

BRFplus for HCM Features a Paradigm Shift in Handling Decisions



Applies to

HCM Features, but used as an example to demonstrate a paradigm shift. A pre-requisite to apply the approaches explained in this document is SAP NetWeaver (ABAP) 7.0 Enhancement Package 2 or above. All screenshots are taken from SAP NetWeaver 7.03/7.31 SP3, where the functionality also is available with older releases.

Summary

The document describes, with the help of an example, how decision services implemented in BRFplus can replace HCM Features. The replacement will provide business-side experts a much better user experience, and empower them to model their business logic.

Author: Wolfgang Schaper

Company: SAP AG

Created on: February 6, 2012

Author Bio

Wolfgang Schaper is the product manager of Business Rule Framework plus. He joined SAP in 1998. Since then he has been working in various positions in development, presales, and product management for various SAP solutions like ERP, CRM, and NetWeaver. Before working with SAP, Wolfgang was an assistant professor at Mathematical Department of Humboldt University Berlin, where he graduated with PhD.

Contents

Purpose of this Document	3
The Example.....	4
Steps in the Implementation of a Decision Service	5
Definition of Task	5
Decision Service Definition	5
Decision Service Trigger Implementation	8
Decision Service Implementation.....	9
Decision Service Testing	11
Conclusion	12
Related Content.....	13
Copyright.....	14

Purpose of this Document

This document describes, with the help of an example, how decision services handling HCM Features can be implemented using BRFplus.

IT professionals will find this document very helpful as it explains and structures an approach that can minimize time and effort when dealing with changes. Once the connection of the process to the decision service has been established, further changes to the decision logic will not require code changes.

Business professionals will find this document helpful as it gives an idea of what is possible with BRFplus. BRFplus will enable them to understand the business logic of a particular decision. Therefore, they are able to better provide input in some cases, or even want to do the changes themselves. Clearly, the approach shown provides increased agility to react to changes in the business context compared to similar logic directly implemented in code.

The example is the implementation of a HCM Feature for screen control in the Organizational Assignment infotype (0001). However the approach is not limited to this particular example. Whenever a task has to be automated, BRFplus may be used to implement a decision service and plug it into the relevant process step - ideally via a BADI, enhancement spot, or any other exit technology.

Remote invocation of BRFplus decision services is also possible by calling a web service or an RFC-enabled function module. Both can be generated from within the BRFplus workbench.

The Example

Features are objects in the SAP HCM system, which by querying different enterprise personnel or data structure fields, determine a specific value by parsing a decision tree. The value is known as a “return code” or “result”. This value is used to determine default values and to control certain system processes. Features serve to enhance system flexibility.

We are looking at feature P0001, which controls the screens flow in the Organizational Assignment infotype (0001). I.e. depending on the current screen, country group, employee group, and other attributes assigned to an employee, the next screen to be processed is being determined.

In HCM standard features are being edited using transaction PE03. Here you can see how feature P0001 is being displayed in that transaction.

Command

Line	Variable key	F	C	Operations
000010			D	TCLAS
000020	A		D	MOLGA
000030	A UN			&P0001=UN,
000040	A 01		D	DYNNR
000050	A 01 2000			&P0001=,
000060	A 01 3000		D	PERSK(1)
000070	A 01 3000 O			&P0001=OE,
000080	A 01 3000 *			&P0001=,
000090	A 07			&P0001=07,
000100	A 09			&P0001=09, " DAENEMARK
000110	A 10			&P0001=10,
000120	A 16			&P0001=16,
000130	A 22			&P0001=22, " JAPAN
000140	A 23			&P0001=23, " SCHWEDEN
000150	A 29			&P0001=29, " ARGENTINA
000160	A 34			&P0001=34, " INDONESIA
000170	A 37			&P0001=37,
000180	A 48			&P0001=48, " PHILIPPINES
000190	A **			&P0001(RPCADD00) ,
000200	B		D	MOLGA
000210	B **			&P0001(RPCADD00) ,

For instance line 70 returns value OE, if Transaction Class (TCLAS) = A, Country Group (MOLGA) = 01, Screen (DYNNR) = 3000, and Employee Subgroup (PERSK) = O.

We want to use BRFplus to model the same logic, but with better readability, usability, and performance.

