



Silvana Kempf,  
SAP SRM  
Development

**Workflow in SAP Supplier Relationship  
Management 7.0  
(SP01)**

October 17, 2008

Version 1.2

## History of Changes

This document is regularly updated in *SAP Service Marketplace* at [service.sap.com/srm-inst](http://service.sap.com/srm-inst).

The following table provides an overview of the most important changes that were made in the latest versions.

<b>Document Version</b>	<b>Important Changes</b>
1.0 (August 26, 2008)	First version
1.1 (October 17, 2008)	Intermediate version
1.2 (October 22, 2008)	Final version

# Contents

<b>1 Introduction .....</b>	<b>8</b>
1.1 Two approval process frameworks .....	8
1.2 Further References.....	10
1.3 Glossary.....	10
<b>2 Process-Controlled Workflow .....</b>	<b>11</b>
2.1 SAP Business Workflow in SAP SRM.....	12
2.2 Business Rule Framework at SRM .....	13
2.3 SAP NetWeaver Portal and UWL.....	14
2.3.1 Notifications in the UWL.....	14
2.4 Business Objects Using the Approval Process Framework.....	15
2.5 Approval Processes in SAP SRM .....	16
2.5.1 Responsibility, Decision Sets and Decision Types .....	16
2.5.2 Fundamental Paradigms .....	17
2.6 The Approval Process in Detail .....	19
2.7 Status and Control .....	19
2.7.1 Process Schema Evaluation.....	19
2.7.2 Approver Determination .....	21
2.7.3 Reviewer Determination.....	23
<b>3 Configuring the System for Process-Controlled Workflow and Extending the Framework .....</b>	<b>25</b>
3.1 Configuration Process .....	25
3.2 Tips for Approval Process Framework Configuration .....	25
3.2.1 Transporting BRF Content .....	25
3.2.2 Features .....	25
3.2.3 Activities .....	26
3.3 Process Level Configuration .....	27
3.4 Customer-Specific Work Items .....	28
3.4.1 Adjust Work Item Text of SRM Standard Tasks .....	28
3.4.2 Adjust Task Description of SRM Standard Tasks.....	28
3.4.3 Customer BRF Expressions .....	29
3.4.4 Programming Guidelines for a Function Module Used in the BRF.....	29
3.5 Customer Responsibility Resolver Strategies .....	32
3.5.1 Interface /SAPSRM/IF_EX_WF_RESP_RESOLVER.....	32

<b>3.6 Customer Strategies for Reviewer Determination .....</b>	<b>36</b>
3.6.1 Interface /SAPSRM/IF_EX_WF_REVIEWER_RULE .....	36
<b>4 Cheat Sheets – Process-Controlled Workflow.....</b>	<b>37</b>
4.1 Extensive Example.....	37
4.2 Upgrade .....	40
<b>5 Troubleshooting – Process-Controlled Workflow .....</b>	<b>41</b>
<b>5.1 Utilities for Troubleshooting.....</b>	<b>41</b>
5.1.1 SRM-Specific Utilities.....	41
5.1.2 SAP NetWeaver Utilities.....	41
5.2 Find out if an approval process is started.....	41
5.3 Find out where and why the approval process got stuck.....	42
5.4 Find Out why Levels Are Not Executed.....	42
5.5 Find Out why Document Is still ‘Awaiting Approval’ .....	42
<b>6 Appendix A: Shipped BRF Expressions.....</b>	<b>44</b>
6.1 Common .....	44
6.2 Process/Workflow attributes.....	44
6.3 Organizational Model access.....	47
6.4 Configurable Calls.....	48
6.5 BO Confirmation.....	49
6.5.1 Document Attributes.....	49
6.6 BO Contract .....	56
6.6.1 Document Attributes.....	56
6.7 BO Invoice .....	64
6.7.1 Document Attributes.....	64
6.8 BO Purchase Order .....	74
6.8.1 Document Attributes.....	74
6.9 BO Quote .....	80
6.9.1 Document Attributes.....	80
6.10 BO RFQ .....	85
6.10.1 Document Attributes.....	85
6.11 BO Shopping Cart.....	90
6.11.1 Document Attributes.....	90
6.11.2 Level Evaluation.....	101

---

<b>7 Appendix B: Approver Determination Strategies in the Standard .....</b>	<b>102</b>
<b>7.1 Overview .....</b>	<b>102</b>
<b>7.2 Shopping Cart (BUS2121) .....</b>	<b>102</b>
7.2.1 Implementation - Specify Employee for Approval.....	102
7.2.2 Implementation - Get Requester's Line Manager for Approval .....	102
7.2.3 Implementation - Get Manager of Previous Approver for Approval.....	103
7.2.4 Implementation - Get Purchasers of all Purchasing Groups for Approval .....	103
7.2.5 Implementation - Get all Employees of Specified Department for Approval .....	104
7.2.6 Implementation - Get all Users of Specified Role for Approval .....	104
7.2.7 Implementation - Get Users of Spending Limit Chain for Approval .....	104
<b>7.3 Purchase Order (BUS2201) .....</b>	<b>106</b>
7.3.1 Implementation - Specify Employee for Approval.....	106
7.3.2 Implementation - Get Requester's Line Manager for Approval .....	106
7.3.3 Implementation - Get Manager of Previous Approver for Approval.....	106
7.3.4 Implementation - Get the Manager of the Purchasing Group for Approval.....	107
7.3.5 Implementation - Get all Employees of Specified Department for Approval .....	107
7.3.6 Implementation - Get all Users of Specified Role for Approval .....	107
7.3.7 Implementation - Get Users of Spending Limit Chain for Approval .....	107
<b>7.4 Purchase Contract (BUS2000113) .....</b>	<b>109</b>
7.4.1 Implementation - Specify Employee for Approval.....	109
7.4.2 Implementation - Get Manager of Document Owner for Approval .....	109
7.4.3 Implementation - Get Manager of Previous Approver for Approval.....	109
7.4.4 Implementation - Get Manager of the Purchasing Group for Approval.....	110
7.4.5 Implementation - Get all Employees of Specified Department for Approval .....	110
7.4.6 Implementation - Get all Users of Specified Role for Approval .....	110
<b>7.5 Confirmation (BUS2203).....</b>	<b>111</b>
7.5.1 Implementation - Specify Employee for Approval.....	111
7.5.2 Implementation - Get Manager of Document Owner for Approval .....	111
7.5.3 Implementation - Get Manager of Previous Approver for Approval.....	111
7.5.4 Implementation - Get Manager of the Purch. Group for Approval .....	112
7.5.5 Implementation - Get Owner of Source Document for Approval .....	112
7.5.6 Implementation - Get all Employees of Specified Department for Approval .....	113
7.5.7 Implementation - Get all Users of Specified Role for Approval .....	113
<b>7.6 Invoice (BUS2205) .....</b>	<b>114</b>
7.6.1 Implementation - Specify Employee for Approval.....	114
7.6.2 Implementation - Get Manager of Document Owner for Approval .....	114
7.6.3 Implementation - Get Manager of Previous Approver for Approval.....	114
7.6.4 Implementation - Get Manager of the Purchasing Group for Approval.....	115
7.6.5 Implementation - Get Owner of Source Document for Approval .....	115
7.6.6 Implementation - Get all Employees of Specified Department for Approval .....	116
7.6.7 Implementation - Get all Users of Specified Role for Approval .....	116
<b>7.7 RFx (BUS2200).....</b>	<b>117</b>
7.7.1 Implementation - Specify Employee for Approval.....	117

- 7.7.2 Implementation - Get Manager of Document Owner for Approval ..... 117
- 7.7.3 Implementation - Get Manager of Previous Approver for Approval..... 117
- 7.7.4 Implementation - Get Manager of the Purchasing Group for Approval..... 118
- 7.7.5 Implementation - Get all Employees of Specified Department for Approval ..... 118
- 7.7.6 Implementation - Get all Users of Specified Role for Approval ..... 118
- 7.8 Supplier Quotation (BUS2202)..... 119**
  - 7.8.1 Implementation - Specify Employee for Approval..... 119
  - 7.8.2 Implementation - Get Manager of Previous Approver for Approval..... 119
  - 7.8.3 Implementation - Get Manager of Purchasing Group (Header) for Approval..... 119
  - 7.8.4 Implementation - Get all Employees of Specified Department for Approval ..... 120
  - 7.8.5 Implementation - Get all Users of Specified Role for Approval ..... 120
- 7.9 Internal Enhancement Implementation /SAPSRM/BD\_WF\_AGENTS ..... 121**
- 7.10 Documentation of Implementing Class for RR\_EMPLOYEE..... 122**
- 7.11 Documentation of Implementing Class for RR\_MANAGER ..... 123**
- 7.12 Documentation of Implementing Class for RR\_MANAGER\_OF\_MANAGER..... 125**
- 7.13 Documentation of Implementing Class for RR\_PURCHASING\_MANAGER ..... 126**
- 7.14 Documentation of Implementing Class for RR\_REPORTING\_LINE\_UNIT..... 127**
- 7.15 Documentation of Implementing Class for RR\_ROLE..... 128**
- 8 Appendix C: Standard Implementation of Dynamic Process Levels ..... 129**
  - 8.1 Implementation /SAPSRM/BD\_WF\_CONFIG\_SC\_SL ..... 129**
    - 8.1.1 General information for shopping cart implementation ..... 129
    - 8.1.2 Implemented logic for shopping cart..... 129
  - 8.2 Implementation /SAPSRM/BD\_WF\_CONFIG\_PO\_SL..... 130**
    - 8.2.1 General information for purchase order implementation..... 130
    - 8.2.2 Implemented logic for purchase order ..... 130
- 9 Appendix D: Standard Implementations of /SAPSRM/BD\_WF\_REVIEWER\_RULE131**
  - 9.1 Overview ..... 131**
  - 9.2 Shopping Cart (BUS2121): ..... 131**
  - 9.3 Purchase Order (BUS2201):..... 131**
  - 9.4 Purchase Contract (BUS2000113): ..... 132**
  - 9.5 Confirmation (BUS2203): ..... 132**
  - 9.6 Invoice (BUS2205): ..... 132**
  - 9.7 Bid Invitation (BUS2200): ..... 132**
  - 9.8 Supplier Quotation (BUS2202):..... 133**

<b>9.9</b>	<b>Documentation of Implementing Class for RV_EMPLOYEE.....</b>	<b>133</b>
<b>9.10</b>	<b>Documentation of Implementing Class for RV_REPORTING_LINE_UNIT.....</b>	<b>133</b>
<b>9.11</b>	<b>Documentation of Implementing Class for RV_ROLE.....</b>	<b>134</b>
<b>9.12</b>	<b>Documentation of Implementing Class for RV_SOURCE_DOCUMENT_RESPONSIBLE.....</b>	<b>134</b>

# 1 Introduction

The operations in SAP SRM 7.0 are mostly workflow-based. The term 'workflow', in this context, does not refer to a workflow engine, but rather to a technical implementation of business processes.

The SRM system provides two types of workflows: implicit and explicit. The main difference between these types is the flexibility required throughout the implementation process.

*Implicit workflow:*

- Is based on the main status of a document
- Does not create a specific work item
- Allows a user to search for their personal work basket, based on their specific role and authorization

Examples include using the Personal Object Worklist (POWL) or the advanced search.

*Explicit workflow:*

- Creates a specific work item, alert, or notification
- Is based on fixed steps in a business process and specific responsibilities
- Is often related to deadlines that are monitored

Despite this, many users consider 'workflow' in the SRM context to simply mean 'approval processes'. An approval process is a subprocess that is typically run when a business document is in its final stages and will soon be released to trigger a follow-on process.

As with earlier releases of SRM, the majority of approval processes are backed by the SAP Business Workflow. However, as one of the major architectural changes in SAP SRM 6.0, a new SRM-specific framework for approval processes has been added. The SRM-specific framework for approval processes of earlier releases cannot be used in SAP SRM 6.0. In SAP SRM 7.0, the SRM-specific framework for approval processes of SAP SRM 5.0 was reactivated. In other words, both frameworks are available.

This document mainly explains why the framework introduced in SRM6.0 is necessary, and how the different parts of this framework interact with each other.

## 1.1 Two approval process frameworks

In SAP SRM 7.0, two business workflow frameworks are supported:

### **Application-controlled workflow**

This refers to the SAP SRM approval workflows, based on SAP Business Workflow, used in releases SAP SRM 5.0 and earlier. With this framework, the application is in control over the SAP Business Workflow templates.

### **Process-controlled workflow**

This refers to the SAP SRM approval workflows, based on SAP Business Workflow, introduced in release SAP SRM 6.0. With this framework, the configured process is in control over the process flow, and not the application itself.

Process-controlled workflows were introduced in SAP SRM 6.0 to replace application-controlled workflows. The newer framework is more robust, more flexible, and more transparent than the older one. Other improvements with process-controlled workflows are that they are easier to maintain and support, they are less sensitive to errors, and they are lighter to configure.

Using the application-controlled workflow framework is only intended in exceptional cases for customers upgrading from release SAP SRM 5.0 or earlier.

The following comparison matrix is intended to help analyzing customer needs regarding upgrade strategy.



<b>Scenario</b>	<b>Process-Controlled Workflow</b>	<b>Application-Controlled Workflow</b>
Team purchasing should be possible	Supported	Not supported
n-step approval should be possible by configuration (without Badl implementation effort on customer side)	Supported	Implementation of Badl BBP_WFL_APPROV_BADI required, possible for PO, SC, RFx Response, RFx, Contract
Approval per item	Available by configuration for SC, PO, Contract	Available for SC
Approval with distributed responsibility (approver has to approve/reject only those items he or she is responsible for)	Available by configuration for Shopping Cart	Implementation of Badl BBP_WFL_APPROV_BADI required and only possible for SC
Changing document by approver during approval anytime	Only possible during approval levels with completion	Supported (depending on security level)
Rule-based reviewer	Supported	Reviewers must be added manually
Add ad hoc approver	Supported	Supported
Delete ad hoc approver	Possible as long as work item is in process	Not possible
Replace ad hoc approver	Ad hoc approver must be deleted; a new one can be inserted	Supported
Forwarding work items	Supported	Supported
Creation of substitution rules	Supported	Supported
Add reviewer manually	Supported	Supported
Delete reviewer	Possible as long as work item is in process	Supported
Replace reviewer	Reviewer must be deleted; a new one can be inserted	Supported
Archiving	Workflow data will be archived when archiving documents	Workflow data must be archived separately
Flexible agent determination	Default Badl implementations available; creation of implementations for Badl definition /SAPSRM/BD_WF_RESP_RESOLVER	Implementation of Badl BBP_WFL_APPROV_BADI required, or modification of SAP coding
Workflow process restart in case of document change	Default behavior can be changed by implementing Badl interface /SAPSRM/IF_EX_WF_PROC_RESTART	Depends on security level of user who changed the document (personalization object BBP_WFL_SECURITY) or on implemented Badl BBP_WFL_APPROV_BADI
Back&forth feature	Available during completion levels for all documents and in case of partial rejection	Available for PO, SC, RFx in case of n-step approval (implementation of Badl BBP_WFL_APPROV_BADI required)

Save document during approval (without work item execution)	Supported	Available for PO, SC, RFx in case of n-step approval (implementation of BAdI BBP_WFL_APPROV_BADI required)
Complete approval history available in Approval Process Overview despite workflow process restart	Supported	Customer-specific BAdI implementation required BBP_WFL_APPROV_BADI (not supported for template-based workflow)
Reviewer actions available in Approval Process Overview	Supported	Not supported
Work item marked in UWL if configured deadline exceeded	Supported	Not supported
Offline approval for forwarded work items	Supported	Not supported
Display of cost center splitting of Shopping Cart items in offline approval	Supported	Not supported

## 1.2 Further References

This document supplements the existing documentation for the SRM approval process framework. For more information, see the following:

- SAP Help Portal at <http://help.sap.com> → *SAP Solutions* → *Overview* → *SAP Business Suite* → *SAP Supplier Relationship Management* → *Functions* → *Approval Process Framework*.
- SAP Solution Manager under *Configuration* → *SAP SRM 2007* → *Basic Settings for SAP SRM* → *Cross-Application Settings* → *Business Workflow*
- SAP Service Marketplace at <http://service.sap.com/instguides> → *SAP Business Suite Applications* → *SAP SRM* → *SAP SRM7.0* → *Upgrade Guide*

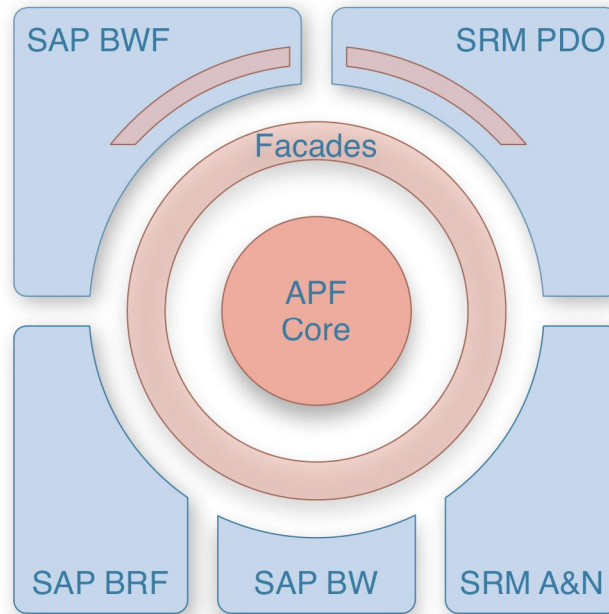
## 1.3 Glossary

English Term	English Abbreviation	Definition	Entry
SAP Business Workflow	SAP BWF		
Business Rules Framework	BRF		
Universal Worklist	UWL		
Business Configuration Set	BC Set		

## 2 Process-Controlled Workflow

Along with the user interface conversion and the underlying PDO layer, development took the opportunity to consolidate the existing approval functionality in SRM.

The following picture outlines the various system parts that are involved:



## 2.1 SAP Business Workflow in SAP SRM

SRM approval process framework uses SAP Business Workflow.

The SAP Business Workflow is a generic workflow engine. It provides:

- A modeling environment ('workflow designer')
- A stable transactional runtime
- A textual/ graphical log analyzer
- Several guarding monitoring tools

While its generic nature is the key to its success across the SAP Business Suite and other consumers, there are more requirements that modern business applications have to meet on top of this.

While the SAP BWF can create and list the execution of a modeled workflow in all technical details, it cannot create an excerpt containing only business-relevant information. In addition to this, a preview of the triggered workflow is often requested by the customer in order to explain the executed process to the user. The more complex the process, the more this feature is wanted. But review for a given business context (document) is generally not available at the moment. And even if it were, there would be a further step required to provide such a preview on the business-only level. For a generic tool like SAP Business Workflow, it is quite hard to generate such a preview without knowing the semantics of the implemented process. This is where a business application must provide additional functions.

For this reason, SAP SRM has its own framework to handle approval processes that have very specific customer requirements.

Up to SAP SRM 5.0, the requirements center on providing:

- A process preview
- An auditable process log
- Ad-hoc agents
- Ad-hoc reviewers
- Overall flexibility in the process

For SAP SRM 6.0, these requirements are still valid, but additional ones have been added:

- Provide a configuration-only setup (for standard approval processes)
- Rule-based insertion of reviewers.
- The ability to integrate completion and approval workflows.

As well as functional requirements on the business level for SAP SRM 6.0, there were also functional requirements on the technical level. Up to SAP SRM 5.0, the internal framework had never been significantly updated, while the requirements have been vastly increasing. This led to a decision to implement a new framework. There was need to:

- Improve the system in terms of stability (concurrency, alignment with SAP Business Workflow, ad hoc agents)
- Improve the area of archiving
- Integrate with UI conversion to Web Dynpro
- Provide a future-oriented core

The new framework still relies on the SAP Business Workflow as an execution engine for the defined workflows. However, in contrast to the various different templates which existed previously (around 20), the new framework is now based on a very generic set of templates.

## 2.2 Business Rule Framework at SRM

Process-controlled workflow uses Business Rule Framework.

The Business Rule Framework (BRF) is provided by SAP NetWeaver. It is a tool which is capable of evaluating business rules. Like the SAP Business Workflow, BRF is a generic framework which does not provide any business logic on its own. Instead, it evaluates complex logical expressions (rules) to a single value. This can be either a Boolean result (true or false) or any other non-multi-value, e.g. a string.

Any other SAP or customer component can make use of the BRF for a variety of purposes. It is not limited to the way SRM approval process framework utilizes it.

The strengths and weaknesses of the framework are, to some extent, related. While it can evaluate any (error-free) expression, it does not provide business content or business logic by itself. This has to come from the business application. So, if the user wants to create rules to be evaluated, they must define all the atomic expressions that are required in that context.

Within SAP SRM 6.0, the BRF is used to control specific aspects of the standards implementation for approval processes. It is not a workflow or a workflow engine. The BRF, with its events and expressions, is used for the following:

- To evaluate process schemas
- To evaluate process levels
- To define defaults for ad hoc process levels
- To determine whether the contact person for a document receives a work item when the approval process is finished
- To define rules based on which reviewers are added to the approval process

Access to document attributes is typically required for evaluation of these BRF events. Information about the current approval process is also sometimes necessary. In SAP SRM 6.0, all major attributes which are available at the BOR instance of a document can be accessed via BRF expression (shipped in the standard). See also the appendix. If required, IBUs, partners and customers can create their own expressions and use them in the above context.

## **2.3 SAP NetWeaver Portal and UWL**

The work items from SAP Business Workflow are delivered to the SAP Business Workplace Inbox. This is also the case in the SRM context. In previous releases, SRM was strongly related to the SAP Business Workplace Inbox, although it provided a modified version with own UI and feature set.

With SAP SRM 6.0, the application has completely moved into the SAP NetWeaver Portal. With this shift, the SRM-specific inbox has been abandoned and the functionality is now provided by the Universal Worklist (UWL). This is the standard means of the SAP NetWeaver Portal. The provided Business Packages include a special configuration for the UWL which allows SRM to provide specifically tailored functionality. For example, using the short cut decision feature, approvers can decide on the entire document without viewing the details screens of a document.

### **2.3.1 Notifications in the UWL**

Out of the box the UWL supports work items created by SAP Business Workflow, alerts (and a special type of notifications created by Knowledge Management system). In addition to work items and alerts, SAP SRM also creates standard notifications (for example, notifications resulting from deadline monitoring of a work item). To be able to access these in the UWL see SAP Note 1051787.

## 2.4 Business Objects Using the Approval Process Framework

As previously mentioned, the approval process framework exists to fulfill several specific requirements. These include previewing of the process, adding an ad hoc agent, adding a reviewer, and providing configurable document owner interaction. The adoption of the framework is not necessarily required to run a workflow in SAP SRM.

All major business objects in SAP SRM 6.0 adopt the approval process framework. These include:

- Shopping cart
- Request for quotation (RFQ)
- Quotation
- Contract
- Purchase order
- Confirmation
- Invoice

The following application areas have been updated to consider the new framework:

- Workload (re-)distribution
- Mass changes of contract
- Selective release (Procurement for Public Sector/Public Sector)

There are other objects in the system whose approval is not driven by the framework but continue to use plain SAP Business Workflow templates:

- Purchase order response
- Procurement card
- Business partner

## 2.5 Approval Processes in SAP SRM

All approval processes in SRM are hierarchical. This is also the case for documents backed by the approval process framework. Before a document (for example, a shopping cart) can reach the next process level, a decision has to be made for all items of the document. You cannot pass items independently through the approval process.

### 2.5.1 Responsibility, Decision Sets and Decision Types

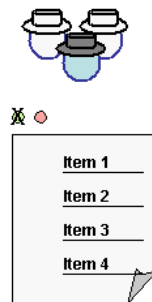
#### 2.5.1.1 Decision Sets

A decision set is a group of items of a purchasing document in an approval process. If the items of a purchasing document belong to different areas of responsibility, for instance, different product categories, the system can create the corresponding number of disjoint decision sets. For each decision set, one workflow work item is created.

Approval Decision Types:

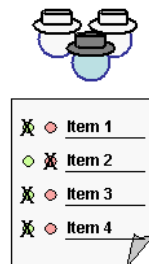
##### 2.5.1.1.1 Decision for Entire Document [Type 1]

Decision for entire document: The document is approved by a single agent as a single, indivisible unit of decision.



##### 2.5.1.1.2 Item-Based Decisions for Entire Document [Type 2]

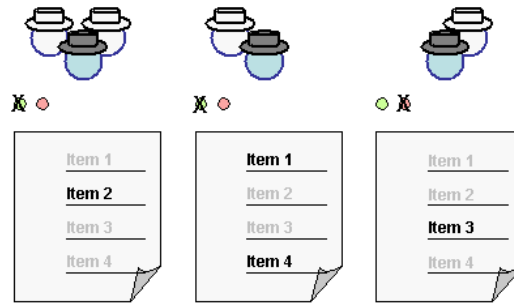
Item-based decision for entire document: The document is approved by a single agent. The decision is made individually for each item of the document.



##### 2.5.1.1.3 Overall Decision for Partial Document [Type 3]

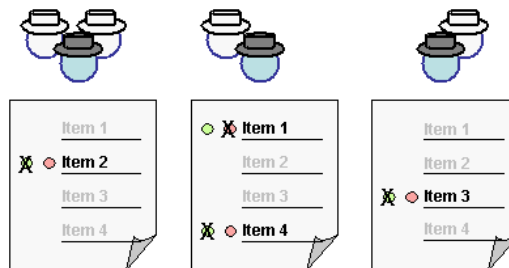
Overall decision for partial document: The items of the document are grouped into decision sets. Each decision set is approved by a separate agent. The decision applies to the entire decision set.





#### 2.5.1.1.4 Item-Based Decision for Partial Document [Type 4]

Item-based decision for partial document: The items of the document are grouped into decision sets. Each decision set is approved by a separate agent. The decision is made individually for each item of the decision set.



