

# **Adding Flexibility to Validation & Substitution Rules**

by **Rohana Gunawardena, Exium Inc.**

A<sup>b</sup>  
C

Originally published in SAPtips magazine  
Brought to you by Exium Inc.

# SAPtips

## Adding Flexibility to Validation & Substitution Rules

By Rohana Gunawardena, Exium, Inc.

*Editor's Note: In the perfect SAP® world, there would be no need for custom ABAP and technical modifications to R/3. But in the real world, there are times when the selective customization of R/3 can help the SAP users to greatly increase their efficiency. In the FI/CO area, one such example is developing table-based logic for validation and substitution using SAP's validation and substitution exits. Once developed, the techniques that FI/CO Editor Rohana Gunawardena recommends in this article can be re-used, saving the trouble of writing code from scratch or altering reports table-by-table each time a custom report must be modified. As usual, Rohana has beefed up his article with numerous screen shots that will allow an FI/CO team to follow each step in the methods he recommends.*

### 1. Introduction

Often substitution and validation rules are used to ensure that correct data is posted to FI-GL and FI-SL. This functionality is very flexible and powerful, however, it can end up being overly complex and it's not always a flexible way for the rules to be updated. You may have the same business logic as your substitution rule embedded in a custom report — if you change your substitution rule, can you be sure your custom report will reflect this change?

In most of standard SAP the business logic is table-driven, so many processes can share the same business logic and reports. This allows easy updating of the business logic. This type of devel-

opment may not come to the user's mind at first, and after a few such developments you will wish there was an easier way to change business logic. You will ask yourself: why isn't there a single place to change similar settings, rather than having to change configuration in several places and also change hard coding in ABAPs?

In this article, I will show you how to develop table-based logic for validation and substitution using validation and substitution exits that will allow the same logic to be easily reused in other parts of SAP and in your own custom ABAPs. There is an assumption in this article that you have knowl-

**Using the  
validation function,  
you can check  
values and value  
intervals entered  
in the  
R/3 environment.**

edge of creating basic validation and substitution rules. If your functional team finds aspects of this article a bit too technical, make sure one of your ABAP folks is reading over their shoulders. The tips described here can be utilized in all supported versions of R/3, but the screen shots are from a 4.5 system.

### 2. What is Validation & Substitution

Here is a quick summary of what validation and substitution rules do. These rules are primarily seen as FI-SL functionality, but they are designed by SAP to work for FI-GL, FI-AA, FI-LC, CO-CCA, CO-PS, EC-PCA, EC-CS, and IS-RE as well. Have a look at the list of call up points in transactions GGB0 and GGB1 to see where they can be used.

#### 2.1. Validation Rules

Using the validation function, you can check values and value intervals entered in the R/3 environment. When data is entered into the system, SAP checks the data against validation rules. As the validation takes place before posting, only valid data can be posted.

Validation rules are maintained using transaction GGB0.

#### 2.2. Substitution Rules

Substitution is the process of checking values against a user-defined rule. If the rule is fulfilled, the system replaces the specified values with other values, and the substituted values are posted.

Substitution rules are maintained using transaction GGB1.

# SAPtips

Code	Template	Description	Notes
GBLR	RGGBR000	Custom code for Boolean rules	Can only return a yes/no answer.
GBLS	RGGBS000	Custom code for substitution rules	Can have subroutines that return a variety of values, e.g., yes/no answer, single field, or class.
GBRU	RGLVU000	Custom code for roll-ups	Subroutine can set the return code SY-SUBRC and/or change values in the standard data structures, e.g., GLU1, GLU2.
GIMV	RGIVU000	Custom code for use in FI-SL field movements	Can have subroutines that return a variety of values.
KROU	RKROU000	Custom code for use in cost center summarization	Subroutine can set the return code SY-SUBRC and/or change values in the standard data structures, e.g., CCSS.
GIMF	RGIFU000	Custom code for fixed field movement	Can have subroutines that return a variety of values, e.g., yes/no answer, single field, or class.
GSVO	RGSVU000	Custom code to set formula values for sets	Can have subroutines that return a variety of values, e.g., yes/no answer, single field, or class.