Steps in the Implementation of a Decision Service

Definition of Task

First clarify the task to be accomplished with the decision service. Perfect candidates are tasks that are free from side effects, this means you have a set of attributes and based on the values of those attributes you want to derive or calculate the values of one or several other attributes. Often tasks are related to one of the following:

- Validations
- Derivations
- Defaulting
- Classification
- Calculation
- Mapping
- Boolean decisions

Typically these are micro decisions. Micro decisions occur very frequently but the impact of a particular decision is not big compared with decisions that affect the complete enterprise such as strategic decisions about starting a new product line. However, micro decisions tend to occur very frequently and to be well-structured. Cumulatively they can have a big impact on the financial result of a company as they directly impact the efficiency of the operations.

In the example of this document the decision service is called for every screen change in the Organizational Assignment infotype (0001). Incorrect or missing screens result in wrong or inconsistent personnel data, which has major impact for personnel administration, payroll, as well may be a violation of legal regulations.

Decision Service Definition

Knowing the task we can start to define the decision service:

- Name "Feature P0001"
- Description "Screen Control in the Organizational Assignment Infotype"

This definition is not yet complete. We also need to define inputs (the context for the decision rules) and outputs (the results) of the service. Usually business people can derive that from the rules they have in mind. It is not a problem if this list is not complete upfront as more parameters can be easily added at a later point in time. Ideally, the equivalent Data Dictionary data elements (shown in brackets in the example below) are already known so that technical attributes as well as business semantics can be derived from the dictionary.

In that example all values are well known from the already existing feature. For feature P0001 we have the following parameters:

- Inputs/Context: Structure PME04
- Outputs/Results: Return Value which is a string data element

Parameters are not always elementary (i.e. single value). This is why BRFplus also supports structure parameters, table parameters, and even deeply nested complex types (e.g. a structure with a component being a table).

The service definition is shown in the next screenshot. In BRFplus a service is implemented with the help of a BRFplus Function. The Signature tab contains the context, which provides the inputs for the service call. It also contains a Result Data Object. In that example the Result Data Object is a string data element. There are other HCM features that return a table of strings.

The following screenshot shows the function in the BRFplus workbench (transaction code BRFplus).

Function: FEATURE_P0001, Feature P0001

Back | Edit | Check | Save | Activate | Transport | Delete | More

General

Detail

Start Simulation | Show Traces | Generate | Create Code Template | Advanced Checks

Properties | **Signature** | Assigned Rulesets | Code Generation

Context

Add Existing Data Object | Add New Data Object | Remove | [Navigation]

Component Name	Text	Type
▼ PME04	Field String for Feature - Screen Control	Structure
▪ BUKRS	CoCode	Element (Text)
▪ WERKS	Pers.area	Element (Text)
▪ BTRTL	Subarea	Element (Text)
▪ PERSG	EE group	Element (Text)
▪ PERSK	EE subgrp	Element (Text)
▪ INFTY	Infotype	Element (Text)
▪ SUBTY	SType	Element (Text)
▪ ITBLD	Screen control	Element (Text)
▪ DYNNR	Screen	Element (Text)
▪ FCODE	F code	Element (Text)
▪ ITYGR	Info group	Element (Text)
▪ MASSN	Action	Element (Text)
▪ MASSG	Act.Reason	Element (Text)
▪ MOLGA	Ctry Grpg	Element (Text)
▪ MOABW	PS grpg	Element (Text)
▪ TCLAS	Tr.Class	Element (Text)
▪ USERG	User group	Element (Text)
▪ PERNR	Pers.No.	Element (Text)
▪ ANSVH	Contract	Element (Text)

Result Data Object

Data Object: [Return Value](#)

Data objects are used in BRFplus to define parameters. The parameter PME04, for example, is defined by a data object shown in the next screenshot. This can be reached by selecting the hyperlink on PME04 in the Signature tab. You can see that properties such as element type (structure), comprised elements, and texts, are taken over from structure PME04 which is defined in the data dictionary. BRFplus is also able to find possible values in domain lists or check tables so that a value help based on those values can be provided when modeling rules.

Note: The implementation of the rule, i.e. how it calculates the return value, will be shown a little later in this article.

Structure: PME04, Field String for Feature - Screen Control

Back | Edit | Check | Save | Activate | Transport | Delete | More

General

Detail

Define Data Binding

Select Bind Type:

DDIC Type Name: Refresh Binding

Structure Definition

	Component Name	Text	Type
▪	BUKRS	CoCode	Element (Text)
▪	WERKS	Pers.area	Element (Text)
▪	BTRTL	Subarea	Element (Text)
▪	PERSG	EE group	Element (Text)
▪	PERSK	EE subgrp	Element (Text)
▪	INFTY	Infotype	Element (Text)
▪	SUBTY	SType	Element (Text)
▪	ITBLD	Screen control	Element (Text)
▪	DYNNR	Screen	Element (Text)
▪	FCODE	F code	Element (Text)
▪	ITYGR	Info group	Element (Text)
▪	MASSN	Action	Element (Text)
▪	MASSG	Act.Reason	Element (Text)
▪	MOLGA	Ctry Grpg	Element (Text)
▪	MOABW	PS grpg	Element (Text)
▪	TCLAS	Tr.Class	Element (Text)
▪	USERG	User group	Element (Text)
▪	PERNR	Pers.No.	Element (Text)
▪	ANSVH	Contract	Element (Text)

