

Indirect Evaluation Based on Different Parameters



Copyright

- No part of this brochure 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®, NT®, EXCEL®, Word® and SQL Server® are registered trademarks of Microsoft Corporation.
- IBM®, DB2®, OS/2®, DB2/6000®, Parallel Sysplex®, MVS/ESA®, RS/6000®, AIX®, S/390®, AS/400®, OS/390®, and OS/400® are registered trademarks of IBM Corporation.
- ORACLE® is a registered trademark of ORACLE Corporation, California, USA.
- INFORMIX®-OnLine for SAP is a registered trademark of Informix Software Incorporated.
- UNIX®, X/Open®, OSF/1®, and Motif® are registered trademarks of The Open Group.
- HTML, DHTML, XML, XHTML are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Laboratory for Computer Science NE43-358, Massachusetts Institute of Technology, 545 Technology Square, Cambridge, MA 02139.
- SAP, SAP Logo, mySAP.com, mySAP.com Marketplace, mySAP.com Workplace, mySAP.com Business Scenarios, mySAP.com Application Hosting, WebFlow, R/2, R/3, RIVA, ABAP, SAP Business Workflow, SAP EarlyWatch, SAP ArchiveLink, BAPI, SAPPHIRE, Management Cockpit, SEM, are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other products mentioned are trademarks or registered trademarks of their respective companies.

Contents

1. Objective and Scope	2
1.1 Introduction.....	2
1.2 Implementation in Detail with an example.....	3
1.2.1 Customization.....	3
1.2.2 Sample Program	6
1.2.3 Test Scenarios.....	10

1. Objective and Scope

1.1 Introduction

Indirect valuation module is used to determine value of a wage type based on different parameters. These parameters, as part of 'INVAL' module for India, comprise pay scale parameters maintained in 'Basic Pay' Infotype of employee and employee subgroup grouping for collective agreement provision (ESG for CAP). Pay scale parameters include pay scale area, pay scale type, pay scale group and pay scale level. These parameters also define 'Allowance Grouping' ('ALGRP') in India payroll localization.

Above mentioned information is usually interpreted as: Whenever there is a need to have different valuation of a particular wage type, a new Allowance Grouping is typically required. This is not true. If business process to determine wage type value does not involve pay scale parameters, it is not required to mandatorily define new Allowance Grouping just to get different valuation of a wage type. Indirect valuation of wage type could be independent of Allowance Grouping.

This can be achieved in two ways:

(1) Standard delivery of indirect valuation module 'TARIF' has many different variants. These variants provide valuation of wage type based on subset of 4 pay scale parameters mentioned above. Means a wage type can be evaluated just based on Pay scale area and pay scale type while keeping other two pay scale parameters a blank (module TARIF variant 'D'). For further information, refer to documentation at:

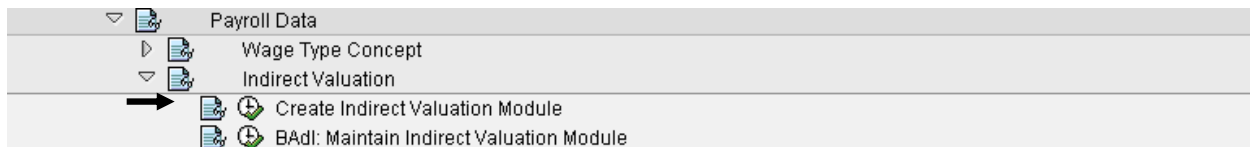
IMG: Payroll: India – Basic Settings – Environment of Wage Type Maintenance – Check Wage type Characteristics – Country grouping = 40 – Double click any wage type – Refer to help documentation (F1) of field 'Indirect eval. module' – Select link 'TARIF'

(2) Second way is to value a wage type on parameters completely different from pay scale parameters. This document explains the process of implementing a new module for indirect evaluation which will maximum take 2 days of effort including the unit testing. A sample implementation is provided below with detailed explanation.