Figure 1: Form Pool Templates

Result	Description	Usage	Param
Yes/No value	A boolean rule which returns a True/False value	Substitution and validation rules	C_EXIT_PARAM_NONE
Single field value	Form returns a single field value for update	Substitution rules only	C_EXIT_PARAM_FIELD
Multiple field values	Form returns a parameter defined as a type. Can be a structure, a table or a combination of these	Substitution and validation rules	C_EXIT_PARAM_CLASS

## 3. How to Add Custom Code

### 3.1. Form Pool

SAP allows users to create custom code for validation and substitution rules using form pools. The form pool is a copy of an SAP standard template that is modified by the users to contain their own code. See Figure 1 for a list of form pools and their usage.

### 3.2. Types of Forms

There are three types of forms which can be defined in the form pools. Depending on the form pool the types of forms maybe restricted.

## 4. Example – Rule Written Without Table

A common use of substitution rules is to populate the functional area, FKBER, a field used by EC-PCA based on existing data in the

FI document. In the example we cover in this article, the functional area will be derived from the cost center category. When using EC-PCA, the functional area is a key field for reporting, often matching the revenue and expense categories used for external reporting, e.g., COS, SG&A, R&D, and interest expenses.

To start, let's look at how this could be implemented as a substitution rule using existing functionality. Go to transaction GGB1, IMG -> Financial Accounting -> Special Purpose Ledger -> Tools -> Maintain Validation/Substitution/Rules -> Maintain Substitution. (See Figure 2.)

**Change Substitution: Initial Screen**

Substitution: 01-UKV

Usage:

Applicatn area	FI	Financial accounting
Callup point	0005	Cost of sales accounting

Figure 2: Select Substitution 01-UKV for Cost of Sales Accounting

# SAPtips

In this case, the substitution rule has many steps, each setting a fixed value for a specific cost center category. (See Figure 3.)

With this type of rule the risk of error by a typo is high, especially if there are many categories; also, the information cannot be shared with other modules. In this step if the cost center category is F, manufacturing cost center, then the functional area is set as 0100 for Production costs. (See Figure 4.)

In the steps 2 to 5, similar rules are defined.

## 5. Example – Rule Written With Table

Now, let's look at implementing this functionality using a custom table. Normally, I always recommend using standard SAP functionality to minimize the number of customizations in the system. However, in some cases customization, if done thoughtfully, can provide greater flexibility than using standard SAP only.

### 5.1. Copy Sample ABAP Form Pool

SAP provides sample form pool RGGBS000. Copy this to a client specific Z module pool, e.g., ZGGBS800; the last three digits are the client number.

### 5.2. Define Client-Specific Form Pool

Transaction GCX2, IMG -> Financial Accounting -> Special Purpose Ledger -> Basic Settings -> User Exits -> Maintain Client-Specific User Exits. Table T80D. (See Figure 5.)

### 5.3. Create Table

Create a custom table to store your substitution rules. In this case, I have called it ZFPCA, and

Step	Description
<input type="checkbox"/> 001	Production
<input type="checkbox"/> 002	Administration
<input type="checkbox"/> 003	Sales
<input type="checkbox"/> 004	Research and development
<input type="checkbox"/> 005	Revenue

Figure 3: The Substitution Rule Has One Step per Cost Center Category

Field	is substituted by:
<input type="checkbox"/> Functional area	Constant value 0100

Figure 4: Each Step Sets a Specific Constant Value

Appl. area	Ex.prog.	Application area
GBLR	Z66BR800	Val/sub: Exits for rules
GBLS	Z66BS800	Val/sub: Exits for substitutio
GBRU	Z6LVU800	Rollup: User exits
GIMV	Z6IVU800	Variable field movement
KROU	ZKROU800	Cst ctr summariztn: User exit

Figure 5: Define Client-Specific Form Pool

the table contains the cost center category to functional area mapping I previously defined, using the steps in the substitution rules.

(See Figure 6.) The name and fields used for your table will depend on the business logic you are trying to implement.