For each level of the decision, the system determines:

- The process level type (approval, approval with completion)
- The assignment of document or items to responsibility areas
- The agents, based on the responsibility areas

## 2.5.2 Fundamental Paradigms

There are several fundamental aspects of SAP SRM which a user must be familiar with.

*When a document is handed over for approval, the SAP Business Workflow owns and controls the process.*

Any other operation on the underlying data may potentially endanger either the consistency of the data or of the process itself.

*Without a work item, you cannot edit or approve a document.*

*Users must never approve their own document*

From a business perspective, this appears to be obvious. If the user has the necessary authorization, then no approval is required. If not, then they require approval by authorized persons. Even if acting as a substitute, users must not approve their own documents.

*A document must not be changeable if the responsibility for approval is split.*

The system cannot prevent approvers from changing content they are not responsible for. Approvers could also change the document so that they would no longer be responsible for the content, but could still approve it. This scenario would mean that the stability of the process could not be guaranteed.

*Approval Levels 'with Completion' always precede Approval 'only' levels.*

## 2.6 The Approval Process in Detail

The framework has been designed to allow various forms of flexibility. For the standard product release, a specific implementation is available to support a configuration-only setup for the 80:20 customer situations.

The approach in the standard release is based on a set of generic templates created with the SAP Business Workflow Design tool. These templates reflect the idea of a hierarchical approval process for all business objects in SRM. In SRM, the templates are backed by a generic process interface. The implementation of this process interface is connected to the BRF.

The framework was implemented using ABAP Objects and the utilizing concept of persistent classes.

They are designed to pass process control to and from SAP SRM. They implement the hierarchical approval idea (based on a generic loop), including Document Owner and Approver interaction. They create work items and then monitor their deadlines.

## 2.7 Status and Control

### 2.7.1 Process Schema Evaluation

The workflow is controlled by process schemas which are determined at runtime, and which define the necessary process levels. When the approval process for a document starts, the SAP Business Workflow starts a new instance of the main workflow template WS40000014, which is responsible for the overall process control. The template works through the process levels defined in the process schema.

The subworkflow template WS40000015 is responsible for process level evaluation. It determines whether the process schema contains further process levels to be carried out. The process level evaluation runs in loop cycles and is repeated each time a process level has been finished, provided that no level restart has been requested. A new process level is started only if a new valid level is determined. The process levels are derived from the process schemas defined in Customizing.

You can find these process schemas in Customizing for SAP Supplier Relationship Management under:

▸SRM Server ▸Cross-Application Basic Settings ▸Business Workflow ▸Process-Controlled Workflow ▸SRM Workflow Settings ▸Business Process Configuration ▸Define Process Levels ◀.

The first step in this Customizing activity is to provide an evaluation ID for each business object. This evaluation ID is a BRF event, which contains a BRF expression. The BRF expression is evaluated at runtime and returns the valid process schema. This expression can contain a function module, a BAdI, a formula or even a constant, if only one process schema for a business object exists.

*Example:* The approval process is different depending on the product categories of the items in the shopping cart. This is implemented using different process schemas for each approval process. The determination of the process schema is done in a function module called by the BRF expression assigned to the evaluation ID of the shopping cart. The function module analyzes the product categories in the shopping cart and derives the appropriate process schema.

To assign process levels, first choose a business object, and then select a process schema (folder "Process Schema Definition"). For a selected business object and process schema, the process levels are defined in the step "Process Level Configuration".

The process schema defines the possible process levels. For each process schema, the process levels are configured in the same customizing step. The following data is required:

- **No.:** Numbering of the process levels
- **Level Type:** There are 3 possible entries. Approval with completion, Approval and Automatic. Within a completion step the document can be changed by the approver if the 'Changeable' flag is set. Within the approval phase, the document can only be approved or rejected but not changed. It is not possible to mix the two phases of completion and approval. The completion phase is always before the approval phase. The automatic step is a final means of validating the approval process, and to verify that at least one step was returned without any error (example: event 0EV999). Depending on the task ID assigned to the automatic step, the document will then be either approved or rejected.

- **Resp. Resolver Name:** Assigned filter value to determine the BAdI implementation. The allowed values are customized:

►SRM Server ►Cross-Application Basic Settings ► Business Workflow ►Process-Controlled Workflow ►Business Process Configuration ►Define Filter Values for BAdI “Define Agents”◄.

It is possible to define new values for the Resp. Resolver Name (for example Z\_COST\_CENTER\_RESP), which can be used later as filter values to identify a customer implementation for a BAdI.

- **Resp. Resolver Parameter:** Specifies the return value(s) of the BAdI filter. This parameter specifies an additional parameter for a process level, which can be used for evaluation in the BAdI methods. Whether this parameter is used or not depends on the specific BAdI implementation. An example would be the implementation for the filter value RR\_ROLE (see chapter [Documentation of implementing class for RR\\_ROLE](#)). The parameter contains in this example an activity group for which all assigned users are returned as agents.
- **Eval. ID:** BRF event to determine if the process level step is executed or not. The event is evaluated at runtime. The expression assigned to the event can contain a function module, a BAdI or a formula. If the result of the expression is 'FALSE', the step is executed.
- **Task ID:** Workflow dialog task assigned to this step. The task given here contains basically the text for the work item the agent receives. The ID of the task is evaluated at runtime. It is possible to assign a newly created task here, but the new task must be incorporated into the UWL; that is, XML files have to be adjusted for the new task IDs and the associated actions. The existing dialog tasks are assigned to package /SAPSRM/WF\_CFG. They can be found using the object navigator (transaction SE80) for this package under *Business Engineering -> Workflow Standard Tasks*.
- **Decision Type:** Select one of the defined decision types.
- **Changeable checkbox:** If this checkbox is selected for a completion step, you can change the document in this step

*Example:* Consider an approval process with several approval levels depending on the value of the shopping cart (n-step spending limit approval). All managers in the organizational hierarchy have to approve the shopping cart until a level is reached for which the total value of the shopping cart no longer exceeds the approval limit of the responsible manager. In addition to the automatic approval step, only one additional step in the process schema is necessary:

- Level type: Approval
- Resp. Resolver Name: RR\_SPENDING\_LIMIT\_APPROVER
- Eval. ID: 0EV\_SC\_SL2: The assigned expression checks the approval limit of the approver of the previous process level. As long as this limit is smaller than the total value of the shopping cart, a new process level is executed.
- Task ID: 40007953 (SRM Shopping Cart Approval 1)

The design time version of the process schema is used as a template for the runtime version. The runtime version can include ad hoc agents and additional process levels. These are added using a BAdI described in the next chapter.

## 2.7.2 Approver Determination

During the approval process, a document can be split into decision sets (depending on the chosen decision type). The created decision sets are then handled in parallel. For each decision set, a new instance of the workflow template WS40000016 (approval) or WS40000017 (approval with completion) starts. In order to execute an approval step, the appropriate agents who receive a work item must be determined. This depends on the current decision set, since the context for each decision set might be different. For example, decision sets can be built on the basis of assigned cost centers.

The responsible agents are determined in three steps:

1. An area of responsibility is assigned to the items. All items with the same area of responsibility form a decision set. An area of responsibility can be, for example, a department, a product category or a cost center. For each decision set, a new sub-workflow is started.
2. For each area of responsibility, the agents for the approval tasks are determined. All agents found in this step get a work item. If the area of responsibility is a department, for example, all employees belonging to that department could be returned as agents. If the area of responsibility is a cost center, the person responsible for the cost center could be returned as agent.
3. If no agent was found for an area of responsibility, fallback agents can be provided to avoid the workflow getting stuck.

The decision sets are determined in task TS40007938 - in the workflow template WS40000015, the n level approval loop.

The assignment of agents is done in task TS40007943, which is called in both subworkflow templates, the template for approval (WS40000016), and the template for approval with completion (WS40000017). For the agent determination, a rule (40000138) assigned to the task TS40007943 has been created. This rule calls the function module /SAPSRM/WF\_CORE\_RESP\_RESOLVER, which determines the responsible agents for the actual decision set.

To ensure the flexibility of assigning agents, the actual strategy to find the correct agents is implemented by BAdIs. The BAdIs are filter-dependent. This implies that you can have several BAdI implementations for the same enhancement implementation. At runtime, the system decides, using the filter values, which implementation is actually. The evaluation of the filter values has to be unique; that is, only one active BAdI implementation is allowed for a given filter. If more than one implementation is found at runtime, the BAdI is deactivated. The filter values associated with a BAdI are defined in the Object Navigator, along with the implementing class for the BAdI.

The implementing class for the BAdI definition /SAPSRM/BD\_WF\_RESP\_RESOLVER must be provided in Customizing:

►SRM Server ►Business Add-Ins (BAdIs) ►Business Workflow ►Process-Controlled Workflow ►Define Agents◄.

Here, the enhancement implementation and the BAdI Implementation must be provided. In the SRM standard, the following enhancement implementations are defined for each business object type:

- Shopping cart: /SAPSRM/BD\_WF\_AGENTS\_SC
- Purchase order: /SAPSRM/BD\_WF\_AGENTS\_PO
- Purchase contract: /SAPSRM/BD\_WF\_AGENTS\_CTR
- Confirmation: /SAPSRM/BD\_WF\_AGENTS\_CONF
- Invoice: /SAPSRM/BD\_WF\_AGENTS\_INV
- Request for quotation /RFQ): /SAPSRM/BD\_WF\_AGENTS\_RFQ
- Supplier quotation: /SAPSRM/BD\_WF\_AGENTS\_QTE
- In addition, there is an internal enhancement implementation: /SAPSRM/BD\_WF\_AGENTS

For each of these enhancement implementations, several BAdI implementations for the determination of agents exist. An implementation is assigned to a process level at runtime through the process level configuration (see chapter [Process schema evaluation](#)) using the following level parameters as filter values:

- Document type of purchasing document (filter value always required)
- Responsibility Resolver Name (filter value always required)
- Responsibility Resolver Parameter: The implementations for RR\_EMPLOYEE and RR\_ROLE use this parameter as required input. Whether a value is needed or not depends on the actual implementing class.

The BAdI provides an interface - /SAPSRM/IF\_EX\_WF\_RESP\_RESOLVER - which must be implemented by the enhancement implementation. The interface provides five methods which must be used to determine the agents for the current step of the process:

- /SAPSRM/IF\_EX\_WF\_RESP\_RESOLVER~GET\_AREA\_TO\_ITEM\_MAP: This method assigns areas of responsibility to the items in a document. This method is called in the approval workflow when the document is split into decision sets (Task TS40007938 in workflow template WS40000015).
- /SAPSRM/IF\_EX\_WF\_RESP\_RESOLVER~GET\_APPROVERS\_BY\_AREA\_GUID: This method determines the agents for a given area of responsibility for each decision set in the document. This method is called in task TS40007943.
- /SAPSRM/IF\_EX\_WF\_RESP\_RESOLVER~GET\_FALLBACK\_AGENTS: This method is optional and can be used to define default agents if no others were determined in method GET\_APPROVERS\_BY\_AREA\_GUID. The default agents can approve or reject the items in a document. In addition, the work item can be forwarded to another processor.
- /SAPSRM/IF\_EX\_WF\_RESP\_RESOLVER~GET\_RESP\_RESOLVER\_PARAMETER\_F4: This method is optional, and can be used to provide an F4 help for the field RESP. RESOLVER PARAMETER in the customizing view "Process level configuration" in the IMG step "Define process levels" of the new workflow.
- /SAPSRM/IF\_EX\_WF\_RESP\_RESOLVER~CHECK\_RESP\_RESOLVER\_PARAMETER: This method is also optional, and can be used to check the user input for the field as described in the previous method. If a value is required for the field "Responsibility Resolver Parameter", implement the check here.

## 2.7.3 Reviewer Determination

### 2.7.3.1 Overview

The approval framework allows reviewers to be added to the process. The reviewers receive a work item and can add notes and attachments. The review process is carried out parallel to the approval process. The review process is also controlled by the main workflow template: WS40000014.

In the first step, a list of reviewers is determined. In order to ensure that the assignment of reviewers is flexible, the list is determined in a BAdI. If reviewers are assigned, an event is generated which starts the reviewer branch in the approval framework. In this branch for each reviewer a workflow item is generated.

In the main workflow template, a container operation (number 000382) returns the list of reviewers. The assigned method determines if a BAdI implementation is active. Just like for the agent determination, the BAdI implementations are filter dependent, meaning different implementations can be assigned depending on the process schema.

You can find these process schemas in Customizing for SAP Supplier Relationship Management under:

► *SRM Server* ► *Cross-Application Basic Settings* ► *Business Workflow* ► *Business Process Configuration* ► *Define Process Levels* ◀.

The reviewers are defined for each process schema in the step "Reviewer". The following values must be provided:

- **Rule:** Assigned filter value to determine the BAdI implementation. The allowed values are customized:  
 ► *SRM Server* ► *Cross-Application Basic Settings* ► *Business Workflow* ► *Business Process Configuration* ► *Define Filter Values for BAdI 'Define Reviewer'* ◀.
- **Rule Parameter:** Specifies the return value of the BAdI filter; that is, the value for the rule.
- **Evaluation ID:** BRF event which determines if the reviewer step is executed or not. The event is evaluated at runtime. The expression assigned to the event can contain a function module, a BAdI or a formula. The reviewer step is executed if the result of the expression is 'TRUE'.
- **Task ID:** Assigned workflow task for this step

In order to add reviewers at runtime, an enhancement spot is provided. The relevant BAdI definition is /SAPSRM/BD\_WF\_REVIEWER\_RULE. If you want to add reviewers at runtime, an implementation must be created for that BAdI. The enhancement spot is accessed with the Object Navigator (SE80) using package /SAPSRM/MWF\_CFG. Here you can create new implementations. In addition to implementing the BAdI, the implementations for the enhancement spot must be activated also in Customizing:

► *SRM Server* ► *Business Add-Ins (BAdIs)* ► *Business Workflow* ► *Process-Controlled Workflow* ► *Define Reviewers* ◀.

Here you provide the names of the implementations. The enhancement implementation for the BAdIs delivered with the Standard is: /SAPSRM/BD\_WF\_REVIEWER.

The Business Add-Ins are filter-dependent, meaning that you can have several BAdI implementations for the same enhancement implementation. At runtime, the system uses the filter values and decides which implementation is called. The filter values are defined in the Object Navigator along with the implementing class for the BAdI.

Here, the enhancement implementation and the BAdI Implementation must be provided. In the SRM standard only one enhancement implementation is defined for reviewers: /SAPSRM/BD\_WF\_REVIEWER. This enhancement implementation has several BAdI implementations. An implementation is assigned at runtime through the reviewer configuration (see beginning of this paragraph) using the following level parameters as filter values:

- Document type of purchasing document (filter value always required)
- Rule (filter value always required)

- Rule Parameter: the return value of the BAdI filter is not required for all rules

The BAdIs are defined for single use only. In other words, each filter value can only be assigned to one implementation, that is, at runtime, only one implementation should be found. Otherwise, the BAdI is deactivated.

The BAdI provides an interface - /SAPSRM/IF\_EX\_WF\_REVIEWER\_RULE - which must be implemented by the enhancement implementation. The interface provides three methods which determine the reviewers for the current process step:

- /SAPSRM/IF\_EX\_WF\_REVIEWER\_RULE~GET\_REVIEWER\_BY\_DOCUMENT: This method determines the agents for a given area of responsibility for each decision set in the document. This method is called in container operation 000382.
- /SAPSRM/IF\_EX\_WF\_REVIEWER\_RULE~GET\_REVIEWER\_RULE\_PARAM\_F4: This method is optional, and can be used to provide F4 help for the field RULE PARAMETER. This field is found in the customizing view "REVIEWER" in the IMG step "Define process levels" of the new workflow.
- /SAPSRM/IF\_EX\_WF\_REVIEWER\_RULE~CHECK\_REVIEWER\_RULE\_PARAM: This method is also optional. It checks the user input for the field as described in the previous method. If a value is required for the field "Rule Parameter", implement the check in this method.



## 3 Configuring the System for Process-Controlled Workflow and Extending the Framework

### 3.1 Configuration Process

The workflow framework switch is done in SAP Solution Manager under Configuration → SAP SRM 7.0 → Basic Settings for SAP SRM → Cross-Application Settings → Business Workflow → Select Workflow Framework. The process-controlled workflow framework is configured by system default (on delivery from SAP). However, if the application-controlled workflow framework was subsequently configured, and you later want to use process-controlled workflows, you must switch back to the process-controlled workflow framework.

### 3.2 Tips for Approval Process Framework Configuration

#### 3.2.1 Transporting BRF Content

The most common method of transporting BRF content to the target systems is to use SAP transport and report 'BRF\_TRANSPORT\_SIMPLE'.

The transportation process can be improved by using BC Sets. These are SAP standard technology which has the following benefits:

- Documentation and analysis of the customizing settings
- Simplified implementation at large companies
- Can be created by customers themselves
- Application of BC Sets is traced in the system, providing success and error reports
- You can easily compare the current system setup against a reference BC Set (local, remote, cross client).

For more information on BC Sets, see the related documentation for SAP NetWeaver 7.0 in SAP Help Portal at <http://help.sap.com> ► *SAP NetWeaver Library* ► *SAP NetWeaver by Key Capability* ► *Solution Life Cycle Management by Key Capability* ► *Customizing* ► *Business Configuration Sets (BC-CUS)*.

#### 3.2.2 Features

BC Sets are available for the following business objects:

- Shopping cart
- Purchase order
- Confirmation
- Invoice
- Contract
- Quote
- RFx

For each of these business objects, there is one BC Set to customize a process without approval and one to customize a process with one-step manager approval. When you activate a BC Set, the system automatically creates a process schema and configures all necessary process levels. For each business object, you can activate one BC Set only.

##### 3.2.2.1 BC Sets for Processes Without Approval

The process schema configured when you activate a BC Set for processes without approval contains only one process level, in which the items of the document are approved automatically.

In these processes, there are no reviewers, no defaults for ad hoc approvers, and the contact person does not receive a work item for acceptance when the approval process is finished.

If you use only BC Sets for processes without approval and you have not configured any other process schemas, SAP Business Workflow is not involved and you do not have to perform any Customizing activities for SAP Business Workflow.

BC Sets that configure processes without approval use the naming convention /SAPSRM/C\_<BO>\_600\_000\_SP04.

For enabling processes without approval event type linkage in transaction SWE2 of class /SAPSRM/CL\_WF\_PDO, Event: *READY\_FOR\_WORKFLOW* to workflow *WS40000014* has to be active.

### 3.2.2.2 BC Sets for Processes with One-Step Manager Approval

The process schema configured when you activate a BC Set for processes with one-step manager approval contains one process level for approval by the document creator's manager, and one for automatic decision. If the system cannot determine a responsible agent, the document is rejected automatically.

In these processes, there are no reviewers, no defaults for ad hoc approvers, and the contact person does not receive a work item for acceptance when the approval process is finished.

BC Sets that configure processes with one-step manager approval use the naming convention /SAPSRM/C\_<BO>\_600\_001\_SP04.

### 3.2.2.3 Available BC Sets

The following BC Sets are available:

Business Object	BC Set	Process Schema	Description
Shopping cart	/SAPSRM/C_SC_600_000_SP04	9C_SC_600_000_SP04	Without approval
	/SAPSRM/C_SC_600_001_SP04	9C_SC_600_001_SP04	One-step manager approval
Purchase order	/SAPSRM/C_PO_600_000_SP04	9C_PO_600_000_SP04	Without approval
	/SAPSRM/C_PO_600_001_SP04	9C_PO_600_001_SP04	One-step manager approval
Confirmation	/SAPSRM/C_CO_600_000_SP04	9C_CO_600_000_SP04	Without approval
	/SAPSRM/C_CO_600_001_SP04	9C_CO_600_001_SP04	One-step manager approval
Invoice	/SAPSRM/C_IN_600_000_SP04	9C_IN_600_000_SP04	Without approval
	/SAPSRM/C_IN_600_001_SP04	9C_IN_600_001_SP04	One-step manager approval
Contract	/SAPSRM/C_CT_600_000_SP04	9C_CT_600_000_SP04	Without approval
	/SAPSRM/C_CT_600_001_SP04	9C_CT_600_001_SP04	One-step manager approval
Quote	/SAPSRM/C_QT_600_000_SP04	9C_QT_600_000_SP04	Without approval
	/SAPSRM/C_QT_600_001_SP04	9C_QT_600_001_SP04	One-step manager approval
RFx	/SAPSRM/C_RQ_600_000_SP04	9C_RQ_600_000_SP04	Without approval
	/SAPSRM/C_RQ_600_001_SP04	9C_RQ_600_001_SP04	One-step manager approval

## 3.2.3 Activities

To activate BC Sets, use transaction SCPR20.

### 3.3 Process Level Configuration

In order to add process levels at runtime, an enhancement spot is provided. The relevant BAdI definition is /SAPSRM/BD\_WF\_PROCESS\_CONFIG. The BAdI is called each time the workflow evaluates the process levels, and checks whether or not to start a new process level. This is done in the subworkflow WS40000015 (N level approval loop). When an active implementation of the BAdI is found, it calls the method CREATE\_PROCESS\_LEVEL\_CONFIG, which creates a new process level if required.

In order to add new process levels at runtime, create an implementation for that particular BAdI. Access the enhancement spot with the Object Navigator (SE80) using package /SAPSRM/WF\_CFG. Here, you can create new implementations. In addition to implementing the BAdI, the implementations for the enhancement spot must be activated also in Customizing:

►SRM Server ►Business Add-Ins (BAdIs) ►Business Workflow ►Process-Controlled Workflow ►Process Level Configuration◄.

Provide the names of the implementations here.

The Business Add-Ins are filter-dependent, meaning you can have several BAdI implementations for the same enhancement implementation. At runtime, the system decides using the filter values which implementation is actually called. The filter values are defined in the Object Navigator along with the implementing class for the BAdI. The filter values are the document type, and the Resp. Resolver Name of the actual process level evaluated at runtime.

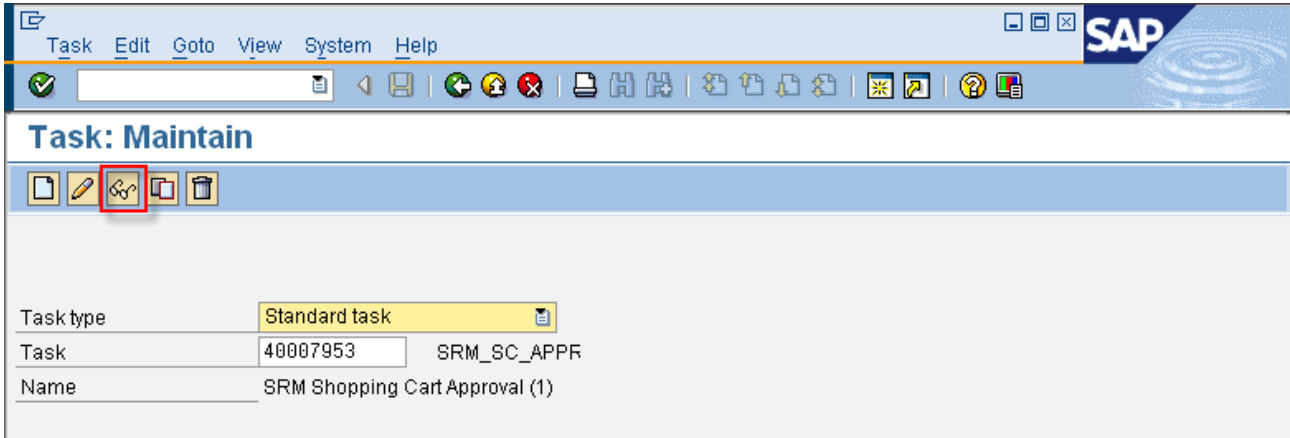
In the standard system, two BAdI implementations are active:

- /SAPSRM/BD\_WF\_CONFIG\_SC\_SL - Shopping Cart: Dynamic Process Schema for Spending Limit
- /SAPSRM/BD\_WF\_CONFIG\_PO\_SL - Purchase Order: Dynamic Process Schema for Spending Limit

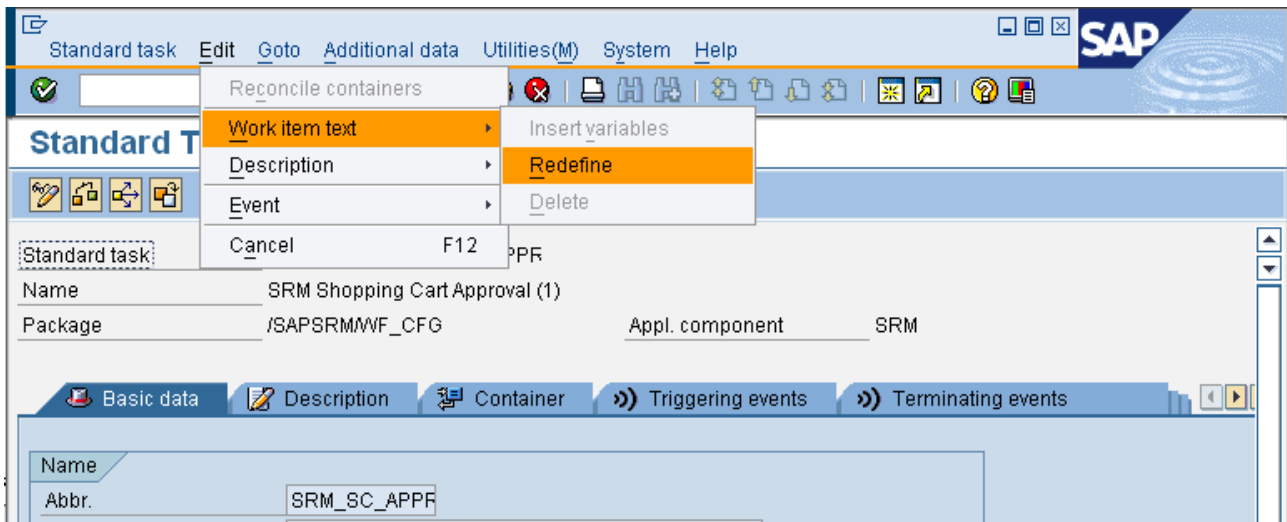
### 3.4 Customer-Specific Work Items

#### 3.4.1 Adjust Work Item Text of SRM Standard Tasks

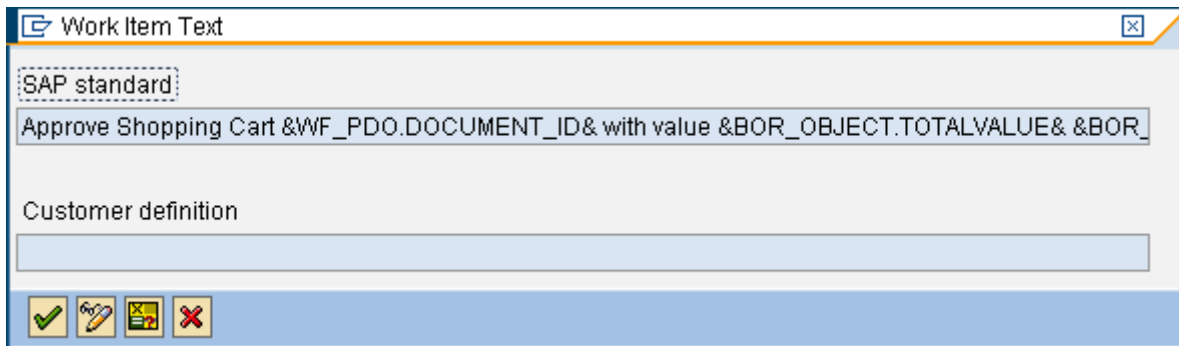
You have the option to adjust the work item text. To do so, open the SAP Business Workflow related transaction PFTC. Enter the ID of the task that you want to adjust:



Display the task. From the menu select Edit -> Work item text -> Redefine:



You can now enter the required text:



#### 3.4.2 Adjust Task Description of SRM Standard Tasks

This cannot be achieved without modification or creation of your own task.