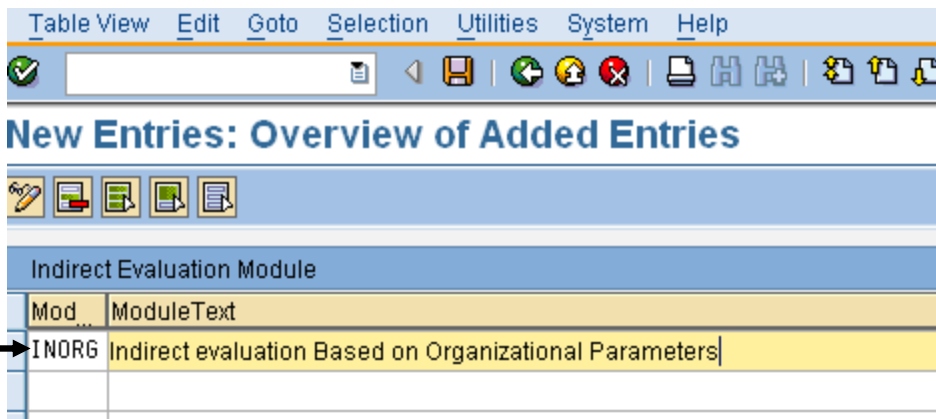
1.2 Implementation in Detail with an example

1.2.1 Customizing

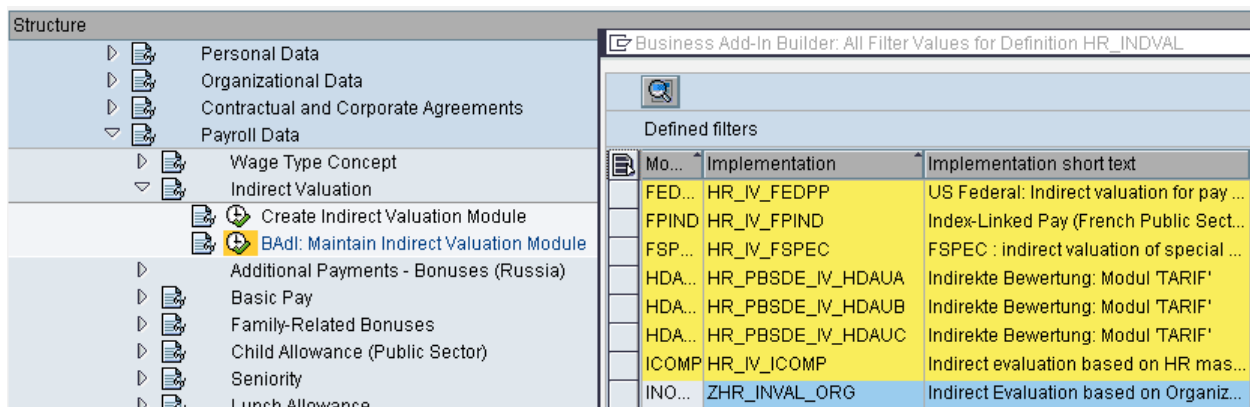
Step 1: In transaction SPRO, go to the nodes under Personnel Management -> Personnel Administration -> Payroll Data -> Indirect Evaluation-> Create Indirect Valuation Module



