Rules in Workflow

Applies to:

SAP ECC 6.0 and upwards.

Summary

This document is prepared to give the beginners a brief idea about Rules in SAP Business Workflow. The document contains basics of Rules and also shows how and where rules can be applied in Workflow with necessary screenshots.

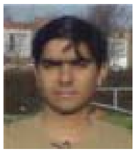
Authors: Reshmi Das, Sunil Ojha and Anindita Kar.

Company: Capgemini Consulting India Private Ltd.

Created on: 3 January, 2008.

Author Bio

Reshmi Das is an ABAP/Workflow consultant. She is currently working with Capgemini Consulting India Pvt. Ltd.



Sunil Ojha is a XI certified workflow consultant. His current engagement is with Capgemini Consulting India Pvt. Ltd.

Anindita Kar is an ABAP/Workflow consultant. She is currently working with Capgemini Consulting India Pvt. Ltd.

Table of Contents

Definition:

- Workflow rules allow us to define and enforce an issue handling process.
- Rules allow us to specify data at runtime according to a template defined at definition time.
- Together with e-mail notifications, workflow rules help us automate the tracking and management of Issues.

What is the difference between Rules and Roles?

Rule	Role
1. A Rule is a general way to 'define' your agents. 2. A rule is maintained with Transaction code **PFAC** and is used to programmatically define the agents. For example, if you want a work item to go to all managers who worked for the company for 5 years or more, then you would have to create a rule that handles this custom scenario. Typically this is done by creating a custom function module that performs your selection logic and passes the results back to the rule. 3. You can define a container element that contains the object of Organizational Management at runtime or a rule for dynamic agent determination.	1. A Role is a general way to 'assign' your agents. 2. A role is maintained with Transaction code **PFCG** and is usually assigned to a User (someone with a valid login). For example, you can have the role of ABAP_DEVELOPER and assign this role to all users who are developers. This is a good method to use if you want ALL developers to receive a workflow work item. 3. You can specify not only a system user, but also a role or an object of Organizational Management such as a position, a Job, an organizational unit or a work center.

Rule Basics

Transaction: PFAC_INS (Create).

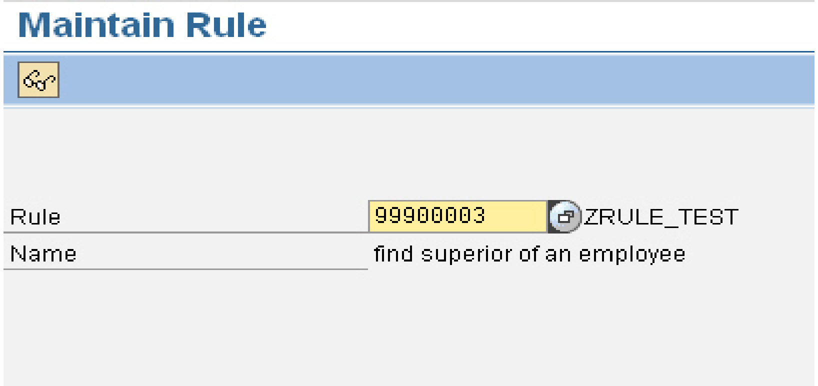
PFAC_CHG (Change).

PFAC_DIS (Display).

PFAC_DEL (Delete).

Menu Path:

Tools **–>** Business Workflow **–>** Development **–>** Definition tools **–>**Rules for Agent Assignment **–>**Create/Change/Display.



Maintain Rule

Rule	99900003	ZRULE_TEST
Name	find superior of an employee	

Note:

- **Each rule gets a 8 digit generated by the system prefixed with AC**
- **Rules can be tested using PFAC or by function module 'RH_GET_ACTORS'.**

Rule Container

- The rule container contains a rule's parameters.
- Each rule has just one rule container.
- At runtime, the rule parameters contain the current, context-specific information that forms the basis of rule resolution. Therefore, the rule parameters constitute "input" for rule resolution.
- In SAP Business Workflow, binding feeds the rule parameters with values from the workflow container. If a task contains default rules, the system fills the rule container from the task container using binding.
- If a rule is used in a workflow to determine the responsible agent, the result of rule resolution is stored in the container element _RULE_RESULT and can be transferred to the workflow container by binding.
- The Rule container includes the following information depending on the Rule Category.
  The online help is here

Standard Rules.

- A rule container is defined for each rule.
- The rule parameters determine which information is required for the rule resolution to be executed at runtime.
- If an error occurs during rule resolution, setting the termination indicator for the rule resolution determines how the workflow will continue:

  Flag is selected - the workflow assumes an Error status and a notification is sent to the workflow administrator.

  Indicator is not activated - the workflow forwards this work item to ALL possible agents.

Rule Resolution

Use

The following types of Rules can be resolved:

Agent determination rules

Agent determination rule is used in SAP Business Workflow for example, determine the responsible agents of a work item who have certain characteristics.

WebFlow Rules

WebFlow rules are used within a workflow in the basic data and in the Web activity step type.

Using a WebFlow rule you can individually execute the following parts of a Web activity:

- Determination of URL
- Preparation of the XML document
- Determination of the authentication data
- Execution of the send procedure
- Determination of transfer format

For more help click here

Defining Rules Using Function to Be Executed

Use

You use function modules for defining agent determination rules when very complex selection criteria, which cannot be depicted by other rule categories, are required for agent determination.

Note:

- **You have decided what you want the workflow to do. Depending on that, you decide whether you want to use a function module delivered by SAP or a function module of your own that suits your requirements exactly.**
- **The function module must enable the following process:**
  a) **The rule container transferred as the table parameter AC_CONTAINER is read using the macro commands SWC_GET_ELEMENT and SWC_GET_TABLE. Note: If you want to have the macro commands available, you must integrate the Include <CNTN01> as a sub-report for shared use. <CNTN01> mainly contains the macro command definitions for creating and processing a container instance.**
  b) **The rule parameters are used to determine the responsible agent(s). At its simplest, this sub-program consists of a loop on a (Customizing) table, from which the agent is selected.**
  c) **The table ACTOR_TAB is filled.**

Procedure:

How to create custom rules in SAP Workflow?

Concepts:

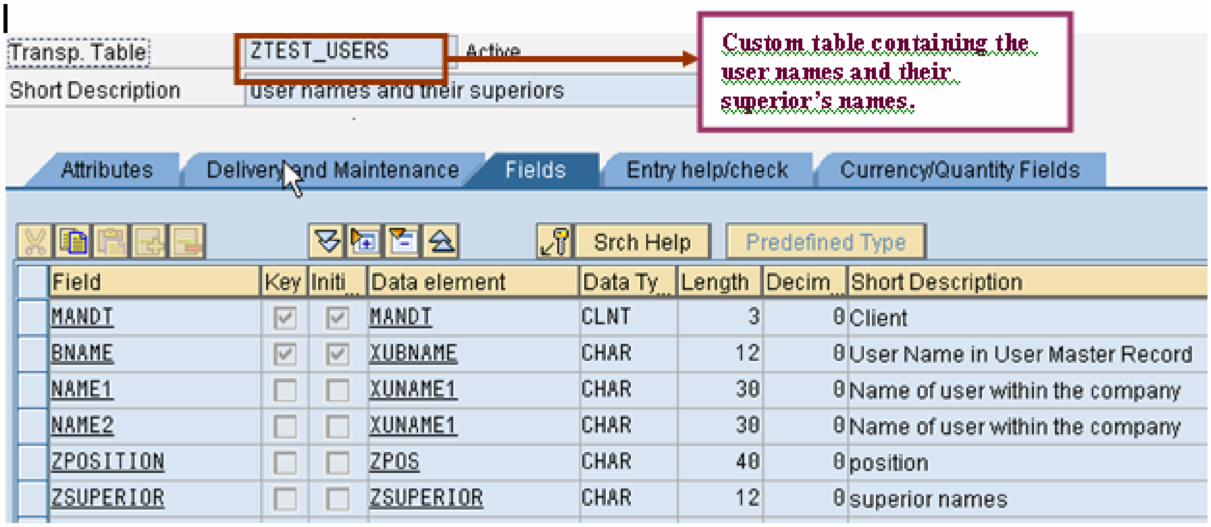
The rules can be created or changed using the standard SAP transaction **PFAC**. Once the rules are created you can call these rules in any workflow via the rule container.

Scenario:

Create a rule which will find the superior for any user/agent.

Design:

1. Create a custom table **ZTEST_USERS** which will contain the names, positions of each user along with their superiors' name.



Transp. Table	ZTEST_USERS Active
Short Description	user names and their superiors

Custom table containing the user names and their superior's names.

