many companies with multiple SAP components are now facing. Roger details the key steps to accessing your new functionality where it otherwise might not work, by using technology you already have, thus getting more value from your SAP system!*

Introduction

I work for a mid-sized chemical distributor operating in the Canadian market. There was a time when our business ran end-to-end on SAP R/3. We did all our order processing, inventory management, financial reporting, etc., from within our SAP R/3 system.

A few years back, SAP started splitting off components and adding more complexity to the landscape. Now we have NetWeaver™ platform with a Service Oriented Architecture.

Many customer sites no longer employ SAP as a single instance, end-to-end solution. Many dedicated components have been designed to stand on their own, as well as connect to the R/3 - ERP backend. Business Warehouse (now BI), was introduced to handle analytics and reporting. Customer Relationship Management (CRM) was introduced to look after the customer interactions and sales side of the equation. Supplier Relationship Management (SRM) was dedicated to the supply side of the business, and so on...

These days, it is not uncommon for mid-sized firms to support a landscape composed of multiple SAP components. It is also very likely that many (or each) of these systems is on a different release level. For example, the core SAP ERP system may still be 4.6C, CRM could be version 4.0 on a 6.20 Web Application Server (WebAS), the BI engine might be running on a 6.40 WebAS, —there are many possibilities. Of course, all of these systems are designed and configured to talk to one another seamlessly in order to perform their specific missions.

The catch is the system that has the functionality doesn't have the data you need to do the work.

The challenge is what happens when your users require functionality not supported by the system they are working in but supported in one of the other SAP instances within your landscape. The catch is the system that has the functionality doesn't have the data you need to do the work.

The good news is you can gain access to newer standard functionality, available in other SAP systems, within your current landscape by using a tried and true approach to development: the "Remote Function Call (RFC)".

Now, I'll venture to guess that many of you are rolling your eyes and wondering why are we doing articles on RFCs in 2007, when there are now myriad connectors and system-to-system integration points, etc. The reason is that in order to bridge the disconnects in the landscape, I have found RFCs are relatively easy to implement, use traditional ABAP programming skills, and are well within the reach of any ABAP developer. No Java or XML skills are required.

In this article, I'll demonstrate how you can more fully leverage your SAP investment to overcome functional gaps and, therefore, get you where you need to go.

Business Case Study

Let's look at a common scenario. Company ABC is still running R/3 4.6C. They required a Smart Form with barcodes not delivered in 4.6C, in order to meet an important customer's requirements. The transactional volumes for this account were low, and the time lines were tight.

The application developer assigned to the project was not well acquainted with all of the details needed to implement barcodes. After spending a considerable amount of time looking at support notes, running demo programs, and trying to understand what was available in 4.6C, he was still not finding what was needed and time was running out.

The developer then tried something different. He logged into his CRM system running the 6.20 WebAS and found the barcode he needed was delivered in the 6.20 release. He discovered that the data is in a 4.6C system and the Smart Form with the required barcode is in the 6.20 system; this leads to the realization that this seemingly irresolvable issue is no big deal. He can simply create an RFC connection between the two systems and pass the data to the form along the RFC channel.

Getting Started

Let's get started. There are four key steps we'll discuss in detail. Our ultimate goal is rendering data collected in one system to a form in another.

1. The first step is to create the RFC destination in the data source system. This establishes the communication link between the source and target.
2. The second is to compose a data collection/transfer routine in the source system to pass data to the target system.
3. Thirdly, we need to create a helper function module to receive the transferred data.
4. The final step is to create the Smart Form, to render the exchanged information.

Step 1: Create the RFC Destination

In order for the two systems to communicate with one another, an RFC destination must be created in the sending system. The transaction for doing so is "SM59". Setting up the RFC connection between the two systems is not a big deal in terms of difficulty. However, since there are security considerations involved, the setup of the connection is best left to your Basis team. The process for creating an RFC destination is well documented in SAP's online help:

The screenshot shows the SAP SM59 transaction for creating an RFC destination. The title bar reads 'RFC Destination MY_REMOTE_SYSTEM'. The 'Test connection' tab is active, showing the RFC destination as 'MY_REMOTE_SYSTEM' and the connection type as 'H' (HTTP Connection to R/3 System). The 'Description' tab shows 'My Remote System'. The 'Technical settings' and 'Logon/Security' tabs are also visible. Under 'Logon/Security', the 'Security options' section shows 'Logon procedure' set to 'SAP Standard', 'SSL' set to 'Inactiv', and 'Authorization' set to 'DEFAULT'. The 'Logon' section shows 'Language' as 'EN', 'Client' as '010', 'User' as 'MY_RFC_USER', and 'Password' as '*****'. There is a checkbox for 'Current User' which is unchecked.

Figure 1: Example RFC Destination

http://help.sap.com/saphelp_46c/helpdata/en/22/042638488911d189490000e829fbbd/frameset.htm

Typically, a communication user is created in the target system and assigned to the RFC in the source system (see Figure 1). The actual execution of the helper function module on the remote system occurs in the name of this user.

Note: The RFC user will need the authority to run the required programs in the target system.

Step 2: Data Collection

The second step is to collect the data to be rendered in the Smart Form. A program was written to collect the product codes for items on a delivery in the source 4.6C system. The 4.6C data collection routine and the call to the remote helper function module are shown in Figure 2.

Tip: Pay attention to the data definitions in your remote helper function module's interface. If you reference tables and fields in your form interface, make sure they exist in both systems; otherwise you may get a short dump resulting from type conflicts. In our example, the data elements exist in both systems.

The program, as shown in Figure 2, is simple enough: a couple of selects and a call to a function module. The only difference between this function call and any other is the inclusion of the phrase “destination ‘<rfc_name>’”. This is the instruction to the system to call the function module in the external system defined in the RFC destination shown earlier.

Step 3: Create the Helper Function Module

Next, we need to create our helper function module “my_helper” in the remote system. Its role is to simply receive the data from the 4.6C system and pass the data to the Smart Form with the barcodes.

- The first step is to create a data structure in the remote system that we can use to hold the table data coming from the 4.6C system. Use transaction “SE11”, select data type, and enter a name for the structure (see Figure 3).

REPORT zzzzzz.

TYPES: BEGIN OF ty_lips,
vbeln TYPE vbeln,
posnr(6) posnr,
matnr(18) matnr,
END OF ty_lips.

DATA: t_lips TYPE TABLE OF ty_lips,
t_return TYPE TABLE OF bapireturn.

DATA: kunnr type kunnr.

PARAMETERS: delivery LIKE likp-vbeln.

START-OF-SELECTION.

SELECT * FROM lips INTO CORRESPONDING FIELDS OF TABLE t_lips
WHERE vbeln = delivery.

SELECT SINGLE kunnr from likp into kunnr
WHERE vbeln = delivery.

call function 'MY_REMOTE_FM' destination 'MY REMOTE SYSTEM' "RFC dest. Name

Exporting

kunnr = kunnr

Tables

t_lips = t_lips

t_return = t_return.

Figure 2: Data Collection Routine

Dictionary object Edit Goto Utilities Environment System Help

ABAP Dictionary: Initial Screen

☐ Database table
☐ View
☒ Data type
☐ Type Group
☐ Domain
☐ Search help
☐ Lock object

ZZTLIPS

Display Change Create

Figure 3: Select the Data Type

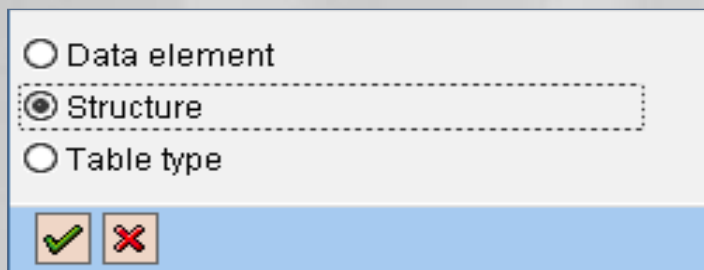


Figure 4: Select Structure

- Click on “Create”.
- In the popup that appears, select “Structure”. See Figure 4.
- Next, add the fields that you are passing in the internal table t_lips: vbeln, posnr, and matnr (see Figure 5).

The helper function module source code is shown in Figure 6. Note the call to the Smart Form.