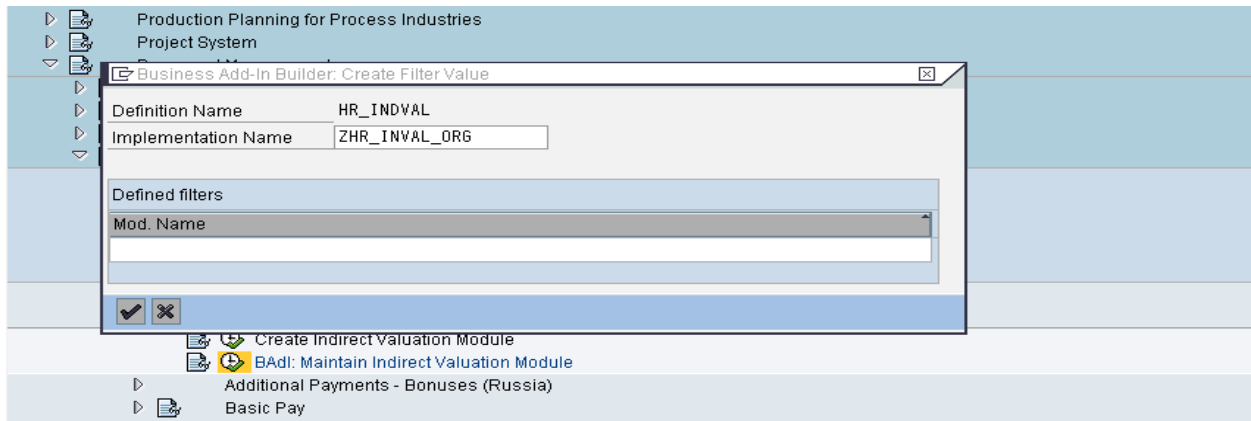
Step 2: Choose 'New Entries'. Enter the name of the Module which is to be implemented and a suitable short description. Save the entry.



Step 3: Go back to IMG screen, and select the next node 'BAdI: Maintain Indirect Valuation Module'. To create the new implementation click on create.

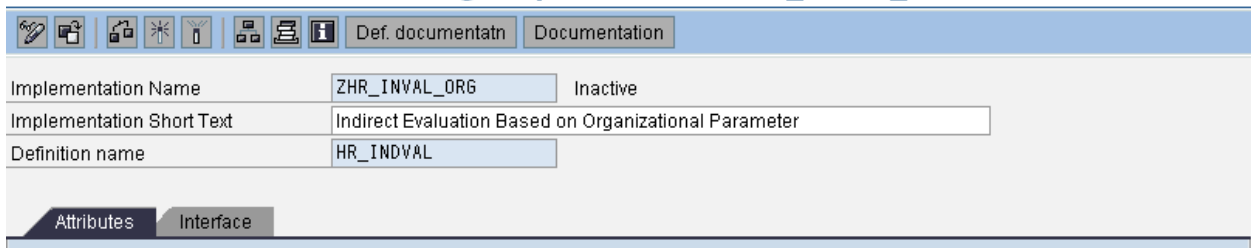


Step 4: Enter the Implementation name and choose enter.

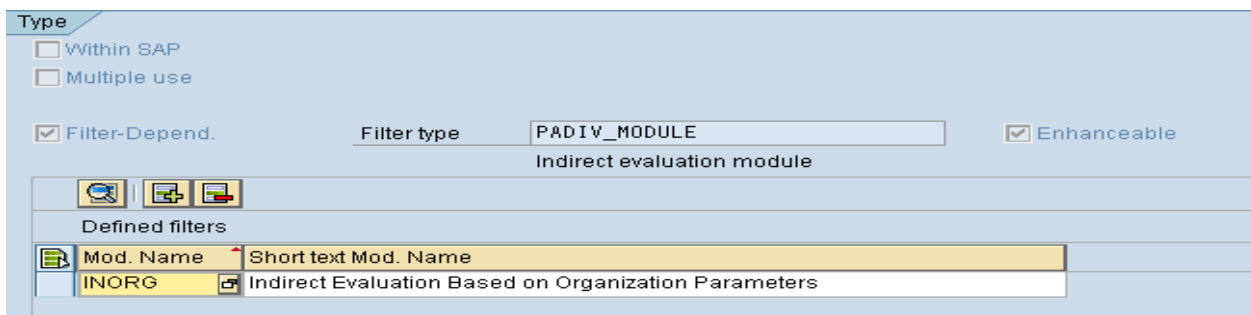


Step 5: Enter a short description about the implementation in the 'Implementation Short Text' field.

Business Add-In Builder: Change Implementation ZHR_INVALID_ORG



Step 6: Create a filter by selecting the option with '+' icon. Enter the module name same as the one entered in step 2.