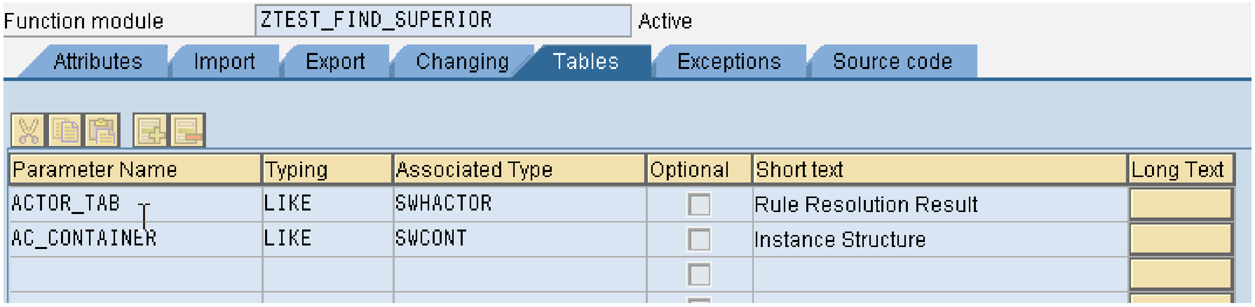
Field	Key	Initi...	Data element	Data Ty...	Length	Decim...	Short Description
MANDT	☑	☑	MANDT	CLNT	3	0	Client
BNAME	☑	☑	XUBNAME	CHAR	12	0	User Name in User Master Record
NAME1	□	□	XUNAME1	CHAR	30	0	Name of user within the company
NAME2	□	□	XUNAME1	CHAR	30	0	Name of user within the company
ZPOSITION	□	□	ZPOS	CHAR	40	0	position
ZSUPERIOR	□	□	ZSUPERIOR	CHAR	12	0	superior names

2. Create a custom function module **ZTEST_FIND_SUPERIOR** which will return back the table containing all the superior information for a particular user/agent.

Note: This custom function module created should have the same interface as that of the SAP standard function module RH_GET_ACTORS.

The following table's parameters have to exist in the custom function module.

ACTOR_TAB STRUCTURE SWHACTOR
AC_CONTAINER STRUCTURE SWCONT



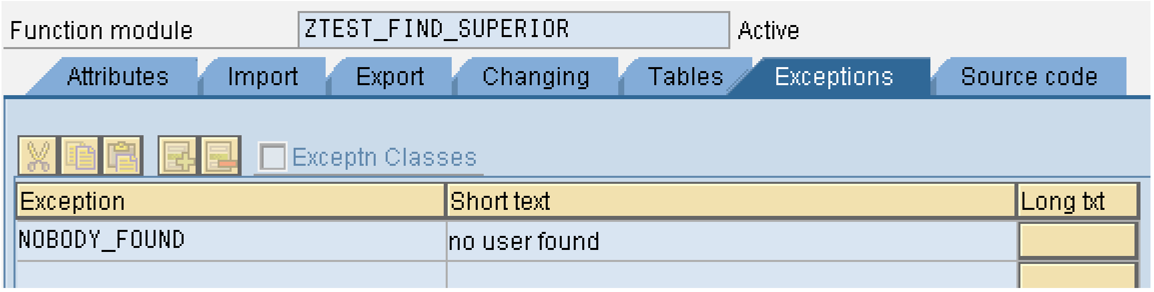
Function module	ZTEST_FIND_SUPERIOR Active

Parameter Name	Typing	Associated Type	Optional	Short text	Long Text
ACTOR_TAB	LIKE	SWHACTOR	□	Rule Resolution Result	
AC_CONTAINER	LIKE	SWCONT	□	Instance Structure	
			□		

Source Code

```
FUNCTION ZTEST_FIND_SUPERIOR.

*"----------------------------------------------------------------------

*"*"Local Interface:

*"  TABLES

*"     ACTOR_TAB STRUCTURE  SWHACTOR
*"       AC_CONTAINER STRUCTURE  SWCONT
*"  EXCEPTIONS
*"       NOBODY_FOUND
*"----------------------------------------------------------------------
  INCLUDE <cntn01>.
  TYPES : BEGIN OF ty_users.
          INCLUDE STRUCTURE ztest_users.
  TYPES : END OF ty_users.

  DATA: org_agent LIKE wfsyst-agent,
              lt_holders TYPE STANDARD TABLE OF swhactor,
              lwa_holders TYPE swhactor,
              lt_users TYPE STANDARD TABLE OF ty_users,
              lwa_users TYPE ty_users,
              v_len TYPE i,
              v_bname LIKE ztest_users-bname,
              num_lines TYPE i.
*Read values assigned to the rule criteria
  swc_get_element ac_container 'org_agent' org_agent.
*Get the superior
  SELECT SINGLE * FROM ztest_users INTO CORRESPONDING FIELDS OF lwa_users
  WHERE bname = org_agent.

 IF NOT lwa_users IS INITIAL.
    REFRESH lt_holders[].
    lwa_holders-otype = 'US'.
    lwa_holders-objid = lwa_users-zsuperior.
    APPEND lwa_holders TO lt_holders.
    APPEND LINES OF lt_holders TO actor_tab.
  ENDIF.
  DESCRIBE TABLE actor_tab LINES num_lines.
  IF num_lines IS INITIAL.
    RAISE nobody_found.
  ENDIF.
ENDFUNCTION.
```

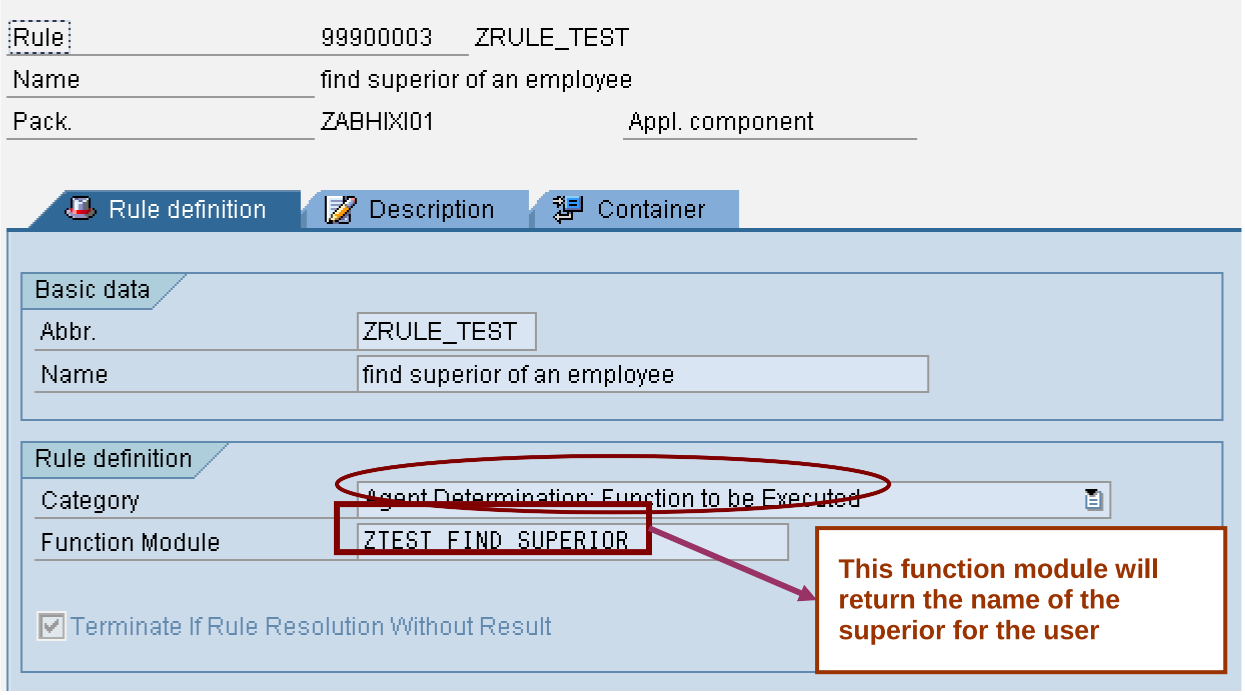
Exception- Goto the exception tab of the Custom Function Module and enter the exceptions as required. Here we have added the Exception NOBODY_FOUND.



Function module ZTEST_FIND_SUPERIOR Active

Exception	Short text	Long txt
NOBODY_FOUND	no user found	

3. Now we have created the Function module **ZTEST_FIND_SUPERIOR**. We will now assign this Function module on the Rule.
4. Go to transaction **PFAC** to create rule. Create rule **ZRULE_TEST**.



Rule	99900003 ZRULE_TEST
Name	find superior of an employee
Pack.	ZABHIXI01
Appl. component	

Basic data

Abbr.	ZRULE_TEST
Name	find superior of an employee

Rule definition

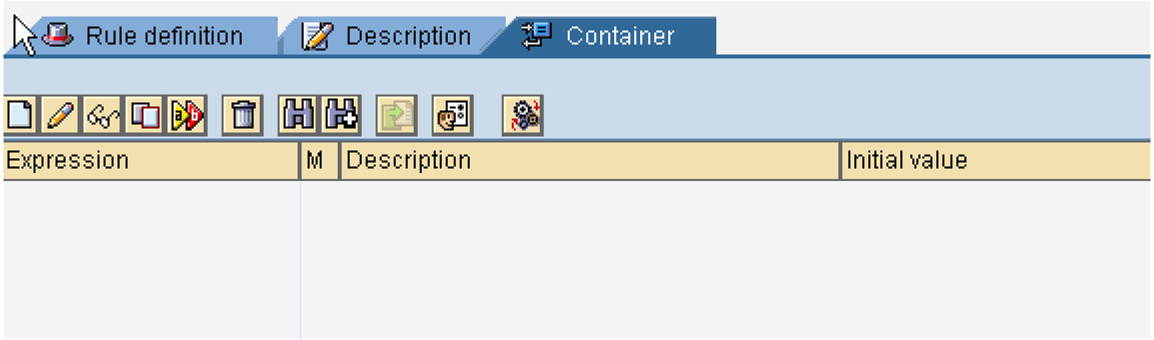
Category	Agent Determination: Function to be Executed
Function Module	ZTEST_FIND_SUPERIOR