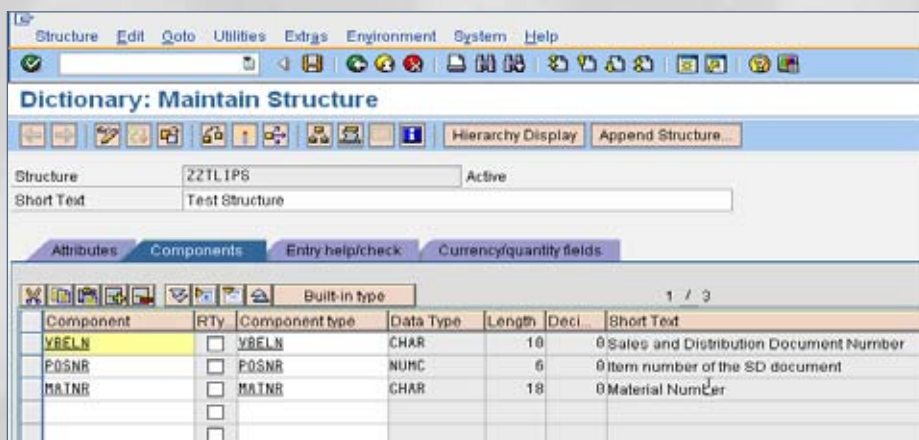


Figure 5: Add the Fields to be Passed

```

**.....
**"Local interface:
** IMPORTING
**   REFERENCE(KUNNR) TYPE KUNNR
** TABLES
**   T_LIPS STRUCTURE ZZTLIPS
**   T_RETURN STRUCTURE BAPIRETURN
**.....

data: w_ssfcompop type ssfcompop,
      w_ssfctrlrop type ssfctrlrop,
      w_return type bapireturn,
      ip_smart_form type tdsfname,
      function_name type rs38l_fnam..

w_ssfcompop-tdiexit = 'X'.
*Set the correct printer
w_ssfcompop-tddest = '<my_Printer>'. " Printer
w_ssfcompop-tdnewid = 'X'.
w_ssfcompop-tdimmed = 'X'.
w_ssfcompop-tdnprev = 'X'.
w_ssfcompop-tdcopies = '1'. " Number of copies
w_ssfctrlrop-no_dialog = 'X'. "No dialog
ip_smart_form = '<My Smart Form>'. "Name of the smart form
* get the function name for this smart form

```

Figure 6: Function Module Source Code – continued on next page

```

call function 'SSF_FUNCTION_MODULE_NAME'
  exporting
    formname      = ip_smart_form
  importing
    fm_name       = function_name
  exceptions
    no_form       = 1
    no_function_module = 2
    others        = 3.
* call the form
call function function_name
  exporting
    control_parameters = w_ssfctrlpop
    output_options     = w_ssfcompop
    user_settings      = space
* Pass in data from external system
  kunnr              = kunnr
  tables
    t_lips            = t_lips
    t_return          = t_return
  exceptions
    formatting_error  = 1
    internal_error    = 2
    send_error        = 3
    user_canceled     = 4
    others            = 5
.
if sy-subrc <> 0.
  w_return-code = sy-subrc.
  w_return-message = sy-msgv1.
  append w_return to t_return.
endif.
endfunction.
  
```

Figure 6: Function Module Source Code
— continued from previous page

Most importantly, make sure the “Remote-enabled module” radio button (in the function module’s attributes) is checked (see Figure 7).

Step 4: Building the Smart Form

The next step is to build the Smart Form (see Figure 8). The key point here is that the form’s interface lines up with the data being pulled in from the remote system. For our example, the

Processing type

☐ Normal function module
☒ Remote-enabled module
☐ Update module

☒ Start immed.
☐ Immediate start, no restart
☐ Start delayed
☐ Coll.run

Figure 7: Select the “Remote-enabled module” Radio Button

SAP Form Builder: Change Form ZZZZZ

Form ZZZZZ
Description New Form

Import Export Tables Exceptions

Parameter Name	Type Assignment	Associated Type	Default Value	Optional	Pass Val.
ARCHIVE_INDEX	TYPE	TGA_DARA		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ARCHIVE_INDEX_TAB	TYPE	TSFDARA		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ARCHIVE_PARAMETERS	TYPE	ARC_PARAMS		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
CONTROL_PARAMETERS	TYPE	SSFCTRLPOP		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
MAIL_APPL_OBJ	TYPE	SWTOBJID		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
MAIL_RECIPIENT	TYPE	SWTOBJID		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
MAIL_SENDER	TYPE	SWTOBJID		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
OUTPUT_OPTIONS	TYPE	SSFCOMPPOP		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
USER_SETTINGS	TYPE	TODDOL	%C	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
KUNNR	TYPE	KUNNR		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figure 8: Build the Smart Form

additions to the Smart Form interface reflect the same elements as the helper function module (see Figure 9).

We have collected the data in our source system and enabled the RFC to pass the data to the remote target system. The collected data is now available for subsequent processing by the Smart Form in the target system. Now you are ready to start coding the form layout using the data pulled in from the remote system.

Parameter Name	Type Assignment	Associated Type
T_LIPS	LIKE	ZZTLIPS

Figure 9: Identifying Data Between Modules

Conclusion

In our example, we have gained access to a standard piece of functionality available in the 6.20 system but not available in 4.6C.

The RFC can connect you to SAP functionality you already own that may be disconnected from the data you need to work with. If the data is not where it needs to be, the RFC can provide the means to quickly (and relatively easily) get it there. On the flip side, once you do upgrade, you'll only need to copy your code and form back to your core system, and the RFC connection can be eliminated.

Writers note: For our example, there may have been other options available to accomplish what we achieved using the RFC channel. Our objective was to simply illustrate an often-overlooked option to work around functional gaps when something needs to be done. Code snippets and system configuration are presented as examples only and are not meant to be used as is in a production environment.

Roger Myers, eBusiness Specialist, Brenntag Canada Inc. Roger been working with SAP since 1996 as a senior SD consultant and application developer. He spent ten years in industrial sales and marketing before moving into information technology. Skilled in ABAP, SAP workflow, and Web development, Roger is always on the hunt for cost-effective solutions to the day-to-day issues facing the user community. Roger is currently the Program Chair for the ASUG Small and Medium Enterprise (SME) Special Interest Group and would love to here from any SME's out there. You may contact Roger at Saptips.Authors@ERPtips.com. Be sure to mention the author's name and/or the article title.



SAP® Mail Function Module Cookbook: A Guide for Developers and Consultants

By Rehan Zaidi, Siemens Pakistan

Editor's Note: *You've got mail! Even better, you've got capabilities in SAP to program and trigger email notifications with some options embedded within ABAP. Rehan Zaidi explains the functions needed to accomplish this, along with the structure of the programs that employ these functions and the key parameters you'll need to use.*

Introduction

The ABAP language provides a variety of options for meeting complex business needs. One such option includes a set of function modules that allow you to program and trigger emails. These modules may be used for sending mail notifications to both internal and external users. Due to the ease with which mail-sending options may be incorporated in custom programs, learning the basics of these modules is essential for SAP developers.

The aim of this article is to provide you with an explanation of these function modules, and how to use them in order to achieve your purpose. These are some of the questions this article will address:

- What are the various function modules available for managing mail from SAP?
- What is the structure of programs that employ these function modules?
- What are the most important parameters of the commonly used function modules?

I will start with an explanation of how mail is displayed within SAP, and the different modules that may generate it. Then, I will discuss in detail the importing and exporting parameters of the associated modules. I will also cover the coding required for creating the body of the mail text. Finally, I will discuss a practical example in order to illustrate the discussion.

This paper is primarily intended for ABAP developers, but may also be of interest to SAP functional consultants and users. I will assume that the reader is familiar with ABAP programming as well as the execution of function modules in an SAP environment. For more information, refer to the SAP documentation on <http://help.sap.com>.

All the coding examples have been taken from Release 4.7.

SAP Mail Documents: An Overview

SAP provides internal mail functionality for notifying users of any operation performed in the system or any action that needs to be taken. You may generate mail notifications from the SAP system that are addressed to both internal and external users. The external users may be users of any other system or any user with an Internet email address.

The SAP internal mail appears as "Documents" in the Inbox of the user's workplace. This may be accessed via the Business Workplace (transaction SBWP). Each document is denoted internally by an object ID, and is comprised of two portions – the document contents and attributes. Initially the mail appears as a document in the "Unread Documents" portion of the Inbox (see Figure 1). After it has been opened and read by the user, the document appears in the "Documents" area of the Inbox.