Step 7: Switch to the 'Interface' tab. The method to be implemented is DO_INDIRECT_VALUATION.

Implementation Name: ZHR_INVAL_ORG Inactive
 Implementation Short Text: Indirect Evaluation Based on Organizational Parameter
 Definition name: HR_INDVAL

Attributes Interface

Interface name: IF_EX_HR_INDVAL
 Name of implementing class: ZCL_IM_HR_INVAL_ORG

Method	Implement..	Description
DO_INDIRECT_VALUATION	ABAP Code	Execution of Indirect Valuation

Step 8: Double Click on the method, to go to the ABAP editor to implement the Method. Click on the signature button on the top, to view the parameters which are available and returned to/from the method.

Class Builder: Class ZCL_IM_HR_INVAL_ORG Change

Class Builder: Class ZCL_IM_HR_INVAL_ORG Change

Method IF_EX_HR_INDVAL-DO_INDIRECT_VALUATION Inactive

Ty.	Parameter	Type spec.	Description
▶	VALUE(FLT_VAL)	TYPE PADIV_MODULE	Parameter FLT_VAL of GET_AMOUNT Method
▶	VALUE(MODULE_SPEC)	TYPE PADIV_MODULE_SPEC	Module Variant
▶	VALUE(MOLGA)	TYPE T001P-MOLGA	Country Grouping
▶	VALUE(TCLAS)	TYPE TCLAS DEFAULT 'A'	Transaction Class (HR Master Data/Applicant Data)
▶	VALUE(PERNR)	TYPE PREL-PERNR	Personnel Number
▶	VALUE(INFTY)	TYPE PREL-INFTY	Infotype
▶	VALUE(LGART)	TYPE T511-LGART	Wage Type
▶	VALUE(BEGDA)	TYPE PREL-BEGDA	Start Date
▶	VALUE(PNNNN)	TYPE PRELP OPTIONAL	Obsolete (This parameter cannot be used)
▶	VALUE(LGART_TAB)	TYPE PADIV_LGART_TAB OPTIONAL	Wage Type Table
▶	VALUE(VALUATION_INPUT)	TYPE PADIV_VALUATION_STRCT OPTIONAL	Entry Values
▶	VALUE(VALUEES_TAB)	TYPE PADIV_VALUEES_TAB OPTIONAL	Table for Value Transfer in Indirect Valuation

```

method IF_EX_HR_INDVAL-DO_INDIRECT_VALUATION
endmethod.
    
```

1.2.2 Sample Implementation

This sample program implements two module variants for the Module INORG.

- 1) Variant 1 : Evaluates the wagetype as 800 INR if the personnel Area is 0001 and 400 INR if the personnel Area as IN02
- 2) Variant 2: Evaluates the wagetype based on the age of the employee. The amount is defaulted by the age multiplied with 1000.
Amount = Age * 1000.

Flow of the program is as follows:

Add the following code between 'method' and 'endmethod' of the BAdi implementation created in step 8.

```

DATA: VALUATION TYPE PADIV_VALUATION_STRCT.
DATA: TMPMSGID LIKE SY-MSGID.
...
describe table IF_EX_HR_INDVAL~WGTP_STACK.

VALUATION = VALUATION_INPUT.

DATA: p0001 TYPE TABLE OF P0001,
      wa_pa0001 LIKE LINE OF p0001.
DATA: sy_subrc TYPE sy-subrc.

DATA: p0002 TYPE TABLE OF P0002,
      wa_pa0002 LIKE LINE OF p0002.

* Read infotype 0001 for the employee
CALL FUNCTION 'HR_READ_INFOTYPE'
  EXPORTING
    TCLAS           = 'A'
    PERNR           = pernr
    INFYTY          = '0001'
    BEGDA           = begda "'18000101'"
    ENDDA           = endda "'99991231'"
  * BYPASS_BUFFER  = ''
  * LEGACY_MODE     = ''
  IMPORTING
    SUBRC           = sy_subrc
  TABLES
    INFYTY_TAB      = p0001[]
  * EXCEPTIONS
  * INFYTY_NOT_FOUND = 1
  * OTHERS          = 2
.
IF SY-SUBRC <> 0.
* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
* WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
ENDIF.

* Read infotype 0002 for the employee
CALL FUNCTION 'HR_READ_INFOTYPE'