Decision Service Trigger Implementation

After having created and activated the data objects and functions, the new decision service is ready to be plugged into the right place in the process. Often you can find BADIs, enhancement spots or other exit options for exactly that purpose.

For feature P0001 the code snippet for calling the respective function module looks like this:

```
CALL FUNCTION 'HR_FEATURE_BACKFIELD'
  EXPORTING
    feature                = 'P0001'
    struc_content          = ls_pme04
  IMPORTING
    back                   = ev_back
  EXCEPTIONS
    dummy                  = 1
    error_operation        = 2
    no_backvalue           = 3
    feature_not_generated  = 4
    invalid_sign_in_funid  = 5
    field_in_report_tab_in_pe03 = 6
  OTHERS                   = 7.
```

You may replace that by implementing the call of BRFplus function “Feature P0001”:

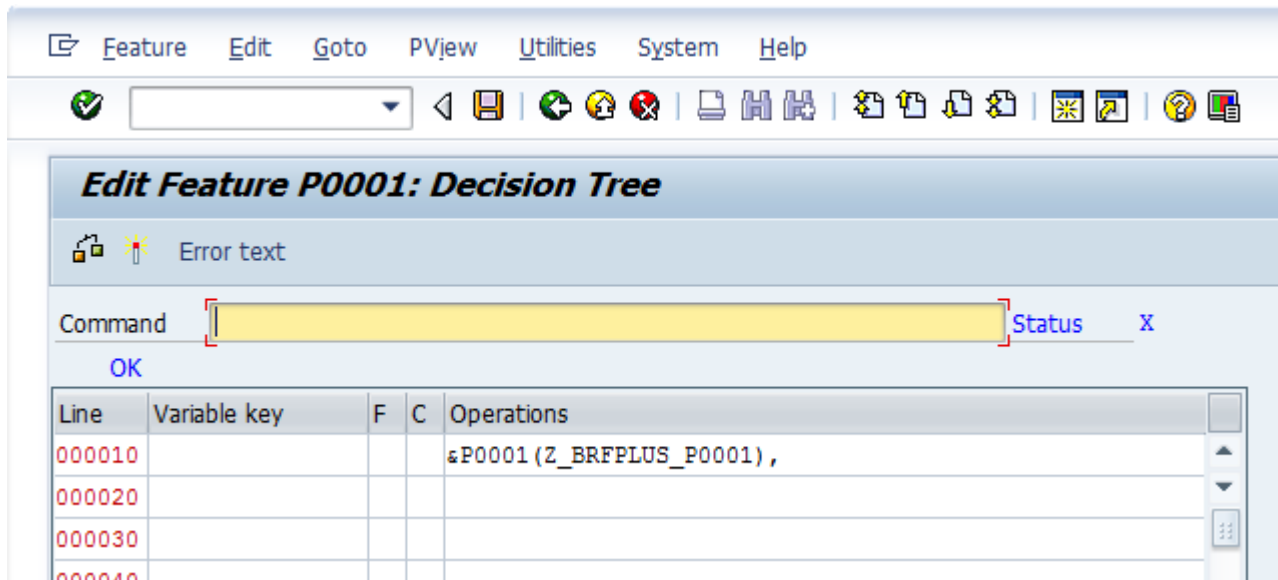
```
GET TIME STAMP FIELD lv_timestamp.
ls_name_value-name = 'PME04'.
GET REFERENCE OF ls_pme04 INTO lr_data.
cl_fdt_function_process=>move_data_to_data_object(
  EXPORTING ir_data          = lr_data
            iv_function_id   = '3AC51E4F98025D01E10000000A42888D'
            iv_data_object   = '24C61E4F98025D01E10000000A42888D'
            iv_timestamp     = lv_timestamp
            iv_trace_generation = abap_false
            iv_has_ddic_binding = abap_true
  IMPORTING er_data          = ls_name_value-value ).
INSERT ls_name_value INTO TABLE lt_name_value.
cl_fdt_function_process=>process(
  EXPORTING iv_function_id = '3AC51E4F98025D01E10000000A42888D'
            iv_timestamp   = lv_timestamp
  IMPORTING ea_result      = ev_back
  CHANGING ct_name_value  = lt_name_value ).
```

The BRFplus workbench provides a code template to be copy/pasted to the backend.