### 3.4.3 Customer BRF Expressions

#### 3.4.4 Programming Guidelines for a Function Module Used in the BRF

If the start condition for a workflow or the selection of a certain process schema is too complex to be defined with BRF expressions alone you can delegate the task to a function module. Create this function module in the customer namespace and assign it to a BRF expression (transaction BRF) using the expression type OCF001 of application class SRM\_WF. The data access is of type 'Function module'. The name of the function module is entered in the field 'Access FM'.

The interface of the function module must be as follows:

##### **Import parameters:**

IT_EXPRESSIONS	TYPE	SBRF260A_T
IO_EVENT	TYPE REF TO	IF_EVENT_BRF
IO_EXPRESSION	TYPE REF TO	IF_EXPRESSION_BRF

None of the import parameters are optional. All are passed by reference.

##### **Export parameters:**

EV_VALUE	TYPE	BRF_RESULT_VALUE
EV_TYPE	TYPE	BRF_RESULT_TYPE
EV_LENGTH	TYPE	BRF_RESULT_LENGTH
EV_CURRENCY	TYPE	BRF_CURRENCY
EV_OUTPUT_LENGTH	TYPE	BRF_RESULT_OUTPUT_LENGTH
EV_DECIMALS	TYPE	BRF_RESULT_DECIMALS
EV_DATA_MISSING	TYPE	BRF_DATA_MISSING

All export parameters are passed by reference.

The import parameters allow the properties of a document currently in the process level to be accessed. A typical implementation to determine a process schema looks like:

```

DATA lo_wf_brf_event      TYPE REF TO /sapsrm/cl_wf_brf_event.
DATA lo_context_provi der TYPE REF TO /sapsrm/cl_wf_context_provi der.
DATA lv_document_gui d   TYPE /sapsrm/wf_document_gui d.
DATA lv_document_type    TYPE /sapsrm/wf_document_type.

DATA lo_pdo_sc           TYPE REF TO /sapsrm/if_pdo_bo_sc.

DATA lt_document_owner  TYPE /sapsrm/t_wf_agent_i d.
DATA lr_document_owner  TYPE REF TO /sapsrm/wf_agent_i d.

DATA ls_header          TYPE bbp_pds_sc_header_d.
DATA lt_header_gui d    TYPE bbpt_gui d.
DATA ls_header_gui d    TYPE bbp_gui d_tab.
DATA lt_item_gui d      TYPE /sapsrm/t_pdo_hi er_gui d_l i st.
DATA lr_item_gui d      TYPE REF TO /sapsrm/s_pdo_hi er_gui d_l i st.

DATA lt_item            TYPE bbpt_pd_sc_i tem_d.
DATA lt_item_i          TYPE bbpt_pd_sc_i tem_d.
DATA lr_item            TYPE REF TO bbp_pds_sc_i tem_d.

*=====
* Preset return parameters
*=====
ev_data_mi ssi ng = 'X'.
ev_val ue         = 0.
ev_type          = 'C'.

```

```

ev_length          = 32.
CLEAR ev_currency.
ev_output_length  = 32.
ev_decimals       = 0.

```

```

*=====
* Get purchasing document
*=====

* get event object
lo_wf_brf_event ?= io_event.

* get context container from BRF event
lo_context_provider = lo_wf_brf_event->get_context_provider( ).

* get document
CALL METHOD lo_context_provider->get_document
IMPORTING
  ev_document_guid = lv_document_guid
  ev_document_type = lv_document_type.

* get instance
lo_wf_pdo ?= /sapsrm/cl_wf_pdo_impl_factory=>get_instance(
  lv_document_guid = lv_document_guid
  lv_document_type = lv_document_type
).

*=====
* Get Document Owner
*=====
lt_document_owner = io_wf_pdo->get_document_owner( ).
READ TABLE lt_document_owner REFERENCE INTO lr_document_owner INDEX 1.
ASSERT sy-subrc = 0.

*=====
* Specialize to Shopping Cart
*=====
IF io_wf_pdo->get_document_type( ) NE /sapsrm/if_pdo_obj_types_c=>gc_pdo_shop.
  ev_data_mising = 'X'.
  EXIT.
ENDIF.

lo_pdo_sc ?= io_wf_pdo->get_pdo( ).

*=====
* Get Shopping Cart Details for Scheme Evaluation
*=====
TRY.

  CALL METHOD lo_pdo_sc->get_header_detail
  IMPORTING
    es_header = ls_header.

  ls_sc_total_value-value = ls_header-total_value.
  ls_sc_total_value-currency = ls_header-currency.

  ls_header_guid-guid = ls_header-guid.
  APPEND ls_header_guid TO lt_header_guid.

  CALL METHOD lo_pdo_sc->/sapsrm/if_pdo_base-get_item_list
  EXPORTING
    it_parent_guid = lt_header_guid
  IMPORTING
    et_item_guid = lt_item_guid.

  CATCH /sapsrm/cx_pdo_wrong_bus_type

```

```

    /sapsrm/cx_pdo_pd_read_error
    /sapsrm/cx_pdo_lock_failed
    /sapsrm/cx_pdo_no_authorization
    /sapsrm/cx_pdo_parameter_error
    /sapsrm/cx_pdo_status_error
    /sapsrm/cx_pdo_incons_user
    /sapsrm/cx_pdo_abort
    /sapsrm/cx_pdo_error.
ev_data_mising = 'X'.
EXIT.

ENDTRY.

CLEAR lt_item.

LOOP AT lt_item_guid REFERENCE INTO lr_item_guid WHERE leaf = abap_true.

    TRY.

        CALL METHOD lo_pdo_sc->get_item_detail
            EXPORTING
                iv_item_guid = lr_item_guid->guid
            IMPORTING
                et_item       = lt_item_i .
**                et_account = lt_account.

        APPEND LINES OF lt_item_i TO lt_item.

    CATCH /sapsrm/cx_pdo_no_authorization
          /sapsrm/cx_pdo_abort .
        ev_data_mising = 'X'.
        EXIT.

    ENDTRY.

ENDLOOP.

*-----
* Implement Logic to determine Scheme
* You can use e.g. the static methods of class /SAPSRM/CL_WF_CONFIG_USER
* to access some information from the org. management
*-----

ev_value = 'YOUR_PROCESS_SCHEME'.
CLEAR ev_data_mising.

ENDFUNCTION.
```

## 3.5 Customer Responsibility Resolver Strategies

### 3.5.1 Interface /SAPSRM/IF\_EX\_WF\_RESP\_RESOLVER

#### 3.5.1.1 Method GET\_AREA\_TO\_ITEM\_MAP

The items are assigned to areas of responsibilities. The available input parameters are:

- Document type and ID
- Attributes of the current process level, such as responsibility resolver, parameter or decision type

For more information on how to access information about the purchasing document, see chapter [Access purchasing documents](#).

The return table contains the list of item GUIDs and the associated GUID of the area of responsibility. If all items are assigned to the same area of responsibility, that is, the document is not split into decision sets, the table may contain only one line with the area GUID and the initial item GUID; that is, '00000000000000000000000000000000'. If more than one area of responsibility is found, the association of items to responsibility areas in the return table is used to create the decision sets.

*Example:* The document is split into decision sets, one for each purchasing group in the document. In this case, the leading objects for the areas of responsibility are the purchasing groups, and the items are assigned to the area GUIDs created for each purchasing group in the document.

The standard BADI implementations that will be described later use a dedicated subclass of the base class /SAPSRM/CL\_WF\_AREA both to create the correct area context and to handle the persistency. Since the information cannot be passed through the workflow container, the information is stored in the database for later retrieval in the method GET\_APPROVERS\_BY\_AREA\_GUID.

A typical implementation looks like:

```

DATA lo_area          TYPE REF TO /sapsrm/if_wf_area.
DATA ls_area_to_item_map TYPE /sapsrm/s_wf_item_to_area.

* Input checks
IF is_document IS INITIAL.
  RAISE EXCEPTION TYPE /sapsrm/cx_wf_abort.
ENDIF.

* Get ID of leading object, e.g. the responsibility resolver name
lv_leading_object = is_current_process_level-resolver_param.

* Create responsibility area
lo_area = /sapsrm/cl_wf_area=>/sapsrm/if_wf_area~create_instance(
  iv_area_type      = 'Name of subclass of /SAPSRM/CL_WF_AREA'
  iv_leading_object_id = lv_leading_object
  ).

* All items are assigned to a single area
ls_area_to_item_map-area_guid = lo_area->get_guid( ).

* Return responsibility area GUID ... w/o item assignment (nil GUID)
ls_area_to_item_map-item_guid = /sapsrm/if_wf_process_c=>gc_nil_guid.
APPEND ls_area_to_item_map TO rt_item_to_area_map.

```

To create a responsibility area, provide a leading object ID. This could be, for example, the responsibility resolver parameter, if it is not empty for the current process level. The leading ID can also be derived from the attributes of the document items, such as the assigned purchasing group. The leading ID, together with the newly created area GUID, is then saved to the database table /SAPSRM/D\_WF\_012.



The ID of the leading object is used in the method GET\_APPROVERS\_BY\_AREA\_GUID to derive the actual agents for the approval step.

The subclass creates the instance and a new GUID, which must be returned in the item map together with the associated item GUID. This return table is used to define the decision sets.

### 3.5.1.2 Method GET\_APPROVERS\_BY\_AREA\_GUID

The list of all responsible agents for a given area of responsibility is returned using the table parameter RT\_APPROVER. The two fields of the table structure contains the user name and the corresponding object type 'US'. The input parameter of the method is the GUID of the responsibility area.

As described in the previous chapter, the information about the responsibility area is written to the database (table /SAPSRM/D\_WF\_012). This information must be retrieved first. For most of the standard BAdI implementations described later, a subclass of the base class /SAPSRM/CL\_WF\_AREA handles the persistency and the determination of the responsible agents. A typical implementation of the method looks as follows:

```
DATA lo_area TYPE REF TO /sapsrm/if_wf_area.
```

```
* Input checks
```

```
IF is_area IS INITIAL.  
    RETURN.  
ENDIF.
```

```
* Get responsibility area reference for given area GUID
```

```
lo_area = /sapsrm/cl_wf_area=>/sapsrm/if_wf_area-get_instance_by_guid(  
    iv_area_type = 'Name of subclass of /SAPSRM/CL_WF_AREA'  
    iv_area_guid = is_area-area_guid  
    ).
```

```
* Return all responsible users assigned to that area
```

```
rt_approver = lo_area->get_responsible_approvers( ).
```

If the area GUID is provided, a new instance of the area is created. In this step, the data is also retrieved from the database. The method GET\_RESPONSIBLE\_APPROVERS determines the list of agents. This method is not inherited from the base class /SAPSRM/CL\_WF\_AREA, but has to be redefined in each subclass. Examples of this logic are found in the documentation of the available standard BAdI implementations.

*Example:* In the example of the previous chapter, the areas of responsibility have been the purchasing groups. The area GUID passed to this method gives the purchasing group of the decision set back as leading object. All purchasers belonging to the purchasing group are then returned as responsible agents by accessing the organizational management.

### 3.5.1.3 Method GET\_FALLBACK\_AGENTS

The implementation of this method is optional. The method returns the workflow agents in case no responsible agents have been found by method GET\_APPROVERS\_BY\_AREA\_GUID using a table parameter. This means that more than one agent can be returned. The user name of the agents must be returned. The appropriate object type is 'US'.

The properties of the current purchasing document can be used to determine fallback agents. The document type and ID are passed as variables into the method. For details on accessing the properties of the purchasing document, refer to [Access purchasing documents](#).

#### 3.5.1.4 Method GET\_RESP\_RESOLVER\_PARAMETER\_F4

The implementation of this method is optional. The method can be used to display a search help to aid the user in finding a correct value for the responsibility resolver parameter. A search help can be called in the method using the function module F4IF\_FIELD\_VALUE\_REQUEST. The function module then calls the search help. It returns a table of selected values. One of these values must be mapped to the changing parameter of the method.

A new search help must be provided if none exists for the chosen parameter.

#### 3.5.1.5 Method CHECK\_RESP\_RESOLVER\_PARAMETER

The current value of the responsibility resolver parameter is passed as input value to the method. The validity of this value can be checked. If the value is invalid the exception /SAPSRM/CX\_WF\_NOT\_FOUND must be raised.

#### 3.5.1.6 Parameters

The method CREATE\_PROCESS\_LEVEL\_CONFIG has the following parameters:

- Importing IS\_DOCUMENT: Purchasing document (type and GUID) to access the current document in the method
- Importing IO\_LEVEL\_SERVICE: Process level service to create a new process level if required
- Changing CS\_PROCESS\_LEVEL: Process level attributes of the newly created process level

#### 3.5.1.7 Access Purchasing Documents

To access a purchasing document in the BAdI, first create a new instance of the document:

```
DATA lo_wf_pdo TYPE REF TO /sapsrm/if_wf_pdo.
```

```
lo_wf_pdo ?= /sapsrm/cl_wf_pdo_impl_factory=>get_instance(  
    iv_document_guid = is_document-document_guid  
    iv_document_type = is_document-document_type  
    ).
```

The interface /SAPSRM/IF\_WF\_PDO, which is implemented for all document types, offers several methods of accessing the attributes of a purchasing document. This can be used to determine if a new process level is necessary.

#### 3.5.1.8 Create a New Process Level

In order to create a new process level, the method must contain the following call to set the attributes:

```
CALL METHOD io_level_service->create_process_level  
    CHANGING  
        cs_process_level = cs_process_level.
```

The return parameter is also the return parameter of method CREATE\_PROCESS\_LEVEL\_CONFIG.

#### 3.5.1.9 Agent Determination

If you decide to determine the approvers already in this method for each process level, the approvers have to be stored in database tables for later retrieval in the BAdI for the agent finding (method GET\_APPROVERS\_BY\_AREA, see chapter [Definition of agents](#)). This is necessary because the respective information is not passed between the various workflow tasks.

To store the approvers, use the function module /SAPSRM/WF\_CORE\_ADHOC\_SAVE. This stores the approvers in the tables /SAPSRM/D\_WF\_015 and /SAPSRM/D\_WF\_016. Retrieve the information with the function module /SAPSRM/WF\_CORE\_ADHOC\_READ.

An example of how to use the function module to save approvers can be found in method /SAPSRM/IF\_EX\_WF\_PROC\_CONFIG~CREATE\_PROCESS\_LEVEL\_CONFIG for the implementing class /SAPSRM/CL\_IM\_WF\_CONFIG\_SL (see next chapter).

An example of how to use the function module to read approves can be found in method /SAPSRM/IF\_WF\_AREA~GET\_RESPONSIBLE\_APPROVERS for the implementing class /SAPSRM/CL\_WF\_AREA\_ADHOC (see chapter [Implementation - Get Users of Spending Limit Chain for Approval](#)).

## 3.6 Customer Strategies for Reviewer Determination

### 3.6.1 Interface /SAPSRM/IF\_EX\_WF\_REVIEWER\_RULE

#### 3.6.1.1 Method GET\_REVIEWER\_BY\_DOCUMENT

This method returns the list of reviewers for the document. The available input parameters are:

- Document type and ID
- Current process value of the reviewer rule parameter

To access information about the purchasing document, see chapter [Access purchasing documents](#).

The return table is a list of all reviewers. It contains the user name and the user type ('US') of each reviewer.

#### 3.6.1.2 Method GET\_REVIEWER\_RULE\_PARAM\_F4

The implementation of this method is optional. The method displays a search help to help the user find a correct value for the responsibility resolver parameter. A search help is called in the method using the function module F4IF\_FIELD\_VALUE\_REQUEST. The function module then calls the search help. It returns a table of selected values. One of these values must be mapped to the changing parameter of the method.

A new search help must be provided if none exists for the chosen parameter.

#### 3.6.1.3 Method CHECK\_REVIEWER\_RULE\_PARAM

The current value of the reviewer rule parameter is passed as input value to the method. The validity of this value can be checked. If the value is invalid the exception /SAPSRM/CX\_WF\_NOT\_FOUND has to be raised.

## 4 Cheat Sheets – Process-Controlled Workflow

### 4.1 Extensive Example

The implementation of a new approval process involves customizing steps, the usage and/or creation of BRF events and expressions as well as programming tasks, if the standard BAdI implementations do not cover the requirements.

Before starting the configuration, the following decisions must be made:

- How to evaluate the correct process scheme. If there is more than one process scheme for a business object type, and the determination is complex, it might be necessary to delegate the decision to a function module called from the BRF ([Programming guidelines for a function module used in the BRF](#)).
- Dynamic versus static determination of process levels. You can use a BAdI to add process levels dynamically at runtime ([Process level configuration](#)), if the number of levels cannot be defined statically at design time. It is important to note that the BRF event assigned to each defined level is always evaluated at runtime. Because of this it is possible in both cases to either execute or skip a defined process level for any given document.
- Is it possible to use the existing BAdI implementations for:
  - Process level configuration (if necessary)
  - Agent determination
  - Reviewer determination

The documentation of the existing implementations can be found in the chapters [Process level configuration](#), [Standard BAdI implementations](#) and [Standard implementations of /SAPSRM/BD\\_WF\\_REVIEWER\\_RULE](#).

In the following, it is assumed that the complete process is configured from the beginning. If BAdI implementations or BRF expressions already exist that meet your requirements, you do not have to create them and the corresponding steps can be omitted.

1. Define filter value for BAdIs for process level configuration and agent determination in Customizing:

▮ *SRM Server* ▸ *Cross-Application Basic Settings* ▸ *Business Workflow* ▸ *Process-Controlled Workflow* ▸ *Business Process Configuration* ▸ *Define Filter Values for BAdI “Define Agents”* ⏪.

All filter values used later on must be entered here. Information on the use of filter values is found in [Overview](#) and [Definition of agents](#).

2. Create the BRF event for the process scheme evaluation using transaction BRF. The event contains an expression that returns the process scheme. If more than one process scheme for a business object is necessary, a function module can be used to derive the correct process scheme.
3. Create the BRF event for each process level, which is evaluated at runtime and determines if the level is executed or not. If the existing BRF expressions contained in BRF object SRM\_WF are not adequate, new expressions can be created.
4. Define process scheme:

▮ *SRM Server* ▸ *Cross-Application Basic Settings* ▸ *Business Workflow* ▸ *Process-Controlled Workflow* ▸ *Business Process Configuration* ▸ *Define Process Levels* ⏪.

Provide the evaluation ID for the business object type. In the next step, define the process levels for each process scheme for a given business object type. Details can be found in [Process schema evaluation](#).

5. If process levels have to be added at runtime, create a new enhancement implementation for the enhancement spot /SAPSRM/BD\_WF\_PROCESS (see [Process level configuration](#)):

- Display package /SAPSRM/WF\_CFG with transaction SE80
  - Go to enhancement spot /SAPSRM/BD\_WF\_PROCESS
  - Create a new implementation with name in customer namespace (choose 'Implement' from menu for this enhancement spot)
  - Create a new BAdI implementation on tab 'Enh. Implementation Elements'. Provide a class name for the implementing class and define the filter values.
  - Implement the method /SAPSRM/IF\_EX\_WF\_PROC\_CONFIG~CREATE\_PROCESS\_LEVEL\_CONFIG (see [Programming guidelines](#)).
6. Activate the new enhancement implementation and BAdI implementation in the IMG:  
 ▮SRM Server ▸Business Add-Ins (BAdIs) ▸ Business Workflow ▸Process-Controlled Workflow ▸Configure Process Levels ⚡.
7. For agent determination, create a new enhancement implementation for the enhancement spot /SAPSRM/BD\_WF\_AGENTS (see [Agent assignment](#)):
- Display package /SAPSRM/WF\_CFG with transaction SE80
  - Go to enhancement spot /SAPSRM/BD\_WF\_AGENTS
  - Create a new implementation with name in customer namespace (choose 'Implement' from menu for this enhancement spot). Use the BAdI definition /SAPSRM/BD\_WF\_RESP\_RESOLVER
  - Create a new BAdI implementation on tab 'Enh. Implementation Elements'. Provide a class name for the implementing class and define the filter values.
  - Implement at least the methods GET\_APPROVERS\_BY\_AREA\_GUID and GET\_AREA\_TO\_ITEM\_MAP (see [Programming guidelines for the methods of interface /SAPSRM/IF\\_EX\\_WF\\_RESP\\_RESOLVER](#)).
8. Activate the new enhancement implementation and BAdI implementation in Customizing:  
 ▮SRM Server ▸Business Add-Ins (BAdIs) Business Workflow ▸Process-Controlled Workflow ▸Define Agents ⚡.

The following steps are necessary in order to define Reviewers who get work items in parallel to approvers:

1. Define filter values for reviewers:  
 ▮SRM Server ▸Cross-Application Basic Settings ▸ Business Workflow ▸Process-Controlled Workflow ▸Business Process Configuration ▸Define Filter Values for BAdI 'Define Reviewer' ⚡  
 .
  2. Create the BRF event for the reviewer evaluation using transaction BRF.
  3. Define reviewer in the IMG:  
 ▮SRM Server ▸Cross-Application Basic Settings ▸ Business Workflow ▸Process-Controlled Workflow ▸Business Process Configuration ▸Define Process Levels ⚡.
- Details can be found in [Overview](#).
4. For reviewer determination create a new enhancement implementation for the enhancement spot /SAPSRM/BD\_WF\_AGENTS (see Agent assignment):
    - Display package /SAPSRM/WF\_CFG with transaction SE80
    - Go to enhancement spot /SAPSRM/BD\_WF\_AGENTS
    - Create new implementation with name in customer namespace (choose 'Implement' from menu for this enhancement spot). Use the BAdI definition /SAPSRM/BD\_WF\_REVIEWER\_RULE.

- Create a new BAdI implementation on tab 'Enh. Implementation Elements'. Provide a class name for the implementing class and define the filter values.
5. Implement at least the method GET\_REVIEWER\_BY\_DOCUMENT (see Programming guidelines for the methods of interface /SAPSRM/IF\_EX\_WF\_REVIEWER\_RULE)
  6. Activate the new enhancement implementation and BAdI implementation in Customizing:  
▮SRM Server ▮Business Add-Ins (BAdIs) ▮ Business Workflow ▮Process-Controlled Workflow ▮  
Define Reviewers↵.

## 4.2 Upgrade

The upgrade process from application-controlled workflow to process-controlled workflow is described in *Upgrade Guide SAP Supplier Relationship Management Server 7.0 ABAP* in chapter 6.26.



## 5 Troubleshooting – Process-Controlled Workflow

### 5.1 Utilities for Troubleshooting

There are several administrative utilities provided by the SAP Business Workflow on the one hand and some SRM specific ones on the other hand.

#### 5.1.1 SRM-Specific Utilities

In order to find details about a specific document the transaction BBP\_PD can be used. The GUID of the document for example is required to identify the related workflow instance. Note that for each change version of a document, a dedicated GUID exists which is related to separate workflow instance.

##### 5.1.1.1 Document Approval Process Analysis

How to start the tool:

- Run Transaction SE80
- Select 'Package' view

Navigate to **/SAPSRM/WF\_CFG\_ANALYSIS** ► Web Dynpro Components ► **/SAPSRM/WDC\_WF\_ANCFG\_MAIN** ► Web Dynpro Applications ► **/sapsrm/wda\_wf\_anlysis\_cfg**

Use the context menu on this entry to execute the application via 'test'

Alternatively use the URL that is provided in the Administration Data of the application properties.

For example "http://<appserver>:<port>/sap/bc/webdynpro/sapsrm/wda\_wf\_anlysis\_cfg"

#### 5.1.2 SAP NetWeaver Utilities

Common transactions are SWI1 or SWI6 respectively:

They can be used to find the workflow log of a specific workflow instance. The log reveals details about potential errors particularly in the chronologically sorted 'technical details' view (see menu 'Extras -> All errors').