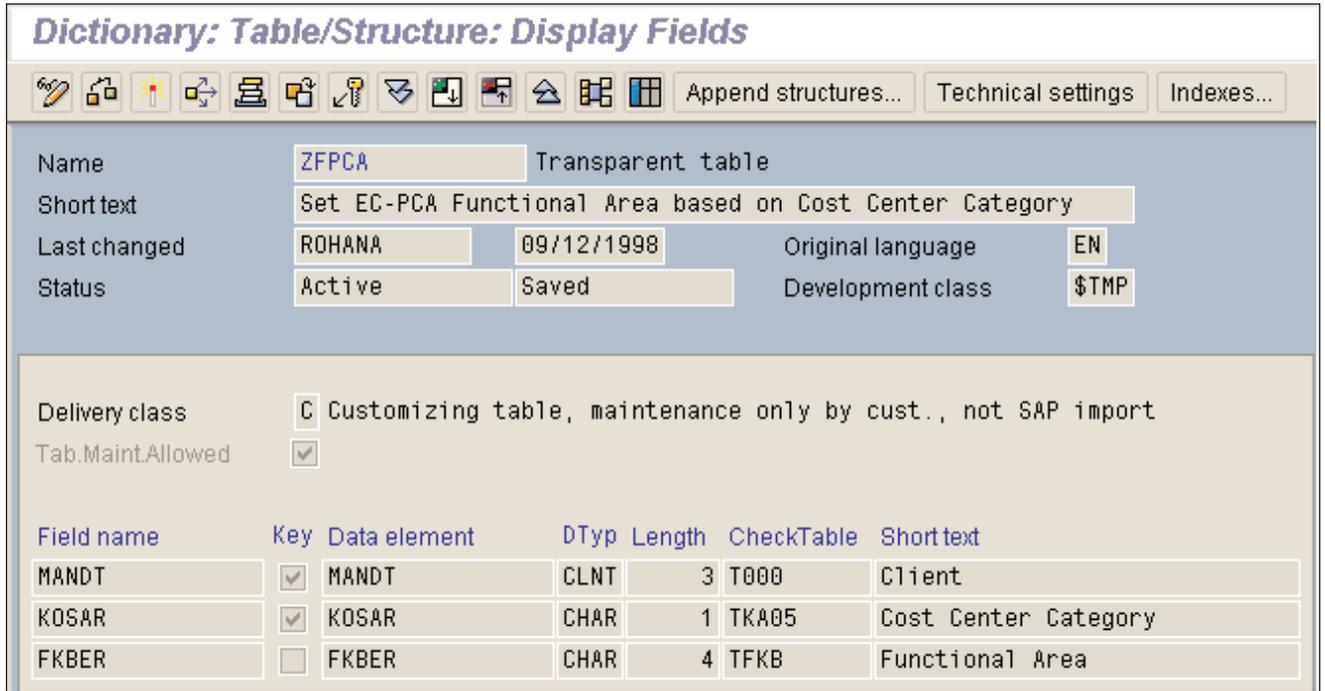


Figure 6: Custom Table ZFPCA to Contain Cost Center to Functional Area Mapping

#### 5.4. Populate Custom Table

Enter the translation values in your custom table. (See Figure 7.)

#### 5.5. Add Code

Add custom code to the client-specific form pool. In the form pool, you will need to change the code in three places.

Add the new tables you will need to use at the beginning of the form pool (see Figure 8).

Add the name of the new subroutine in form GET\_EXIT\_TITLES (see Figure 9). This enables selection of the user routine with F4 help when creating the substitution rule (see Figure 15).

Add your custom code as a new form at the end of the form pool. (See Figure 10.) The actual code you will enter will depend on the specific business logic you are trying to implement.

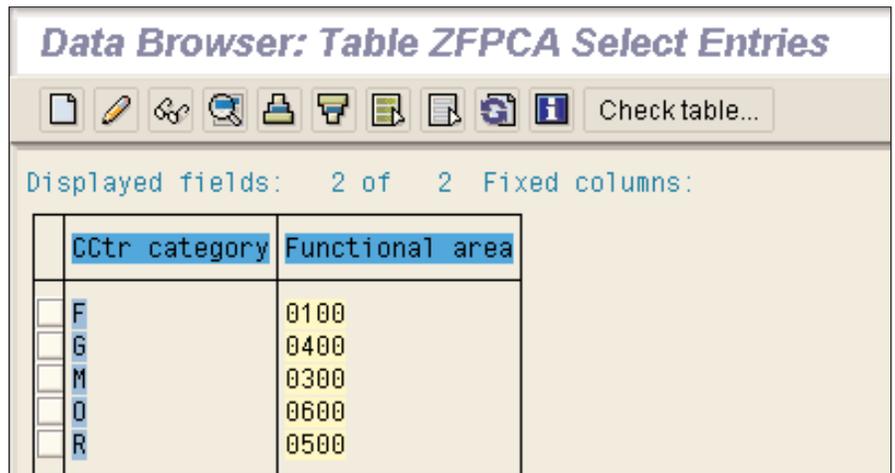


Figure 7: Contents of Table ZFPCA



Figure 8: Additional Table Definitions in the Form Pool

# SAPtips

```

FORM GET_EXIT_TITLES TABLES ETAB.
...
...
...
EXITS-NAME = 'U900'.
EXITS-PARAM = C_EXIT_PARAM_FIELD.
EXITS-TITLE = TEXT-900. "Functional Area From Cost Center Ctg

APPEND EXITS.
...
...
...
ENDFORM
    
```