```



```

EXPORTING
*   TCLAS                = 'A'
      PERNR              = pernr
      INFYTY             = '0002'
      BEGDA              = begda "'18000101'"
      ENDDA              = endda "'99991231'"
*   BYPASS_BUFFER       = ' '
*   LEGACY_MODE          = ' '
IMPORTING
      SUBRC              = sy_subrc
TABLES
      INFYTY_TAB         = p0002[]
*   EXCEPTIONS
*   INFYTY_NOT_FOUND    = 1
*   OTHERS              = 2

      IF SY-SUBRC <> 0.
*   MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
*           WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
      ENDIF.

```

**Call the perform routine where the amount is calculated which implemented in another program called 'ZTKM_SAPLRPIB'.*

Perform (FLT_VAL) **in program** ZTKM_SAPLRPIB **TABLES** P0001 p0002 LGART_TAB **USING** BEGDA ENDDA MODULE_SPEC CHANGING valuation.

** output the evaluated values for amount rate and num. These values are returned from the subroutine.*

valuation_output = valuation.

3) The routine implemented evaluates the wagetypes as follows based on variants:

- If variant 'A' then
 - o Read the record from infotype 0001 with 'BEGDA as the key from all records of that employee.
 - o Read Personnel area, from infotype 0001.
 - if personnel area is IN01 then set the amount value of evaluation structure to 400 and currency as INR.
 - If the personnel area is IN02 then set the amount field of evaluation structure to 800 INR.
- If variant 'B' then
 - o Read the record from Infotype 0002 with BEGDA as the key from all records of that employee.
 - o Compute the age of the employee based on date of birth and multiply it with thousand. Set the value of amount field of evaluation structure to the evaluated value.

Create a report program with the name 'ZTKM_SAPLRPIB' and implement the following program given below in that report program.

```

*&-----*
*&      Form  INORG
*&-----*
*      text
*-----*
*      -->P0001      text
*      -->WAG_TAB    text
*      -->VARNT      text
*      -->EVALUATION text
*-----*
FORM INORG TABLES p0001 p0002 wag_tab USING Bgda TYPE BEGDA EDDA TYPE ENDDA v
arnt TYPE C CHANGING evaluation TYPE PADIV_VALUATION_STRCT.

DATA: wa_p0001 TYPE p0001,
      wa_p0002 TYPE p0002.
DATA: b_day TYPE sy-datum,
      lv_age TYPE i,
      im_ref_date TYPE sy-datum.
DATA: it_p0001 TYPE p0001 OCCURS 0,
      it_p0002 TYPE p0002 OCCURS 0.

it_p0001[] = p0001[].
it_p0002[] = p0002[].
im_ref_date = sy-datum.

* variant A
IF varnt = 'A'.

    SORT it_p0001 DESCENDING.
* Read Record of Infotype 0001 with key date.
    LOOP AT it_p0001 INTO wa_p0001 WHERE BEGDA <= BGDA AND ENDDA >= BGDA.
        EXIT.
    ENDLOOP.
* according to personnel area / sub area return the amount
    IF wa_p0001-WERKS = 'IN01' AND wa_p0001-BTRTL = '0001'.

        evaluation-amount = 40000.
* evaluation-unit = 0.
        evaluation-number = 0.
        evaluation-currency = 'INR'.
    ELSE.

        evaluation-amount = 80000.
* evaluation-unit = 0.
        evaluation-number = 0.
        evaluation-currency = 'INR'.
    ENDIF.
* variant B
ELSEIF varnt = 'B'.
    SORT it_p0002 DESCENDING.
* Read Record of Infotype 0002 with key date.
    LOOP AT it_p0002 INTO wa_p0002 WHERE BEGDA <= BGDA AND ENDDA >= BGDA.
        EXIT.
    ENDLOOP.