For transaction SWI6 the following parameters are relevant:

- Object Type Category: *ABAP Class*
- Class/Interface: */SAPSRM/CL\_WF\_PDO\_<BO>*  
(where <BO> has to be replaced by one the following: SC, PO, CONF, INV, CTR, RFQ or QTE)
- Key: *The GUID of the document*

SWE2: Check the existing event type linkages

SM58: Check status of transactional RFCs related to user WF-BATCH

SM13: Check status of update requests

ST22: Check for potential ABAP runtime errors

SLG1: View application log related to object */SAPSRM/*

### 5.2 Find out if an approval process is started

Use SWI6 or SWI1 to check whether a workflow instance for the given document has been created.

Sometimes event coupling is disabled – to be checked via SWE2: Coupling of class /SAPSRM/CL\_WF\_PDO, Event: *READY\_FOR\_WORKFLOW* to workflow *WS40000014* has to be active.

### 5.3 Find out where and why the approval process got stuck

No obvious error occurred, but no workitem delivered

- For each workflow task a list of possible agents has to be assigned. The system establishes the recipients of a work item at runtime. The recipients are derived from the intersection of possible agents and responsible agents barring excluded agents. If this intersection is empty, no users receive the work item.
  - Usually the SRM related workflow tasks should be defined as 'general task' via transaction /SAPSRM/WF\_GEN, that is, each user is eligible to process it.
  - Under no circumstances the requester(document owner) must approve his own document. Therefore he is on the list of excluded agents. Please check via Transaction SWI6.
- UWL connector runs on an error and can not update the users view. Please check the details on the users screen, a message is provided in the UWL.

An error occurred during background processing.

- The Update Task System did not finish on the last commit operation. Please check transaction SM13
- The transactional RFC (tRFC) related to a workflow background step might have failed. Check for core dumps via transaction ST22. Check the RFCs via SM58 by providing the user WF-BATCH as selection parameter.
- If an exception is raised instead of a core dump the workflow instance is put into error state. Check transaction SWI6 to find details about the error in the technical workflow log. The administrative view of the failed work item allows a restart of the work item execution.
- If a core dump occurred during background processing the workflow instance gets stuck however it is still in status 'STARTED'. Check the exact start time of the latest work item of this workflow in the workflow log. Check ST22 to find the corresponding core dump at this point in time. There is currently no means to recover from this error state.

### 5.4 Find Out why Levels Are Not Executed

Check the application log (transaction SLG1) filtered for the object /SAPSRM/ to see BRF specific log entries.

By restricting the timeframe the details of a rule-based process level evaluation can be determined. Furthermore the result of a process schema evaluation can also be checked.

### 5.5 Find Out why Document Is still 'Awaiting Approval'

If a document appears to be in status 'Awaiting approval' even though all process steps are processed. Typically, this is an issue with the 'handle process finish' call.

Please check in SWI6 if there is already a message available that allows to understand the failure. The overall workflow status will be 'error' and the background step 'handle process finish' will also be in status 'error'. Navigate to the background step and check the message tab in the technical view for any message.

If you don't have a message in the workflow or don't understand the message, you can also try to debug the final call.

- Start transaction SE24
- Create instance of /SAPSRM/CL\_WF\_PDO\_<BO>, where <BO> denotes the concrete business object

- Debug the message handle process finish. Please change sy-uname variable from your debug user to 'WF-BATCH' to consider also authorization problems for the technical workflow user.

## 6 Appendix A: Shipped BRF Expressions

### 6.1 Common

#### 6.1.1.1 Constant Value FALSE

Class ID	Expression	Result Type
0CN001	0B_WF_FALSE	Bool

'FALSE' in BRF.

#### 6.1.1.2 Constant Value True

Class ID	Expression	Result Type
0CF001	0B_WF_TRUE	Bool

'TRUE' in BRF

#### 6.1.1.3 Random Number

Class ID	Expression	Result Type
0RV001	0V_WF_RANDOMNUMBER	Float

A generated random number is returned (range 0-1).

## 6.2 Process/Workflow attributes

#### 6.2.1.1 Automatic Decision Required by SAP System

Class ID	Expression	Result Type
0FB001	0B_WF_SYSTEMAPPROVAL	Bool

Rule composed of other expressions:

Logical NOT of expression 0B\_WF\_VALIDLEVAVAILA. If 0B\_WF\_VALIDLEVAVAILA is 'TRUE', the result is 'FALSE' and vice versa.

#### 6.2.1.2 At Least One Process Level Is Valid

Class ID	Expression	Result Type
0CF001	0B_WF_VALIDLEVAVAILA	Bool

Evaluation of workflow process status of previous level. Result:

- 'TRUE': previous level has status ACTIVE, COMPLETED or VALID
- 'FALSE': otherwise

### 6.2.1.3 Status of WF\_PDO Object

Class ID	Expression	Result Type
OCF001	0V_WF_PDOSTATUS	Bool

Evaluation of workflow process status of current document version. The possible values are:

- INIT: No process exists or it is still in initial status (for example, preview before order)
- ACTV: Process still active and document is not rejected
- FINI: Document rejected or last level of process completed.

### 6.2.1.4 Status of Previous Level

Class ID	Expression	Result Type
OCF001	0V_WF_PREVLEVELSTAT	Bool

Evaluation of workflow process status of previous level in approval process. The possible values are:

- ABORTED: process ended negative
- ACTIVE: assigned process is running
- COMPLETED: process ended positive
- INVALID: not valid (in the sense of the BRF framework)
- VALID: valid (in the sense of BRF framework).

### 6.2.1.5 Is last process level

Class ID	Expression	Result Type
OCN001	0B_WF_ISLASTLEVEL	Bool

If the last process level is reached, the result is set to 'TRUE'; otherwise 'FALSE'.

### 6.2.1.6 Actual Process Level

Class ID	Expression	Result Type
OCF001	0V_WF_ACTLEVELSTAT	Char

The status of the current process level is evaluated. Possible return values are:

- ABORTED: Aborted (process ended negative)
- ACTIVE: Active (assigned process is running)
- COMPLETED: Completed (process ended positive)
- INITIAL: Initial by template
- INVALID: Not valid (in the sense of the BRF framework)
- VALID: Valid (in the sense of the BRF framework)

### 6.2.1.7 Person Responsible for the document

Class ID	Expression	Result Type
OCF001	0V_WF_DOCRESPONSIBLE	Char

The SAP user names of the persons responsible for the confirmation are determined. The following rules are applied in the given order:

- 1.) The creator, if the creator is a professional user or if the creator is one of the requestors and belongs to the purchasing organization
- 2.) The requestor of the confirmation, if there is a single requestor, the requestor belongs to the purchasing organization and the requestor is identical to the creator or it is not an external request.
- 3.) If no person responsible is found so far, the previous document is searched for a person responsible:
  1. The creator of the source document, if the creator is assigned to the purchasing organization
  2. If the source document is a purchase order in the back-end system, the creator of the purchase order is returned if he/she belongs the purchasing organization
  3. Finally, the responsible purchasing groups are determined for all product categories found in the items of the confirmation. The user names of all purchasers assigned to those purchasing groups are returned.

## 6.3 Organizational Model access

### 6.3.1.1 Is Top Manager

Class ID	Expression	Result Type
OCF001	0B_WF_ISTOPMANAGER	Bool

First, the person responsible for the document is determined using expression 0V\_WF\_DOCRESPONSIBLE. From the organizational management it is then determined whether the person responsible is a manager who has no manager himself/herself. In this case, the person is a top manager and the output is set to 'TRUE'. Otherwise, it is 'FALSE'.

### 6.3.1.2 Preceeding process level has only top manager

Class ID	Expression	Result Type
OCF001	0B_WF_PRVLVLNLYTPMGR	Bool

The previous approval step in the workflow is analyzed. If this process level has the status 'Completed', 'Active' or 'Valid' the list of approvers is checked. If all approvers are top managers (derived from the organizational management), the result is set to 'TRUE'. If at least one is not a top manager, the result is 'FALSE'.

## **6.4 Configurable Calls**

The corresponding information is available via SAP Note 1015211



## 6.5 BO Confirmation

### 6.5.1 Document Attributes

#### 6.5.1.1 Active Statuses Exist

Class ID	Expression	Result Type
0CF001	0B_CO_STATUSTABLE	Bool

The table of active statuses is determined for the confirmation. This list is compared to the status passed as constants in the expression. If one of them is found, the result is set to 'TRUE', otherwise it is 'FALSE'. The constant expressions contain the following statuses:

- I1009: Held (HELD)
- I1021: Created (CRTD)
- I1039: Incomplete (INCM)

The constant expressions have to contain the internal representation of the system status, which should be checked against the attribute table. The list of system statuses can be found in table TJ02T.

#### 6.5.1.2 Gross Amount As String

Class ID	Expression	Result Type
0CF001	0C_CO_GROSSAMOUNTSTR	Char

The net amount of the confirmation is calculated as the sum of the net values of all items not marked for deletion. The net value of an item is calculated as follows: net price \* quantity / price unit. The net amount is returned as a character string with the actual decimal notation from the user data and the actual currency representation. The value is displayed in the currency of the confirmation.

#### 6.5.1.3 Posting Date for a Business Transaction

Class ID	Expression	Result Type
0CF001	0C_CO_GROSSAMOUNTSTR	Date

The posting date of the confirmation from the document header is passed as the result in the format YYYYMMDD

#### 6.5.1.4 Gross Confirmation Amount in Document Currency

Class ID	Expression	Result Type
0CF001	0F_CO_GROSSAMOUNT	Float

The posting date of the confirmation from the document header is passed as the result in the format YYYYMMDD

### 6.5.1.5 Gross Confirmation Amount in Document Currency

Class ID	Expression	Result Type
0CF001	0V_CO_BEREF0BJ	Char

The document number of the reference document in the back-end system is passed as the result. Leading zeroes are not deleted. The length is determined by the number range in the back-end system.

### 6.5.1.6 Bill of Lading Number

Class ID	Expression	Result Type
0CF001	0V_CO_BILLOFLADING	Char

The number of the bill of lading is returned as the result.

### 6.5.1.7 Business Process

Class ID	Expression	Result Type
0CF001	0V_CO_BUSINESSPROC	Char

The header GUID of the current confirmation is passed as the result.

### 6.5.1.8 Changed By

Class ID	Expression	Result Type
0CF001	0V_CO_CHANGEDBY	Char

The last person who changed the confirmation is selected from the document header. The SAP user name is passed as the result.

### 6.5.1.9 Created By

Class ID	Expression	Result Type
0CF001	0V_CO_CREATEDBY	Char

The person who created the confirmation is selected from the document header. The SAP user name is passed as the result.

### 6.5.1.10 Security Level of Creator

Class ID	Expression	Result Type
0CF001	0V_CO_CREATUSRSECLVL	Char

The security level is first determined from the user management using the creator of the document. If a value is found, this value is passed as the result. If no value is found, the security levels associated with all the roles to which the user is assigned are identified. The highest security level found is passed as the result.

### 6.5.1.11 Activity Group of Creator

Class ID	Expression	Result Type
0CF001	0V_CO_CREAUSRACTGRP	Char

The creator of the confirmation is selected from the document header. An instance of the business object BUS4101(EC User) for the creator is created and the activity groups to which the user is assigned are determined. If one of the activity groups provided in the additional constants nn\_C4\_C\_ACTGRP\_\* is found in the table of activity groups, the result is set to 'FALSE'. Otherwise, it is set to 'TRUE'.

### 6.5.1.12 Currency Key

Class ID	Expression	Result Type
0CF001	0V_CO_CURRENCY	Char

The currency key from the confirmation document header is passed as the result

### 6.5.1.13 Business Transaction Description

Class ID	Expression	Result Type
0CF001	0V_CO_DESCRIPTION	Char

The description from the confirmation document header is returned as the result.

### 6.5.1.14 Object Type Name

Class ID	Expression	Result Type
0CF001	0V_CO_DOCTYPENAME	Char

The name of the document type in logon language is passed back:

If the subtype is RT, the result is 'Return delivery' for the language English. In all other cases the result is 'Confirmation'. These texts are stored in the message 049(bbp\_wfl) and 050(bbp\_wfl).

### 6.5.1.15 Event of the Object

Class ID	Expression	Result Type
0CF001	0V_CO_EVENT	Char

Constant expression. The return value is always 'SAVED'.

### 6.5.1.16 External Requirement

Class ID	Expression	Result Type
0CF001	0V_CO_EXTERNALDEMAND	Char

If the confirmation belongs to an external demand, the result is 'X'; otherwise it is ''. The external demand is identified by a missing requester (for example, when the PO has been created in the back-end), the subtype of the document is 'ER' or the user has no assigned roles (indicating a technical user).

### 6.5.1.17 Responsible Person from Source Document

Class ID	Expression	Result Type
0CF001	0V_CO_GETSRCDOCRSP	Char

The SAP user names of the persons responsible for the confirmation are determined. The following rules are applied in the given order:

1. The creator if the creator is a professional user or if the creator is one of the requesters and belongs to the purchasing organization
2. The requester of the confirmation if the following conditions are fulfilled:
  - single requester AND
  - requester belongs to the purchasing organization AND
  - requester is identical to the creator or it is not an external request
3. If no responsible person has been found so far, the preceding documents are searched for a person responsible:
  1. The creator of the source document if the creator is assigned to the purchasing organization
  2. If the source document is a purchase order in the back-end system, the creator of the purchase order is returned if he/she belongs to the purchasing organization
  3. As a last step, the responsible purchasing groups are determined for all product categories found in the items of the confirmation. The user names of all purchasers assigned to those purchasing groups are returned.

### 6.5.1.18 Goods Receipt / Issue Slip Number

Class ID	Expression	Result Type
0CF001	0V_CO_GRGISLIPNO	Char

The goods receipt/issue slip number is returned. Leading zeroes are not deleted.

**6.5.1.19 Transaction Number**

Class ID	Expression	Result Type
0CF001	0V_CO_OBJECT_ID	Char

The transaction number of the confirmation is returned determined from the document header. Leading zeroes are not deleted.

**6.5.1.20 Transaction Type with Transaction Number**

Class ID	Expression	Result Type
0CF001	0V_CO_PROCESSID	Char

The result value is the description of the business transaction type (in logon language of the user) concatenated with the confirmation number (leading zeroes are deleted). The complete list of business transaction types together with the descriptions can be found in table BBPC\_PROC\_TYPE\_T. A confirmation has the bus. trans. type CONF with description 'Confirmation' in English.

**6.5.1.21 Lean PO Number**

Class ID	Expression	Result Type
0CF001	0V_CO_PURCHASEORDER	Char

If the confirmation refers to a local purchase order (GUID of source document), the transaction number for this PO is determined and passed back as the result. Leading zeroes are not deleted. If the purchase order is a back-end document, the returned value is initial.

**6.5.1.22 Code of Rejection Reason**

Class ID	Expression	Result Type
0CF001	0V_CO_REASONCODE	Char

The output is the code from the document header indicating the reason of rejection. The length is 4 characters. The list of possible values can be found in table BBP\_IVREA, which is accessible through the customizing of the 'Confirmation and Invoice Verification'

**6.5.1.23 Supplier Invoice Number**

Class ID	Expression	Result Type
0CF001	0V_CO_REFERENCE	Char

The reference number from the confirmation document header is determined and passed back as the result.

#### 6.5.1.24 Characteristic

Class ID	Expression	Result Type
OCF001	0V_CO_SPECIFICATION	Char

Subtype of document derived from the invoice header. The valid subtypes are:

- CF (Goods/services confirmation)
- RT (Return delivery confirmation)

#### 6.5.1.25 Requester User Name from Source Document

Class ID	Expression	Result Type
OCF001	0V_CO_SRCDOCREQUER	Char

The requester of the source document is determined. The history of the confirmation is searched for a source document of type shopping cart. If a shopping cart is found, the shopping cart partners are selected. For the partner of type requester the SAP user name is derived. This user name is passed as the result.

#### 6.5.1.26 Completion Status

Class ID	Expression	Result Type
OCF001	0V_CO_STATUSCOMPLETE	Char

It is checked if the status I1038 (Complete) is active for the confirmation. If yes, the result is 'X'; if no, the result is ''.

#### 6.5.1.27 Status Object Number

Class ID	Expression	Result Type
OCF001	0V_CO_STATUSOBJNR	Char

The GUID of the confirmation document header is returned.

#### 6.5.1.28 Object Type for Status Management

Class ID	Expression	Result Type
OCF001	0V_CO_STATUSOBJTYPE	Char

The type of the status object is determined (from table CRM\_JSTO). Using the header GUID the result for a confirmation is 'CCH'. All possible status object types can be found in table TJ03.

**6.5.1.29 Status Profile**

Class ID	Expression	Result Type
0CF001	0V_CO_STATUSPROFILE	Char

The status profile for the status object is determined (from table CRM\_JSTO). The header GUID of the invoice is used. The value is passed as the result.

**6.5.1.30 Type/Role of Creator of Procurement Document**

Class ID	Expression	Result Type
0CF001	0V_CO_WFLCREATORATYPE	Char

The type of the person who started the workflow is returned. Possible values are:

- V: Vendor
- D: Desktop user
- P: Professional user
- R: Requester
- M: Invoice assistant (mass entry)

**6.5.1.31 Posting Date**

Class ID	Expression	Result Type
0CF001	0D_CO_POSTINGDATE	Date

Reads the posting date from CO BUS2203 object.

## 6.6 BO Contract

### 6.6.1 Document Attributes

#### 6.6.1.1 Is Change Version

Class ID	Expression	Result Type
OCF001	0B_CT_ISCHANGEVERS	Bool

If the version type of the document is 'C' (=change version), the result is set to 'TRUE'; otherwise 'FALSE'.

#### 6.6.1.2 Product category exists

Class ID	Expression	Result Type
OCF001	0B_CT_PRCAT_EX_CHECK	Bool

The product categories of all items in the contract are checked against the list of product categories in the constant expression 9C\_C4\_C\_PRCAT\_001. A single value has the format 'CatID\Logsys' (ID of product category and logical system of source system. The logical system is necessary to ensure uniqueness). Multiple values are separated by ";". If a product category is found, the result is 'FALSE'. Otherwise, it is 'TRUE'.

#### 6.6.1.3 Simple List of Changes exists

Class ID	Expression	Result Type
OCF001	0B_CT_SIMPLELISTOCHG	Bool

If change versions exist for the contract, the current version and the original version are compared. Changes are collected in the table attribute SimpleListOfChanges, which contains key words indicating which information has been changed (it does not contain the content of the changes). If the list contains one of the key words ACC\_MODIFY (accounting information was changed) or PRT\_MODIFY (partner information was changed), the output is set to 'TRUE'; otherwise it is 'FALSE'. The key words which are checked against the table are passed as constant expressions in the Boolean expression.

A complete list of possible key words for changes, which can appear in the output table of the BOR attribute SimpleListOfChanges, can be found in function module BBP\_PDH\_WFL\_MAP\_CHANGES. The list of fields which are actually compared can be extracted from table BBPD\_COMP\_FIELDS for comparison method 'VUI'. For the comparison in the attribute only a part of the structures is used:

1. Header
2. Items
3. Accounting information
4. Partner information
5. Confirmation
6. Conditions
7. Pricing information
8. Org. data



9. Tax information
10. Dynamic attributes

#### 6.6.1.4 Active Statuses Exist

**Attribute not implemented. Do not use expression.**

Class ID	Expression	Result Type
OCF001	0B_CT_STATUSTABLE	Bool

#### 6.6.1.5 List Of Changed Technical Fields

Class ID	Expression	Result Type
OCF001	0B_CT_TECLSTOFCHG	Bool

If change versions exist for the contract, the current version and the original version are compared. Changes are collected in the table attribute TechnicalListOfChanges, which contains an object name and the field names of the changed fields separated by a space (it does not contain the content of the changes). If the list contains one of the key words provided as constant expressions nn\_C4\_C\_nnnn, the return value is set to 'TRUE'. Otherwise, it is 'FALSE'.

The list of fields which are actually compared can be extracted from table BBPD\_COMP\_FIELDS for comparison method 'VUI'. For the comparison in the attribute only a part of the structures is used:

1. Header (no object name used)
2. Items (no object name used)
3. Accounting information (object name ACCOUNTING)
4. Partner information (object name PARTNER)
5. Confirmation (object name CONFIRMATIONS)
6. Conditions (object name CONDITION\_COM)
7. Pricing information (object name PRIDOC)
8. Org. data (object name ORGMAN)
9. Tax information (object name TAX)
10. Dynamic attributes (object name DYN\_ATTR)

The list of object names can be found in table BBPD\_OBJECTS.

#### 6.6.1.6 Target Value

Class ID	Expression	Result Type
OCF001	0F_CT_TARGETVALUE	Bool

The target value of the contract is determined from the document header. The value is displayed in the document currency. A different currency can be passed in the expression using a constant expression nn\_C5\_C\_WHG\_nnn. The total value will then be converted to this currency using type rate 'M' and the current local date.

### 6.6.1.7 Changed By

Class ID	Expression	Result Type
0CF001	0V_CT_CHANGEDBY	Char

SAP user name of last person who changed the contract, extracted from the document header.

### 6.6.1.8 Changed By

Class ID	Expression	Result Type
0CF001	0V_CT_CHANGEDON	Char

The date of last change is passed back as the result. Date and time are concatenated. The resulting string has the format YYYYMMDDHHMMSS.

### 6.6.1.9 Creator User Name

Class ID	Expression	Result Type
0CF001	0V_CT_CREATORUSER	Char

The creator of the contract is selected from the document header. An instance of the business object BUS4101(EC User) for the creator is created and the SAP user name is derived. This user name is passed as the result.

### 6.6.1.10 Current System Status

*Attribute not implemented. Do not use expression.*

Class ID	Expression	Result Type
0CF001	0V_CT_CURSYSTAT	Char

### 6.6.1.11 Current User Status

*Attribute not implemented. Do not use expression.*

Class ID	Expression	Result Type
0CF001	0V_CT_CURUSERSTAT	Char

**6.6.1.12 Error Free Condition**

Attribute not implemented. Do not use expression.

Class ID	Expression	Result Type
0CF001	0V_CT_ERRORFREEFLG	Char

**6.6.1.13 Event of the Object**

Class ID	Expression	Result Type
0CF001	0V_CT_EVENT	Char

The method checks if the current document version is a change version (version type 'C'). If yes, the result is 'CHANGEVERSIONSAVED'. Otherwise, the result is 'SAVED'

**6.6.1.14 Last Changed By**

Class ID	Expression	Result Type
0CF001	0V_CT_LSTDOCPRCESR	Char

The last person who changed the contract is selected from the document header. An instance of the business object BUS4101(EC User) for this user is created and the SAP user name is derived. This user name is passed as the result.

**6.6.1.15 Send Medium for Contract Output**

Class ID	Expression	Result Type
0CF001	0V_CT_MEDIUM	Char

The type of the output medium (for example, XML, PRN, FAX, MAI, etc.) for the supplier of the contract is passed.

**6.6.1.16 Partner Number of Message Recipient**

Class ID	Expression	Result Type
0CF001	0V_CT_MSGRECIKEY	Char

GUID of the partner of type supplier in the contract is passed as the result.

**6.6.1.17 Transaction Number**

Class ID	Expression	Result Type
0CF001	0V_CT_OBJECT_ID	Char

Contract number from the document header is passed as the result. The number is 10 digits long and the leading zeroes are not deleted.

**6.6.1.18 Original Document**

Class ID	Expression	Result Type
0CF001	0V_CT_ORIGINALDOC	Char

Contract number from the document header of the original document is passed as the result. The number is 10 digits long and the leading zeroes are not deleted.

**6.6.1.19 Object Type of Previous Document**

Class ID	Expression	Result Type
0CF001	0V_CT_PRCRSRDCMNTTYP	Char

The last preceding document is determined. The business object type is passed as the result, for example, BUS2121 for a shopping cart. If no previous document can be found, the result is initial.

**6.6.1.20 Priority**

Attribute not implemented. Do not use expression.

Class ID	Expression	Result Type
0CF001	0V_CT_PRIORITY	Char

**6.6.1.21 Transaction Type with Transaction Number**

Class ID	Expression	Result Type
0CF001	0V_CT_PRCRSRDCMNTTYP	Char

The result value is the description of the business transaction type (in logon language of the user) concatenated with the contract number (leading zeroes are deleted). The complete list of business transaction types together with the descriptions can be found in table BBPC\_PROC\_TYPE\_T. For example, a purchase contract has the business transaction type PCTR with description 'Purchase Contract' in English.

### 6.6.1.22 Business Transaction Type

Class ID	Expression	Result Type
OCF001	0V_CT_PROCESSTYPE	Char

The business transaction type from the document header is determined and passed as the result. A purchase contract, for example, has the bus. trans. type PCTR. The complete list of business transaction types together with the descriptions can be found in table BBPC\_PROC\_TYPE\_T

### 6.6.1.23 Responsible Purchasing Group

Class ID	Expression	Result Type
OCF001	0V_CT_PURCHASNGGROUP	Char

The ID of the purchasing group from the organization management is passed as the result. The format is object type 'O' + object ID (for example, 'O 50000019').

### 6.6.1.24 Responsible Purchasing Department

Class ID	Expression	Result Type
OCF001	0V_CT_PURCHASNGORG	Char

The ID of the purchasing organization from the organization management is passed as the result. The format is object type 'O' + object ID (for example, 'O 50000019').

### 6.6.1.25 Status Object Number

Attribute not implemented. Do not use expression.

Class ID	Expression	Result Type
OCF001	0V_CT_STATUSOBJNR	Char

### 6.6.1.26 Status Object Type

Attribute not implemented. Do not use expression.

Class ID	Expression	Result Type
OCF001	0V_CT_STATUSOBJTYPE	Char

**6.6.1.27 Status Profile**

Attribute not implemented. Do not use expression.

Class ID	Expression	Result Type
0CF001	0V_CT_STATUSPROFILE	Char

**6.6.1.28 Transaction can be printed**

Attribute not implemented. Do not use expression.

Class ID	Expression	Result Type
0CF001	0V_CT_TOBEPRINTED	Char

**6.6.1.29 Supplier for this contract**

Class ID	Expression	Result Type
0CF001	0V_CT_VENDORBP	Char

The business partner number of the supplier is passed back as the result. The number is found searching for the business partner of type Supplier from the partner data. The number has 10 digits and leading zeroes are not deleted.