Figure 9: Add Exit Name in Form Pool

```

*-----*
* FORM U900 *
*-----*
* Select the functional area based on the cost center category *
*-----*
FORM U900 USING FKBER.

CLEAR: ZFPCA.
SELECT SINGLE * FROM ZFPCA WHERE KOSAR = CSKSV-KOSAR.

FKBER = ZFPCA-FKBER

ENDFORM.
    
```

Figure 10: Add Custom Code in Form for Exit.

To add greater flexibility, you could create this code as a function module and have the subroutine call the function module. Then, the function module could be repeatedly used in other form pools or your custom ABAPs. This method is highly beneficial if you add custom logic, e.g., if the cost center category is not found in the table, then you can set the value to 0900. This makes future changes to the logic much easier to implement across many custom ABAPs, and you can re-use the function module if the same logic needs to be used in different form pools.

### 5.6. Create Substitution Rule

Create the substitution rule using transaction GGB1, menu path Substitution -> Create. (See Figure 11.) In this

case, there will be only one step in the substitution rule. The single step makes use of user rule U900 that looks up table ZFPCA to select the functional area for all cases of cost center category.

Enter a name for the substitution rule at the step overview screen. (See Figure 12.) The new substitution rule has no steps — just press the "Insert entry" button to enter the new step.

### Create Substitution: Initial Screen

Substitution	Z1-UKV
<b>Usage</b>	
Applicatn area	FI      Financial accounting
Callup point	0005      Cost of sales accounting
<b>Copy from</b>	
Substitution	

Figure 11: Create a New Substitution Rule

# SAPtips

Figure 12: Insert a Single Step for the New Substitution Rule

Table	Field	Description	Type	000000
		Only exit		000000
ACCIT	FKBER	Functional area	CHAR	000004

Figure 13: Select the Field to be Substituted in the Step

Figure 14: Select "Exit" as the Substitution Method

Enter a name for the substitution rule at the step overview screen. (See Figure 12.) The new substitution rule has no steps — just press the "Insert entry" button to enter the new step.

Now, choose the field, Functional area, to be substituted. (See Figure 13.)

Define your step as using an exit. (See Figure 14.)

At the step detail screen go to the exit field, press F4, and select exit U900 from the selection popup. (See Figure 15.) Also enter a name for the step.

At the step detail screen go to the exit field, press F4, and select exit U900 from the selection popup. (See Figure 15.) Also enter a name for the step.

The definition of the step is now complete. (See Figure 16.) Next, press the green arrow to go back to the step overview. The prerequisite logic "TRUE" is added automatically by SAP.

The substitution rule is now complete and you can save it. (See Figure 17.)