In the code first the parameters are passed. Then the BRFplus function is being processed, meaning the decision service is called, and the result is copied back to the ev_back field. Note that the ID can be found in the BRFplus workbench (and is included in the generated code template).

Maybe you want to implement some logic that checks if a BRFplus function for feature P0001 is available. In that case it is being processed. Otherwise the traditional HCM feature is being executed.

Alternatively to implementing the BRFplus call in the backend, you can configure a traditional HCM feature in a way that for any parameter values a report is invoked that calls the respective BRFplus rules. Here you can see an example for such a minimal feature configuration, where Z_BRFPLUS_P0001 is the name of a report calling the BRFplus function as described above.

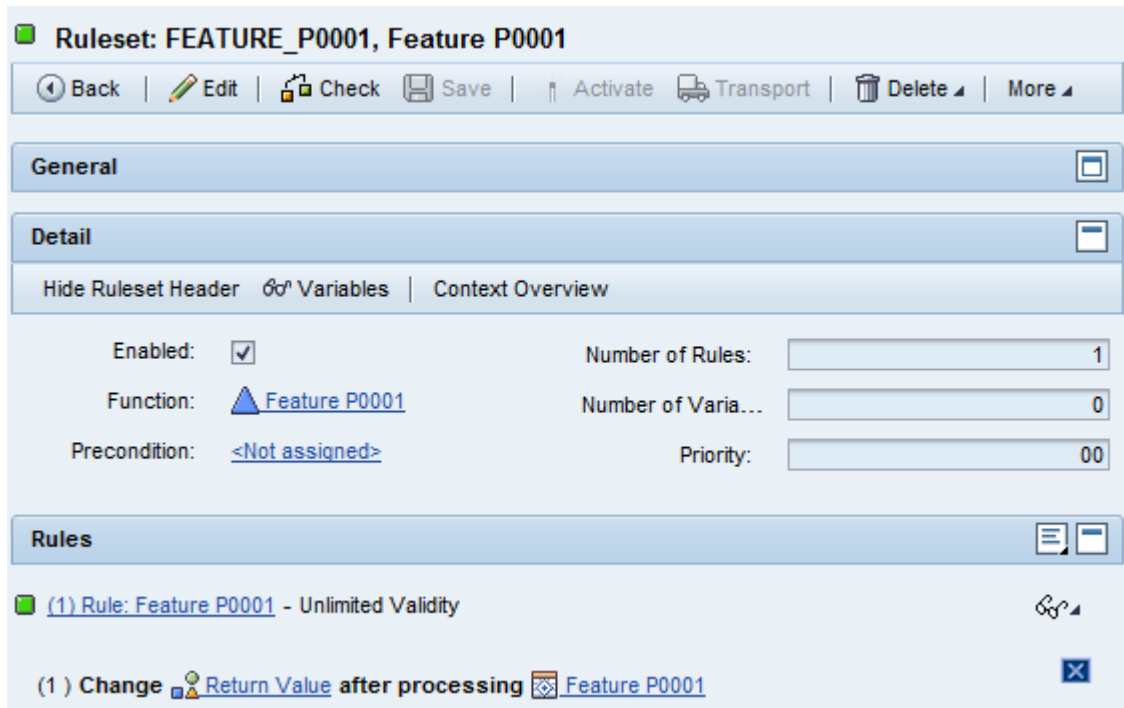


Decision Service Implementation

The service is defined and plugged into the right place in the process, but no business logic has yet been defined.

Business logic can be implemented in Rulesets in BRFplus. One or more rulesets can be created from the function screen. Rulesets can have pre-conditions and priorities to define the order of execution. In the feature P0001 example there is one Ruleset, where customers may add more Ruleset to extend or overrule the decision logic.

The Ruleset itself is very simple. It only uses one expression of type Decision Table, and it provides its result as the Return Value.



Rulesets can vary in complexity. To manage this complexity, rules can draw on different expressions based on a wide range of expression types like decision trees, mathematical calculations, data base lookups, analyzing tabular data, loop expressions, calling ABAP methods, triggering workflows, and many more. Actually all HCM features only require very simple Rulesets.

The following screenshot shows the decision table that derives the Return Value from the fields of structure PME04. Checking on the hyperlink on the expression takes the user to the decision table. On the left side there are several condition columns specifying matching conditions. These columns not only may contain values, but also can bear ranges, patterns, or even more complex logic. For instance the third row combines several values with an "OR" operator. The result column at the right provides the value that is returned if a row is matched. It can contain direct values as well as context parameters (to be seen in the third row), or call other expressions (to be seen in the fourth row where an ABAP method is being called, and its return value is being provided as a result).