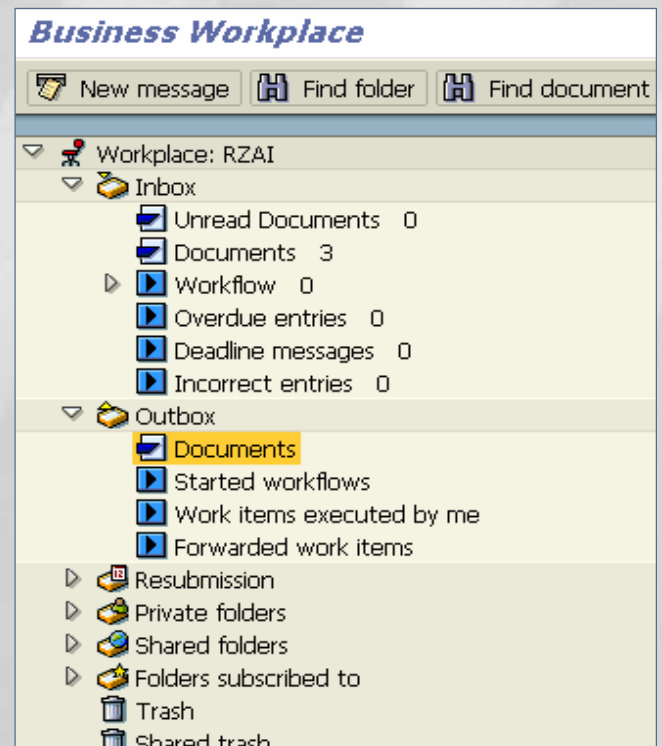


Figure 1: Folders in the SAP Business Workplace (SBWP)

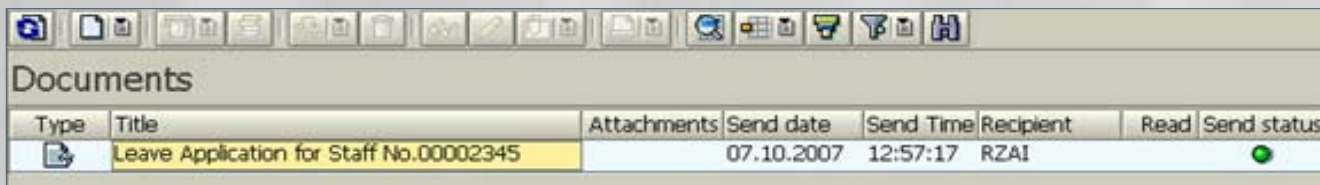


Figure 2: Documents Shown in the Right Pane of SAP Business Workplace (SBWP)

The mail document appears in the right pane, as shown in Figure 2.

Double-clicking the relevant document line displays the document contents. You may click the “Attributes” tab in order to view the properties of the document in question, as shown in Figure 3.

The SAP documents are managed by function modules provided in the R/3 system. Most of these function modules reside in the function group SO11. They may be used for sending mail documents to users in the local SAP system or to external users. Figure 4 provides an overview of some of the function modules that may be used in ABAP programs.

Via these modules, you may either create entirely new documents or change existing ones. You may also send an express message to the user to whom the mail is sent. This lets you immediately notify logged users that mail has arrived in their inboxes (see Figure 5).

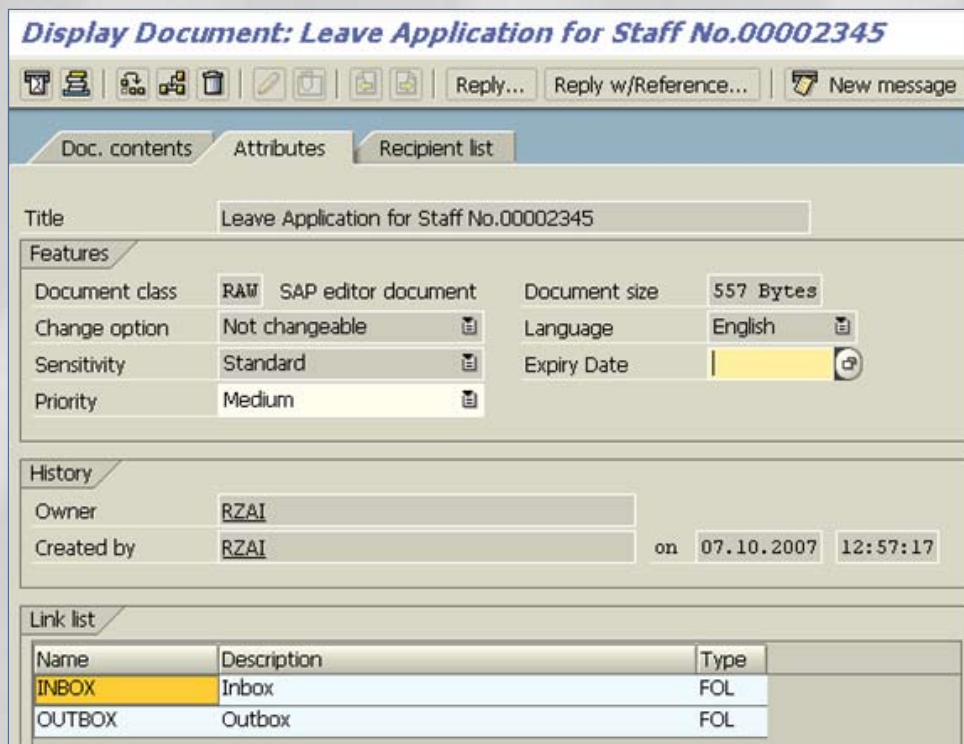


Figure 3: Attributes of an SAP Mail Document

Function Module	Purpose
SO_NEW_DOCUMENT_SEND_API1	Used for creating a new document and sending it to specified users
SO_DOCUMENT_UPDATE_API1	Used for changing an existing document
SO_OLD_DOCUMENT_SEND_API1	Used to send existing documents
SO_FOLDER_READ_API1	Retrieves contents of folders (like Outbox, etc.); the object ID is used for searching documents
SO_NEW_DOCUMENT_ATT_SEND_API1	Creates a new document and sends it along with attachments
SO_OLD_DOCUMENT_ATT_SEND_API1	Used to send existing documents along with attachments

Figure 4: Function Modules Used for Generating Mail Documents

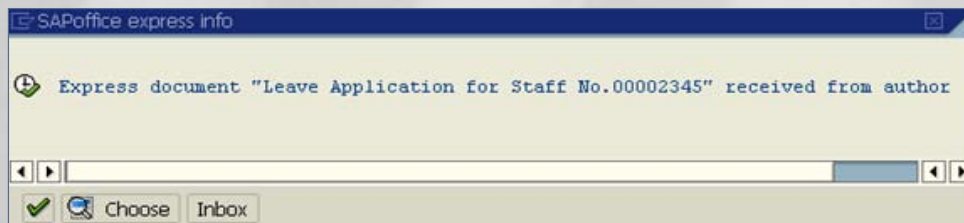


Figure 5: Express Messages for User Notification

A commonly used function module for creating mail documents is SO_NEW_DOCUMENT_SEND_API1. This function module lets you send a new document (mail) to SAP users as well as external users. A simple call of the function module is shown in Figure 6.

CALL FUNCTION 'SO_NEW_DOCUMENT_SEND_API1'

EXPORTING

DOCUMENT_DATA = MYDOC

PUT_IN_OUTBOX = 'X'

TABLES

OBJECT_CONTENT = MAIL_CONTENT

RECEIVERS = MYREC

Figure 6: Call of Function Module SO_NEW_DOCUMENT_SEND_API1

At the time of mail dispatching, the document is created based on the specified attributes and contents. You must define, through programming, the receivers to whom the SAP office mail will be sent. You may program logic for dynamically determining the names of the potential users.

Importing Parameters of SO_NEW_DOCUMENT_SEND_API1

Now, let's discuss the key importing parameters of the function module SO_NEW_DOCUMENT_SEND_API1:

DOCUMENT_TYPE: As the name indicates, this parameter denotes the type of the document. The default value for this parameter is "RAW". You may pass other values as well, with the exception of FOL (folders) and DLI (distribution lists)

PUT_IN_OUTBOX: If you check this indicator, a copy of the sent document is also placed in the Outbox of the user who launched the Email triggering program.

DOCUMENT DATA: This is an important structure and lets you specify the data pertinent to the newly created document. The most commonly used fields of this structure are shown in Figure 7.

Let's discuss the use of the key ones:

a) **OBJ_NAME:** This field is used to specify the name of the document object.