☑ Terminate If Rule Resolution Without Result

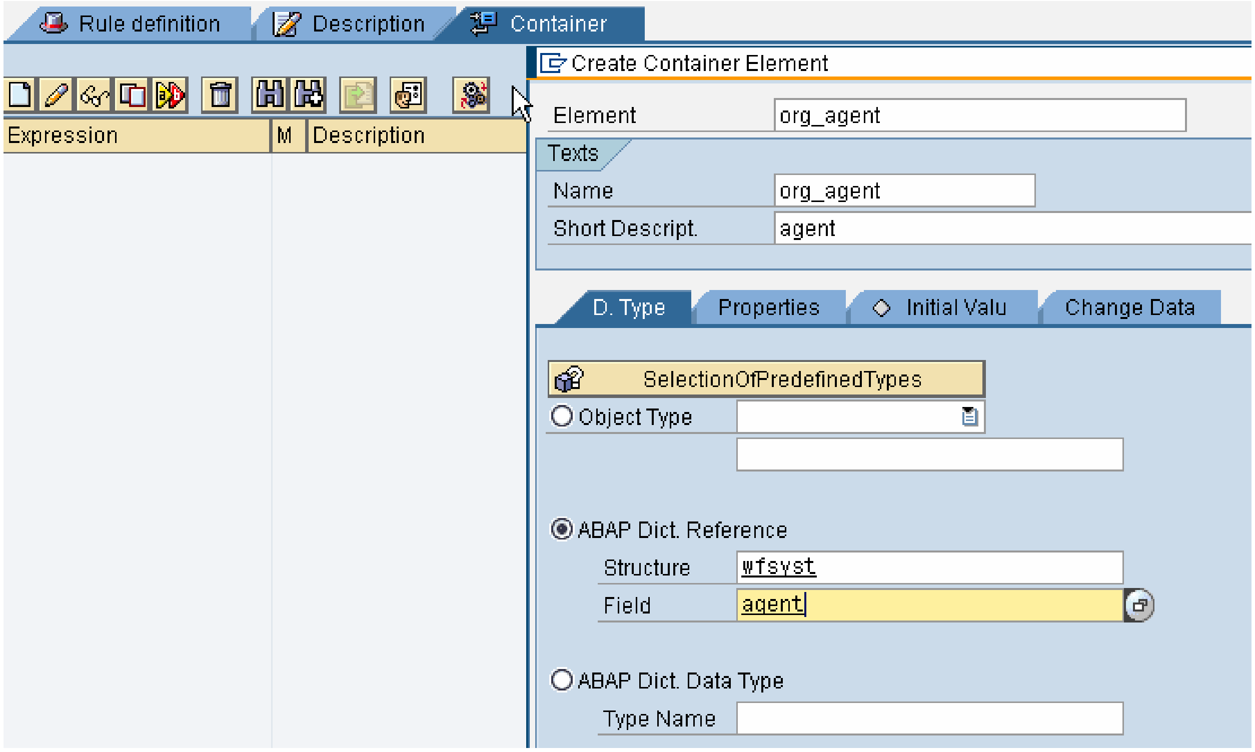
This function module will return the name of the superior for the user

In the Rule Definition tab select the **Agent Determination: Function to be Executed** in the Category field and provide the name of the Function module created (**ZTEST_FIND_SUPERIOR**).

Go to the Container tab and create a container element for the agent/user which will be passed to the function module by clicking the **Create** Button and enter the values as shown.



Expression	M	Description	Initial value



Create Container Element

Element	org_agent

Texts

Name	org_agent
Short Descript.	agent

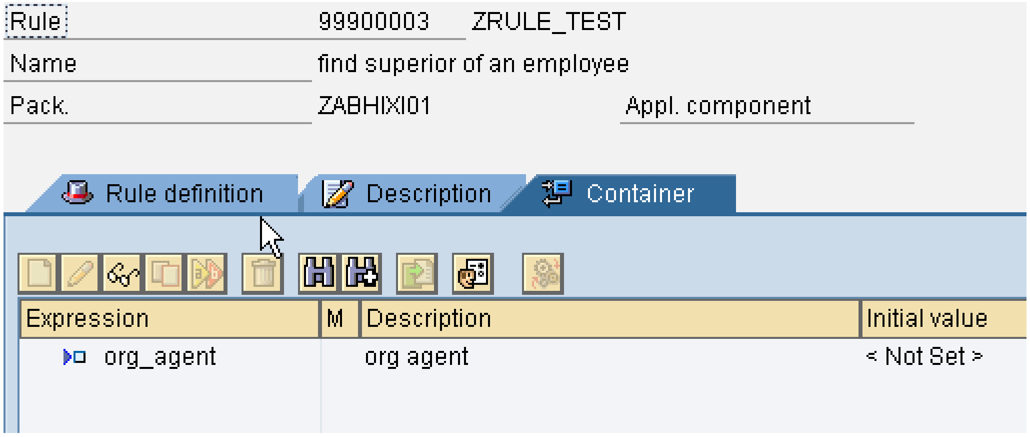
○ Object Type

● ABAP Dict. Reference

Structure	wfsyst
Field	agent

○ ABAP Dict. Data Type

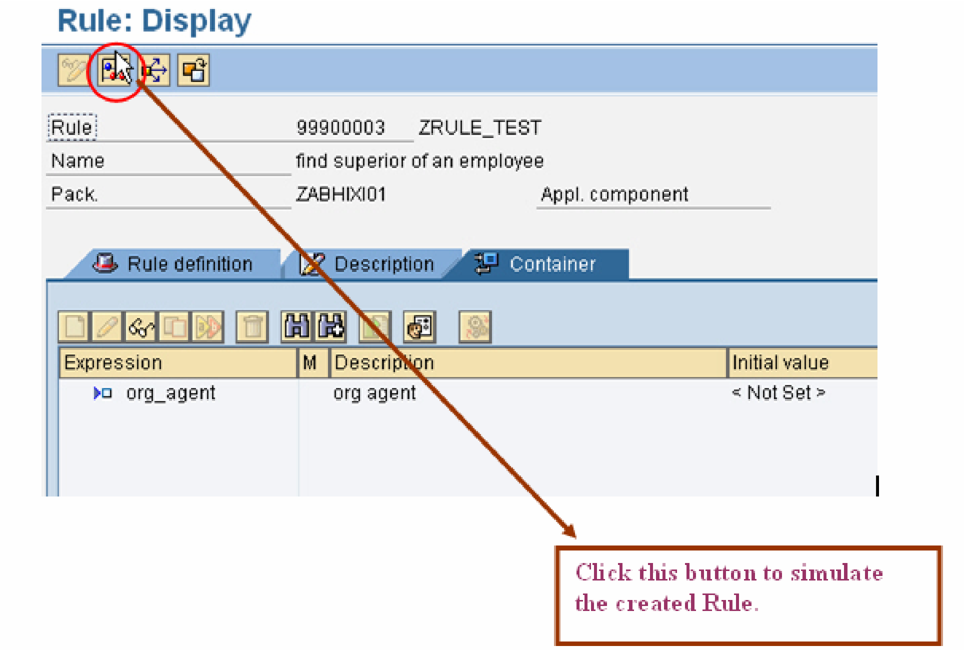
Type Name	



Rule	99900003	ZRULE_TEST
Name	find superior of an employee	
Pack.	ZABHIXI01	Appl. component

Expression	M	Description	Initial value
org_agent		org agent	< Not Set >

5. The rule is now created and we can test this rule in the PFAC transaction by clicking the “**Simulation**” button on the application toolbar. Enter the user name in the Container Element value section and press enter.



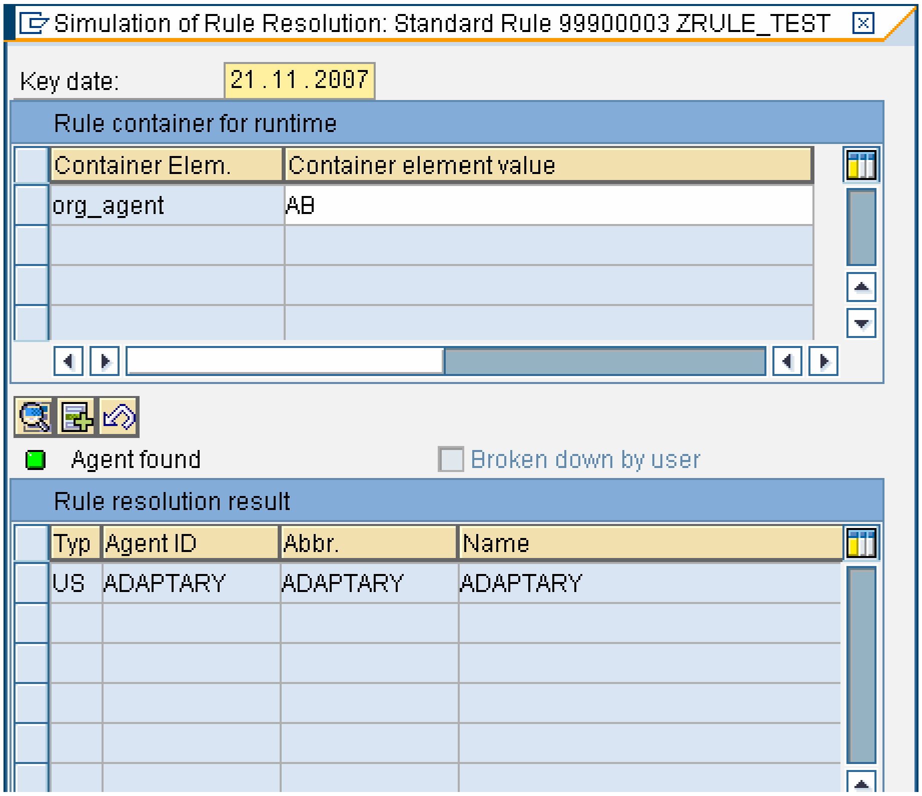
Rule: Display

Rule	99900003 ZRULE_TEST	
Name	find superior of an employee	
Pack.	ZABHIXI01	Appl. component

Expression	M	Description	Initial value
org_agent		org agent	< Not Set >

Click this button to simulate the created Rule.