# SAPtips

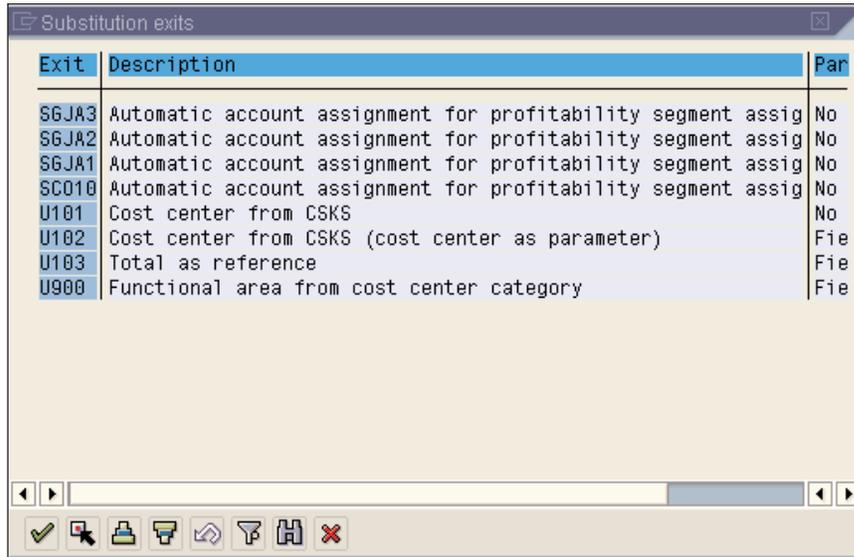


Figure 15: Select the Substitution Exit U900

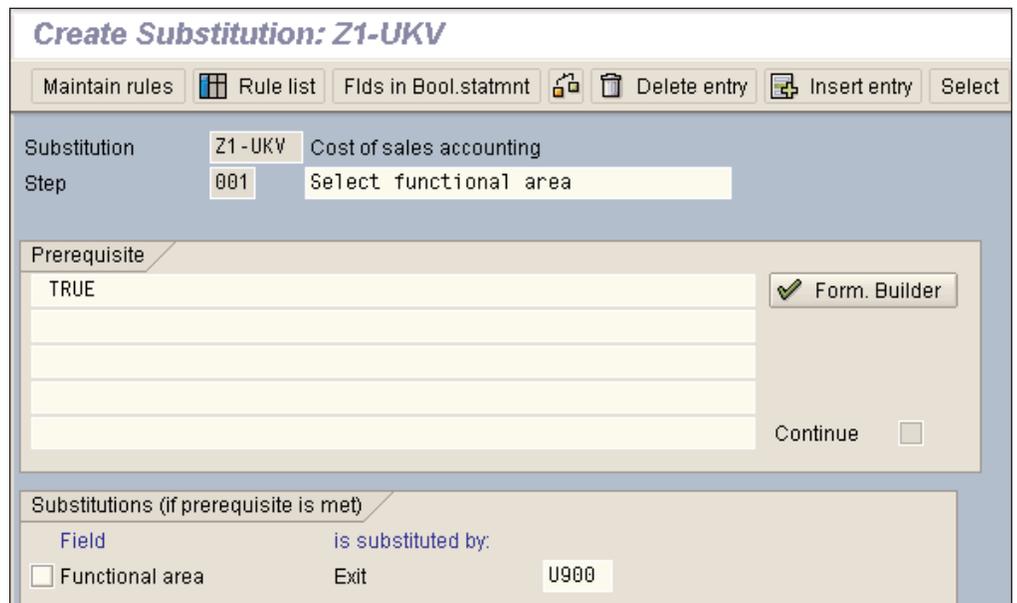


Figure 16: Step 001 Is Very Simple as All Logic Is in the Exit U900.



Figure 17: Completed Substitution Rule

# SAPtips

## 5.7. Test Rule in Simulation Mode

Go to the change or display mode of transaction GGB1 and select Substitution -> Simulate. (See Figure 18.) In the simulation mode screens, you will be able to enter the cost center category and see the selected functional area value.

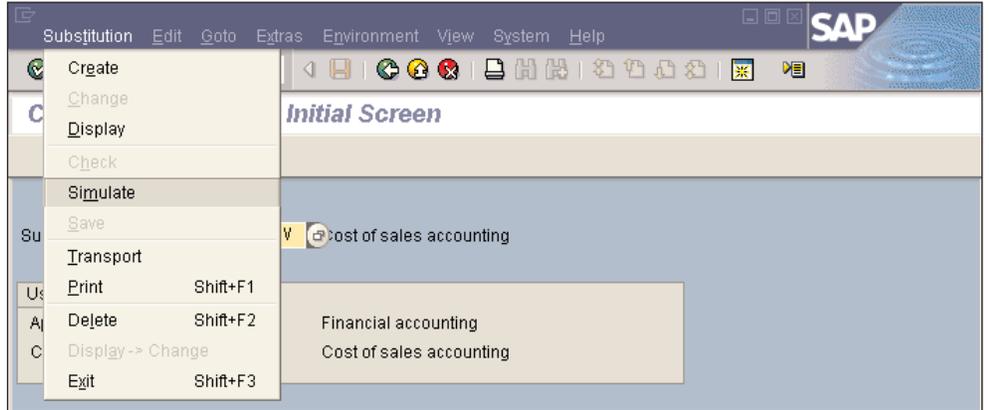


Figure 18: Use the Simulation Mode to Test Your Substitution Rule

The data entry screen for the substitution rule simulation appears. (See Figure 19.) Enter the input values for the GL account and cost center category.