Attributes Components Entry help/check Currency/quantity fields						
Built-in type 1 / 20						
Component	RT	Component type	Data Type	Length	Decim	Short Description
.INCLUDE	<input type="checkbox"/>	SO OBJ CHG11	---	0		0Data of an object which can be changed
OBJ_NAME	<input type="checkbox"/>	SO OBJ NAM	CHAR	12		0Name of document, folder or distribution list
OBJ_DESCR	<input type="checkbox"/>	SO OBJ DES	CHAR	50		0Short description of contents
OBJ_LANGU	<input type="checkbox"/>	SO OBJ LA	LANG	1		0Language in Which Document Is Created
OBJ_SORT	<input type="checkbox"/>	SO OBJ SRT	CHAR	10		0Sort field
OBJ_EXPDAT	<input type="checkbox"/>	SO OBJ EDT	DATS	8		0Expiration date of object
SENSITIVITY	<input type="checkbox"/>	SO OBJ SNS	CHAR	1		0Object: Sensitivity (private, functional, ...)
OBJ_Prio	<input type="checkbox"/>	SO OBJ PRI	CHAR	1		0Document priority
NO_CHANGE	<input type="checkbox"/>	SO OBJ CP	CHAR	1		0Document cannot be changed
PRIORITY	<input type="checkbox"/>	SO REC PRI	CHAR	1		0Priority
EXPIRY DAT	<input type="checkbox"/>	SO EXP DAT	DATS	8		0SAPoffice: Expiration Date for this Folder Entry
.INCLUDE	<input type="checkbox"/>	SO PROCES11	---	0		0SAPoffice: Processing parameters for document
PROC_TYPE	<input type="checkbox"/>	SO VM TYP	CHAR	1		0Execute: Type (Report, Dialog Module, ...)
PROC_NAME	<input type="checkbox"/>	SO TEXT030	CHAR	30		0SAPoffice: Text field
PROC_SYST	<input type="checkbox"/>	SO PROC SY	CHAR	10		0System in Which the Document Is to Be Processed
PROC_CLINT	<input type="checkbox"/>	SO PROC CL	CLNT	3		0Client in Which Document Is to Be Processed
SKIP_SCREEN	<input type="checkbox"/>	SO SKIPS	CHAR	1		0Execute: Process First Screen in Background
TO DO OUT	<input type="checkbox"/>	SO ACOUT	CHAR	1		0ToDo: To Be Completed Outside SAPoffice
FREE_DEL	<input type="checkbox"/>	SO DEL ALL	CHAR	1		0Object Can Be Deleted by All Users with an Application
DOC_SIZE	<input type="checkbox"/>	SO DOC_SIZ	CHAR	12		0Size of SAPoffice Document (for API1)

Figure 7: Fields of the Document Data Structure

- b) **OBJ_DESCR:** The value populated in this field is displayed as the subject of the document or SAP office email. For example, if you want to display the value Prompt Action Required, the field DESCR must be assigned this value, as shown below:

DOCUMENT_DATA-OBJ_DESR = 'Prompt Action Required'.

It is not necessary for the subject lines to contain only hard-coded data. You may also display SAP fields in it. For example, you may display the subject as Prompt Action Needed For Employee Number 852. In this case, the employee number is not a constant, but needs to be assigned dynamically so that for each employee, the subject line displays a different employee number. This may be achieved by concatenating the text with the variable containing the employee number value.

CONCATENATE 'Prompt Action Required for' EMP_NO INTO DOCUMENT_DATA-OBJ_DESCR.

- c) **SENSITIVITY:** This denotes the sensitivity of the document in question. The values permitted in this field are shown in Figure 8.

Value	Meaning
O	Normal Sensitivity
F	Functional
P	Confidential

Figure 8: Permissible Values for the Sensitivity

- d) **OBJ_LANGU:** This field denotes the language of the document.
- e) **OBJ_EXPDAT:** This is the expiry date of the document in question.

Exporting Parameters and Exceptions of the Function Module

There are two exporting parameters, namely NEW_OBJECT_ID and SEND_TO_ALL of the mail function module. The NEW_OBJECT_ID returns the ID of the object generated as a result of function module execution. The SENT_TO_ALL parameter, when having a return value of "X", denotes that the document has been sent to all the receivers successfully.

Moreover, there may be exceptions generated as a result of the function module call. These are generated when unexpected or erroneous situations occur. For example,

when the document could not be sent to any of the desired receivers, or when there is a problem of authorization, an exception may occur.

It is not necessary for the subject lines to contain only hard-coded data.

Writing the Body of the Email

The most important part is to formulate the text of the mail document. This is done by populating the internal table to be passed against the parameter OBJECT_CONTENT of the function module. Each row of this internal table represents a line that is displayed as the text of the mail.

This internal table based on the dictionary structure SOLISTI1 has a field LINE denoting each line of the email text. You need to formulate and then append each line of the mail text into the internal table in order for it to be displayed. The text of the mail may be dependent upon a number of factors. You may use an IF statement in order to check conditions and vary the text of the mail document.

For defining the mail text, you first need to define an internal table based on structure SOLISTI1 as shown here:

DATA: MAIL_TEXT LIKE SOLISTI1 OCCURS 0 WITH HEADER LINE.

For adding a blank line, the code is:

**CLEAR MAIL_TEXT.
APPEND MAIL_TEXT.**

Moreover, you may also display constant text as well as variable values in the mail text. For example, if you need to display an employee name within the line of Email, the code will look as follows:

EMPLOYEE_NAME_LENGTH = STRLEN(EMP_NAME).

**CONCATENATE EMP_NAME(EMPLOYEE_NAME_LENGTH)
'has applied for annual leave'
INTO MAIL_TEXT SEPARATED BY SPACE.
APPEND MAIL_TEXT.**

Component	RT	Component type	Data Type	Length	Decim	Short Description
RECEIVER		SO_RECNAME	CHAR	1215		0 SAPoffice: Name of the recipient of a document (also ext.)
REC_TYPE		SO_ESCAPE	CHAR	1		0 Specification of recipient type
REC_ID		SO_REC_ID	CHAR	17		0 Recipient ID
REPLY_DOC		SO_REP_DOC	CHAR	46		0 Object ID of the document being replied to
REC_DATE		SO_DAT_RC	DATS	8		0 Date document was received
PROXY_ID		SO_SUB_ID	CHAR	17		0 ID of the substitute who (also) received the object
RETRN_CODE		SO_RET_COD	CHAR	4		0 Return code
.INCLUDE		SOSNDATT11		0		0 SAPoffice: Send Attributes
EXPRESS		SO_SND_EX	CHAR	1		0 Send express
COPY		SO_SND_CP	CHAR	1		0 Send: As copy
BLIND_COPY		SO_SND_BC	CHAR	1		0 Send as a blind copy
NO_FORWARD		SO_FORFB	CHAR	1		0 You cannot forward this message
NO_PRINT		SO_PRIFB	CHAR	1		0 Object cannot be printed
TO_ANSWER		SO_OBJ_RR	CHAR	1		0 A reply must be sent for this object

Figure 9: Dictionary Structure SOMLRECI1

In this example, we first determine the length of the employee name. The employee name is then concatenated with the hard-coded text and appended into the MAIL_TEXT internal table.

Specifying the Receivers

The table parameter RECEIVERS allows you to specify the information pertinent to the receiver(s) of the document. For each receiver or receiver group, a separate row must be defined. The internal table parameter is based on the dictionary structure SOMLRECI1, as shown in Figure 9.

The internal table has the following significant fields:

- RECEIVER:** The name of the receiver(s) may be specified using this field of table RECEIVERS: The receiver may either be an SAP user name, a shared distribution list, or an Internet address of the receiver.
- ADR_TYPE:** In order to specify what type has been entered in the RECEIVER field, a suitable value must be entered in the ADR_TYPE field. Figure 10 shows the permissible values and their meaning.

Value	Meaning
B	SAP user ID
F	Fax number
C	Shared Distribution List
U	Internet Email Address

Figure 10: Permissible Values of ADR_TYPE

- REC_DATE:** This is used to specify the date on which the document is to be sent to the receivers. If no date is specified, the document is dispatched when the function module program is executed.
- EXPRESS:** If the value "X" is passed, the express flag is set for the document in question. An express pop-up message is sent to the recipient if he or she is logged into the SAP server. The recipient of the document is logged. This is used if you want to notify the recipient about the document that has been received in his or her Inbox.
- COPY and BLIND_COPY:** If you like to send a document with the attribute having value Copy, you need to check the COPY parameter. Similarly, if the BLIND_COPY is passed the value "X", the attribute of the document is set as Secret Copy (i.e., it may

not be forwarded or printed from the Business Workplace transaction).

- f) **NO_FORWARD:** This parameter is used for controlling the forwarding of the document in question. If the value "X" is passed for this parameter, the receiver may not be able to forward the document using the SAP Business Workplace.
- g) **NO_PRINT:** If you want to restrict the user from printing the document, the NO_PRINT parameter may be used. You may simply pass the value "X".
- h) **TO_ANSWER and TODO_EXPL:** You may also impose certain restrictions with respect to the deletion of the documents. The TO_ANSWER parameter makes replying to the document mandatory before it can be deleted from the SAP Inbox. Likewise, the TODO_EXPL parameter makes it necessary for the document to be processed in order for it to be deleted.