```

```
* according to B'date of employee
  b_day = wa_p0002-gbdat.

* calculate the age.
  lv_age = im_ref_date(4) - b_day(4).

*b_day not passed yet, deduct one year
  IF im_ref_date+4(2) <= b_day+4(2).
    lv_age = lv_age - 1.
  ENDIF.

*b_day not yet, but in this month.
  IF im_ref_date+4(2) = b_day+4(2).
    IF im_ref_date+6(2) > b_day+6(2).
      lv_age = lv_age + 1.
    ENDIF.
  ENDIF.

* Multiply the age with the amount to be returned
  evaluation-amount = 100000 * lv_age .

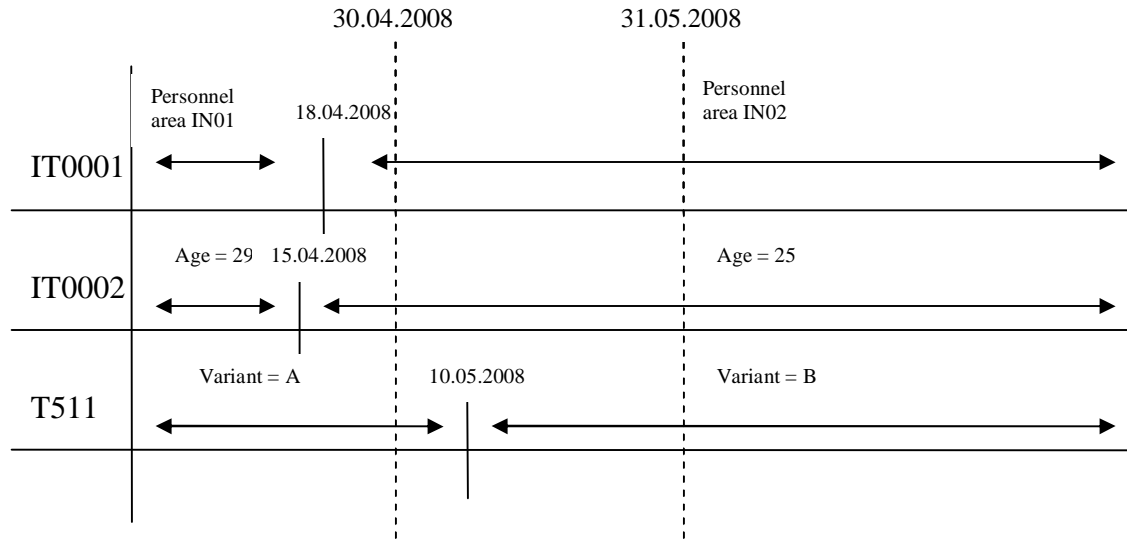
* evaluation-unit = 0.
  evaluation-number = 0.
  evaluation-currency = 'INR'.

ENDIF.

ENDFORM.                " INORG
```

1.2.3 Test Scenario

SCENARIO 1:



For Period 01/2008:

Splits due to changes in IT0001 – (Change in Personnel Area)

Split	Period		Personnel Area	Module Name	Module Variant	Amount
	From	To				
1	01.04.2008	17-04-2008	IN01	INORG	A	400
2	18.04.2008	31.12.9999	IN02	INORG	A	800

Final Values after factoring:

First Split – $MB10 = 400 * 17/30 = 226.67$

Second Split - $MB10 = 800 * 13/30 = 346.67$

For Period 02/2008:

Splits due to change in T511 – (Change in Variant)

Split	Period		Personnel Area	Module name	Module Variant	Amount
	From	To				
1	01.05.2008	09.05.2008	IN02	INORG	A	800
2	10.05.2008	31.05.2008	IN02	INORG	B	25000

First split – based on Personnel area (IN02) - Variant A

$MB10 = 800 * 9/31 = 232.26$

Second Split – Based on age (P0002) – Variant B

$MB10 = 25000 * 22 / 31 = 17741.94$