**6.6.1.30 Version Type**

Class ID	Expression	Result Type
0CF001	0V_CT_VERSIONTYPE	Char

The version type of the change version is passed back. If the document is the active version, the result is ' '. Historical versions are marked with 'H', change versions with 'C':

**6.6.1.31 Name of Version Type**

Class ID	Expression	Result Type
0CF001	0V_CT_VERSTYPENAME	Char

The text in logon language of the user for the version types given above is displayed. The result in English for the three types would be:

- ' ': Contract
- 'C': Change version of contract
- 'H': Contract version

### 6.6.1.32 Transaction Type and Transaction Number

Class ID	Expression	Result Type
0CF001	0V_CT_PROCESSID	Char

Gets Transaction Type With Trans. Number e.g. "Glob.Outline Agrmnt 5600004398".

## 6.7 BO Invoice

### 6.7.1 Document Attributes

#### 6.7.1.1 Active Statuses Exist

Class ID	Expression	Result Type
0CF001	0B_IN_STATUSTABLE	Bool

The table of active statuses is determined for the invoice. This list is compared to the status passed as constants in the expression. If one of them is found, the result is set to 'TRUE'; otherwise it is 'FALSE'. The constant expressions contain the following statuses:

- 1.) I1009: Held (HELD)
- 2.) I1021: Created (CRTD)
- 3.) I1039: Incomplete (INCM)

The constant expressions have to contain the internal representation of the system status, which should be checked against the attribute table. The list of system statuses can be found in table TJ02T.

#### 6.7.1.2 Posting Date

Class ID	Expression	Result Type
0CF001	0D_IN_POSTINGDATE	Char

The posting date of the invoice from the document header is passed as the result in the format YYYYMMDD

#### 6.7.1.3 Posting Date in Accounting

Class ID	Expression	Result Type
0CF001	0D_IN_POSTINGDATE1	Char

The posting date of the invoice in accounting is passed as the result. The format is YYYYMMDD.

#### 6.7.1.4 Gross Invoice Amount

Class ID	Expression	Result Type
0CF001	0F_IN_GROSSINVAMOUNT	Float

The gross amount of the invoice is passed. The value is displayed in the currency of the invoice.



**6.7.1.5 Gross Invoice Amount as String**

Class ID	Expression	Result Type
OCF001	0F_IN_GROSSINVAMTSTR	Char

The gross amount of the invoice is passed back as character string N293.

**6.7.1.6 Backend Logical System**

Class ID	Expression	Result Type
OCF001	0V_IN_BCKENDLOGSYS	Char

The logical system of the logistics back-end system is determined.

**6.7.1.7 BE Acc. Required**

Attribute not implemented. Do not use expression.

Class ID	Expression	Result Type
OCF001	0V_IN_BEACCQUIRE	Char

**6.7.1.8 Business Process**

Class ID	Expression	Result Type
OCF001	0V_IN_BUSINESSPROC	Char

Header GUID of invoice is retrieved.

**6.7.1.9 BusTransDescriptn**

Class ID	Expression	Result Type
OCF001	0V_IN_BUSTRANSDESC	Char

Description of invoice from document header is returned as the result.

**6.7.1.10 Cardholder Name**

Class ID	Expression	Result Type
OCF001	0V_IN_CARDHOLDRNAM	Char

Name of the procurement card holder from the invoice document header is passed as the result.

### 6.7.1.11 Category

Class ID	Expression	Result Type
0CF001	0V_IN_CATEGORY	Char

GUID of product category from invoice document header.

### 6.7.1.12 Category ID

Class ID	Expression	Result Type
0CF001	0V_IN_CATEGORYID	Char

ID of product category from invoice document header.

### 6.7.1.13 Changed By

Class ID	Expression	Result Type
0CF001	0V_IN_CHANGEDBY	Char

The last person who changed the invoice is selected from the document header. The SAP user name is passed as the result.

### 6.7.1.14 Changed On

Class ID	Expression	Result Type
0CF001	0V_IN_CHANGEDON	Char

The date of last change is passed back as the result. Date and time are concatenated. The resulting string has the format YYYYMMDDHHMMSS.

### 6.7.1.15 Company Code

Class ID	Expression	Result Type
0CF001	0V_IN_COMPANYCODE	Char

The company code from the invoice header is passed as the result.

**6.7.1.16 Created By**

Class ID	Expression	Result Type
0CF001	0V_IN_CREATEDBY	Char

The person who created the invoice is selected from the document header. The SAP user name is passed as the result.

**6.7.1.17 Created On**

Class ID	Expression	Result Type
0CF001	0V_IN_CREATEDON	Char

The date of creation is passed back as the result. Date and time are concatenated. The resulting string has the format YYYYMMDDHHMMSS.

**6.7.1.18 Created On (Date)**

Class ID	Expression	Result Type
0CF001	0V_IN_CREATEDONDATE	Char

The posting date from the invoice document header is selected. The result is passed in the format YYYYMMDD.

**6.7.1.19 Creator – User Name**

Class ID	Expression	Result Type
0CF001	0V_IN_CREATORUSER	Char

The creator of the invoice is selected from the document header. An instance of the business object BUS4101(EC User) for the creator is created and the SAP user name is derived. This user name is passed as the result.

**6.7.1.20 Currency**

Class ID	Expression	Result Type
0CF001	0V_IN_CURRENCY	Char

The currency key from the invoice document header is passed as the result.

**6.7.1.21 Evaluated Receipts Settlement (ERS)**

Class ID	Expression	Result Type
0CF001	0V_IN_ERS	Char

The creation indicator of the document is checked. If the indicator is 3 (=ERS, evaluated receipt settlement), the result is set to 'X'. Otherwise it is ''.

**6.7.1.22 Event of the Object**

Class ID	Expression	Result Type
0CF001	0V_IN_EVENT	Char

Constant expression. The return value is always 'SAVED'.

**6.7.1.23 Start external approval**

Class ID	Expression	Result Type
0CF001	0V_IN_EXTAPPRWFLSTRT	Char

If the status I1360 (Awaiting external approval) is active for the invoice, the result value is 'X'. Otherwise it is ''.

**6.7.1.24 External Requirement**

Class ID	Expression	Result Type
0CF001	0V_IN_EXTERNALDEMAND	Char

If the invoice belongs to an external demand, the result is 'X'. Otherwise it is ''. The external demand is identified by a missing requestor (for example, when the PO was created in the back-end), the subtype of the document is 'ER' or the user has no assigned roles (indicating a technical user).

**6.7.1.25 InvoiceCreditMemoIndicator (Subtype)**

Class ID	Expression	Result Type
0CF001	0V_IN_INVCREDMEMIND	Char

Subtype of document derived from the invoice header. The valid subtypes are:

- IV (Invoice)
- CM (Credit memo)

**6.7.1.26 Logical System**

Class ID	Expression	Result Type
OCF001	0V_IN_LOGICALSYSTEM	Char

The result is the logical system for the invoice.

**6.7.1.27 Logical System of FI System**

Class ID	Expression	Result Type
OCF001	0V_IN_LOGICSYSFISYS	Char

The result is the logical system of the FI system.

**6.7.1.28 Transaction No.**

Class ID	Expression	Result Type
OCF001	0V_IN_OBJECT_ID	Char

The invoice document number from the document header is passed as the result. The number has 10 digits, leading zeroes are not deleted.

**6.7.1.29 Object ID in Backend System**

Class ID	Expression	Result Type
OCF001	0V_IN_OBJIDBCKENDSYS	Char

The document number of the follow-on document in the back-end is passed as the result. Leading zeroes are not deleted. The length is determined by the number range in the back-end system.

**6.7.1.30 Object Type in Backend System**

Class ID	Expression	Result Type
OCF001	0V_IN_OBJTYPBCKENDSY	Char

The business object type of the follow-on document in the back-end system (for example, BUS2081) is passed as the result.

### 6.7.1.31 Payment Terms

Class ID	Expression	Result Type
0CF001	0V_IN_PAYMENTTERMS	Char

The key of the payment terms from the document header is passed as the result (the length is 4 characters).

### 6.7.1.32 Procurement Card Number

Class ID	Expression	Result Type
0CF001	0V_IN_PCARDNO	Char

The procurement card number determined from the document header is passed back as the result. Leading zeroes are not deleted.

### 6.7.1.33 Procurement Card Company

Class ID	Expression	Result Type
0CF001	0V_IN_PROCCARDCOMP	Char

The key term for the procurement card company is passed back as the result (if found in document header). The length of the result is four characters. The list of procurement companies can be found in table BBP\_PCINS or in the customizing for procurement cards.

### 6.7.1.34 Transaction Type with Transaction Number

Class ID	Expression	Result Type
0CF001	0V_IN_PROCESSID	Char

The result value is the description of the business transaction type (in logon language of the user) concatenated with the invoice number (leading zeroes are deleted). The complete list of business transaction types together with the descriptions can be found in table BBPC\_PROC\_TYPE\_T. An invoice has the bus. trans. type INV with description 'Invoice' in English.

### 6.7.1.35 Reason Code

Class ID	Expression	Result Type
0CF001	0V_IN_REASONCODE	Char

The output is the code from the document header indicating the reason for the invoice rejection. The length is 4 characters. The list of possible values can be found in table BBP\_IVREA, which is accessible through the customizing of the 'Confirmation and Invoice Verification'.

**6.7.1.36 Reference**

Class ID	Expression	Result Type
OCF001	0V_IN_REFERENCE	Char

The reference number (external invoice number) from the invoice document header is determined and passed back as the result.

**6.7.1.37 Completion Status**

Class ID	Expression	Result Type
OCF001	0V_IN_STATUSCOMPLETE	Char

It is checked if the status I1038 (Complete) is active for the invoice. If yes, the result is 'X'; if no, the result is ' '.

**6.7.1.38 Status Object Number**

Class ID	Expression	Result Type
OCF001	0V_IN_STATUSOBJNR	Char

The GUID of the invoice is returned.

**6.7.1.39 Object Type for Status Management**

Class ID	Expression	Result Type
OCF001	0V_IN_STATUSOBJTYPE	Char

The type of the status object is determined (from table CRM\_JSTO). Using the header GUID, the result for an invoice is 'IVH'. All possible status object types can be found in table TJ03.

**6.7.1.40 Status Profile**

Class ID	Expression	Result Type
OCF001	0V_IN_STATUSPROFILE	Char

The status profile for the status object is determined (from table CRM\_JSTO). The header GUID of the invoice is used. The value is passed as the result.

#### 6.7.1.41 Tolerance Group

Class ID	Expression	Result Type
0CF001	0V_IN_TOLGROUP	Char

The tolerance group assigned to the invoice is passed back as the result. The tolerance group has a length of 4 characters. The possible values can be found in customizing (Set tolerance groups) or in the table BBP\_TOL\_GRP

#### 6.7.1.42 Transaction Type

Class ID	Expression	Result Type
0CF001	0V_IN_TRANSACTIONTYP	Char

The business transaction type of the invoice is returned as the result. For an invoice the type is 'INV'.

#### 6.7.1.43 Transaction Number

Class ID	Expression	Result Type
0CF001	0V_IN_TRANSACTNO	Char

The invoice document number from the document header is passed as the result. The number has 10 digits, leading zeroes are not deleted.

#### 6.7.1.44 Validity Date of Procurement Card

Class ID	Expression	Result Type
0CF001	0V_IN_VALIDDATEPCARD	Char

The validity date of the procurement card is passed back as the result. The format is YYYYMMDD.

#### 6.7.1.45 Type/Role of Creator of Procurement Document

Class ID	Expression	Result Type
0CF001	0V_IN_WFLCREATORTYPE	Char

The type of the person who started the workflow is returned. Possible values are:

- V: Vendor
- D: Desktop user
- P: Professional user



- R: Requester
- M: Invoice assistant (mass entry)

**6.7.1.46 Name of Object Type**

Class ID	Expression	Result Type
0CF001	0V_IN_DOCTYPENAME	Char

Retrieves the name of the object type (invoice or credit memo), for example, "Invoice".

**6.7.1.47 User Name of Requester of Source Document**

Class ID	Expression	Result Type
0CF001	0V_IN_SRCDOCREQUSER	Char

Gets the user name of the requester of source document Shopping Cart.

## 6.8 BO Purchase Order

### 6.8.1 Document Attributes

#### 6.8.1.1 Is Change Version

Class ID	Expression	Result Type
OCF001	0B_PO_ISCHANGEVERS	Bool

If the version type of the document is 'C' (=change version) the result is set to 'TRUE', otherwise 'FALSE'.

#### 6.8.1.2 Checks Whether Specified Product Category Exists

Class ID	Expression	Result Type
OCF001	0B_PO_PRCAT_EX_CHECK	Bool

The product categories of all items in the purchase order are checked against the list of product categories in the constant expression 9C\_C4\_C\_PRCAT\_001. A single value has the format 'CatID\Logsys' (ID of product category and logical system of source system. The logical system is necessary to ensure uniqueness). Multiple values are separated by ";". If a product category is found, the result is 'FALSE'. Otherwise it is 'TRUE'.

#### 6.8.1.3 Simple List of Changes Contains PRT\_MODIFY or ACC\_MODIFY

Class ID	Expression	Result Type
OCF001	0B_PO_SIMPLELISTOCHG	Bool

If change versions exist for the purchase order, the current version and the original version are compared. Changes are collected in the table attribute SimpleListOfChanges, which contains key words indicating which information has been changed (it does not contain the content of the changes). If the list contains one of the key words ACC\_MODIFY (accounting information was changed) or PRT\_MODIFY (partner information was changed) the output is set to 'TRUE', otherwise it is 'FALSE'. The key words which are checked against the table are passed as constant expressions in the boolean expression.

A complete list of possible key words for changes, which can appear in the output table of the BOR attribute SimpleListOfChanges, can be found in function module BBP\_PDH\_WFL\_MAP\_CHANGES.

The list of fields which are actually compared can be extracted from table BBPD\_COMP\_FIELDS for comparison method 'VUI'. For the comparison only some of the structures are used:

1. Header
2. Items
3. Accounting information
4. Partner information
5. Confirmation
6. Conditions
7. Pricing information
8. Org. data

- 9. Tax information
- 10. Dynamic attributes

#### 6.8.1.4 Active Statuses Exist

Class ID	Expression	Result Type
0CF001	0B_PO_STATUSTABLE	Bool

The table of active statuses is determined for the purchase order. This list is compared to the status passed as constants in the expression. If one of them was found, the result is set to 'TRUE'; otherwise it is 'FALSE'. The constant expressions contain the following statuses:

- 1.) I1009: Held (HELD)
- 2.) I1021: Created (CRTD)
- 3.) I1039: Incomplete (INCM)

The constant expressions have to contain the internal representation of the system status, which should be checked against the attribute table. The list of system statuses can be found in table TJ02T.

#### 6.8.1.5 Changed By

Class ID	Expression	Result Type
0CF001	0V_PO_CHANGEDBY	Char

SAP user name of last person who changed the purchase order extracted from document header

#### 6.8.1.6 Changed On

Class ID	Expression	Result Type
0CF001	0V_PO_CHANGEDON	Char

The date of last change is passed back as the result. Date and time are concatenated. The resulting string has the format YYYYMMDDHHMMSS.

#### 6.8.1.7 Creator – User Name

Class ID	Expression	Result Type
0CF001	0V_PO_CREATORUSER	Char

The creator of the purchase order is selected from the document header. An instance of the business object BUS4101(EC User) for the creator is created and the SAP user name is derived. This user name is passed as the result.

### 6.8.1.8 Currency

Class ID	Expression	Result Type
0CF001	0V_PO_CURRENCY	Char

Currency of purchase order extracted from document header.

### 6.8.1.9 Event of the Object

Class ID	Expression	Result Type
0CF001	0V_PO_EVENT	Char

The method checks if the current document version is a change version (version type 'C'). If yes, the result is 'CHANGEVERSIONSAVED'. Otherwise the result is 'SAVED'.

### 6.8.1.10 Last Changed By – User Name

Class ID	Expression	Result Type
0CF001	0V_PO_LSTDOCPRCESR	Char

The last person who changed the purchase order is selected from the document header. An instance of the business object BUS4101(EC User) for this user is created and the SAP user name is derived. This user name is passed as the result.

### 6.8.1.11 Sending Medium for Purchase Order Output

Class ID	Expression	Result Type
0CF001	0V_PO_MEDIUM	Char

The type of the output medium (e.g. XML,PRN,FAX;MAI,etc.) for the purchase order is passed.

### 6.8.1.12 Transaction number

Class ID	Expression	Result Type
0CF001	0V_PO_OBJECT_ID	Char

The purchase order number from the document header is passed as result.

**6.8.1.13 Purchase Order Total Value Increased (Boolean char1)**

Class ID	Expression	Result Type
0CF001	0V_PO_POTOTALVALINCR	Char

Sign of the difference of the total values between the actual PO and the original document in case of existing change versions. The result is 'X' if the actual value is higher, otherwise ' '.

**6.8.1.14 Net Total Amount of Purchase Order**

Class ID	Expression	Result Type
0CF001	0V_PO_POTOTALVALUE	Float

The value of all items of the purchase order is summed. The total value is passed as the result. The value is in PO currency.

**6.8.1.15 Difference Between Current Total Value and Original**

Class ID	Expression	Result Type
0CF001	0V_PO_POTOTLVLUEDIFF	Float

The total value of the purchase order is compared to the total value of the original purchase order in case of existing change versions. If the currencies differ, the value of the original document is converted to the current currency using type rate 'M' and current local date. The two values are compared. The difference is always passed as a positive value. The sign of the difference is passed using the attribute POTotalValueIncreased of BUS2201, which is set to 'X' if the current total value is higher (see expression 0V\_PO\_POTOTLVALINCR)

**6.8.1.16 Object Type of Previous Document Type**

Class ID	Expression	Result Type
0CF001	0V_PO_PRCRSRDCMNTTYP	Char

The last preceding document is determined. The business object type is passed as the result (for example, BUS2121 for a shopping cart). If no previous document can be found, the result is initial.

**6.8.1.17 Transaction Type with Transaction Number**

Class ID	Expression	Result Type
0CF001	0V_PO_PROCESSID	Char

The result value is the description of the business transaction type (in logon language of the user) concatenated with the purchase order number (leading zeroes are deleted). The complete list of business

transaction types together with the descriptions can be found in table BBPC\_PROC\_TYPE\_T. A purchase order, for example, can have the bus. trans. type ECLP, which has the description 'Local PO' in English.

#### 6.8.1.18 Business Transaction Type

Class ID	Expression	Result Type
0CF001	0V_PO_PROCESSTYPE	Char

The business transaction type from the document header is determined and passed as the result. The complete list of business transaction types together with the descriptions can be found in table BBPC\_PROC\_TYPE\_T. For example, a purchase order can have the bus. trans. type ECLP.

#### 6.8.1.19 Responsible Purchasing Group

Class ID	Expression	Result Type
0CF001	0V_PO_PURCHASNGGROUP	Char

The ID of the purchasing group from the organization management is passed as the result. The format is object type 'O ' + object ID (for example, 'O 50000019').

#### 6.8.1.20 Responsible Purchasing Organization

Class ID	Expression	Result Type
0CF001	0V_PO_PURCHASNGORG	Char

The ID of the purchasing organization from the organization management is passed as the result. The format is object type 'O ' + object ID (for example, 'O 50000019').

#### 6.8.1.21 Status Object Number

Class ID	Expression	Result Type
0CF001	0V_PO_STATUSOBJNR	Char

The header GUID of the current purchase order is passed as the result. This also identifies the status object.

#### 6.8.1.22 Status Profile

Class ID	Expression	Result Type
0CF001	0V_PO_STATUSPROFILE	Char

The status profile for the status object is determined (from table CRM\_JSTO). The header GUID of the purchase order is used. The value is passed as the result.

**6.8.1.23 Time Of Last Output**

Class ID	Expression	Result Type
0CF001	0V_PO_TIMELASTOUTPUT	Char

The date of the last output is passed back as the result. Date and time are concatenated. The resulting string has the format YYYYMMDDHHMMSS. If the document has not been generated, the result is 0.

**6.8.1.24 Purchase Order Total**

Class ID	Expression	Result Type
0CF001	0V_PO_VALUESUM	Char

The values of all items of the purchase order are summed. The sum is passed as the result in the currency of the purchase order.

**6.8.1.25 Version Type (Active Document, 'Normal', Work Version)**

Class ID	Expression	Result Type
0CF001	0V_PO_VERSIONTYPE	Char

The version type of the change version is passed back. If the document is the active version, the result is ' '. Historical versions are marked with 'H', change versions with 'C':

**6.8.1.26 Name of Version Type**

Class ID	Expression	Result Type
0CF001	0V_PO_VERSIONTYPNAME	Char

The text in logon language of the user for the version types given above is displayed. The result in English would be:

- ' ': Purchase order
- 'C': Change version of purchase order
- 'H': Purchase order version

**6.8.1.27 Object Type for Status Management**

Class ID	Expression	Result Type
0CF001	0V_PO_STATUSOBJTYPE	Char

Gets the object type for status management e.g. "POH".

## 6.9 BO Quote

### 6.9.1 Document Attributes

#### 6.9.1.1 Active Statuses Exist

Class ID	Expression	Result Type
OCF001	0B_QT_STATUSTABLE	Bool

The table of active statuses is determined for the quotation. This list is compared to the status passed as constants in the expression. If one of them is found, the result is set to 'TRUE'; otherwise it is 'FALSE'. The constant expressions contain the following statuses:

1. I1009: Held (HELD)
2. I1021: Created (CRTD)
3. I1039: Incomplete (INCM)

The constant expressions have to contain the internal representation of the system status, which should be checked against the attribute table. The list of system statuses can be found in table TJ02T.

#### 6.9.1.2 Person who last changed bid - activity group

Class ID	Expression	Result Type
OCF001	0V_QT_ACTDOCACTGROUP	Char

The last person who changed the quotation is selected from the document header. An instance of the business object BUS4101(EC User) for this user is created and the activity groups to which the user is assigned are determined. If one of the activity groups provided in the additional constants nn\_C4\_C\_ACTGRP\_\* is found in the table of activity groups, the result is set to 'FALSE'. Otherwise it is set to 'TRUE'.

#### 6.9.1.3 Changed By

Class ID	Expression	Result Type
OCF001	0V_QT_CHANGEDBY	Char

The last person who changed the quotation is selected from the document header. The SAP user name is passed as the result.

#### 6.9.1.4 Time of last change of bid

Class ID	Expression	Result Type
OCF001	0V_QT_CHANGEDON	Char



The date of last change is passed back as the result. Date and time are concatenated. The resulting string has the format YYYYMMDDHHMMSS.

#### 6.9.1.5 Name of Creator of Bid

Class ID	Expression	Result Type
0CF001	0V_QT_CREATEDBY	Char

The SAP user name of the creator of the quotation is returned as the result.

#### 6.9.1.6 Time at which Bid was created

Class ID	Expression	Result Type
0CF001	0V_QT_CREATEDON	Char

The date of creation is passed back as the result. Date and time are concatenated. The resulting string has the format YYYYMMDDHHMMSS.

#### 6.9.1.7 Bid last changed by – User Name

Class ID	Expression	Result Type
0CF001	0V_QT_LSTDOCPRCESR	Char

The last person who changed the quotation is selected from the document header. The SAP user name (determined as an attribute of BUS4101) is passed as the result.

#### 6.9.1.8 Send medium for Bid output

Class ID	Expression	Result Type
0CF001	0V_QT_MEDIUM	Char

The type of the output medium (e.g. XML,PRN,FAX,MAI,etc.) for the recipient of the message is passed.

#### 6.9.1.9 Partner Number of Message Recipient

Class ID	Expression	Result Type
0CF001	0V_QT_MSGRECIPKEY	Char

GUID of the partner of type vendor in the quotation is passed as the result.

**6.9.1.10 Description / Name of Bid**

Class ID	Expression	Result Type
0CF001	0V_QT_NAME	Char

The description from the document header is passed as the result.

**6.9.1.11 Transaction Number**

Class ID	Expression	Result Type
0CF001	0V_QT_OBJECT_ID	Char

Quotation number from the document header is passed as the result. The number is 10 digits long and the leading zeroes are not deleted.

**6.9.1.12 Transaction Type With Trans. Number**

Class ID	Expression	Result Type
0CF001	0V_QT_PROCESSID	Char

The result value is the description of the business transaction type (in logon language of the user) concatenated with the quotation number (leading zeroes are deleted). The complete list of business transaction types together with the descriptions can be found in table BBPC\_PROC\_TYPE\_T. A quotation has the bus. trans. type QUOT with description 'Quotation' in English.

**6.9.1.13 Business Transaction Type**

Class ID	Expression	Result Type
0CF001	0V_QT_PROCESSTYPE	Char

The business transaction type of the bid invitation (and not the quotation itself) is returned as result. The bid invitation is determined from the source document (stored in the quotation header).

**6.9.1.14 Status Object Number**

Class ID	Expression	Result Type
0CF001	0V_QT_STATUSOBJNR	Char

The GUID of the quotation is returned.

### 6.9.1.15 Object Type for Status Management

Class ID	Expression	Result Type
0CF001	0V_QT_STATUSOBJTYPE	Char

The type of the status object is determined (from table CRM\_JSTO). Using the header GUID the result for a quotation is 'QOH'. All possible status object types can be found in table TJ03.

### 6.9.1.16 Status Profile

Class ID	Expression	Result Type
0CF001	0V_QT_STATUSPROFILE	Char

The status profile for the status object is determined (from table CRM\_JSTO). The header GUID of the quotation is used. The value is passed as the result.

### 6.9.1.17 Business Partner Number of Supplier who Submitted the Bid

Class ID	Expression	Result Type
0CF001	0V_QT_VENDORBP	Char

The business partner number of the vendor is passed back as the result. The number is found searching for the business partner of type vendor from the partner data. The number has 10 digits, leading zeroes are not deleted.