6. It displays the name of the superior for the agent/user and the Agent Found is displayed in Green color.



Simulation of Rule Resolution: Standard Rule 99900003 ZRULE_TEST

Key date: 21.11.2007

Rule container for runtime

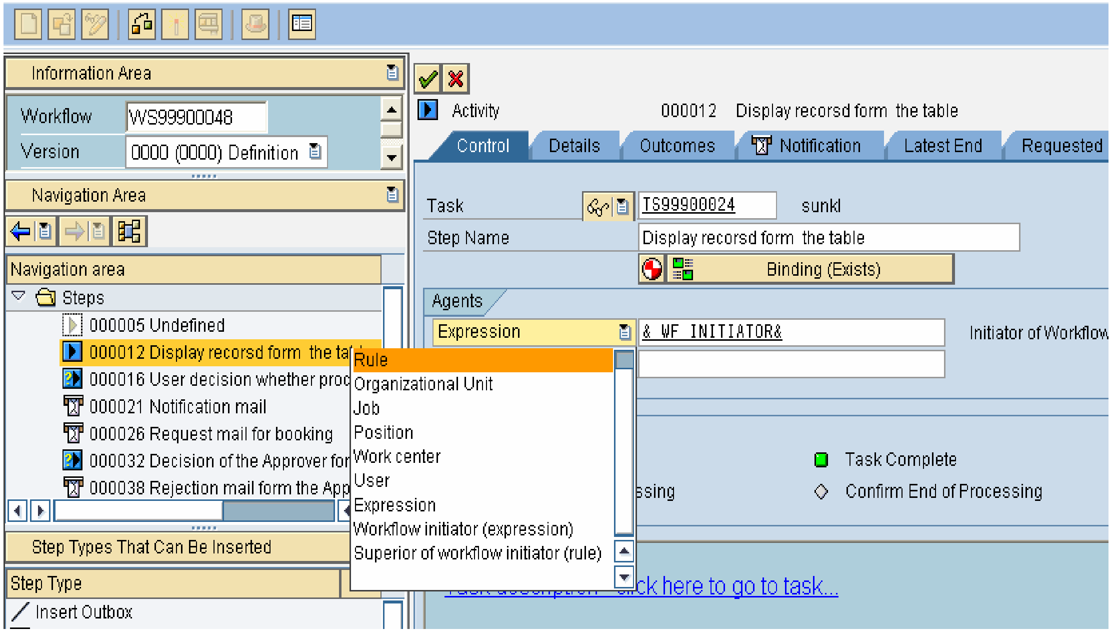
Container Elem.	Container element value
org_agent	AB

🟩 Agent found □ Broken down by user

Rule resolution result

Typ	Agent ID	Abbr.	Name
US	ADAPTARY	ADAPTARY	ADAPTARY

7. Now we can use this rule in any of your workflow definition as required.



Workflow WS99900048

Version 0000 (0000) Definition

Activity 000012 Display recorsd form the table

Task TS99900024 sunkl

Step Name Display recorsd form the table

Binding (Exists)

Agents

Expression &_WF_INITIATOR& Initiator of Workflow

Information Area
Navigation Area
Navigation area
Steps
000005 Undefined
000012 Display recorsd form the ta
000016 User decision whether proc
000021 Notification mail
000026 Request mail for booking
000032 Decision of the Approver for
000038 Rejection mail form the App
Step Types That Can Be Inserted
Step Type
Insert Outbox
Control
Details
Outcomes
Notification
Latest End
Requested
Rule
Organizational Unit
Job
Position
Work center
User
Expression
Workflow initiator (expression)
Superior of workflow initiator (rule)
ssing
Task Complete
Confirm End of Processing
Task description - click here to go to task...

Definition of an Agent Determination Rule

Use:

You use this function if you want to define more rules in addition to the agent determination rules delivered by SAP.

Features:

The following things should be specified while defining an Agent Determination Rule:

- Which information must be available so that rule resolution can be performed when the workflow is executed? — This information constitutes the ***rule parameters***. They are defined as elements of the rule container.
- The regulations in ***rule resolution*** that are used to determine the appropriate employees.

The rule resolution procedure is specified by the ***rule category***.

Rule Category Agent Determination: Responsibilities

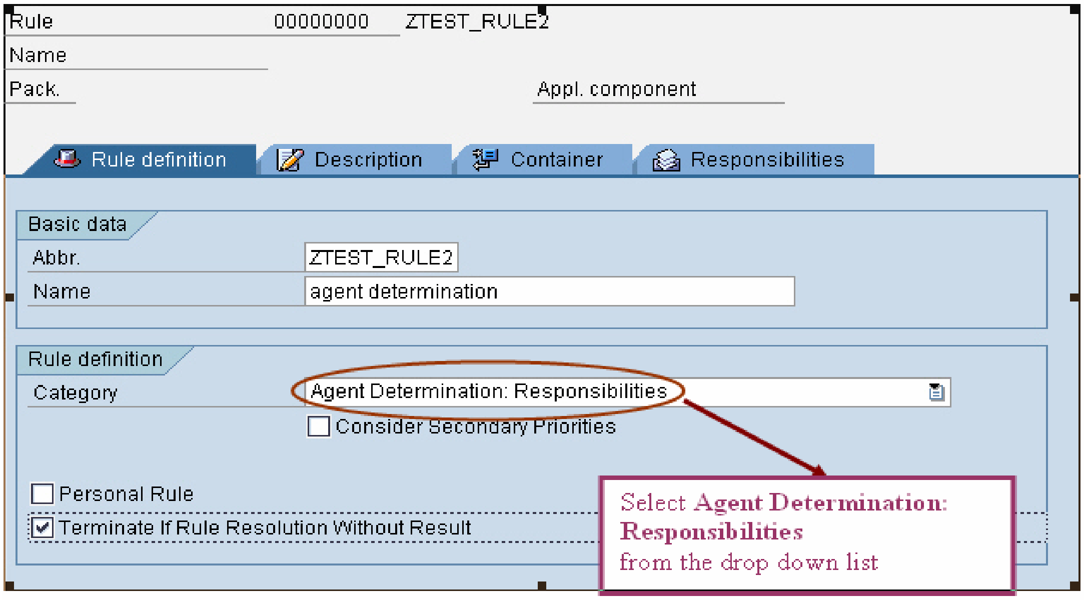
Use:

With this rule category you can select agents based on values from the workflow or the task. To do this, you transfer container elements from the task or workflow container to the rule container using binding. In the rule you define responsibilities that relate Organizational Management objects (job, position, organizational unit, user, and so on) with possible values from container elements in the rule container.

- A responsibility is itself an organizational object that requires no Customizing settings and no ABAP source text.
- Defined Value ranges are assigned to Agents.
- Data is filled from workflow container at runtime.

Procedure:

1. Go to **PFAC_ INS/PFAC_CHG** transaction for create/change rule.
2. On the Rule Definition tab page, choose the category **Agent Determination: Responsibilities**, and assign an abbreviation and a name for the rule.