Putting It All Together

In this section, I will use the concepts mentioned in this article to show how a simple company requirement may be fulfilled. Suppose you need to write a program that checks the employees who are to attend training courses commencing in the following week. An SAP office mail is to be sent to the supervisor of the employees in question.

The FORM routine within the program that deals with this requirement is shown in Figure 11.

```
FORM SEND_EMAIL USING P_EMP_NO P_EMP_NAME
                        P_UNAME.
```

```
DATA: MYREC LIKE SOMLREC1 OCCURS 0 WITH HEADER LINE.
```

```
DATA: MYDOC LIKE SODOCCHG11.
```

```
DATA: MAIL_CONTENT LIKE SOLISTI1 OCCURS 0 WITH HEADER LINE.
```

```
DATA : SUBJECT LIKE SODOCCHG11-OBJ_DESCR.
```

```
DATA : EMPLOYEE_NAME_LENGTH TYPE I.
```

```
.....
```

```
.....
```

```
MYREC-RECEIVER = P_UNAME.
```

```
MYREC-EXPRESS = 'X'.
```

```
APPEND MYREC.
```

```
MYDOC-OBJ_NAME = 'NAME'.
```

```
CONCATENATE 'SAP-Related Training for Employee No. ' P_EMP_NO INTO SUBJECT.
```

```
MYDOC-OBJ_DESCR = SUBJECT.
```

```
MYDOC-SENSITIVITY = 'O'.
```

```
EMPLOYEE_NAME_LENGTH = STRLEN( P_EMP_NAME ).
```

```
CONCATENATE P_EMP_NAME(EMPLOYEE_NAME_LENGTH)
```

```
'has to be sent to an SAP Training Course. '
```

```
INTO MAIL_CONTENT-LINE SEPARATED BY SPACE.
```

```
APPEND MAIL_CONTENT.
```

```
CALL FUNCTION 'SO_NEW_DOCUMENT_SEND_API1'
```

```
EXPORTING
```

```
DOCUMENT_DATA          = MYDOC
```

```
PUT_IN_OUTBOX          = 'X'
```

```
TABLES
```

```
OBJECT_CONTENT          = MAIL_CONTENT
```

```
RECEIVERS                = REC
```

```
EXCEPTIONS
```

```
TOO_MANY_RECEIVERS      = 1
```

```
DOCUMENT_NOT_SENT       = 2
```

```
DOCUMENT_TYPE_NOT_EXIST = 3
```

```
OPERATION_NO_AUTHORIZATION = 4
```

```
PARAMETER_ERROR         = 5
```

```
X_ERROR                  = 6
```

```
ENQUEUE_ERROR            = 7
```

```
OTHERS                    = 8
```

```
.
```

```
ENDFORM.
```

Figure 11: Mail Sending Form Routine

The routine inputs the name of the SAP user to whom the mail is to be sent, along with the employee number and name. At the beginning of the routine, the data objects for specifying receivers, document attributes, and contents are declared. The receiver table MYREC is then populated with the information of the SAP user. Since an express message is to be sent to the receiver, the EXPRESS indicator is assigned a value of "X".

The document data is then set and the subject of the email is specified, along with the hard-coded text and employee number variable. The text of the mail body is then formulated and inserted into the internal table MAIL_CONTENT. Finally, the function module NEW_DOCUMENT_SEND_API1 is called. Since the business process requires that the sent document must be placed in the Outbox of the user from whose ID the mail program is run, the parameter PUT_IN_OUTBOX is passed the value "X".

This FORM routine when executed sends an email to the respective supervisor of the employee in question.

Conclusion

In this article, I have discussed the various function modules needed for triggering emails from ABAP programs. I covered in detail the important parameters of these function modules. At the end, I showed a working example based on the mentioned concepts. I hope this article proves to be a precious resource for your ABAP team, and will help them in easily incorporating the email triggering option in their programs.

Rehan Zaidi, *Senior SAP Consultant, Siemens Pakistan. Rehan has been involved in both ABAP development and functional configuration for SAP HR implementations at multinational and local companies, and also has experience with SAP Workflow. He has contributed articles to the SAP Professional Journal, the HR Expert newsletter, and to the TechRepublic Website. He is currently working on his first book, specifically designed for SAP HR Users and Managers, as well as a guide for ABAP/Workflow Consultants titled "201 Interview Questions on Workflow". Rehan is the founder of the Web site www.siteofSAP.com. You may contact the author at SAPtips.Authors@ERPtips.com. Be sure to mention the author's name and/or the article title.*



Turning ERP on Its Head:

Can SAP® eSOA Change Companies' Approaches to ERP?

By Jon Reed, JonERP.com

***Editor's Note:** It's still in its infancy, but according to Jon Reed, it's a trend worth watching. eSOA could be the solution of the future for those companies that need to incorporate customization to their SAP system, but are skittish about doing so for a multitude of reasons. Jon discusses the opportunities that exist with eSOA and provides some preliminary feedback from companies who are using this technology to enhance their systems, without the issues previously encountered when customizing SAP.*

Introduction

Attendees of SAPHIRE 2007 got one message loud and clear: Enterprise SOA (eSOA), SAP's proprietary version of Service Oriented Architecture (SOA), has arrived. Or has it? True, we saw a number of case studies from SAP customers who have taken the Enterprise SOA plunge. But as of today, most companies are in either the "hypothetical" or the pilot project phase. That makes this an ideal time to assess the value of eSOA before more companies dip their toes into that powerful current.

In this article, I'll take a look at the reported benefits from some actual SAP eSOA projects that have taken place thus far. We'll examine the true merits of eSOA, and by the end of this piece, we'll have a better handle on what eSOA can do and why it may actually be one technical advancement that lives up to the hype.

There may not be many eSOA projects in the works yet, but SAP professionals certainly believe that the eSOA era is coming. On a recent survey I did on JonERP.com, an SAP professional cited "Enterprise SOA" as the skill area he wanted to learn next, ahead of NetWeaver™ Portals and NetWeaver BI. There seems to be a growing sense across the SAP market that eSOA is "for real." In this article, I'll explain why.

Before we get into the benefits of eSOA, there's a question we have to tackle first: if eSOA is ready for SAP customers, then why isn't the usage of eSOA more commonplace yet? It's no longer a matter of waiting on the technology from SAP, though it's true that some of the biggest eSOA innovations, such as a fully loaded Enterprise Services Repository (ESR), are still very recent.

That makes this an ideal time to assess the value of eSOA before more companies dip their toes into that powerful current.

The release of NetWeaver 7.1 in September 2007 marks the first edition of NetWeaver with a fully stocked ESR, which now serves as the metadata repository of all service objects for Enterprise SOA. But even before the release of 7.1, there were enough Web-enabled SAP services, xApps, and pre-delivered business content to give SAP customers a great start on eSOA. So why aren't more customers in the middle of eSOA projects?

The answer involves a paradox of sorts. Once the structure for utilizing eSOA is in place, it's not hard to put eSOA into action. But the eSOA structure does have some prerequisites, and it turns out those prerequisites are not simple to put into place. I'm not going to detail all the prerequisites in this article, but preparing for eSOA does require both a technical and business transformation.

The obvious major requirement is an SAP NetWeaver platform, preferably 7.1 or at least 2004s. Most SAP customers are not yet live on ERP 6.0 (the ERP suite that runs on these NetWeaver releases), so we can expect companies to remain focused on the core upgrades before they spend too much time on eSOA.

Other pre-SOA requirements that SAP customers have cited include:

- a coherent strategy for managing master data throughout the enterprise
- a "governance" program to manage the services that are being created and ensure they are compliant with business regulations

Another aspect of governance is ensuring that services are centrally managed so that they can be accessed across the enterprise. "Governance" is really a fancy way of saying, "Let's make sure to have a process in place so that you and I aren't working on the same project at the exact same time."

To add to the ramp-up obstacles, the skills involved in running eSOA projects are not the same as those needed for a classic SAP implementation. Companies have achieved eSOA success by retraining internal employees and mixing in some expert outside consultants, but suffice to say, anticipating eSOA skills needs is another key consideration before launching.

When we take all these factors together, we can understand why the move to eSOA is more of a slow adoption creep than an all-out charge. But the move is still happening because there are benefits to eSOA that can't be achieved in traditional ERP environments.

Proven Benefits of eSOA

1. Customization Without the Upgrade Hassles

The biggest benefit of eSOA is the return of customized functionality without the hassles. In my last article for SAPtips, which focused on midsize SAP implementations (CIO Corner, August/September 2007), I talked about how the best-of-breed era was essentially over for SAP customers who were standardizing on SAP throughout the enterprise. There was a bit of confusion over that statement that I hope to clear up this time around.