### 6.9.1.18 Business Partner Number of Supplier who Submitted the Bid

Class ID	Expression	Result Type
0CF001	0V_QT_VENDORBP	Char

The business partner number of the supplier is passed back as the result. The number is found searching for the business partner of type supplier from the partner data. The number has 10 digits, leading zeroes are not deleted.

### 6.9.1.19 Role of Last Changing Person is SAP\_BBP\_STAL\_PURCHASE\_MANAGER

Class ID	Expression	Result Type
0CF001	0V_QT_LSTDCACTGROUP	Bool

If the last document changing person has the role "SAP\_BBP\_STAL\_PURCHASE\_MANAGER" return value is set to 'X', '' otherwise.

### 6.9.1.20 Event of the Object

Class ID	Expression	Result Type
OCF001	OV_QT_EVENT	Char

Constant expression. The return value is always 'TOBEACCEPTED'.

## 6.10 BO RFQ

### 6.10.1 Document Attributes

#### 6.10.1.1 Active Statuses exist

Class ID	Expression	Result Type
OCF001	0B_RQ_STATUSTABLE	Bool

The table of active statuses are determined for the RFx. This list is compared to the status passed as constants in the expression. If one of them is found, the result is set to 'TRUE'; otherwise it is 'FALSE'. The constant expressions contain the following statuses:

- 1.) I1009: Held (HELD)
- 2.) I1021: Created (CRTD)
- 3.) I1039: Incomplete (INCM)

The constant expressions have to contain the internal representation of the system status, which should be checked against the attribute table. The list of system status can be found in table TJ02T.

#### 6.10.1.2 Submission Deadline for Bid

Class ID	Expression	Result Type
OCF001	0D_RQ_QUOTDEADLINE	Date

The submission deadline from the document header is passed as the result. The format is YYYYMMDD.

#### 6.10.1.3 Changed by

Class ID	Expression	Result Type
OCF001	0V_RQ_CHANGEDBY	Char

SAP user name of last person who changed the RFx extracted from document header.

#### 6.10.1.4 Time of last change

Class ID	Expression	Result Type
OCF001	0V_RQ_CHANGEDON	Char

The date of last change is passed back as the result. Date and time are concatenated. The resulting string has the format YYYYMMDDHHMMSS.

### 6.10.1.5 Security Level of Creator

Class ID	Expression	Result Type
0CF001	0V_RQ_CREATUSRSECGRP	Char

The security level is first determined from the user management using the creator of the document. If a value is found, this value is passed as the result. If no value is found, the security levels associated with all the roles to which the user is assigned are identified. The highest security level found is passed as the result.

### 6.10.1.6 Creator User Activity Group

Class ID	Expression	Result Type
0CF001	0V_RQ_CREAUSRACTGRP	Char

The creator of the RfX is selected from the document header. An instance of the business object BUS4101(EC User) for the creator is created and the activity groups to which the user is assigned are determined. If one of the activity groups provided in the additional constants nn\_C4\_C\_ACTGRP\_\* is found in the table of activity groups, the result is set to 'FALSE'. Otherwise it is set to 'TRUE'.

### 6.10.1.7 Name (Bid Invitation or Bid Invitation Change)

Class ID	Expression	Result Type
0CF001	0V_RQ_DOCTYPENAME	Char

The text in logon language of the user for the version types given above is displayed. The result in English for the three types would be:

- ' ': RfX
- 'C': Change version of RfX
- 'H': Version of RfX

### 6.10.1.8 Event of the Object

Class ID	Expression	Result Type
0CF001	0V_RQ_EVENT	Char

Constant expression. The return value is always 'SAVED'.

### 6.10.1.9 Last Changed By – User Name

Class ID	Expression	Result Type
0CF001	0V_RQ_LSTDOCPRCESR	Char

The last person who changed the RFx is selected from the document header. An instance of the business object BUS4101(EC User) for this user is created and the SAP user name is derived. This user name is passed as the result.

#### 6.10.1.10 Send Medium for RFx Output

Attribute not implemented. Do not use expression.

Class ID	Expression	Result Type
0CF001	0V_RQ_MEDIUM	Char

#### 6.10.1.11 Partner Number of Message Receipt

Attribute not implemented. Do not use expression

Class ID	Expression	Result Type
0CF001	0V_RQ_MSGRECIKEY	Char

#### 6.10.1.12 PDO Status

Class ID	Expression	Result Type
0CF001	0V_RQ_PDO_STATUS	Char

Evaluation of workflow process status of actual document version. The possible values are:

- INIT: No process exists or it is still in initial status
- ACTV: Process still active and document is not rejected
- FINI: Document rejected or last level of process completed

#### 6.10.1.13 Business Transaction Type

Class ID	Expression	Result Type
0CF001	0V_RQ_PROCESSTYPE	Char

The business transaction type from the document header is determined and passed as the result. An RFx may have the business transaction type BID. The complete list of business transaction types together with the descriptions can be found in table BBPC\_PROC\_TYPE\_T

#### 6.10.1.14 Responsible Purchasing Group

Class ID	Expression	Result Type
0CF001	0V_RQ_PURCHASNGGROUP	Char

The ID of the purchasing group from the organization management is passed as the result. The format is object type 'O' + object ID (for example, 'O 50000019').

#### 6.10.1.15 Responsible Purchasing Organization

Class ID	Expression	Result Type
0CF001	0V_RQ_PURCHASNGORG	Char

The ID of the purchasing organization from the organization management is passed as the result. The format is object type 'O' + object ID (for example, 'O 50000019').

#### 6.10.1.16 Status Object Number

Class ID	Expression	Result Type
0CF001	0V_RQ_STATUSOBJNR	Char

Document header GUID is passed as the result.

#### 6.10.1.17 Object Type for Status Management

Class ID	Expression	Result Type
0CF001	0V_RQ_STATUSOBJTYPE	Char

The type of the status object is determined (from table CRM\_JSTO). Using the header GUID the result for a bid invitation is 'BIH'. All possible status object types can be found in table TJ03.

#### 6.10.1.18 Status Profile

Class ID	Expression	Result Type
0RFQ	0V_RQ_STATUSPROFILE	Char

The status profile for the status object is determined (from table CRM\_JSTO). The header GUID of the bid invitation is used. The value is passed as the result.

#### 6.10.1.19 Creator User Name

Class ID	Expression	Result Type
0CF001	0V_RQ_CREATUSRNAME	Char

SAP user name of person who created the RFx.



### 6.10.1.20 Transaction Type and Number

Class ID	Expression	Result Type
0CF001	0V_RQ_PROCESSID	Char

Retrieves transaction type and number, for example, "RFX external service 1211749".

## 6.11 BO Shopping Cart

### 6.11.1 Document Attributes

#### 6.11.1.1 Contains External Requirement Item

Class ID	Expression	Result Type
OCF001	0B_SC_CONTSPITEM	Bool

Method not yet supported. Do not use expression. The method will look for service procurement items in the shopping cart. If found, the result will be 'TRUE'; otherwise 'FALSE'.

#### 6.11.1.2 Cost Center List

Class ID	Expression	Result Type
OCF001	0B_SC_COSTCENTERLIST	Bool

The account assignments for all items relevant for the current approval step are selected. The cost centers are collected (not restricted to the account assignment category for cost centers). This list is searched for the cost centers given in the constant expression 9C\_C4\_C\_CONTATTRVAL2. If at least one of the cost centers is found, the value is set to 'TRUE'.

The constant expression can contain a list of cost centers separated by ' ; '. Leading zeroes have to be provided.

#### 6.11.1.3 Item has no Description

Class ID	Expression	Result Type
OCF001	0B_SC_ITEM_DESR_MIS	Bool

Method checks if all items in the shopping cart have a description. If at least one description is missing, the result is set to 'FALSE'. If all items have a description, the result is 'TRUE'.

#### 6.11.1.4 No Limit

Class ID	Expression	Result Type
OCF001	0B_SC_NOLIMIT	Bool

Return value is 'TRUE' if the shopping cart has limit items without limit. Otherwise the result is 'FALSE'.

### 6.11.1.5 Checks Whether Specified Product Category Exists

Class ID	Expression	Result Type
OCF001	0B_SC_PRCAT_EX_CHECK	Bool

The product categories of all items in the shopping cart are checked against the list of product categories in the constant expression 9C\_C4\_C\_PRCAT\_002. A single value has the format 'CatID\Logsys' (ID of product category and logical system of source system. The logical system is necessary to ensure uniqueness). Multiple values are separated by ";". If a product category is found, the result is 'FALSE'. Otherwise it is 'TRUE'.

### 6.11.1.6 An Item Exists from a Free Entry

Class ID	Expression	Result Type
OFB001	0B_SC_XSTFRTXTLNTM	Bool

Rule composed of other expressions:

If spending limit of requester < total value and approval limit of previous agent < total value, the result is set to 'TRUE'; otherwise 'FALSE'.

### 6.11.1.7 Item Exists Without Price

Class ID	Expression	Result Type
OCF001	0B_SC_XSTNOPRCLNTM	Bool

If no item has a price and no limit item exists, the return value is set to 'TRUE'. Otherwise the result is 'FALSE'.

### 6.11.1.8 Item Exists Without Supplier

Class ID	Expression	Result Type
OCF001	0B_SC_XSTNVNDRLNTM	Bool

If there is at least one item which has no partner of type vendor associated to it, the value is set to 'TRUE'. Otherwise the result is 'FALSE'.

### 6.11.1.9 Changed On

Class ID	Expression	Result Type
OCF001	0D_SC_CHANGEDATE	Date

Date of last change of the shopping cart extracted from document header

**6.11.1.10 Created On**

Class ID	Expression	Result Type
OCF001	0D_SC_CREATEDATE	Date

Date of creation of the shopping cart extracted from document header.

**6.11.1.11 Account Assignment Category**

Class ID	Expression	Result Type
OCF001	0V_SC_ACCTASSCAT	Char

Determination of the account category of all line items relevant for the current approval step. For item-based approval, only items to be approved are considered. The account category is only returned if it is unique for all items; otherwise the result variable is initial.

**6.11.1.12 Annual Budget of the Processor**

Class ID	Expression	Result Type
OCF001	0V_SC_ANNUALBUDGET	Float

The annual budget for the requester of the shopping cart is determined. If the budget is defined for another period (monthly, quarterly) the value is annualized. The SAP user of type requester from the partners of the shopping cart is selected. For this user the budget will be searched from three different sources:

- From the user personalization (su01): if a value is found, this value is used.
- If no value is found, the budget from the organization management is derived (field: output limit).
- In addition, a budget is searched from the role assignment of the user.
- If values from both sources are found, the higher value on an annual basis is taken as the budget.

The annual budget is converted to the currency of the document header using type rate 'M' and the current local date. It is also possible to pass a different currency as additional constant expression (nn\_C5\_C\_WHG\_nnn). The spending limit will then be converted into this currency.

**6.11.1.13 Approval Limit of the Processor**

Class ID	Expression	Result Type
OCF001	0V_SC_APPROVAL_LIMIT	Float

In case of dynamic approval, the approval limit of the current approver in document currency is determined. It is possible to pass a different currency as additional constant expression (nn\_C5\_C\_WHG\_nnn). The previous approval limit will then be converted into this currency using type rate 'M' and the current local date.

**6.11.1.14 Catalog ID (if the same for all items)**

Class ID	Expression	Result Type
0CF001	0V_SC_CATALOGID	Char

Determination of the catalog ID of all line items relevant for the current approval step. For item-based approval only items to be approved are considered. The catalog ID is only returned if it is unique for all items; otherwise the result variable is initial.

**6.11.1.15 Changed By**

Class ID	Expression	Result Type
0CF001	0V_SC_CHANGEDBY	Char

SAP user name of last person who changed the shopping cart extracted from document header.

**6.11.1.16 Cost Center List**

Class ID	Expression	Result Type
0CF001	0V_SC_COSTCENTERLIST	Char

The account assignments for all items relevant for the current approval step are selected. The cost centers are collected (not restricted to the account assignment category for cost centers). This list is searched for the cost centers given in the constant expression 0C\_C4\_C\_COSTCENTER. If the cost center is found, the value is passed as the result. This expression is only an example of how to use the attribute COSTCENTERLIST of the shopping cart, because the constant expression is not in the user space and cannot be changed.

**6.11.1.17 Document Creator**

Class ID	Expression	Result Type
0CF001	0V_SC_CREATEDBY	Char

SAP user name of person who created the shopping cart, extracted from document header.

**6.11.1.18 Document Creator with Leading US**

Class ID	Expression	Result Type
0CF001	0V_SC_CREATEDBY_US	Char

SAP user name of person who created the shopping cart, extracted from document header. Prefix US is added, for example, USSAPUSER.

**6.11.1.19 Currency**

Class ID	Expression	Result Type
0CF001	0V_SC_CURRENCY	Char

Currency of shopping cart, extracted from document header. The currencies of the items are not considered.

**6.11.1.20 Document Type in Backend System**

Class ID	Expression	Result Type
0CF001	0V_SC_DOCTYPE	Char

Determination of the document type of the requisition in the back-end for all items of type material and relevant for the current approval step. In addition, the document type from the header is determined. The result is not initial if all selected items and the header have the same document type.

**6.11.1.21 Document Name**

Class ID	Expression	Result Type
0CF001	0V_SC_DOCUMENTNAME	Char

Document name extracted from document header.

**6.11.1.22 Event of the Object**

Class ID	Expression	Result Type
0CF001	0V_SC_EVENT	Char

Constant expression. The return value is always 'SAVED'.

**6.11.1.23 Gross Total Value**

Class ID	Expression	Result Type
0CF001	0V_SC_GROSSTOTLVLU	Float

Gross value of shopping cart in the currency of the document header. Tax amounts in a different currency are converted to the shopping cart header currency with rate type 'M' and current local date.

**6.11.1.24 Shopping Cart Limit**

Class ID	Expression	Result Type
OCF001	0V_SC_LIMIT	Float

Sum of limits of limit items in the shopping cart in document header currency. If a limit item has no limit, the expected value is used. Amounts in a different currency are converted to the shopping cart header currency with rate type 'M' and current local date. (Only items relevant for the current approval step are considered.)

**6.11.1.25 Most Expensive Item**

Class ID	Expression	Result Type
OCF001	0V_SC_LNTMMSTXPNSV	Float

Highest value (price\*quantity) of all material and service items in the shopping cart in document header currency. Amounts in a different currency are converted to the shopping cart header currency with rate type 'M' and current local date. (Only items relevant for the current approval step are considered.)

**6.11.1.26 Material Group**

Class ID	Expression	Result Type
OCF001	0V_SC_MTERILGROUP	Char

If all items of the shopping cart have the same product category ID (restricted to the first nine characters), this ID is passed as the result. Otherwise, the result is initial. (Only items relevant for the current approval step are considered.)

**6.11.1.27 Most Expensive Material**

Class ID	Expression	Result Type
OCF001	0V_SC_MTRLMSTXPNSV	Float

The item with the highest price/unit of all items of type 'material(01)' in the shopping cart is selected (only items relevant for the current approval step are considered; service and limit items are not considered). The price/unit is converted to the document header currency and passed as the result. The conversion is done with rate type 'M' for the current local date.

**6.11.1.28 Plant (if the same for all items)**

Class ID	Expression	Result Type
OCF001	0V_SC_PLANT	Char

If all items of type 'material(01)' relevant for the current approval step of the shopping cart have the same plant, this plant is passed as the result. Otherwise, the result is initial.

#### 6.11.1.29 Responsible Purch. Organization

Class ID	Expression	Result Type
0CF001	0V_SC_PRCHSNGRGNZTN	Char

If all items relevant for the current approval step of the shopping cart have the same purchasing organization, this value is passed as the result. Otherwise, the result is initial.

#### 6.11.1.30 Approval Process of the Previous Processor

Class ID	Expression	Result Type
0CF001	0V_SC_PREVAPPROVLIMI	Char

The previous approval step in the workflow is analyzed. The approval limit and currency of the previous approver is determined. A different currency can be passed as an additional constant expression (nn\_C5\_C\_WHG\_nnn). The previous approval limit will then be converted into this currency using type rate 'M' and the current local date.

#### 6.11.1.31 Responsible Purchasing Group

Class ID	Expression	Result Type
0CF001	0V_SC_PURCHSNGGRUP	Char

If all items relevant for the current approval step of the shopping cart have the same purchasing group, this value is passed as the result. Otherwise, the result is initial.

#### 6.11.1.32 Organizational Unit

Class ID	Expression	Result Type
0CF001	0V_SC_RPORTNGLNUNT	Char

For the user who created the shopping cart (extracted from the header) the organizational units (object type O) are determined. The ID (8 digits, numerical) of the lowest level to which the user is assigned is returned. If no unit is found, the return value is initial. The hierarchy structure can be displayed with transaction pposa\_bbp.



**6.11.1.33 Name of Requester**

Class ID	Expression	Result Type
0CF001	0V_SC_RQUESTONRNM	Char

SAP user name of requester of shopping cart.

**6.11.1.34 Requester – User ID**

Class ID	Expression	Result Type
0CF001	0V_SC_RQUESTONRUSRNAM	Char

The requester of the shopping cart is selected from the shopping cart partners. An instance of the business object BUS4101(EC User) for the requester is created and the SAP user name is derived. This user name is passed as the result.

**6.11.1.35 SC Value in User Budget Scenario**

Class ID	Expression	Result Type
0CF001	0V_SC_SCVALUESPENT	Float

The total value spent of the shopping cart in document header currency is passed back as the result. The total value spent can be changed by the BAdI BBP\_SC\_VALUE\_GET.

**6.11.1.36 SL Manager User – User ID**

Class ID	Expression	Result Type
0CF001	0V_SC_SLMANAGERUSER	Char

The requester of the shopping cart is selected from the shopping cart partners. An instance of the business object BUS4101(EC User) for the requester is created and the list of approvers is determined (as attributes of business object BUS4101). For each approver found, the approval limit is compared to the total value of the shopping cart. If the approval limit is in a currency different from the document header currency, the approval limit is converted with rate type 'M' for the current local date. The approver with the limit higher and closest to the total value of the shopping cart is selected. For this approver the SAP user name is passed as the result. If no approver with a limit higher than the total value is found, the result value is initial.

### 6.11.1.37 Spending Limit of the Processor

Class ID	Expression	Result Type
0CF001	0V_SC_SPENDING_LIMIT	Float

The spending limit for the requester of the shopping cart is determined. The SAP user of type requester from the partners of the shopping cart is selected. For this user the spending limit will be searched for from three different sources:

- From the user personalization (su01): if a value is found, this value is used.
- If no value is found, the spending limit from the organization management is derived (field: output limit).
- In addition, a spending limit is searched from the role assignment of the user.
- If values from both sources are found, the higher value is taken as the spending limit. The spending limit is converted to the currency of the document header using type rate 'M' and the current local date. A different currency can be passed as an additional constant expression (nn\_C5\_C\_WHG\_nnn). The spending limit will then be converted to this currency.

### 6.11.1.38 Purchase Limit of Requester

Class ID	Expression	Result Type
0CF001	0V_SC_SPNDNGLMTRQST	Float

Identical to 0V\_SC\_SPENDING\_LIMIT

### 6.11.1.39 Overall Value

Class ID	Expression	Result Type
0CF001	0V_SC_TOTALVALUE	Float

The total value of the shopping cart in document header currency is passed back as the result.

### 6.11.1.40 Tax Amount

Class ID	Expression	Result Type
0CF001	0V_SC_TOTLTXMOUNT	Float

The sum of all taxes on item level in document header currency is computed. If the currency differs from the shopping cart currency, the value is converted using rate type 'M' and the current local date.

**6.11.1.41 Budget Defined for User**

Class ID	Expression	Result Type
OCF001	0V_SC_USRBUDGTDFND	Float

The requester of the shopping cart is selected from the shopping cart partners. An instance of the business object BUS4101 (EC User) for the requester is created. For the requester the defined budget is determined from the Org. management and the assigned roles. If values are found from both sources, the higher value is chosen taking into account different validity periods and different currencies. The budget is converted to the currency of the shopping cart using type rate 'M' and the current local date.

**6.11.1.42 Procurement Expenditure**

Class ID	Expression	Result Type
OCF001	0V_SC_USRBUDGTSPNT	Float

The requester of the shopping cart is selected from the shopping cart partners. An instance of the business object BUS4101 (EC User) for the requester is created. The total budget spent is determined for the user in the current budget period as defined (month, quarter or year). The current shopping cart is included with the total value spent (see expression 0V\_SC\_SCVALUESPENT). The value is converted to the currency of the shopping cart using type rate 'M' and the current local date.

**6.11.1.43 An Item Exists from a Free Entry**

Class ID	Expression	Result Type
OCF001	0V_SC_XSTFRTXTLNTM	Char

If at least one item relevant for the current approval step is a free text item, the return value is set to 'X'. Otherwise, the result is ' '.

**6.11.1.44 Contains Position with Accounting Type "AS"**

Class ID	Expression	Result Type
OCF001	0B_SC_ACCTASSCAT_AS	Bool

This expression returns:

- 'X', if Shopping Cart contains at least one item with accounting type "AS"
- ' ', otherwise

**6.11.1.45 Contains Position with Accounting Type "CC"**

Class ID	Expression	Result Type
OCF001	0B_SC_ACCTASSCAT_CC	Bool

This expression returns:

- 'X', if Shopping Cart contains at least one item with accounting type "CC"
- '', otherwise

#### 6.11.1.46 Contains Position with Accounting Type "CC" or "AS"

Class ID	Expression	Result Type
OCF001	0B_SC_CONTACCTASSCAT	Bool

This expression returns:

- 'X', if Shopping Cart contains at least one item with accounting type "CC" or "AS"
- '', otherwise

#### 6.11.1.47 Value of Decision Set is Bigger than Lowest Approval Limit

Class ID	Expression	Result Type
OCF001	0B_SC_DSVALGTAPPLIM	Char

If the value of one of the available decision sets is greater then the lowest approval limit of responsible approver the return value is set to 'X'. Otherwise, the result is ' '.

#### 6.11.1.48 Position with Follow-On Document RFx Exists

Class ID	Expression	Result Type
OCF001	0B_SC_XSTFLLWNRFAQ	Bool

If at least for one position a follow-on document of type RFx exists, the return value is set to 'X'. Otherwise, the result is ' '.

#### 6.11.1.49 Name of Requester

*The expression is obsolete. Use expression 0V\_SC\_RQUESTONRUSRNAM instead.*

Class ID	Expression	Result Type
OCF001	0V_SC_RQUESTONRUSR	Float

#### 6.11.1.50 SL Manager User – User Name

*The expression is obsolete. Use expression 0V\_SC\_RQUESTONRUSRNAM instead.*

Class ID	Expression	Result Type
OCF001	0V_SC_RQUESTONRUSR	Float

## 6.11.2 Level Evaluation

### 6.11.2.1 Spending Limit Approval One - Step

Class ID	Expression	Result Type
0CF001	0B_SC_SPENDLIMAPPR01	Bool

Rule composed of other expressions. This expression returns:

- 'TRUE', if spending limit of requester < total value and approval limit of agent > total value and no valid process levels exists.
- 'FALSE', otherwise.

### 6.11.2.2 Spending Limit Approval N - Step

Class ID	Expression	Result Type
0FB001	0B_SC_SPENDLIMAPPR02	Bool

Rule composed of other expressions. This expression returns:

- 'TRUE', if spending limit of requester < total value and approval limit of previous agent < total value
- 'FALSE', otherwise.

### 6.11.2.3 Approval process based on Reporting Line and Spending Limit

Class ID	Expression	Result Type
0FB001	0B_SC_SPENDLIMAPPR03	Bool

This expression returns:

- 'TRUE', as long as the value of the decision set is small as the approval limit of agent on current reporting line level.
- 'FALSE', otherwise

## 7 Appendix B: Approver Determination Strategies in the Standard

### 7.1 Overview

The standard BAdI implementations are documented in the following paragraphs for the various document types in SRM. For each document type the available implementations are described. In many cases the implementing classes are not object-specific but are used for several objects. In these cases the documentation is moved to a separate chapter and can be directly accessed through a link.

All implementations implement at least the method to assign the area of responsibility to the items and the method to derive the responsible approvers for the given area of responsibility. The methods to support the field Responsibility Resolver Parameter are only implemented when the activating filter value Responsibility Resolver Name requires an entry in the Parameter field in the process schema. The method to provide a fallback agent is not implemented for the standard BAdI implementations.

### 7.2 Shopping Cart (BUS2121)

The enhancement implementation for the Shopping Cart is /SAPSRM/BD\_WF\_AGENTS\_SC. For this enhancement implementation seven BAdI implementations are defined. Which implementation is called at runtime is decided through the filter values. The filter values are set up using the document type (BUS2121 for shopping carts) and the Responsibility Resolver Name, which is derived from the process level at runtime. The possible values for the latter are defined in the workflow customizing (see chapter [Add Responsibility Resolver Names](#)). The following chapters describe the filter values and the corresponding BAdI implementation.

#### 7.2.1 Implementation - Specify Employee for Approval

The BAdI implementation /SAPSRM/BD\_WF\_SC\_RR\_EMPLOYEE returns an employee (Business Partner) as responsible agent for shopping cart approval in the relevant process level. The document is not split into decision sets, that is, all items are assigned to the same agent.

This BAdI has the following filter values:

Document type	=	BUS2121	AND
Responsibility Resolver Name	=	RR_EMPLOYEE	

The Responsibility Resolver Parameter is used to specify the employee (Business Partner) ID of the person who should be assigned as agent.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_EMPLO. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_EMPLOYEE](#).

#### 7.2.2 Implementation - Get Requester's Line Manager for Approval

The BAdI implementation /SAPSRM/BD\_WF\_SC\_RR\_MANAGER returns all managers of the shopping cart's requester of the previous level as responsible agents for shopping cart approval in the relevant process level. The document is not split into decision sets, that is, all items are assigned to the same agents.

This BAdI has the following filter values:

Document type	=	BUS2121	AND
Responsibility Resolver Name	=	RR_MANAGER	

The Responsibility Resolver Parameter is not used.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_MANAGER. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_MANAGER](#).