Rule 00000000 ZTEST_RULE2
Name
Pack.
Appl. component

Basic data
Abbr. ZTEST_RULE2
Name agent determination

Rule definition
Category Agent Determination: Responsibilities
□ Consider Secondary Priorities
□ Personal Rule
☑ Terminate If Rule Resolution Without Result

Select **Agent Determination: Responsibilities** from the drop down list

3. **Create Rule Container**

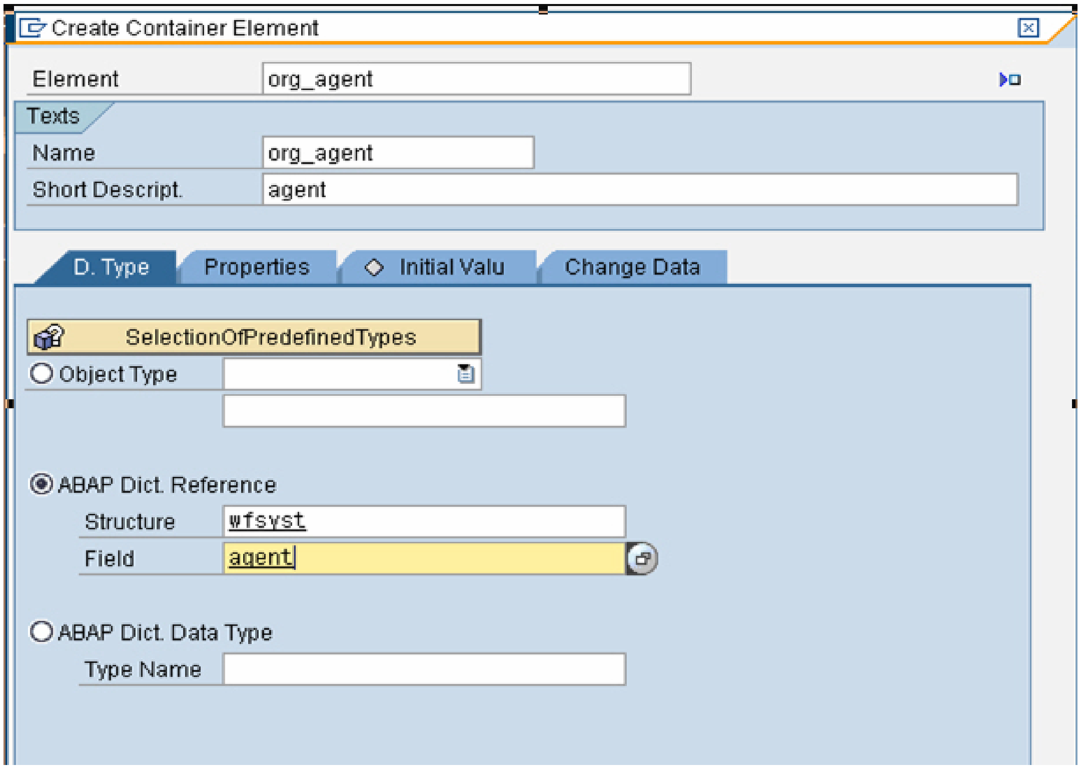
The Rule Container must contain the container elements whose values you want the system to check when the rule is executed. At runtime, binding fills the rule container with data from the task or workflow container. To create the rule container, you therefore need to know the definition of the task or workflow container.

a) Go to the Container tab and Choose 🗋.

b) The **Create container element** dialog box appears. Make entries in the ***Element***, ***Name***, and ***Short Description*** fields.

Create either an object reference or an ABAP dictionary reference for the container element.

As a guide, you can use the data type reference of the container element of the workflow or task container from which the container will be filled at runtime.



Create Container Element

Element: org_agent

Texts

Name: org_agent

Short Descript.: agent

○ Object Type

● ABAP Dict. Reference

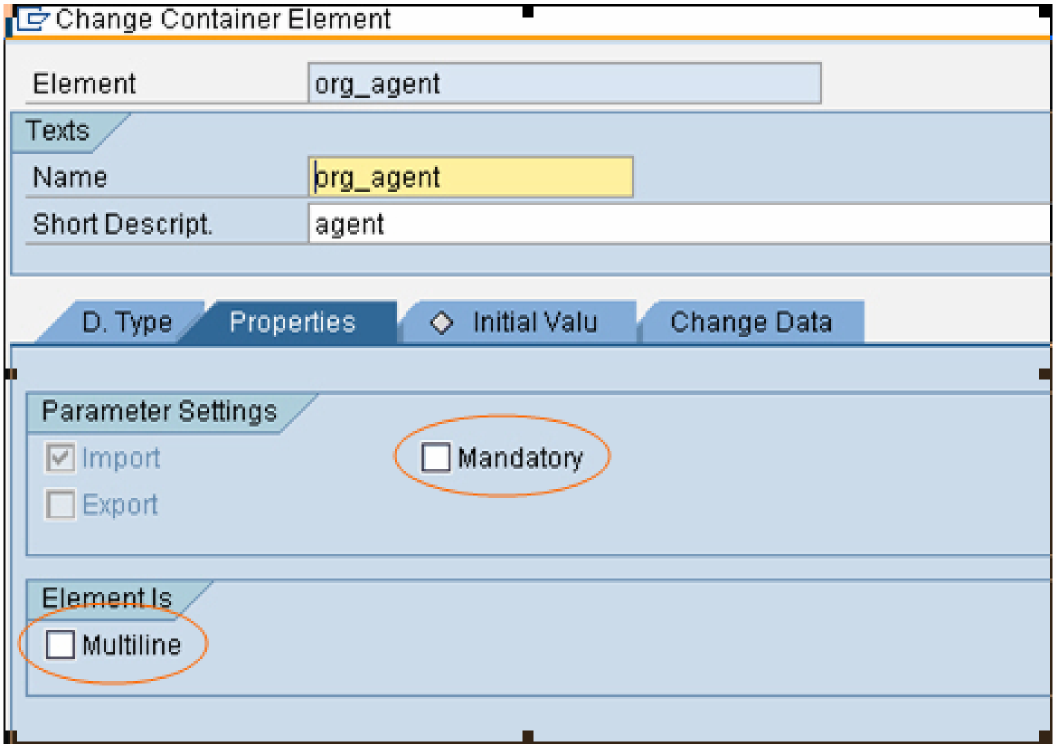
Structure: wfsyst

Field: agent

○ ABAP Dict. Data Type

Type Name:

c) On the ***Properties tab*** page, set, if required, the ***Obligatory*** and/or ***Multiline*** indicators.



Change Container Element

Element	org_agent
Name	org_agent
Short Descript.	agent

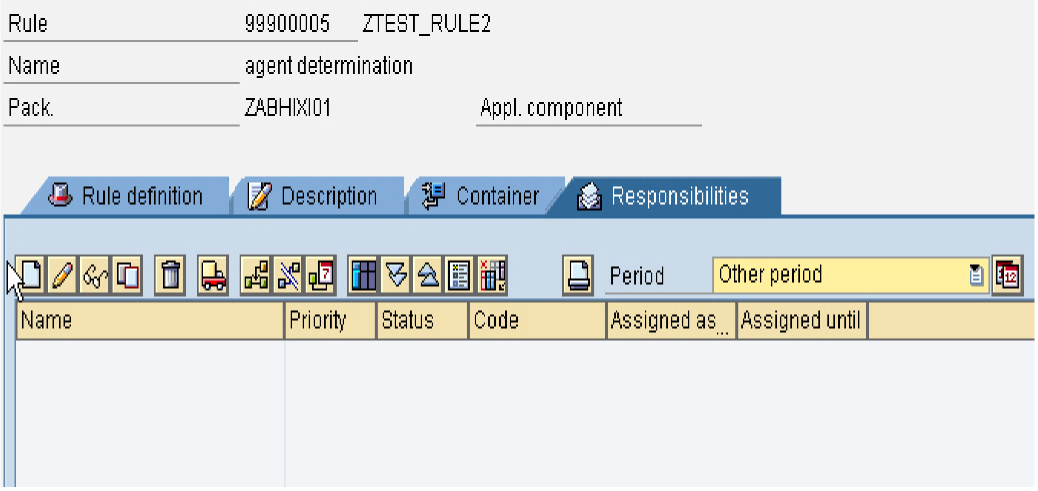
Repeat steps 1 to 4 until you have defined all the necessary elements for the rule container.

4. Creating a Responsibility-

With a responsibility you define the container element values that you want to trigger processing by the same users. In the responsibility, you define values or value ranges for the container elements of the rule container. If you wish, you can assign a priority to each responsibility.

a) Go to the **Responsibilities** tab.

Choose 🗋.



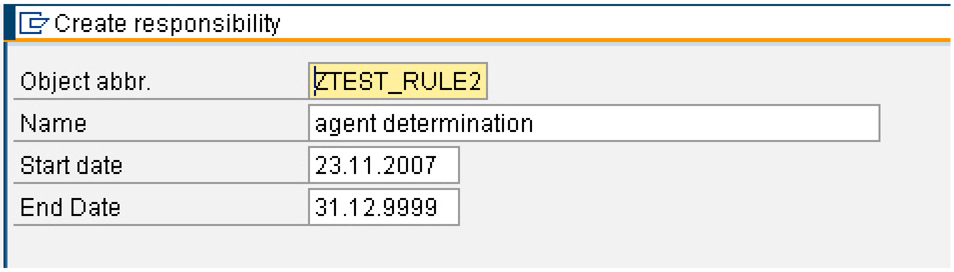
Rule	99900005	ZTEST_RULE2	
Name	agent determination		
Pack.	ZABHIXI01	Appl. component	

Period: Other period

Name	Priority	Status	Code	Assigned as ...	Assigned until

b) The ***Create Responsibility*** dialog box appears.

Enter an abbreviation, a name, and the validity period of the responsibility and confirm your entries.