Decision tables are very powerful and therefore used excessively in some use cases. Business people also love decision tables because they can load them into Microsoft Excel, change them, and upload them into BRFplus again. Decision tables with several hundreds or even thousands of rows are sometimes required by the domain experts. Decision tables can either return a single match or all matching lines as a table. So also features returning tables can be modeled with decision tables.

Decision Table: FEATURE_P0001, Feature P0001

Back | Edit | Check | Save | Activate | Transport | Delete | More

General

Detail

Export To Excel | Context Overview | Start Simulation

Table Contents

Find: Next Previous Table Settings

EE subgrp	Screen	Ctry Grpg	Tr.Class	Return Value
0	3000	01 (Germany)	A (Master Data a...	OE
	2000 ; 3000	01 (Germany)	A (Master Data a...	''
		07 (Canada) ; 09 (Denmark) ; 10 (...)	A (Master Data a...	Ctry Grpg
			A (Master Data a...	RPCADD00

When comparing that representation with the display of the same logic in transaction PE03 (cf. screenshot on page 4), you will notice a significant improvement of usability and understandability. Also you will learn that it is significantly faster to set up the example with BRFplus.

Finally all objects need to be activated. Then the decision service is ready to be used.

Decision Service Testing

A decision service can be tested directly in the BRFplus workbench on the Function user interface with the button Start Simulation.

The screenshot shows the 'Simulation' window titled 'Business Rule Framework plus - Simulation'. It includes a 'Back to Simulation' button, a 'Context Values' section, a 'Result' section showing 'Return Value: OE', and a 'Processing Steps' section. The 'Processing Steps' section contains a table with columns for Step, Value, Type, and Status, showing a detailed execution hierarchy for Feature P0001.

Step	Value	Type	Status
Trace for Feature P0001 started on 06.02.2012 13:51:33 by user D027629		Trace	
Feature P0001		Function	STARTED
Event Processing			
Context			
Field String for Feature - Screen Control		Data Object	
Feature P0001		Ruleset	STARTED
Rule processing: Feature P0001 (position 000001)			
Feature P0001		Rule	STARTED
Feature P0001		Decision Table	STARTED
Evaluating Decision Table Row 1			
Cell Condition of Column 7 in Row 1 is met			
Evaluating Cell Condition of Column 8 in Row 1			
If test parameter Tr.Class (TCLAS); value A			
Include Condition 1 is equal to A			
Cell Condition of Column 8 in Row 1 is met			
All Conditions in Row 1 is met. Evaluating Result of Row 1			
Determined Decision Table Result Value: OE			
Feature P0001	OE	Decision Table	FINISHED
Value Change			
Return Value	OE	Data Object	UPDATED
Feature P0001		Rule	FINISHED
Feature P0001		Ruleset	FINISHED
Feature P0001		Function	FINISHED

Together with the result you can display all processing steps and intermediate results in a well-organized execution hierarchy tree, which also can be downloaded to Microsoft Excel.

Conclusion

So the Screen Control in the Organizational Assignment Infotype task is now solved with help of a decision service implemented in BRFplus. With an effort of just a few hours BRFplus can be used to implement a decision service to cater for automation of many tasks, freeing up time for more value adding work. The business rules that implement the decision service can be understood and even be changed by the domain expert so that IT professional does not need to understand the business details.

For HCM features it is possible to provide a migration tool that translates features from PE03 representation to BRFplus objects. As BRFplus offers a public API, these objects can be loaded to BRFplus without the need of manual effort.

BRFplus objects can be stored as customizing objects, system objects, or master data objects. In addition the SAP transportation management and SAP authorization management also apply for BRFplus objects. So all concepts of shipping default content for HCM features, allowing customers to extend or overwrite that default content, and providing role specific access authorizations can be kept with BRFplus.

Related Content

- BRFplus in SAP Developer Network

<http://www.sdn.sap.com/irj/sdn/nw-rules-management?rid=/webcontent/uuid/90754865-f283-2b10-6d9f-b10f3c28c3a0>

Copyright

© 2012 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, System i, System i5, System p, System p5, System x, System z, System z9, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, Informix, i5/OS, POWER, POWER5, POWER5+, OpenPower and PowerPC are trademarks or registered trademarks of IBM Corporation.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

These materials are provided "as is" without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

Any software coding and/or code lines/strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.