### 7.2.3 Implementation - Get Manager of Previous Approver for Approval

The BAdI implementation /SAPSRM/BD\_WF\_SC\_RR\_MANOFMA returns all managers of approvers of the previous level as responsible agents for shopping cart approval in the relevant process level. The method assigns all items of a decision set to the same area, thus respecting the existing decision sets for the document.

This BAdI has the following filter values:

Document type	=	BUS2121	AND
Responsibility Resolver Name	=	RR_MANAGER_OF_MANAGER	

The Responsibility Resolver Parameter is not used.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_MANOFMA. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_MANAGER\\_OF\\_MANAGER](#).

### 7.2.4 Implementation - Get Purchasers of all Purchasing Groups for Approval

The BAdI implementation /SAPSRM/BD\_WF\_SC\_RR\_PGRP returns all purchasers of an item's purchasing group as responsible agents for shopping cart approval in the relevant process level. The document is split into decision sets according to the assigned purchasing group.

This BAdI has the following filter values:

Document type	=	BUS2121	AND
Responsibility Resolver Name	=	RR_PURCHASING_GROUP	

The Responsibility Resolver Parameter is not used.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_PGRP\_SC.

#### Method GET\_AREA\_TO\_ITEM\_MAP:

First it is checked that the purchasing document is a shopping cart. For other document types an exception is raised.

The purchasing groups of the document is determined. In order to do this an instance of the purchasing document is created to be able to access the attributes. From the organizational data of all the items the assigned purchasing groups are determined. The purchasing group acts as leading object ID and for each purchasing group found in the document an area of responsibility is created. This is done using the class /SAPSRM/CL\_WF\_AREA\_PURCH\_GRP.

All items are assigned to the area of responsibility, which has the corresponding purchasing group as leading object. The document is therefore split into decision sets, that is, for each purchasing group a decision set is created.

#### Method GET\_APPROVERS\_BY\_AREA\_GUID:

The responsible agents are derived by the method /SAPSRM/CL\_WF\_AREA\_PURCH\_GRP~GET\_RESPONSIBLE\_APPROVERS.

For the purchasing group (as stored in the leading object) the attributes from the organizational management are read and the user names of all the users, that is, purchasers, belonging to the purchasing group are returned. These user names are passed back as responsible agents.

Method GET\_FALLBACK\_AGENTS:

Not implemented.

Method GET\_RESP\_RESOLVER\_PARAMETER\_F4:

Not implemented because no parameter is required.

Method CHECK\_RESP\_RESOLVER\_PARAMETER:

Not implemented.

## 7.2.5 Implementation - Get all Employees of Specified Department for Approval

The BAdI implementation /SAPSRM/BD\_WF\_SC\_RR\_RLUNIT returns all employees of a specific reporting-line unit as responsible agents for shopping cart approval in the relevant process level. The document is not split into decision sets, that is, all items are assigned to the same agents.

This BAdI has the following filter values:

Document type	=	BUS2121	AND
Responsibility Resolver Name	=	RR_REPORTING_LINE_UNIT	

The Responsibility Resolver Parameter is used to specify the reporting line unit from which the employees should be assigned as responsible agents.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_RLUNIT. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_REPORTING\\_LINE\\_UNIT](#).

## 7.2.6 Implementation - Get all Users of Specified Role for Approval

The BAdI implementation /SAPSRM/BD\_WF\_SC\_RR\_ROLE returns all user accounts of a specific user role as responsible agents for shopping cart approval in the relevant process level. The document is not split into decision sets, that is, all items are assigned to the same agents.

This BAdI has the following filter values:

Document type	=	BUS2121	AND
Responsibility Resolver Name	=	RR_ROLE	

The Responsibility Resolver Parameter is used to specify the user role from which the users should be assigned as responsible agents.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_ROLE. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_ROLE](#).

## 7.2.7 Implementation - Get Users of Spending Limit Chain for Approval

The BAdI implementation /SAPSRM/BD\_WF\_SC\_RR\_SL returns all spending-limit approvers (as defined in the PPOMA\_BBP attribute "Approver for Value Limit", abbreviation "SLAPPROVAL") of the shopping cart's requester for approval. The selected process level is representing a placeholder for a process-level sequence that will be calculated during process start by the BAdI implementation /SAPSRM/BD\_WF\_CONFIG\_SC\_SL. The sequence (defined by the PPOMA\_BBP attribute SLAPPROVER) ends when the approver has an approval limit that exceeds the document's total value.

This BAdI has the following filter values:



Document type = BUS2121 AND  
Responsibility Resolver Name = RR\_SPENDING\_LIMIT\_APPROVER

The Responsibility Resolver Parameter is not used.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_ADHOC.

Method GET\_AREA\_TO\_ITEM\_MAP:

The method creates one area of responsibility of class /SAPSRM/CL\_WF\_AREA\_ADHOC, which has the current process level GUIID as leading object ID.

If the process level is of type 'Approval with completion' all items are assigned to the created area of responsibility. In all other cases it is first checked if a mapping of the items of the document exist to the current process level. This mapping is retrieved from the database with the function module /SAPSRM/WF\_CORE\_ADHOC\_READ from the table /SAPSRM/D\_WF\_015 and /SAPSRM/D\_WF\_016.

These tables can be filled using e.g. an implementation of the BAdI /SAPSRM/BD\_WF\_PROCESS\_CONFIG which creates new process levels at runtime. As for the standard implementations of this BAdI (/SAPSRM/BD\_WF\_CONFIG\_SC\_SL and /SAPSRM/BD\_WF\_CONFIG\_PO\_SL) they do not save a mapping of items to the process level, so that the all items are assigned to the same area of responsibility.

Method GET\_APPROVERS\_BY\_AREA\_GUID:

The responsible agents are derived by the method /SAPSRM/CL\_WF\_AREA\_ADHOC~GET\_RESPONSIBLE\_APPROVERS.

The leading object ID is the GUID of the actual process level. With this GUID the mapping of the agents to the process level is retrieved from the database tables /SAPSRM/D\_WF\_015 and /SAPSRM/D\_WF\_016 using the function module /SAPSRM/WF\_CORE\_ADHOC\_READ. All users from the mapping table are then assigned as responsible agents.

The information has to be saved when the process levels are created. For the standard implementations the spending limit approvers have been saved and which are retrieved here. Please refer to [Implemented logic for shopping cart](#) for the implemented logic.

Method GET\_FALLBACK\_AGENTS:

Not implemented.

Method GET\_RESP\_RESOLVER\_PARAMETER\_F4:

Not implemented because no parameter is required.

Method CHECK\_RESP\_RESOLVER\_PARAMETER:

Not implemented.

## 7.3 Purchase Order (BUS2201)

The enhancement implementation for the purchase order is /SAPSRM/BD\_WF\_AGENTS\_PO. For this enhancement implementation seven BAdI implementations are defined. Which implementation is called at runtime is decided through the filter values. The filter values are set up using the document type (BUS2201 for purchase orders) and the Responsibility Resolver Name. The possible values for the latter are defined in the workflow customizing (see chapter [Add Responsibility Resolver Names](#)). The following chapters describe the filter values and the corresponding BAdI implementation.

### 7.3.1 Implementation - Specify Employee for Approval

The BAdI implementation /SAPSRM/BD\_WF\_PO\_RR\_EMPLOYEE returns an employee (Business Partner) as responsible agent for approval in the relevant process level. The document is not split into decision sets, that is, all items are assigned to the same agent.

This BAdI has the following filter values:

Document type	=	BUS2201	AND
Responsibility Resolver Name	=	RR_EMPLOYEE	

The Responsibility Resolver Parameter is used to specify the employee (Business Partner) ID of the person who should be assigned as agent.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_EMPLO. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_EMPLOYEE](#).

### 7.3.2 Implementation - Get Requester's Line Manager for Approval

The BAdI implementation /SAPSRM/BD\_WF\_PO\_RR\_MANAGER returns all managers of the document's requester of the previous level as responsible agents for approval in the relevant process level. The document is not split into decision sets, that is, all items are assigned to the same agent.

This BAdI has the following filter values:

Document type	=	BUS2201	AND
Responsibility Resolver Name	=	RR_MANAGER	

The Responsibility Resolver Parameter is not used.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_MANAGER. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_MANAGER](#).

### 7.3.3 Implementation - Get Manager of Previous Approver for Approval

The BAdI implementation /SAPSRM/BD\_WF\_PO\_RR\_MANOFMA returns all managers of approvers of the previous level as responsible agents for approval in the relevant process level. The method assigns all items of a decision set to the same area, thus respecting the existing decision sets for the document.

This BAdI has the following filter values:

Document type	=	BUS2201	AND
Responsibility Resolver Name	=	RR_MANAGER_OF_MANAGER	

The Responsibility Resolver Parameter is not used.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_MANOFMA. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_MANAGER\\_OF\\_MANAGER](#).

### 7.3.4 Implementation - Get the Manager of the Purchasing Group for Approval

The BAdI implementation /SAPSRM/BD\_WF\_PO\_RR\_PURMGR returns all managers of the document's purchasing group as responsible agents for approval in the relevant process level. The document is not split into decision sets, that is, all items are assigned to the same agents.

This BAdI has the following filter values:

Document type	=	BUS2201	AND
Responsibility Resolver Name	=	RR_PURCHASING_MANAGER	

The Responsibility Resolver Parameter is not used.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_PURMGR. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_PURCHASING\\_MANAGER](#).

### 7.3.5 Implementation - Get all Employees of Specified Department for Approval

The BAdI implementation /SAPSRM/BD\_WF\_PO\_RR\_RLUNIT returns all employees of a specific reporting-line unit as responsible agents for approval in the relevant process level. The document is not split into decision sets, that is, all items are assigned to the same agents.

This BAdI has the following filter values:

Document type	=	BUS2201	AND
Responsibility Resolver Name	=	RR_REPORTING_LINE_UNIT	

The Responsibility Resolver Parameter is used to specify the reporting line unit from which the employees should be assigned as responsible agents.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_RLUNIT. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_REPORTING\\_LINE\\_UNIT](#).

### 7.3.6 Implementation - Get all Users of Specified Role for Approval

The BAdI implementation /SAPSRM/BD\_WF\_PO\_RR\_ROLE returns all user accounts of a specific user role as responsible agents for approval in the relevant process level. The document is not split into decision sets, that is, all items are assigned to the same agents.

This BAdI has the following filter values:

Document type	=	BUS2201	AND
Responsibility Resolver Name	=	RR_ROLE	

The Responsibility Resolver Parameter is used to specify the user role from which the users should be assigned as responsible agents.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_ROLE. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_ROLE](#).

### 7.3.7 Implementation - Get Users of Spending Limit Chain for Approval

The BAdI implementation /SAPSRM/BD\_WF\_PO\_RR\_SL returns all spending-limit approvers (as defined in the PPOMA\_BBP attribute "Approver for Value Limit", abbreviation "SLAPPROVAL") of the document's

requester for approval. The selected process level is representing a placeholder for a process-level sequence that will be calculated during process start by BAdI implementation /SAPSRM/BD\_WF\_CONFIG\_PO\_SL. The sequence (defined by the PPOMA\_BBP attribute SLAPPROVER) ends when the approver has an approval limit that exceeds the document's total value.

This BAdI has the following filter values:

Document type = BUS2201 AND  
Responsibility Resolver Name = RR\_SPENDING\_LIMIT\_APPROVER

The Responsibility Resolver Parameter is not used.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_ADHOC. The implemented logic is the same as for the shopping cart and the documentation can be found in [Implementation /SAPSRM/BD\\_WF\\_SC\\_RR\\_SL - Get Users of SL Chain for Approval](#).

## 7.4 Purchase Contract (BUS2000113)

The enhancement implementation for the purchase contract is /SAPSRM/BD\_WF\_AGENTS\_CTR. For this enhancement implementation six BAdI implementations are defined. Which implementation is called at runtime is decided through the filter values. The filter values are set up using the document type (BUS2000113 for purchase contracts) and the Responsibility Resolver Name. The possible values for the latter are defined in the workflow customizing (see chapter [Add Responsibility Resolver Names](#)). The following chapters describe the filter values and the corresponding BAdI implementation.

### 7.4.1 Implementation - Specify Employee for Approval

The BAdI implementation /SAPSRM/BD\_WF\_CTR\_RR\_EMPLOYEE returns an employee (Business Partner) as responsible agent for approval in the relevant process level. The document is not split into decision sets, that is, all items are assigned to the same agent.

This BAdI has the following filter values:

Document type	=	BUS2000113	AND
Responsibility Resolver Name	=	RR_EMPLOYEE	

The Responsibility Resolver Parameter is used to specify the employee (Business Partner) ID of the person who should be assigned as agent.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_EMPLO. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_EMPLOYEE](#).

### 7.4.2 Implementation - Get Manager of Document Owner for Approval

The BAdI implementation /SAPSRM/BD\_WF\_CTR\_RR\_MANAGER returns all managers of the document's requester of the previous level as responsible agents for approval in the relevant process level. The document is not split into decision sets, that is, all items are assigned to the same agent.

This BAdI has the following filter values:

Document type	=	BUS2000113	AND
Responsibility Resolver Name	=	RR_MANAGER	

The Responsibility Resolver Parameter is not used.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_MANAGER. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_MANAGER](#).

### 7.4.3 Implementation - Get Manager of Previous Approver for Approval

The BAdI implementation /SAPSRM/BD\_WF\_CTR\_RR\_MANOFMA returns all managers of approvers of the previous level as responsible agents for approval in the relevant process level. The method assigns all items of a decision set to the same area, thus respecting the existing decision sets for the document.

This BAdI has the following filter values:

Document type	=	BUS2000113	AND
Responsibility Resolver Name	=	RR_MANAGER_OF_MANAGER	

The Responsibility Resolver Parameter is not used.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_MANOFMA. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_MANAGER\\_OF\\_MANAGER](#).

#### 7.4.4 Implementation - Get Manager of the Purchasing Group for Approval

The BAdI implementation /SAPSRM/BD\_WF\_CTR\_RR\_PURMGR returns all managers of the document's purchasing group as responsible agents for approval in the relevant process level. The document is not split into decision sets, that is, all items are assigned to the same agents.

This BAdI has the following filter values:

Document type	=	BUS2000113	AND
Responsibility Resolver Name	=	RR_PURCHASING_MANAGER	

The Responsibility Resolver Parameter is not used.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_PURMGR. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_PURCHASING\\_MANAGER](#).

#### 7.4.5 Implementation - Get all Employees of Specified Department for Approval

The BAdI implementation /SAPSRM/BD\_WF\_CTR\_RR\_RLUNIT returns all employees of a specific reporting-line unit as responsible agent for approval in the relevant process level. The document is not split into decision sets, that is, all items are assigned to the same agents.

This BAdI has the following filter values:

Document type	=	BUS2000113	AND
Responsibility Resolver Name	=	RR_REPORTING_LINE_UNIT	

The Responsibility Resolver Parameter is used to specify the reporting line unit from which the employees should be assigned as responsible agents.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_RLUNIT. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_REPORTING\\_LINE\\_UNIT](#).

#### 7.4.6 Implementation - Get all Users of Specified Role for Approval

The BAdI implementation /SAPSRM/BD\_WF\_CTR\_RR\_ROLE returns all user accounts of a specific user role as responsible agents for approval in the relevant process level. The document is not split into decision sets, that is, all items are assigned to the same agents.

This BAdI has the following filter values:

Document type	=	BUS2000113	AND
Responsibility Resolver Name	=	RR_ROLE	

The Responsibility Resolver Parameter is used to specify the user role from which the users should be assigned as responsible agents.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_ROLE. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_ROLE](#).

## 7.5 Confirmation (BUS2203)

The enhancement implementation for the confirmation is /SAPSRM/BD\_WF\_AGENTS\_CONF. For this enhancement implementation seven BAdI implementations are defined. Which implementation is called at runtime is decided through the filter values. The filter values are set up using the document type (BUS2203 for confirmations) and the Responsibility Resolver Name. The possible values for the latter are defined in the workflow customizing (see chapter [Add Responsibility Resolver Names](#)). The following chapters describe the filter values and the corresponding BAdI implementation.

### 7.5.1 Implementation - Specify Employee for Approval

The BAdI implementation /SAPSRM/BD\_WF\_CONF\_RR\_EMPLOYEE returns an employee (Business Partner) as responsible agent for approval in the relevant process level. The document is not split into decision sets, that is, all items are assigned to the same agent.

This BAdI has the following filter values:

Document type	=	BUS2203	AND
Responsibility Resolver Name	=	RR_EMPLOYEE	

The Responsibility Resolver Parameter is used to specify the employee (Business Partner) ID of the person who should be assigned as agent.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_EMPLO. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_EMPLOYEE](#).

### 7.5.2 Implementation - Get Manager of Document Owner for Approval

The BAdI implementation /SAPSRM/BD\_WF\_CONF\_RR\_MANAGER returns all managers of the document's requester of the previous level as responsible agents for approval in the relevant process level. The document is not split into decision sets, that is, all items are assigned to the same agent.

This BAdI has the following filter values:

Document type	=	BUS2203	AND
Responsibility Resolver Name	=	RR_MANAGER	

The Responsibility Resolver Parameter is not used.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_MANAGER. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_MANAGER](#).

### 7.5.3 Implementation - Get Manager of Previous Approver for Approval

The BAdI implementation /SAPSRM/BD\_WF\_CONF\_RR\_MANOFMA returns all managers of approvers of the previous level as responsible agents for approval in the relevant process level. The method assigns all items of a decision set to the same area, thus respecting the existing decision sets for the document.

This BAdI has the following filter values:

Document type	=	BUS2203	AND
Responsibility Resolver Name	=	RR_MANAGER_OF_MANAGER	

The Responsibility Resolver Parameter is not used.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_MANOFMA. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_MANAGER\\_OF\\_MANAGER](#).

## 7.5.4 Implementation - Get Manager of the Purch. Group for Approval

The BAdI implementation /SAPSRM/BD\_WF\_CONF\_RR\_PURMGR returns all managers of the document's purchasing group as responsible agents for approval in the relevant process level. The document is not split into decision sets, that is, all items are assigned to the same agents.

This BAdI has the following filter values:

Document type = BUS2203 AND  
Responsibility Resolver Name = RR\_PURCHASING\_MANAGER

The Responsibility Resolver Parameter is not used.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_PURMGR. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_PURCHASING\\_MANAGER](#).

## 7.5.5 Implementation - Get Owner of Source Document for Approval

The BAdI implementation /SAPSRM/BD\_WF\_CONF\_RR\_P\_OWNER returns the document owner of the source document as responsible agents for approval in the relevant process level. The document is not split into decision sets.

This BAdI has the following filter values:

Document type = BUS2203 AND  
Responsibility Resolver Name = CONF\_RR\_SOURCE\_DOCUMENT\_OWNER

The Responsibility Resolver Parameter is not used.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_POWNER\_CONF.

### Method GET\_AREA\_TO\_ITEM\_MAP:

One instance of the area of responsibility of type /SAPSRM/CL\_WF\_AREA\_EMPLO\_LIST is created. All items are assigned to this one instance. As leading object the table of responsible users of the confirmation is used. The logic to retrieve the responsible users is as follows:

- Creator of confirmation if user is a professional purchaser
- Else: Partner of type requestor:
  - If exactly one and the user is the creator and belongs to the purchasing organization
  - If not the creator, but an user belonging to the company
  - If not the creator and external, the responsible persons of the source document (purchase order): this is the requestor if the person belongs to the purch. Organization or all purchasers of the relevant purchasing groups in the purchase order

### Method GET\_APPROVERS\_BY\_AREA\_GUID:

The responsible agents are derived by the method /SAPSRM/CL\_WF\_AREA\_EMPLO\_LIST~GET\_RESPONSIBLE\_APPROVERS

The list of approvers is stored as leading object for the area of responsibility. This list is retrieved and all users in the list are assigned as responsible agents.

### Method GET\_FALLBACK\_AGENTS:

Not implemented.



Method GET\_RESP\_RESOLVER\_PARAMETER\_F4:

Not implemented because no parameter is required.

Method CHECK\_RESP\_RESOLVER\_PARAMETER:

Not implemented.

## 7.5.6 Implementation - Get all Employees of Specified Department for Approval

The BAdI implementation /SAPSRM/BD\_WF\_CONF\_RR\_RLUNIT returns all employees of a specific reporting-line unit as responsible agents for approval in the relevant process level. The document is not split into decision sets, that is, all items are assigned to the same agents.

This BAdI has the following filter values:

Document type	=	BUS2203	AND
Responsibility Resolver Name	=	RR_REPORTING_LINE_UNIT	

The Responsibility Resolver Parameter is used to specify the reporting line unit from which the employees should be assigned as responsible agents.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_RLUNIT. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_REPORTING\\_LINE\\_UNIT](#).

## 7.5.7 Implementation - Get all Users of Specified Role for Approval

The BAdI implementation /SAPSRM/BD\_WF\_CONF\_RR\_ROLE returns all user accounts of a specific user role as responsible agents for approval in the relevant process level. The document is not split into decision sets, that is, all items are assigned to the same agents.

This BAdI has the following filter values:

Document type	=	BUS2203	AND
Responsibility Resolver Name	=	RR_ROLE	

The Responsibility Resolver Parameter is used to specify the user role from which the users should be assigned as responsible agents.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_ROLE. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_ROLE](#).

## 7.6 Invoice (BUS2205)

The enhancement implementation for the invoice is /SAPSRM/BD\_WF\_AGENTS\_INV. For this enhancement implementation seven BAdI implementations are defined. Which implementation is called at runtime is decided through the filter values. The filter values are set up using the document type (BUS2205 for invoice) and the Responsibility Resolver Name. The possible values for the latter are defined in the workflow customizing (see chapter [Add Responsibility Resolver Names](#)). The following chapters describe the filter values and the corresponding BAdI implementation.

### 7.6.1 Implementation - Specify Employee for Approval

The BAdI implementation /SAPSRM/BD\_WF\_INV\_RR\_EMPLOYEE returns an employee (Business Partner) as responsible agents for approval in the relevant process level. The document is not split into decision sets, that is, all items are assigned to the same agent.

This BAdI has the following filter values:

Document type	=	BUS2205	AND
Responsibility Resolver Name	=	RR_EMPLOYEE	

The Responsibility Resolver Parameter is used to specify the employee (Business Partner) ID of the person who should be assigned as agent.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_EMPLO. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_EMPLOYEE](#).

### 7.6.2 Implementation - Get Manager of Document Owner for Approval

The BAdI implementation /SAPSRM/BD\_WF\_INV\_RR\_MANAGER returns all managers of the document's requester of the previous level as responsible agents for approval in the relevant process level. The document is not split into decision sets, that is, all items are assigned to the same agent.

This BAdI has the following filter values:

Document type	=	BUS2205	AND
Responsibility Resolver Name	=	RR_MANAGER	

The Responsibility Resolver Parameter is not used.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_MANAGER. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_MANAGER](#).

### 7.6.3 Implementation - Get Manager of Previous Approver for Approval

The BAdI implementation /SAPSRM/BD\_WF\_INV\_RR\_MANOFMA returns all managers of approvers of the previous level as responsible agents for approval in the relevant process level. The method assigns all items of a decision set to the same area, thus respecting the existing decision sets for the document.

This BAdI has the following filter values:

Document type	=	BUS2205	AND
Responsibility Resolver Name	=	RR_MANAGER_OF_MANAGER	

The Responsibility Resolver Parameter is not used.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_MANOFMA. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_MANAGER\\_OF\\_MANAGER](#).

## 7.6.4 Implementation - Get Manager of the Purchasing Group for Approval

The BAdI implementation /SAPSRM/BD\_WF\_INV\_RR\_PURMGR returns all managers of the document's purchasing group as responsible agents for approval in the relevant process level. The document is not split into decision sets, that is, all items are assigned to the same agents.

This BAdI has the following filter values:

Document type = BUS2205 AND  
Responsibility Resolver Name = RR\_PURCHASING\_MANAGER

The Responsibility Resolver Parameter is not used.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_PURMGR. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_PURCHASING\\_MANAGER](#).

## 7.6.5 Implementation - Get Owner of Source Document for Approval

The BAdI implementation /SAPSRM/BD\_WF\_INV\_RR\_P\_OWNER returns the document owner of the source document as responsible agents for approval in the relevant process level.

This BAdI has the following filter values:

Document type = BUS2205 AND  
Responsibility Resolver Name = INV\_RR\_SOURCE\_DOCUMENT\_OWNER

The Responsibility Resolver Parameter is not used.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_POWNR\_INV.

### Method GET\_AREA\_TO\_ITEM\_MAP:

One instance of the area of responsibility of type /SAPSRM/CL\_WF\_AREA\_EMPLO\_LIST is created. All items are assigned to this one instance. As leading object the table of responsible users of the invoice is used. The logic to retrieve the responsible users is as follows:

- Partner of type requestor, if exactly one is found
- Else: creator of the PO (local or back-end) determined from the doc. Header

### Method GET\_APPROVERS\_BY\_AREA\_GUID:

The responsible agents are derived by the method /SAPSRM/CL\_WF\_AREA\_EMPLO\_LIST~GET\_RESPONSIBLE\_APPROVERS

The list of approvers is stored as leading object for the area of responsibility. This list is retrieved and all users in the list are assigned as responsible agents.

### Method GET\_FALLBACK\_AGENTS:

Not implemented.

### Method GET\_RESP\_RESOLVER\_PARAMETER\_F4:

Not implemented because no parameter is required.

### Method CHECK\_RESP\_RESOLVER\_PARAMETER:

Not implemented.

### 7.6.6 Implementation - Get all Employees of Specified Department for Approval

The BAdI implementation /SAPSRM/BD\_WF\_INV\_RR\_RLUNIT returns all employees of a specific reporting-line unit as responsible agents for approval in the relevant process level. The document is not split into decision sets, that is, all items are assigned to the same agents.