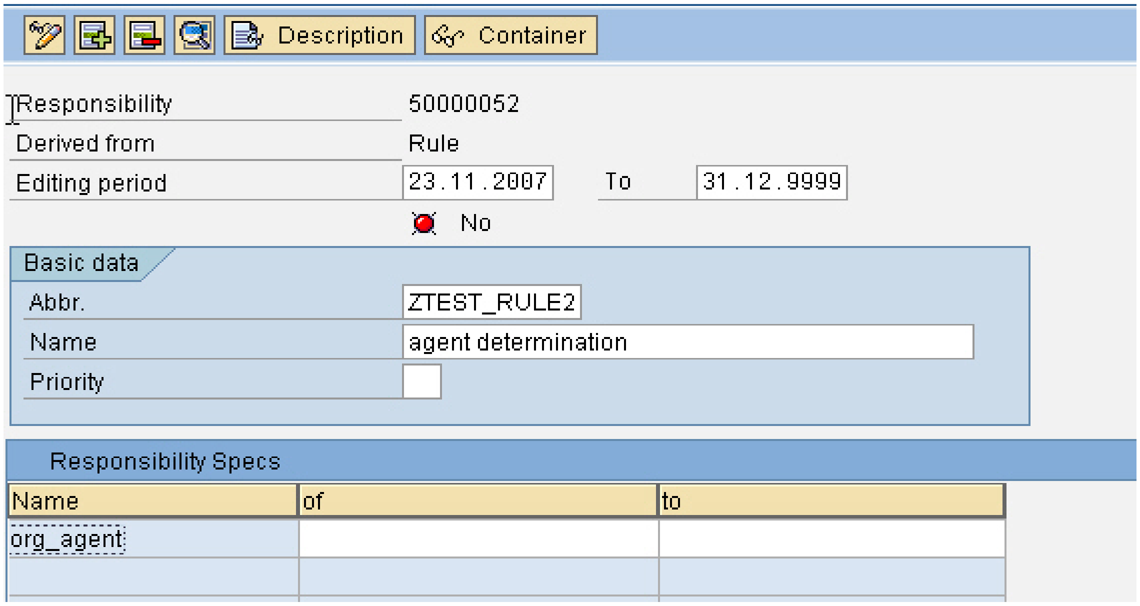


Create responsibility

Object abbr.	ZTEST_RULE2
Name	agent determination
Start date	23.11.2007
End Date	31.12.9999

The Change Responsibility for Rule <.....> screen appears.

In this responsibility editor, all container elements in the rule container are listed.



Responsibility	50000052		
Derived from	Rule		
Editing period	23.11.2007	To	31.12.9999
	● No		

Basic data

Abbr.	ZTEST_RULE2
Name	agent determination
Priority	

Responsibility Specs

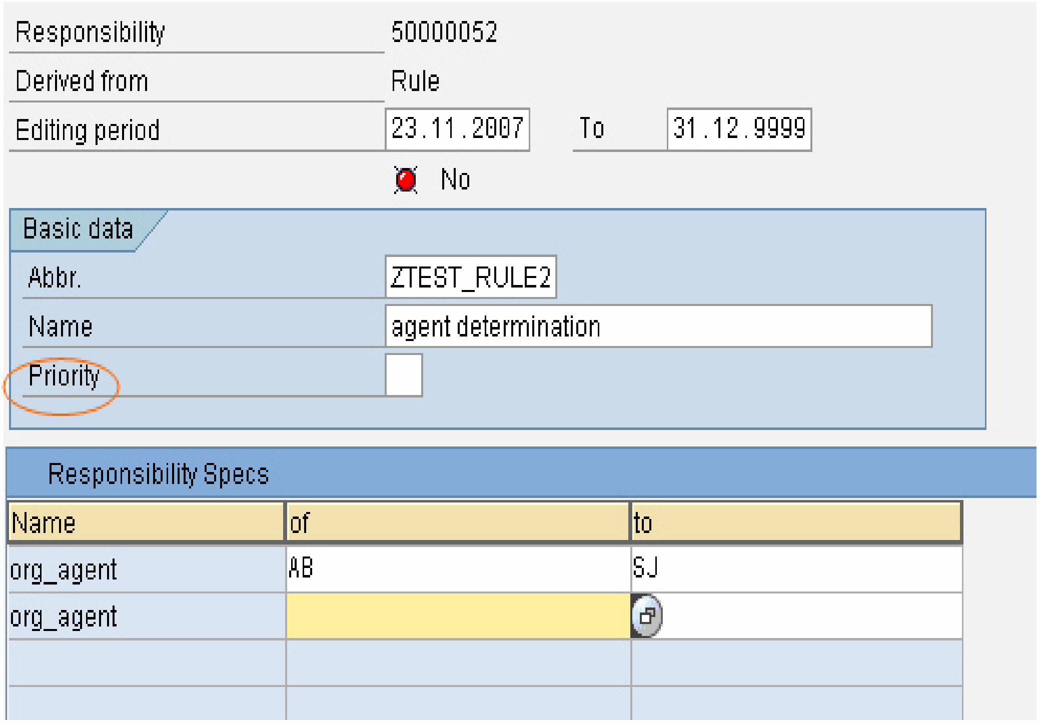
Name	of	to
org_agent		

Enter a single value or a value range for the container elements whose content you want the system to check during rule resolution. If you want to check several single values or value ranges for one container element, you can create new entry rows.

Position the cursor in a row with the container element for which you need a new row, and choose

Note:

- **If you do not want to check a container element for a particular responsibility, leave this line blank. The status of the responsibility is set to Responsibility incomplete, in order to draw your attention to unchecked container elements.**
- **If you wish, give the responsibility a priority.**
- **The responsibilities with the highest priority are evaluated first. Responsibilities with lower priorities are only evaluated if this is defined in the rule definition.**



Responsibility	50000052		
Derived from	Rule		
Editing period	23.11.2007	To	31.12.9999
	● No		

Basic data

Abbr.	ZTEST_RULE2
Name	agent determination
Priority	□

Responsibility Specs

Name	of	to
org_agent	AB	SJ
org_agent		

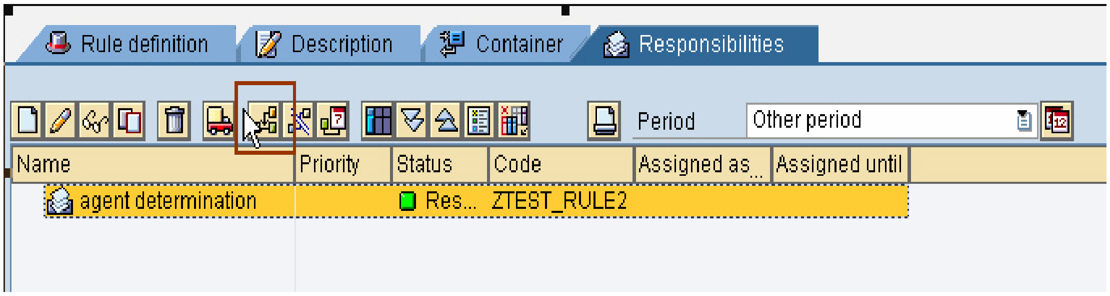
c) Save it.

5. Assigning Users or Organizational Objects to the Responsibility

You assign Organizational Management objects to a responsibility. These are then returned as the result of the rule, if the responsibility check is positive.

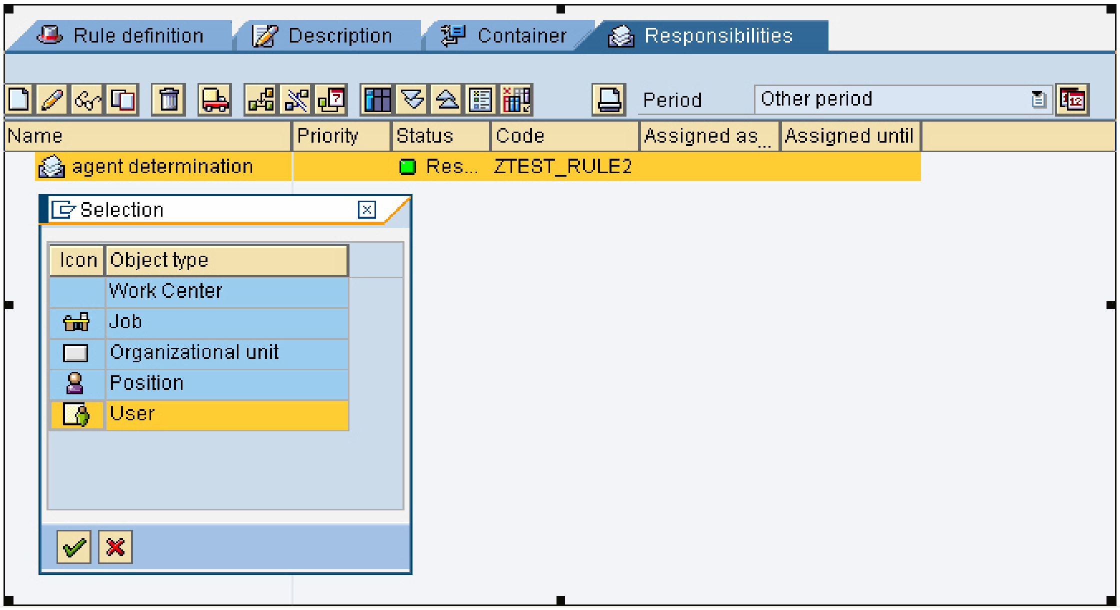
a) In the Responsibility tab select the responsibility to which you want to assign an Organizational Management object, and choose.

The Selection dialog box appears.