The proclamation that "best-of-breed is dead" is certainly true in terms of standardizing on the "back office" or core functional areas of SAP. These days, it's almost unheard of to find a company that is running separate ERP systems for Financials, Human Resources, and Manufacturing and attempting to integrate the three. Those kinds of headaches are the stuff of ERP legend. Today, companies are saving their integration budget for areas beyond the ERP core that offer a competitive advantage in their industry.

Integration in the context of eSOA, however, is much more affordable. One of the benefits of eSOA is that it allows customers to look at a range of customization options without altering the enterprise core. Because eSOA is based on "open standards," the options for plugging best-of-breed applications into SAP have never been better. In that sense, perhaps the true era of hassle-free best of breed is just beginning. Of course, using SAP's own Enterprise Services or xApps, or even

building your own, will probably be easier than a best-of-breed alternative. The point is that eSOA allows more plug-in flexibility without the heavy price.

That price is something we know well from ERP customers who warned us of financial consequences for over-customizing SAP when it comes time to upgrade. These "upgrade horror stories" are not specific to SAP. The difficulty of upgrading any heavily customized ERP package has led to the folk wisdom that if you aren't willing to use an ERP system "out of the box", maybe you shouldn't use an ERP system at all.

eSOA changes these tradeoffs. Properly utilized, eSOA allows companies to stay within the base configurations for the enterprise core, while building custom eSOA apps on top of that core that will not be impacted by the upgrade cycle. And these eSOA apps, or services in some cases, can be re-used throughout the enterprise.

Integration in the context of eSOA, however, is much more affordable.

2. The Return of Business Process Reengineering—Who Would Have Thought?

"Business process reengineering" is a catch phrase so crusty we can barely remember uttering it. In fact, most of us deny ever using the phrase. But look out – process reengineering may be making a comeback. The ability to re-use and customize Enterprise Services has led some ERP visionaries, such as Mitch Morris, Director of Business Intelligence at Deloitte Consulting, to argue that we can now return to the business process reengineering (BPR) approach of the mid-90s.

This approach to ERP was largely abandoned when we learned that you have to accept the bulk of the ERP vendors' built-in business processes for your industry in order for it to be successful. But with the power of eSOA, the business process re-engineering mindset is suddenly relevant again.

Essentially, with eSOA, you can sit down and chart out your optimal business processes. You can then figure out which of those options are configurable within SAP. Every business process you can fit into the SAP enterprise core goes there. Any business process that does not fit into standard SAP can then be a candidate for an Enterprise Service.

During the Home Depot eSOA presentation at SAP-PHIRE '07, Kent Sanders, a Manager with SAP's Customer Advisory Office, described a similar approach that SAP is recommending for early stage eSOA customers. Sanders explained that as companies look at options for composite applications and Enterprise Services, they should also re-evaluate existing business processes to make sure that they aren't using "outdated business rules" based on yesterday's ERP limitations.

Now we know why SAP is calling its new NetWeaver release a "Business Process Platform". The name may sound a bit generic, but the implications are anything but ordinary. The ability to take advantage of SAP's best practices, but without sacrificing the specific business processes that make your company special, has never been easily doable. eSOA has the potential to change all that.

Cardinal Health, a flagship SAP customer, detailed this new approach at SAPPHIRE '07. Brent Stutz, Director of Enterprise Architecture at Cardinal, explained that they set a goal to incorporate 80 percent of their business processes into the SAP enterprise core. The remaining 20 percent of their functionality that could not be fit into SAP then became candidates for "composite applications", which is a common term for building applications made up of Enterprise Services and re-usable components.

Historically, companies in Cardinal's situation would have had a very tough choice with that remaining 20 percent of functionality: live without it, change what they do to fit into SAP's structure, or invest in costly SAP customizations that would put them at risk for future upgrades and pose problems for application support. But using the eSOA framework, Cardinal had a range of options for what to do with that remaining 20 percent.

Wait a minute...does this mark the end of the tried-and-true "80/20 rule"? Well, we probably shouldn't go that far into eSOA evangelism, but so far, this looks like a major change in how companies go about implementing ERP. The fact that companies like Cardinal are

already doing it really gives teeth to Mitch Morris' "BPR is back" viewpoint.

3. Reduced Development Costs

eSOA also has the potential to save big money on development costs. eSOA projects promote the re-usability of services, and allow users to build applications with visual building blocks. Both these techniques tend to save money and time over traditional "cubicle coding". With the release of NetWeaver 7.1, SAP has pulled all these new development tools together into the NetWeaver CE (Composition Environment).

So far, the testimonials on reduced development costs are promising. What are the implications? eSOA custom development not only saves hassles during the upgrade cycle, but the development work itself is a cost savings. In a December 2006 study, Gartner predicted that through 2008, companies that implement "enterprise-wide CBD (component-based development) re-use methods in support of SOAs will increase developer productivity by 40 percent over the next five years". And in Cardinal Health's SAPPHIRE 2007 presentation, they reported that their eSOA projects have proven to be two thirds cheaper than traditional development projects.

NetWeaver-driven SOA development allows technical teams to build Web services, and sometimes even complete business processes, using re-usable "building blocks". Modeling environments like Visual Composer streamline the coding process, and to a certain extent, allow business users to participate in the overall structure of a new program or service. It's not surprising that a programming process based on re-usable pieces would be more efficient and affordable than the classic line-by-line development effort.

4. Extending SAP to the Casual User

Another benefit of eSOA is the ability to extend SAP to casual users. The casual user doesn't want to have to deal with the SAP GUI screen, and CIOs do not want the overhead of installing and maintaining client-based ERP systems. By allowing the casual user to access role-based SAP functions via a Web browser, eSOA saves installation costs while giving users access to the specific SAP information they need in a familiar, browser-based environment.

Of course, eSOA is not the only SAP initiative that serves this function. Enterprise Portals, now called NetWeaver Portals, is inspired by the same user-friendly mentality. SAP's partnerships with Microsoft (Duet) and Adobe (Forms) are also part of SAP's overall goal of

giving users the SAP data they need in the navigation environment in which they are most comfortable.

The early reports are in: SAP customers are reporting success extending SAP to casual users via eSOA and role-based Portals.

At this point, the early reports are in: SAP customers are reporting success extending SAP to casual users via eSOA and role-based Portals (SAP Portals now ships with more than 800 “Business Packages” of built-in, role-based user content). Most importantly, casual users are embracing SAP with unexpected levels of enthusiasm because it is being presented in a familiar browser-based format. Smooth user adoption not only saves on training and change management costs, but it also ensures that the water cooler talk stays focused on sports and American Idol, instead of SAP bashing.

5. With eSOA, You Can Start Small

The strange payoff of eSOA is that once the initial platform is in place, you can start small. A good analogy might be: you have to build the runway (NetWeaver, core upgrades, and all the other prepwork), but then you can fly and land any size plane. What we saw at SAPHIRE '07 was a number of smaller scale eSOA and xApps success stories. For example, I did a one-on-one interview with Gisa, an SAP customer that had implemented an SAP Enterprise Service for Electronic Bill Presentation and Payment (EBPP) in just two total weeks of labor time. This EBPP service integrated seamlessly with Gisa's enterprise core. The catch, of course, is that they were already running on mySAP® ERP 2005. But once you are running on a services-friendly platform, you can roll out eSOA gradually, building on the ROI momentum as you go forward.

6. No More “Interface Parting Trauma” During Upgrades

Last, but not least, a commitment to eSOA means fast, and, dare we say, painless upgrades. In that sense, the real payoff for eSOA will come years down the road, when it's once again time for a major upgrade to the enterprise core. In theory, at that point a company will get to maintain all their SOA-based functionality without having to bid fond farewell to their favorite

hand-coded interfaces that would previously have had to be abandoned. The only catch is a good one: The new eSOA add-ons won't be hand-coded using the traditional approach of a team of ABAP programmers doing the heavy lifting of specialized customizations. Therefore, because the core code base will not have to be altered, upgrading to newer releases of SAP in future years should be much smoother. As companies compose and customize eSOA apps, they will run on top of the ERP code base. The ultimate promise of eSOA is that you don't have to “go vanilla” to be guaranteed a smooth upgrade. If SAP can deliver on that promise, then it will redefine basic assumptions about ERP. If and when we reach that point, SAP R/3 will feel as distant to tomorrow's ERP user as R/2 feels to today's user.

Conclusion

SAP eSOA is a monster topic, far beyond the scope of one article. If there is reader interest, I will continue to touch on eSOA in future issues of SAPtips, and share more case study details on “eSOA in action”. What we focused on this time were the true benefits of eSOA as per the testimonies of SAP customers who have gone this route.