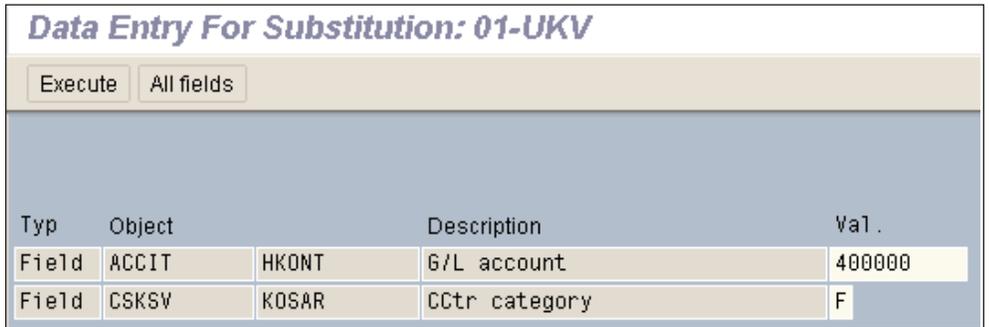


Figure 19: Data Entry for Simulation

## 5.8. Activate Substitution Rule

Finally, activate the substitution rule by assigning it to a company code in the correct call up point. In this case, use transaction OBBZ to enter the substitution rule. (See Figure 21.) You could also make the entry from transaction GGB0/GGB1 using the menu path Environment -> Val./sub. Analysis -> Activate validation/substn (button) -> Financial accounting (radio button) Substitution (button).

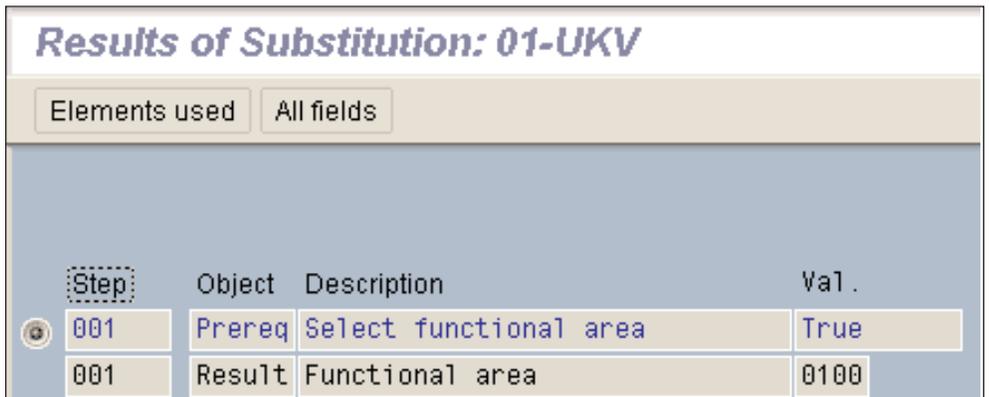


Figure 20: Results for Simulation

## 6. Summary

At this point, we now have a table-driven sub-

# SAPtips

Co...	CallPnt	Substitution	Description	Activtn level
0001	5	Z1-UKV	Cost of sales accounting	1

Figure 21: Activate Substitution Rule for Company Code 0001

stitution rule. This allows easy updating of the rule using table maintenance, e.g., SM31. It also allows the rules to be easily displayed and understood by business users. Just print out the table using SE16 and the business logic encapsulated in table ZFPCA can be used in custom reports that will automatically reflect any changes you make in the table. All of these points add greater flexibility and long-term usability to your SAP system over the short-term solution of adding all of the business logic in the GGB0 and GGB1 transactions. In the long term, this extra effort will pay off for you.

The key concept to take from this example is to introduce flexibility and reusability to the solutions you develop in SAP. Often, solutions are developed in isolation from other users on the system, and problems occur when you try to expand solutions to cover more than one area in SAP. When developing solutions in SAP, think about how they will be used in the future with any proposed expansions in functionality the business is looking for.

The functionality in this article could have also been implemented as a custom field movement routine using transaction GCF1, IMG -> Financial Accounting -> Special Purpose Ledger -> Basic Settings -> Master Data -> Main-

tain Field Movements. The form pool RGIVU000 would be copied to the client-specific version and the custom code entered in it. This goes to show there is always more than one way to do things in SAP — usually there are several options, and you have to select the best solution for your business situation.