Period: Other period

Name	Priority	Status	Code	Assigned as ...	Assigned until
agent determination		🟩 Res...	ZTEST_RULE2		

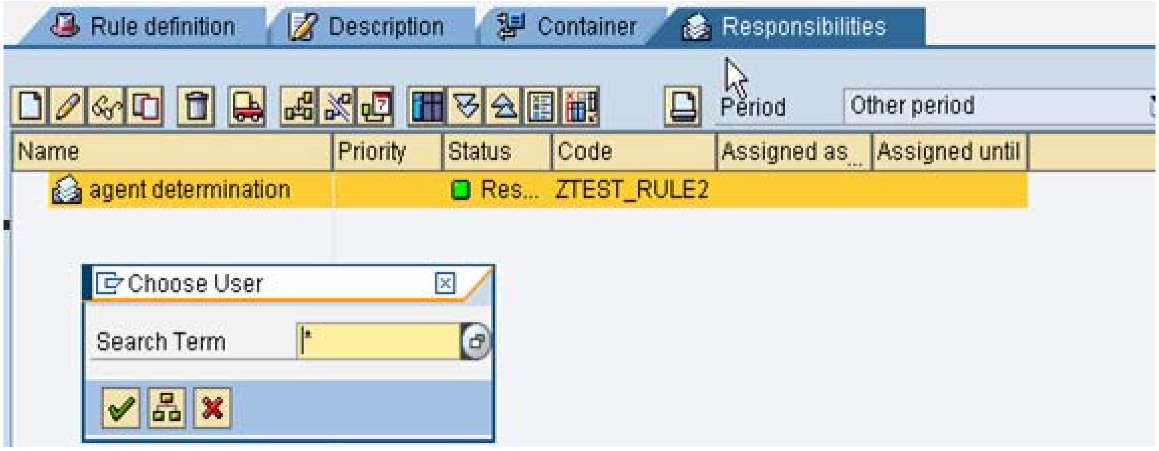


Name	Priority	Status	Code	Assigned as...	Assigned until
agent determination		🟩 Res...	ZTEST_RULE2		

Icon	Object type
	Work Center
	Job
□	Organizational unit
	Position
	User

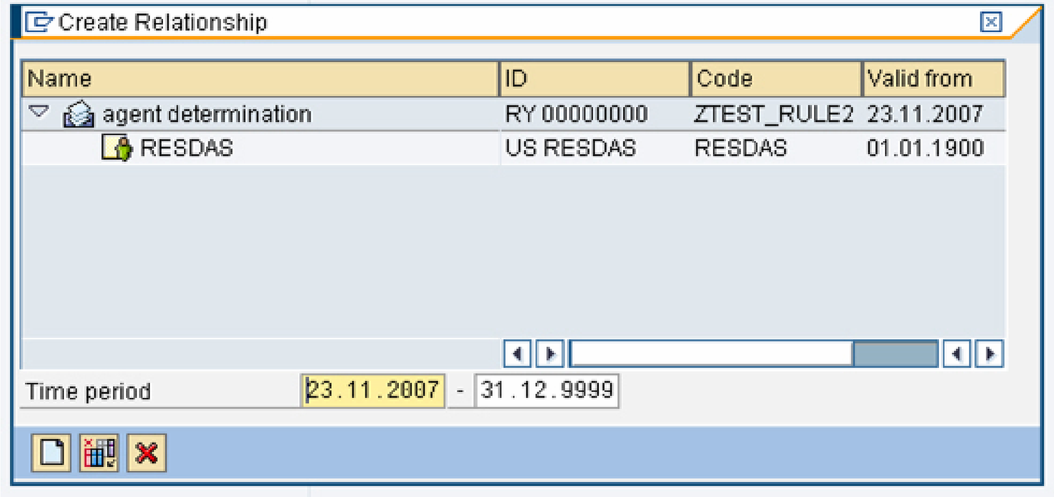
b) Select an object type that you want to assign as an agent and choose *Continue*.

c) The system asks you to enter a search term and to create the relationship between the responsibility and the selected object. Confirm your entries.



Name	Priority	Status	Code	Assigned as...	Assigned until
agent determination		🟩 Res...	ZTEST_RULE2		

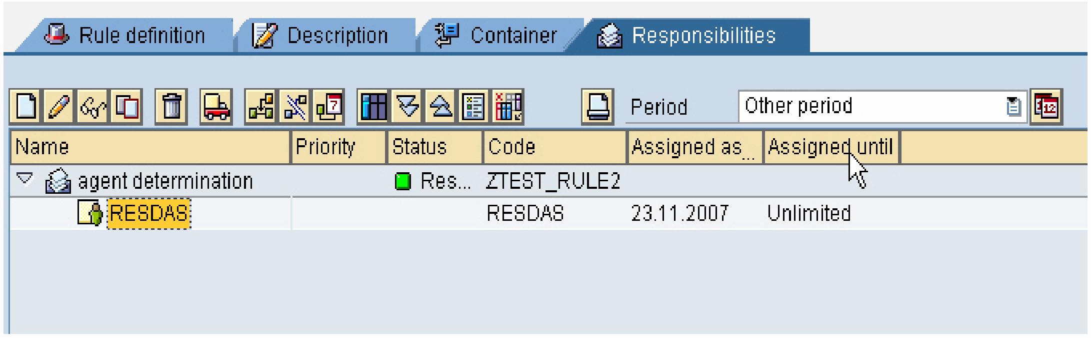
Search Term: *



Name	ID	Code	Valid from
agent determination	RY 00000000	ZTEST_RULE2	23.11.2007
RESDAS	US RESDAS	RESDAS	01.01.1900

Time period 23.11.2007 - 31.12.9999

In the Overall View, you can also specify a validity period for the relationship.



Period: Other period

Name	Priority	Status	Code	Assigned as	Assigned until
agent determination		🟩 Res...	ZTEST_RULE2		
RESDAS			RESDAS	23.11.2007	Unlimited

Rule Resolution Based on Evaluation Paths

Definition:

- A chain of relationships between objects in a hierarchical structure.
- Evaluation path describes how to find one or more organizational objects based

On an initial organizational units.

- The organizational situation of employees within an enterprise is depicted in the organizational plan. Using this existing information, you can perform rule resolution starting from one particular employee to determine one other employee, or several other employees, along specific relationships in the organizational plan.

Example: The evaluation path O-S-P describes the relationship chain:

Organizational unit **->** position **->** person **->**.

Use:

Evaluation paths are used for the selection of objects in evaluations.

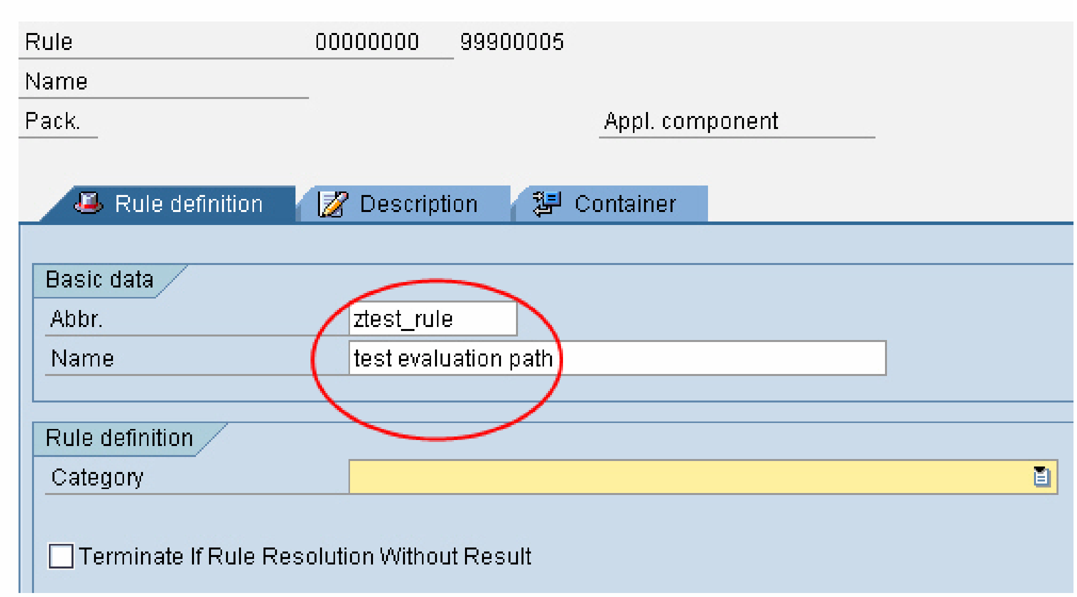
You select an evaluation path and the system evaluates the structure along this evaluation path. The report only takes into account the objects that it finds using the specified evaluation path.

Maintain Evaluation path:

Transaction: OOAW

Procedure

1. Go to PFAC_INS and create a rule and enter the name and short description for the rule.



Rule 00000000 99900005

Name

Pack.