SAP is a visionary company with a well-honed hype machine. In this case, though, the buzz about eSOA does seem to be justified. At minimum, we can say, “so far, so good”. In a few weeks, I'll be heading to SAP TechEd 2007, where I intend to get an earful on the next stages of eSOA in the wake of NetWeaver 7.1.

Jon Reed, *JonERP.com*. *Jon Reed is an independent SAP analyst who writes on SAP consulting trends. He is the President of JonERP.com, an interactive Web site that features Jon's SAP Career Blog and his podcasts for SAP professionals. Jon has been publishing SAP career and market analysis for more than a decade, and he serves as the career expert for SearchSAP's “Ask the Expert” panel. He is the author of the SAP Consultant Handbook. From 2003 to 2006, Jon was the Managing Editor of SAPtips. You may contact Jon at SAPtips.Authors@ERPTips.com. Be sure to mention the author's name and/or the article title.*



So What's Jon Reed Up To These Days?



Starting with the first edition of SAPtips in February of 2003, Jon Reed served as the Managing Editor. Jon assembled the first team of writers for SAPtips and helped to build the Document Library that is such a great resource for SAP® customers today. Jon brought that independent attitude to SAPtips that continues to set us apart from other SAP publications.

Jon moved on from a management role in SAPtips in March of 2007, but you can still catch his “CIO Corner” articles, which appear in SAPtips on a regular basis. We wanted to let you know about Jon’s new Web site, www.JonERP.com, which should be a valuable site for those who are interested in SAP skills trends.

JonERP.com is an interactive Web site that monitors SAP consulting trends and, as per its tagline, provides “Career Answers for SAP Professionals.” Through the JonERP.com “SAP Career Blog”, SAP professionals can pose career questions to Jon across functional and technical skills categories. [JonERP.com](http://www.JonERP.com) also features podcasts with industry insiders, where Jon and his guests examine the “human side of SAP implementations” and identify the skills necessary to implement the latest SAP technology.

JonERP.com will soon launch the “Hiring Trends in SAP” newsletter, which will provide SAP managers with the keys to recruiting and retaining top SAP talent through an analysis of SAP hiring trends in “summary brief” format. [JonERP.com](http://www.JonERP.com) fulfills Jon’s dream of creating his own Web site for SAP professionals who are looking to make the best project choices in the context of a broader career strategy. He hopes that the honest commentary on the site will lead [JonERP.com](http://www.JonERP.com) to become a trusted destination for SAP professionals in the NetWeaver™ era.

We’d like to wish Jon well in his new venture and we look forward to publishing his take on the SAP market in future editions of SAPtips.



To ASP or Not to ASP?

By CJ Rhoads

***Editor's Note:** To ASP or not ASP, that is the question. 'Tis better to "rent" or to "own" applications? Ask CJ Rhoads her opinion of Application Service Providers (in case you were confused as to which ASP we're talking about here), and you'll get a very frank discussion on the pros and cons of this concept. If you're wondering if ASP is for you, then you'll want to read what CJ has to say first. (We'll give you a hint: Neither a borrower or a lender be.)*

What is ASP

The acronym ASP stands for Application Service Provider. This definition is not to be confused with ASP: Active Server Pages, which is a middleware programming language of Microsoft's. In this article, we are only concerned with the Application service provider definition.

Of course, that still does not explain what it is. Basically, companies that offer ASP services "rent" the use of an application across the Internet. For example, instead of purchasing and implementing a Microsoft Exchange Email Server for a few thousand dollars, a company would pay \$25 per month per person for a vendor to provide Microsoft Exchange Email services to all their employees. The idea is an old one, harking back to the timeshare mainframe system days; an organization pays a monthly amount to subscribe to a service instead of purchasing hardware and installing software.

It was thought that the ASP business model might work very well for ERP packages such as JD Edwards®, SAP®, or Oracle E-Business Suite. Sometimes this is referred to as "hosted ERP". Another oft-quoted good example of software in need of ASP is CRM (Customer Resource Management). Over the years, there have been hundreds of vendors who have developed ASP services providing a variety of applications.

Although born of the Internet Boom in the late nineties, the ASP business model seemed to live beyond the dot com bust. In 2003, the industry was estimated at several billion dollars. Tanja Falkowski did a thesis at Braunschweig Institute of Economics called Feasibility Study on the Use of the ASP Business Model for Enterprise Application Software in 2002, at the height of the ASP proliferation. She notes that in 2001 ASPisland

and ASPnews, both aggregator sites for ASP-based vendors, reported between 1814 and 2128 ASP vendors providing services.

Basically, companies that offer ASP services "rent" the use of an application across the Internet.

Advantages of ASP

The vendors of ASP services can identify many benefits you'll derive as a client by allowing them to provide the application services to your employees instead of having an internal information technology team install and implement the system. Utilizing an ASP vendor enables the customer company to:

- Focus on core competencies, and leave the complexity of installing and maintaining applications to the experts.
- Solve the problem of locating and retaining skilled IT staff.
- Reduce total cost of ownership by 30 to 50 percent (according to the Gartner Group).
- Cut the time necessary to deploy comprehensive applications company wide.
- Enable quick implementation (often days instead of months or years) at a very low initial cost.
- Eliminate the problem of forced upgrades.
- Have 24 X 7 technical support, physical security, and disaster recovery services at a very low cost.
- Rely upon a predictable monthly fee instead of dealing with unknown cost surprises and annual forced upgrades.

It is interesting to note that there are many more advantages of the ASP model for the vendor. However, since we are focused on the client or customer needs, we won't go into them here.

Disadvantages of ASP

Of course, nothing is perfect. The disadvantages of ASP to the customer/client are:

- The client must rely upon the security and integrity of the vendor, as they have no direct control over the environment.
- Integration with non-ASP software can be problematic.
- For large implementations and contracts of long durations, the costs can far exceed self-implementation.

Falkowski noted in her thesis that despite the disadvantages, application service providers represent a "viable alternative" to in-house deployment of software applications. She describes several successful ASP vendors as shown in Table 1.

Of course, we have the benefit of time in that we know which of these companies truly became successful since 2002. Out of the seventeen companies listed in Table 1, four have been acquired, five are no longer doing what could be described as ASP services, and two went completely out of business. Only six are still in business using an ASP business model.

Personal Experiences with ASP

Do you remember NetPC - the thin-client, terminal-like system? Much hyped in the late 1990s, it was presented as the death knell for Windows PCs. NetPCs promised simpler implementation, more control, lower costs, and easier-to-use interfaces. In 1995, when they were first introduced with much fanfare, I had immediately published my opinion that they were not going to go anywhere. The fact is that people like to control their own personal computers. NetPCs were going backwards, back to the days of terminals where everything was controlled by pocket-protected mainframe programmers

behind thick glass datacenters. Even the full-force of Larry Ellison could not make that dead horse run. Well, I must admit I have the exact same feeling about the ASP business model. It simply can't work. Here's why:

- The complexity involved in managing two completely different businesses using a single platform drastically increases the costs involved in implementation. That complexity (and cost) increases exponentially with each additional business installed, eliminating any hope of economy of scale.
- Those added costs must be passed onto the customers, who then end up paying for not only their own application services, but also the added cost of complexity for other customers AND the profit margin of the vendor.
- Additionally, the lack of control over the environment, the real risk of losing business critical data should the ASP go out of business (a very real possibility), and a potential inability to customize the software to meet the specific needs of the customer all add up to a decrease in the quality of ASP over in-house service.

Vendor	Services Offered or Provided
LearningStation	Provides Office applications.
McAfee	Provides security and virus protection.
Personable.com	Provides Office applications and file sharing servers.
Agilera	Provides JD Edwards, Lawson, Oracle and PeopleSoft services.
BlueStarSolutions	Provides SAP, Ariba, InterShip, Siebel applications.
Outtask	Offers Siebel, Great Plains, PeopleTask (a Human Resources program).
Frontera	Provides Vignette, E.piphany, and BroadVision access.
Descartes	Provides inventory management applications.
Logistics.com	Provides shipping and carrier management applications.
ShipLogix	Provides transportation management software.
TriZetta Group	Provides healthcare software and services.
Scheduling.com	Provides patient scheduling.
Portera	Provides automation applications for the professional services industries such as accounting, law firms, investment banks, marketing service providers.
OpenAir.com	Provides project management software and services.
Surebridge	Offers ERP and CRM applications.
Iterelate	Offers CRM services.
Salesforce.com	Offers sales automation services.