Keep in mind that if you maintain the cost of sales substitution rule for real in your SAP system, make sure you read OSS note 85799 for some caveats. And always remember: if your functional team is not sure about the best ways to approach these technical customizations, your ABAP team should be able to fill in the gaps you need and help you to make sense of any of the technical concepts I've covered in this article.

**Rohana Gunawardena, Exium Inc.** Rohana is a Senior FI/CO Consultant specializing in global, multi-site rollouts. Rohana has been working with SAP since 1992, focusing on the FI and CO modules with emphasis on business segment reporting, cross-module integration to FI/CO, sales-cycle accounting, and A/R. He also has experience with SD and ABAP. Rohana is a Fellow of the Institute of Chartered Accountants in England and Wales. He can be reached at [Rohana@Exium.com](mailto:Rohana@Exium.com).

# SAPtips

*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.*



### **About The Author**

Rohana Gunawardena heads the SAP practice division at Exium. Exium is a leading business and technology consulting firm that enables companies to achieve their strategic business goals. Exium specializes in delivering superior IT solutions using SAP products.

Rohana has worked on a variety of SAP projects since 1992, focusing on large multi-year implementations of SAP following the full SAP life-cycle from strategic planning to production support. Rohana has a special focus on the Order-to-Cash process and cross-module integration to FI/CO.

Rohana is a Fellow of the Institute of Chartered Accountants in England & Wales. Previously Rohana has worked with the consulting practices of Accenture, Deloitte and PwC.

He has spoken at many SAP conferences and has published more than 30 magazine articles in SAP Financials Expert, SAP SCM Expert and SAPtips on various aspects of SAP. Rohana is widely acknowledged as a leading SAP expert.

You may reach him via e-mail at [Rohana@Exium.com](mailto:Rohana@Exium.com).

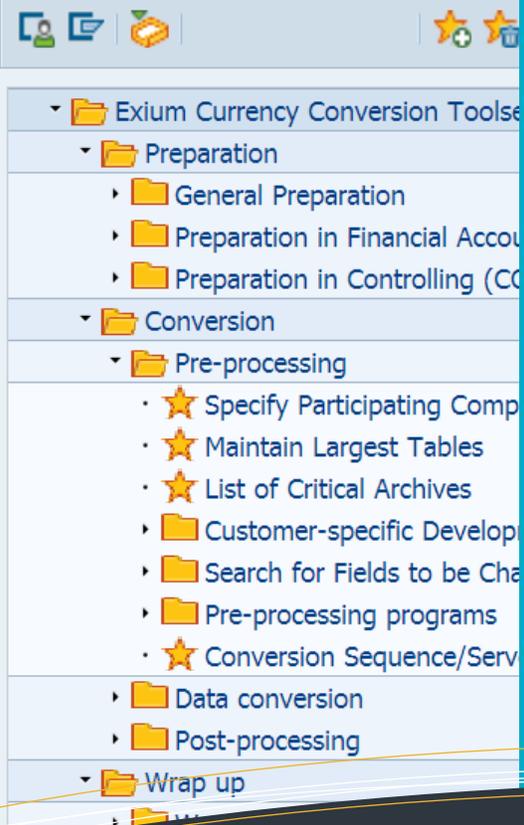
## Solution Highlights

- **Accelerated Transformation**  
Projects as quick as 8 - 12 weeks
- **Smooth and Safe Transition**  
Structured methodology to support transformation
- **Data Integrity**  
Financial data validation before and after transformation
- **Reduced Project Costs**  
Lower cost than re-implementing to correct system settings
- **Success-based Pricing**  
Fixed-bid pricing to avoid cost overruns
- **Active Participation**  
On-site team actively involved

# Group Currency Conversion



## Exium Currency Conversion



## Conversion Scenarios

Our sophisticated solutions allow organizations to maximize their ROI and extend the life of their system by correcting data, which is normally not possible through simple configuration, to reflect the current business environment.

- Post go-live activation of group currency
- Currency type alignments
- Chart of Account rationalization
- Material Ledger alignment
- Controlling Area merger
- Local Currency changeover
- many more.

**EXIUM**<sup>®</sup>  
Technology Solutions