Appl. component

Basic data

Abbr. ztest_rule

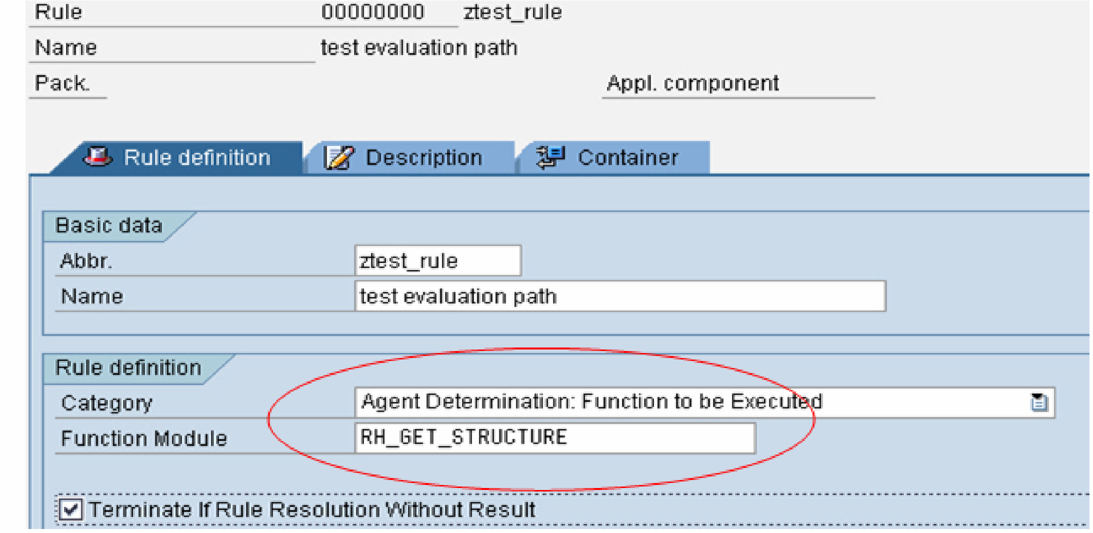
Name test evaluation path

Rule definition

Category

□ Terminate If Rule Resolution Without Result

2. In the Category field choose ***Agent Determination: Function to be Executed*** and enter **RH_GET_STRUCTURE** as the function module.



Rule 00000000 ztest_rule

Name test evaluation path

Pack.

Appl. component

Basic data

Abbr. ztest_rule

Name test evaluation path

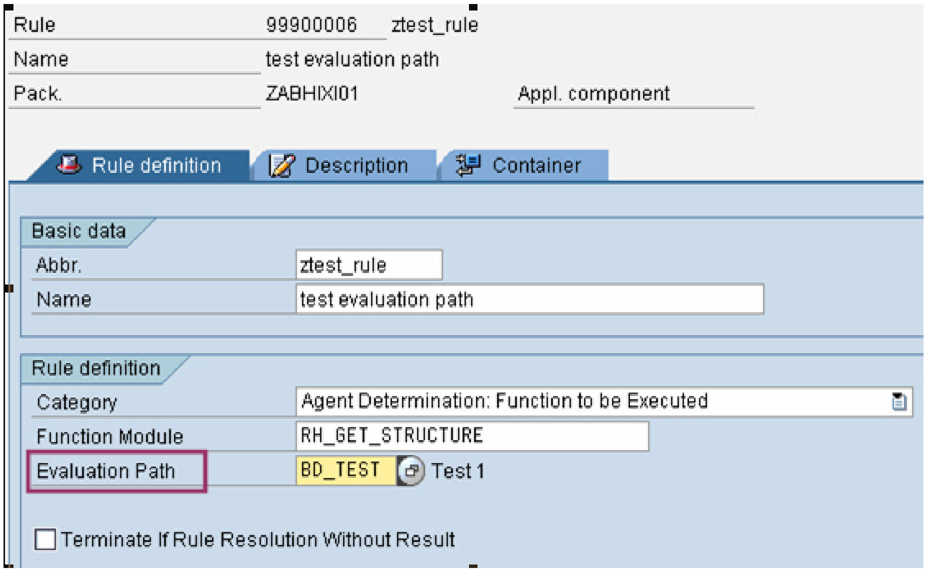
Rule definition

Category Agent Determination: Function to be Executed

Function Module RH_GET_STRUCTURE

☑ Terminate If Rule Resolution Without Result

The evaluation path is now active. Now you can enter the required Evaluation path.



Rule 99900006 ztest_rule

Name test evaluation path

Pack. ZABHIXI01

Appl. component

Basic data

Abbr. ztest_rule

Name test evaluation path

Rule definition

Category Agent Determination: Function to be Executed

Function Module RH_GET_STRUCTURE

Evaluation Path BD_TEST Test 1

□ Terminate If Rule Resolution Without Result

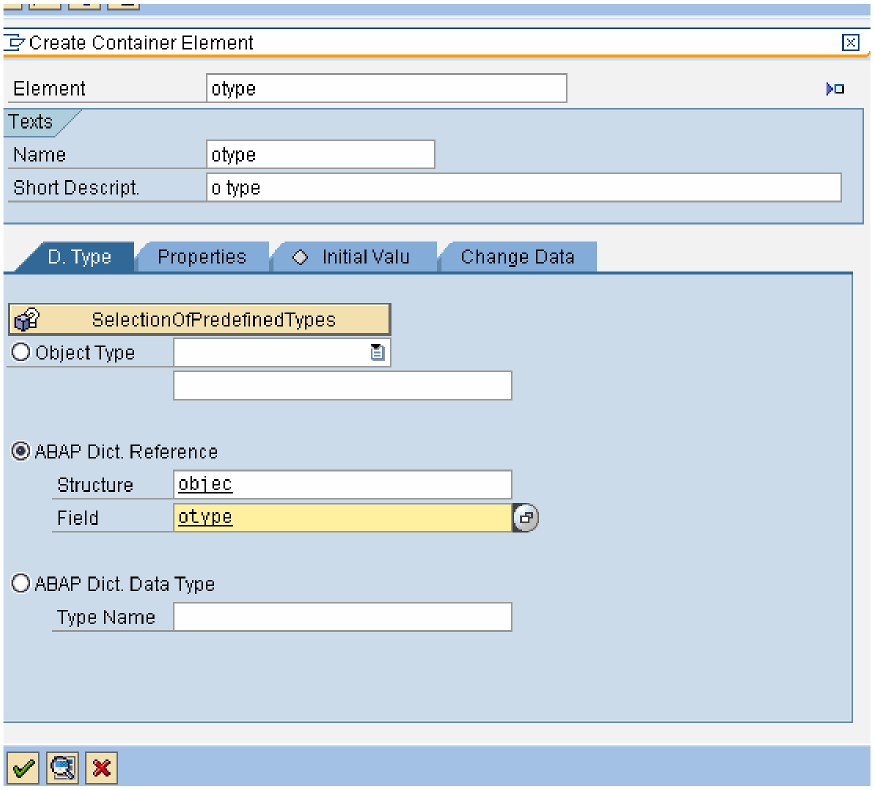
3. Switch to the Container tab page to create the Rule Container.

Note: The rule container for an agent determination rule based on evaluation paths must contain only the *Organizational Management object* for which the evaluation path is to be applied. At runtime the system fills the rule container from the workflow or task container using binding.

Click on the *Container* tab page and create the following container elements.

Name	Data type reference	Description
`OType`	`OBJEC-OTYPE`	Type of the *Organizational Management object*
`ObjID`	`OBJEC-REALO`	ID of the *Organizational Management object*
`Org_Agent`	`WFSYST-AGENT`	Organizational Management object

1. Choose .



Create Container Element

Element	otype

Texts

Name	otype
Short Descript.	o type

○ Object Type

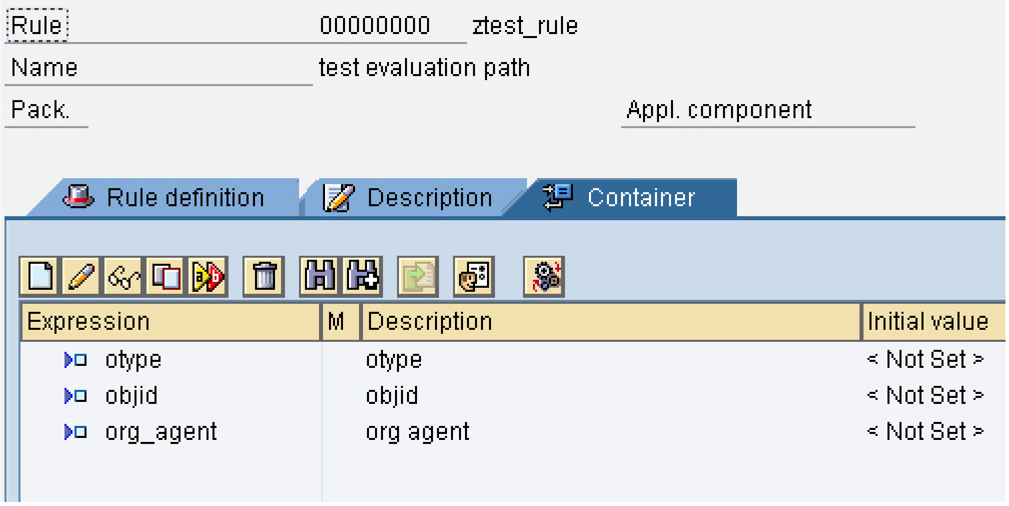
● ABAP Dict. Reference

Structure	objec
Field	otype

○ ABAP Dict. Data Type

Type Name	

2. Make entries in the *Element*, *Name*, and *Short Description* fields.
3. Create the required ABAP dictionary reference for the container element.
4. Repeat steps 1 to 3 until you have defined all the elements for the rule container.



Rule	00000000	ztest_rule
Name	test evaluation path	
Pack.		Appl. component

Expression	M	Description	Initial value
otype		otype	< Not Set >
objid		objid	< Not Set >
org_agent		org agent	< Not Set >

Choose Save.

Note: The rule container contains elements with which an Organizational Management object can be transferred. At runtime either the container elements OType and ObjID or the container element Org_Agent must be filled. If all container elements are filled, only the value transferred in Org_Agent is evaluated.

Related Content

Rules in SAP Business Workflow

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.