Table 1: ASP Vendors and Their Services

- The decrease in quality combined with the increase in costs means that customers will ONLY use the service during the initial, low-cost (for them) implementation. Once they learn how to use the system (on someone else's dime), they might realize what they need customized and move on to their own installation. That would add to the already-too-high fixed costs of the vendor, forcing them to price the services out of the market, increasing the chances that the vendor will not be able to make a profit and stay in the black.

What happens most often is that the vendor only attracts small, unprofitable clients, and only for a short time.

It is interesting to note that in the waning days of the Internet boom, the ASP business model promised to bring vendors of web services back into profitability by selling subscription services. It was one of the last business models still supported heavily by venture capital. That's because, on paper, economies of scale show that if the vendor can attract enough customers for a single installation, they can make a huge profit because, again – on paper – the installation has a fixed cost. What happens most often is that the vendor only attracts small, unprofitable clients, and only for a short time. They never get enough of them to reach break even or, alternatively, they do get enough of them only to discover that the cost of the increasing maintenance blows all the profits before they are even made.

Most of the ASPs declared successful currently are still strongly supported by venture capital – much of it coming directly from Oracle, the vendor for the underlying database that many of them run on. Larry Ellison's support of the ASP industry echoes the earlier days of the NetPC. I don't think he can carry the whole industry on his back, however, so it may be that a rendering is near.

Morphing ASP

If you've been in the technology industry for a long time, then you know that new buzzwords are introduced all the time. ASP as a buzzword first appeared on the scene in 2000. As noted, the venture capitalists had been supporting the ASP business model, but now it seems to be losing some ground. Most of the ASP vendor-member organizations are no longer in business,

or were acquired by Internet.com, a prolific purchaser of failed dot coms. (Bias Alert: Internet.com purchased Cambridge Information Network from Earthweb, to whom Cambridge Technology Partners had sold it at the height of the dot com boom and for which I served on the advisory board). Sad to say, the on-line community of technology executives did not survive the transition.

ASPnews.com, on the other hand, was also acquired by Internet.com, but it seems to have survived the transition. However, there is much more talk about SaaS and SOA than ASP on the ASPnews website. SaaS stands for Software As A Service, and SOA stands for Service Oriented Architecture. (Rather than trying to explain what SOA is, if you don't already know, take my word for it that the definition of SOA changes with each consultant you speak with, rendering it a useless concept and closer to marketecture than any actual products or services. I tell my clients that they can safely ignore all the hype surrounding it.)

Getting back to SaaS, the most recent editorial posted on ASPnews.com is called *SaaS, An Idea Whose Time Has Finally Come...Again*, by Julie Craig. The article starts out with the premise: "It started with ASPs, but they dwindled; now software as a services (SaaS) is poised to do what the ASPs couldn't." Craig notes that Salesforce.com's CEO Jim Steele believes that 25% of software sales will be subscription-services by 2011. She also describes the new company Workday.com started by Dave Duffield, formerly of PeopleSoft. Workday.com is developing ERP subscription services, and expects it to be fully operational by 2008.

Obviously, the terminology ASP is losing its buzz. The concept, therefore, is morphing into on demand software services, hosted software services, software subscription services, or software as a service. No matter what it is called, the question remains whether customers would prefer to purchase, own, and control their own systems, or rely upon a vendor to provide the services that run their operations.

Summary: To ASP or not to ASP?

I believe I know the answer to the question. If given a choice, customers (at least customers in the United States) would prefer to purchase, own, and control their own systems. Consider the difficulty in getting Americans to give up their individual cars and ride the much more efficient busses and trains. We do not give up control easily, and ASP services will undoubtedly be less convenient than having an in-house system.

That doesn't mean to say that it will never work. There are some occasions where ASP (or SaaS or on-demand software or hosted software) may be to the customer's benefit.

One of my clients, a Fortune 500 company, wanted to install JD Edwards in dozens of offices. According to the CIO, since they didn't have any in-house expertise in JD Edwards, it would have been extremely difficult to find the skilled personnel (especially since this decision was made before PeopleSoft®/Oracle® acquired JD Edwards). It would also have been expensive to invest in the installation of the software and hardware. Instead, they partnered with a vendor using the ASP model, to provide the ERP services to their employees.

It worked very well for them. According to the Vice President who managed the project, the service was fantastic. The vendor was an expert at implementation and ensured a very graceful transition from the old system to the new system. They mapped out a plan, worked through all the issues, and got the system running smoothly.

My client kept adding so many users that eventually they passed the break-even point. Within a year, despite the excellent service, and even with the capital expense of buying and implementing a system and the ongoing expense of hiring a certified JDE® expert, they decided to terminate the ASP contract. The CIO recognized that they "used" the vendor to learn about the implementation, stating "Now that we are mature users of the product, we found bringing it in-house to be much more cost effective."


Notice, however, that despite excellent service, the customer chose to cut off the vendor, probably just as the account was getting profitable. From a customer point of view, the ASP service worked great. But from a vendor point of view, I don't see how this can be a winning long-term business model. Some ASP vendors may try to prevent this type of problem by making it difficult, if not impossible, for the client to easily get their data back and imported into an alternative platform. But that's not good for the customer, and has already given the industry a black eye. I have several clients who regret their decision to put their data on a vendor's server.

Last year a company called me - desperate. There was no answer at the corporate office or the technical support line of their ASP vendor. They had no in-house backup of the data, and no access to the system in order

to download their own data. Unfortunately, I could not help them.

The bottom line is this: If you want to implement new software that is available in an ASP model, but don't have the in-house expertise yet, you might consider an ASP. Go in with your eyes fully open, however, and be sure you keep a synchronized on-site backup of all your data at all times.

Otherwise, if I were you, I would stay away. And if you are considering investing in a company with an ASP business model, all I can say is that I wouldn't risk my pension.

Dr. CJ Rhoads is a well-known guru, speaker, and author on how to make better decisions about business strategy and technology. Rhoads is the founder of ETM Associates, Inc., a Douglassville, PA, based enterprise technology management consulting firm (ETMAssociates.com). She's also an Associate Professor in the College of Business at Kutztown University, as well as author of over a hundred published articles, dozens of manuals and whitepapers, and two forthcoming books. She can be reached at CJRhoads@ETMAssociates.com. 

SAPtips Associate Editors

Logistics	Sutrisno Japit, SAP FI/CO Consultant
Pricing	Matthias Liebich, MLI Solutions, LLC
CRM	Scott Cameron, CRM Consultant
Financials	Bill McNaughten, Enterprise Resource Corp.
HR and Payroll.	Satish Badgi, Unisys Corporation
Business Warehouse (BW).	Anurag Barua, Independent SAP Consultant
Basis.	Eric Walter, Senior SAP Consultant
ABAP Development	Rehan Zaidi, Siemens Pakistan
Data Archiving.	Breck Whitten, Halliburton

We are always looking for outstanding SAP® consultants and subject-matter experts who want to write for us on a regular or semi-regular basis. Get involved with the SAPtips editorial team today, by emailing Managing Editor Cheryl Cave at Cheryl.Cave@ERPtips.com



Laura Donovan,
Director of Publications

Laura Donovan is Director of Publications for Klee Associates, Inc., with responsibility for the SAPtips and JDEtips Journals and the editing and production of our Mastery-Level Training Manuals. Laura has over 30 years experience in the software arena, where she has specialized in user documentation and training development and delivery. She's held in-house publications and training management positions and has worked as a consultant.

Laura may be reached at:

Laura.Donovan@ERPtips.com
1.877.832.2594 ext 190

The publisher, Andy Klee, is the President of Klee Associates, Inc., a premier provider of JD Edwards and SAP implementation and training services.

Andy's personal goal for 2007 is to help more clients optimize their use of JD Edwards and SAP, and realize more value for their software investment.

Andy.Klee@ERPtips.com
1.970.856.4811



SAPtips is published six times per year. Copies are available online to paid readers. Preview all content at: www.SAPtips.com

The information in our publications and on our Website is the copyrighted work of Klee Associates, Inc. and is owned by Klee Associates, Inc. NO WARRANTY: This documentation is delivered as is, and Klee Associates, Inc. makes no warranty as to its accuracy or use. Any use of this documentation is at the risk of the user. Although we make every good faith effort to ensure accuracy, this document may include technical or other inaccuracies or typographical errors. Klee Associates, Inc. reserves the right to make changes without prior notice. NO AFFILIATION: Klee Associates, Inc. and this publication are not affiliated with or endorsed by SAP AG. SAP AG software referenced on this site is furnished under license agreements between SAP AG and its customers and can be used only within the terms of such agreements. SAP AG and mySAP are registered trademarks of SAP AG. All other product names used herein are trademarks or registered trademarks of their respective owners.