This BAdI has the following filter values:

Document type	=	BUS2205	AND
Responsibility Resolver Name	=	RR_REPORTING_LINE_UNIT	

The Responsibility Resolver Parameter is used to specify the reporting line unit from which the employees should be assigned as responsible agents.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_RLUNIT. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_REPORTING\\_LINE\\_UNIT](#).

### 7.6.7 Implementation - Get all Users of Specified Role for Approval

The BAdI implementation /SAPSRM/BD\_WF\_INV\_RR\_ROLE returns all user accounts of a specific user role as responsible agents for approval in the relevant process level. The document is not split into decision sets, that is, all items are assigned to the same agents.

This BAdI has the following filter values:

Document type	=	BUS2205	AND
Responsibility Resolver Name	=	RR_ROLE	

The Responsibility Resolver Parameter is used to specify the user role from which the users should be assigned as responsible agents.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_ROLE. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_ROLE](#).

## 7.7 RFx (BUS2200)

The enhancement implementation for the RFx is /SAPSRM/BD\_WF\_AGENTS\_RFQ. For this enhancement implementation six BAdI implementations are defined. Which implementation is called at runtime is decided through the filter values. The filter values are set up using the document type (BUS2200 for Rfx) and the Responsibility Resolver Name. The possible values for the latter are defined in the workflow customizing (see chapter [Add Responsibility Resolver Names](#)). The following chapters describe the filter values and the corresponding BAdI implementation.

### 7.7.1 Implementation - Specify Employee for Approval

The BAdI implementation /SAPSRM/BD\_WF\_RFQ\_RR\_EMPLOYEE returns an employee (Business Partner) as responsible agent for approval in the relevant process level. The document is not split into decision sets, that is, all items are assigned to the same agent.

This BAdI has the following filter values:

Document type	=	BUS2200	AND
Responsibility Resolver Name	=	RR_EMPLOYEE	

The Responsibility Resolver Parameter is used to specify the employee (Business Partner) ID of the person who should be assigned as agent.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_EMPLO. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_EMPLOYEE](#).

### 7.7.2 Implementation - Get Manager of Document Owner for Approval

The BAdI implementation /SAPSRM/BD\_WF\_RFQ\_RR\_MANAGER returns all managers of the document's requester of the previous level as responsible agents for approval in the relevant process level. The document is not split into decision sets, that is, all items are assigned to the same agent.

This BAdI has the following filter values:

Document type	=	BUS2200	AND
Responsibility Resolver Name	=	RR_MANAGER	

The Responsibility Resolver Parameter is not used.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_MANAGER. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_MANAGER](#).

### 7.7.3 Implementation - Get Manager of Previous Approver for Approval

The BAdI implementation /SAPSRM/BD\_WF\_RFQ\_RR\_MANOFMA returns all managers of approvers of the previous level as responsible agents for approval in the relevant process level. The method assigns all items of a decision set to the same area, thus respecting the existing decision sets for the document.

This BAdI has the following filter values:

Document type	=	BUS2200	AND
Responsibility Resolver Name	=	RR_MANAGER_OF_MANAGER	

The Responsibility Resolver Parameter is not used.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_MANOFMA. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_MANAGER\\_OF\\_MANAGER](#).

### 7.7.4 Implementation - Get Manager of the Purchasing Group for Approval

The BAdI implementation /SAPSRM/BD\_WF\_RFQ\_RR\_PURMGR returns all managers of the document's purchasing group as responsible agents for approval in the relevant process level. The document is not split into decision sets, that is, all items are assigned to the same agents.

This BAdI has the following filter values:

Document type = BUS2200 AND  
Responsibility Resolver Name = RR\_PURCHASING\_MANAGER

The Responsibility Resolver Parameter is not used.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_PURMGR. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_PURCHASING\\_MANAGER](#).

### 7.7.5 Implementation - Get all Employees of Specified Department for Approval

The BAdI implementation /SAPSRM/BD\_WF\_RFQ\_RR\_RLUNIT returns all employees of a specific reporting-line unit as responsible agents for approval in the relevant process level. The document is not split into decision sets, that is, all items are assigned to the same agents.

This BAdI has the following filter values:

Document type = BUS2200 AND  
Responsibility Resolver Name = RR\_REPORTING\_LINE\_UNIT

The Responsibility Resolver Parameter is used to specify the reporting line unit from which the employees should be assigned as responsible agents.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_RLUNIT. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_REPORTING LINE UNIT](#).

### 7.7.6 Implementation - Get all Users of Specified Role for Approval

The BAdI implementation /SAPSRM/BD\_WF\_RFQ\_RR\_ROLE returns all user accounts of a specific user role as responsible agents for approval in the relevant process level. The document is not split into decision sets, that is, all items are assigned to the same agents.

This BAdI has the following filter values:

Document type = BUS2200 AND  
Responsibility Resolver Name = RR\_ROLE

The Responsibility Resolver Parameter is used to specify the user role from which the users should be assigned as responsible agents.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_ROLE. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_ROLE](#).

## 7.8 Supplier Quotation (BUS2202)

The enhancement implementation for the supplier quotation is /SAPSRM/BD\_WF\_AGENTS\_QTE. For this enhancement implementation five BAdI implementations are defined. Which implementation is called at runtime is decided through the filter values. The filter values are set up using the document type (BUS2202 for supplier quotation) and the Responsibility Resolver Name. The possible values for the latter are defined in the workflow customizing (see chapter [Add Responsibility Resolver Names](#)). The following chapters describe the filter values and the corresponding BAdI implementation.

### 7.8.1 Implementation - Specify Employee for Approval

The BAdI implementation /SAPSRM/BD\_WF\_QTE\_RR\_EMPLOYEE returns an employee (Business Partner) as responsible agent for approval in the relevant process level. The document is not split into decision sets, that is, all items are assigned to the same agent.

This BAdI has the following filter values:

Document type	=	BUS2202	AND
Responsibility Resolver Name	=	RR_EMPLOYEE	

The Responsibility Resolver Parameter is used to specify the employee (Business Partner) ID of the person who should be assigned as agent.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_EMPLO. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_EMPLOYEE](#).

### 7.8.2 Implementation - Get Manager of Previous Approver for Approval

The BAdI implementation /SAPSRM/BD\_WF\_QTE\_RR\_MANOFMA returns all managers of approvers of the previous level as responsible agents for approval in the relevant process level. The method assigns all items of a decision set to the same area, thus respecting the existing decision sets for the document.

This BAdI has the following filter values:

Document type	=	BUS2202	AND
Responsibility Resolver Name	=	RR_MANAGER_OF_MANAGER	

The Responsibility Resolver Parameter is not used.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_MANOFMA. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_MANAGER\\_OF\\_MANAGER](#).

### 7.8.3 Implementation - Get Manager of Purchasing Group (Header) for Approval

The BAdI implementation /SAPSRM/BD\_WF\_QTE\_RR\_PURMGR returns all managers of the document's purchasing group as responsible agents for approval in the relevant process level.

This BAdI has the following filter values:

Document type	=	BUS2202	AND
Responsibility Resolver Name	=	RR_PURCHASING_MANAGER	

The Responsibility Resolver Parameter is not used.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_PURMGR\_QTE.

Method GET\_AREA\_TO\_ITEM\_MAP:

The purchasing group of the document is determined. In order to this first an instance of the purchasing document is created to be able to access the attributes. From the organizational data of the document header the purchasing group is determined. The purchasing group acts as leading object ID and an area of responsibility is created for the purchasing group. This is done using the same class /SAPSRM/CL\_WF\_AREA\_MANAGER as for RR\_MANAGER.

All items are assigned to the same area of responsibility, that is, purchasing group. The document is not split into decision sets.

Method GET\_APPROVERS\_BY\_AREA\_GUID:

The responsible agents are derived by the method /SAPSRM/CL\_WF\_AREA\_MANAGER~GET\_RESPONSIBLE\_APPROVERS.

For the purchasing group (as stored in the leading object) the attributes from the organizational management are read and the user names of all the managers belonging to the organizational unit are returned. These user names of the managers are passed back as responsible agents.

Method GET\_FALLBACK\_AGENTS:

Not implemented.

Method GET\_RESP\_RESOLVER\_PARAMETER\_F4:

Not implemented because no parameter is required.

Method CHECK\_RESP\_RESOLVER\_PARAMETER:

Not implemented.

## 7.8.4 Implementation - Get all Employees of Specified Department for Approval

The BAdI implementation /SAPSRM/BD\_WF\_QTE\_RR\_RLUNIT returns all employees of a specific reporting-line unit as responsible agents for approval in the relevant process level. The document is not split into decision sets, that is, all items are assigned to the same agents.

This BAdI has the following filter values:

Document type	=	BUS2202	AND
Responsibility Resolver Name	=	RR_REPORTING_LINE_UNIT	

The Responsibility Resolver Parameter is used to specify the reporting line unit from which the employees should be assigned as responsible agents.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_RLUNIT. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_REPORTING\\_LINE\\_UNIT](#).

## 7.8.5 Implementation - Get all Users of Specified Role for Approval

The BAdI implementation /SAPSRM/BD\_WF\_QTE\_RR\_ROLE returns all user accounts of a specific user role as responsible agents for approval in the relevant process level. The document is not split into decision sets, that is, all items are assigned to the same agents.

This BAdI has the following filter values:

Document type	=	BUS2202	AND
---------------	---	---------	-----



Responsibility Resolver Name = RR\_ROLE

The Responsibility Resolver Parameter is used to specify the user role from which the users should be assigned as responsible agents.

The implementing class for these filter values is /SAPSRM/CL\_IM\_WF\_RR\_ROLE. The documentation is not object-specific and can be found in [Documentation of implementing class for RR\\_ROLE](#).

## 7.9 Internal Enhancement Implementation /SAPSRM/BD\_WF\_AGENTS

For this enhancement implementation two BAdI implementations exists:

- /SAPSRM/BD\_WF\_RR\_ADHOC: activated by the filter value RESP\_RESOLVER\_NAME = ADHOC
- /SAPSRM/BD\_WF\_RR\_SYSTEM: activated by the filter value RESP\_RESOLVER\_NAME = SYSTEM

The implementing class for the ADHOC class has already been described in [Implementation /SAPSRM/BD\\_WF\\_SC\\_RR\\_SL - Get Users of Spending Limit Chain for Approval](#).

As for the implementing class for the SYSTEM class only the method GET\_AREA\_TO\_ITEM\_MAP is implemented returning an initial GUID for the area of responsibility since no area is needed for system approval.

## 7.10 Documentation of Implementing Class for RR\_EMPLOYEE

This chapter describes the implementing class /SAPSRM/CL\_IM\_WF\_RR\_EMPLO.

### Method GET\_AREA\_TO\_ITEM\_MAP:

For the responsibility resolver name RR\_EMPLOYEE the process level contains the ID of the employee as responsibility resolver parameter. For this ID the corresponding GUID of the employee is determined. This employee GUID is then used as leading object ID to create the area of responsibility. This is done using the class /SAPSRM/CL\_WF\_AREA\_EMPLOYEE.

The method assigns all items of the document to the same area, that is, the same employee.

### Method GET\_APPROVERS\_BY\_AREA\_GUID:

The responsible agents are derived by the method /SAPSRM/CL\_WF\_AREA\_EMPLOYEE~GET\_RESPONSIBLE\_APPROVERS.

For the GUID of the employee (which is the identifying ID of the area) the attributes from the organizational management are read and the user name is returned. This user name of the employee is passed back as responsible agent.

### Method GET\_FALLBACK\_AGENTS:

Not implemented.

### Method GET\_RESP\_RESOLVER\_PARAMETER\_F4:

This method calls a search help to find the business partner numbers for employees. A screen is displayed to search for the business partner ID of employees by several criteria like last name, user name, organization. The search help is BBP\_BUPA\_EMPLOYEE provided by the function module /SAPSRM/F4IF\_SHLP\_EXIT\_BUPA. The business partner ID of the (first) selected employee is returned.

### Method CHECK\_RESP\_RESOLVER\_PARAMETER:

The provided business partner ID is checked against the organizational management that the business partner is an employee. If the business partner is not an employee or if the field is empty the exception /SAPSRM/CX\_WF\_NOT\_FOUND is raised.

## 7.11 Documentation of Implementing Class for RR\_MANAGER

This chapter describes the implementing class /SAPSRM/CL\_IM\_WF\_RR\_MANAGER.

### Method GET\_AREA\_TO\_ITEM\_MAP:

For the responsibility resolver name RR\_MANAGER the responsible persons of the purchasing document are determined. This table of user names is then used as leading object ID to create the area of responsibility. This is done using the class /SAPSRM/CL\_WF\_AREA\_MANAGER.

The method assigns all items of the document to the same area.

The determination of the responsible persons for a document depends on the object type:

- Shopping cart: all partners of type REQUESTOR of the document
- Purchase order:
  - User who created the PO determined from doc. header, if not a technical user
  - Else: all partners of type REQUESTOR
- Purchase contract:
  - User who created the PO determined from doc. header, if not a technical user
  - Else: current user name if the user is an employee
- Invoice:
  - Partner of type requestor, if exactly one is found
  - Else: creator of the PO (local or back end) determined from the doc. Header
- Confirmation:
  - Creator of confirmation if user is a professional purchaser
  - Else: Partner of type requestor:
    - If exactly one and the user is the creator and belongs to the purchasing organization
    - If not the creator, but an user belonging to the company
    - If not the creator and external, the responsible persons of the source document (purchase order): this is the requestor if the person belongs to the purch. Organization or all purchasers of the relevant purchasing groups in the purchase order
- Bid invitation: Creator of document if the user is an employee and all partners of type requestor.
- Supplier quotation: all partners of type requestor from the parent document (bid invitation)

### Method GET\_APPROVERS\_BY\_AREA\_GUID:

The responsible agents are derived by the method /SAPSRM/CL\_WF\_AREA\_MANAGER~GET\_RESPONSIBLE\_APPROVERS.

For all users found as document responsables (as stored in the leading object) the attributes from the organizational management are read and the user names of all their managers belonging to the organizational unit are returned. These user names of the managers are passed back as responsible agents.

### Method GET\_FALLBACK\_AGENTS:

Not implemented.

Method GET\_RESP\_RESOLVER\_PARAMETER\_F4:

Not implemented because no parameter is required.

Method CHECK\_RESP\_RESOLVER\_PARAMETER:

Not implemented.

## 7.12 Documentation of Implementing Class for RR\_MANAGER\_OF\_MANAGER

This chapter describes the implementing class /SAPSRM/CL\_IM\_WF\_RR\_MANOFMA.

### Method GET\_AREA\_TO\_ITEM\_MAP:

For the responsibility resolver name RR\_MANAGER\_OF\_MANAGER the approvers of the previous level in the approval process of the purchasing document are determined.

First the previous process level of the workflow is determined. If the previous level was not valid the document responsibilities are determined using the same method /SAPSRM/CL\_WF\_AREA\_MANAGER~GET\_AREA\_TO\_ITEM\_MAP for the current process level. For a detailed description see chapter [Documentation of implementing class for RR\\_MANAGER](#).

If a valid level is found the next step is to determine the approvers of the previous level. This is done for each existing decision set.:

- If the previous level was executed, the approver is returned
- If the previous level was not executed, all agents who got a work items are returned.

For each decision set the returned table of user names is then used as leading object ID to create the area of responsibility. This is done using the same class /SAPSRM/CL\_WF\_AREA\_MANAGER as for RR\_MANAGER.

The method assigns all items of a decision set to the same area, thus respecting the existing decision sets for the document.

### Method GET\_APPROVERS\_BY\_AREA\_GUID:

The responsible agents are derived by the method /SAPSRM/CL\_WF\_AREA\_MANAGER~GET\_RESPONSIBLE\_APPROVERS.

For all users found as approvers (as stored in the leading object) the attributes from the organizational management are read and the user names of all their managers belonging to the organizational unit are returned. These user names of the managers are passed back as responsible agents.

### Method GET\_FALLBACK\_AGENTS:

Not implemented.

### Method GET\_RESP\_RESOLVER\_PARAMETER\_F4:

Not implemented because no parameter is required.

### Method CHECK\_RESP\_RESOLVER\_PARAMETER:

Not implemented.

## 7.13 Documentation of Implementing Class for RR\_PURCHASING\_MANAGER

This chapter describes the implementing class /SAPSRM/CL\_IM\_WF\_RR\_PURMGR. This class cannot be used for shopping cars and supplier quotations.

### Method GET\_AREA\_TO\_ITEM\_MAP:

The purchasing group of the document is determined. In order to this first an instance of the purchasing document is created to be able to access the attributes. From the organizational data of the document header the purchasing group is determined. The purchasing group acts as leading object ID and an area of responsibility is created for the purchasing group. This is done using the same class /SAPSRM/CL\_WF\_AREA\_MANAGER as for RR\_MANAGER.

All items are assigned to the same area of responsibility, that is, purchasing group. The document is not split into decision sets.

### Method GET\_APPROVERS\_BY\_AREA\_GUID:

The responsible agents are derived by the method /SAPSRM/CL\_WF\_AREA\_MANAGER~GET\_RESPONSIBLE\_APPROVERS.

For the purchasing group (as stored in the leading object) the attributes from the organizational management are read and the user names of all the managers belonging to the organizational unit are returned. These user names of the managers are passed back as responsible agents.

### Method GET\_FALLBACK\_AGENTS:

Not implemented.

### Method GET\_RESP\_RESOLVER\_PARAMETER\_F4:

Not implemented because no parameter is required.

### Method CHECK\_RESP\_RESOLVER\_PARAMETER:

Not implemented.

## 7.14 Documentation of Implementing Class for RR\_REPORTING\_LINE\_UNIT

This chapter describes the implementing class /SAPSRM/CL\_IM\_WF\_RR\_RLUNIT.

### Method GET\_AREA\_TO\_ITEM\_MAP:

For the responsibility resolver name RR\_REPORTING\_LINE\_UNIT the process level contains the name of the organizational unit as responsibility resolver parameter. The object ID of this unit is then used as leading object ID to create the area of responsibility. This is done using the class /SAPSRM/CL\_WF\_AREA\_RLUNIT.

The method assigns all items of the document to the same area, that is, the same reporting line unit.

### Method GET\_APPROVERS\_BY\_AREA\_GUID:

The responsible agents are derived by the method /SAPSRM/CL\_WF\_AREA\_RLUNIT ~GET\_RESPONSIBLE\_APPROVERS.

The users assigned to the organizational unit (defined as leading object ID) are determined from the organizational management. All users found are returned as responsible agents.

### Method GET\_FALLBACK\_AGENTS:

Not implemented.

### Method GET\_RESP\_RESOLVER\_PARAMETER\_F4:

This method calls the input help of the organizational management to find departments (organizational units).

### Method CHECK\_RESP\_RESOLVER\_PARAMETER:

It is checked that the provided organizational unit exists. If not the exception /SAPSRM/CX\_WF\_NOT\_FOUND is raised.

## 7.15 Documentation of Implementing Class for RR\_ROLE

This chapter describes the implementing class /SAPSRM/CL\_IM\_WF\_RR\_ROLE.

### Method GET\_AREA\_TO\_ITEM\_MAP:

For the responsibility resolver name RR\_ROLE the process level contains the name of the activity group as responsibility resolver parameter. This activity group is then used as leading object ID to create the area of responsibility. This is done using the class /SAPSRM/CL\_WF\_AREA\_ACTIV\_GRP.

The method assigns all items of the document to the same area, that is, the same employee.

### Method GET\_APPROVERS\_BY\_AREA\_GUID:

The responsible agents are derived by the method /SAPSRM/CL\_WF\_AREA\_ACTIV\_GRP ~GET\_RESPONSIBLE\_APPROVERS.

The users assigned to the activity group (defined as leading object ID) are determined from the organizational management. All users found are returned as responsible agents.

### Method GET\_FALLBACK\_AGENTS:

Not implemented.

### Method GET\_RESP\_RESOLVER\_PARAMETER\_F4:

This method calls a search help to find the activity groups. A screen is displayed to choose an existing activity group by name. The elementary search help is PRGN\_AGR. The selected value is returned.

### Method CHECK\_RESP\_RESOLVER\_PARAMETER:

It is checked that the provided activity group exists. If not the exception /SAPSRM/CX\_WF\_NOT\_FOUND is raised.



## 8 Appendix C: Standard Implementation of Dynamic Process Levels

### 8.1 Implementation /SAPSRM/BD\_WF\_CONFIG\_SC\_SL

#### 8.1.1 General information for shopping cart implementation

The enhancement implementation is an implementation for the shopping cart and the name of the implementation is /SAPSRM/BD\_WF\_PROCESS\_SC.

For this enhancement implementation one BAdI implementation exists:

/SAPSRM/BD\_WF\_CONFIG\_SC\_SL: Shopping Cart: Dynamic Spending-Limit Process-Level Configuration

This BAdI has the following filter values:

Document type	=	BUS2121	AND
Responsibility Resolver Name	=	RR_SPENDING_LIMIT_APPROVER	

The Resp. Resolver Name is taken from the corresponding entry in the customizing table for the current level in the process schema at runtime.

The implementing class is /SAPSRM/CL\_IM\_WF\_CONFIG\_SL. This class implements the Method CREATE\_PROCESS\_LEVEL\_CONFIG, which is called when the defined filter values match the values at runtime.

*Example:* In the example of the N-Step approval process for a shopping cart there is one step defined in the process schema. The document type of a shopping cart is BUS2121 and the process schema contains as Resp. Resolver Name RR\_SPENDING\_LIMIT\_APPROVER. Since these are the filter values for the defined BAdI implementation the implemented method of this BAdI is called in the workflow.

#### 8.1.2 Implemented logic for shopping cart

The BAdI determines the list of approvers starting with the document owner. For each new approver a new process level is created and the list of approvers is stored in database tables.

The steps in detail are:

- 1) Create instance of document
- 2) The document owner is determined from the partners of the document with partner type requestor. The requestors are checked to be employees of the company. (If a requestor is not an employee this requestor is not considered). If more than one requestor is found, the first one is chosen.
- 3) For the chosen user the organizational management is analyzed to find the responsible approver for the value limit (attribute: SLAPPROVER). It is checked that the approver is an employee.
- 4) For each new approver a new process level is added.
- 5) The approver is added to the list of all approvers (agents for the workflow).
- 6) This procedure (steps 3, 4 and 5) is repeated for each new approver until no new approver can be found.
- 7) The list of approvers is stored in the database for access in the BAdI for agent determination. The information is stored in a mapping table of the GUIDs of the determined process levels and the responsible agents for a level. The mapping table of the items of the document to the process levels is empty.

## 8.2 Implementation /SAPSRM/BD\_WF\_CONFIG\_PO\_SL

### 8.2.1 General information for purchase order implementation

The enhancement implementation is an implementation for the purchase order and the name of the implementation is /SAPSRM/BD\_WF\_PROCESS\_PO.

For this enhancement implementation one BAdI implementation exists:

/SAPSRM/BD\_WF\_CONFIG\_PO\_SL: Shopping Cart: Dynamic Spending-Limit Process-Level Configuration

This BAdI has the following filter values:

Document type	=	BUS2201	AND
Responsibility Resolver Name	=	RR_SPENDING_LIMIT_APPROVER	

The Resp. Resolver Name is taken from the corresponding entry in the customizing table for the current level in the process schema at runtime.

The implementing class is /SAPSRM/CL\_IM\_WF\_CONFIG\_SL. This is the same class as is used for the shopping cart.

### 8.2.2 Implemented logic for purchase order

The implementing class is the same as for the shopping cart. For implementation details please look at chapter [Implemented logic for Shopping Cart](#).

## 9 Appendix D: Standard Implementations of /SAPSRM/BD\_WF\_REVIEWER\_RULE

### 9.1 Overview

In the SRM standard four different rules are defined for the determination of reviewers:

- **RV\_EMPLOYEE:** The employee specified in the parameter for the rule is assigned as reviewer.
- **RV\_REPORTING\_LINE\_UNIT:** All users assigned to the department (unit in the organizational management) provided as rule parameter are returned as reviewers.
- **RV\_ROLE:** All users which have been assigned to the role as provided in the rule parameter are returned as reviewers.
- **RV\_SOURCE\_DOCUMENT\_RESPONSIBLES:** The responsible persons for a document are returned as reviewers. The rule parameter is not used here.

For each of these four rules an implementing class exists. These classes are independent of the object types. Nevertheless there are BAdI implementations for these rules for each business object type in order to be able to activate or deactivate them for each business object type individually.

The following BAdI implementations exist:

#### Default:

- **/SAPSRM/BD\_WF\_REVIEWER:** default BADI impl. for reviewer finding by role
  - Filter: RV\_ROLE

### 9.2 Shopping Cart (BUS2121):

- **/SAPSRM/BD\_WF\_SC\_RV\_EMPLOYEE:** reviewer finding via user id for shopping cart.
  - Filter: DOCUMENT\_TYPE = BUS2121 AND REVIEW\_RUL\_NAME = RV\_EMPLOYEE.
- **/SAPSRM/BD\_WF\_SC\_RV\_RLUNIT:** find reviewers by reporting line unit for shopping cart.
  - Filter DOCUMENT\_TYPE = BUS2121 AND REVIEW\_RUL\_NAME = RV\_REPORTING\_LINE\_UNIT.
- **/SAPSRM/BD\_WF\_SC\_RV\_ROLE:** reviewer finding by role for shopping cart.
  - Filter DOCUMENT\_TYPE = BUS2121 AND REVIEW\_RUL\_NAME = RV\_ROLE.

### 9.3 Purchase Order (BUS2201):

- **/SAPSRM/BD\_WF\_PO\_RV\_EMPLOYEE:** reviewer finding via user id for purchase order.
  - Filter DOCUMENT\_TYPE = BUS2201 AND REVIEW\_RUL\_NAME = RV\_EMPLOYEE.
- **/SAPSRM/BD\_WF\_PO\_RV\_RLUNIT:** find reviewers by reporting line unit for purchase order.
  - Filter DOCUMENT\_TYPE = BUS2201 AND REVIEW\_RUL\_NAME = RV\_REPORTING\_LINE\_UNIT.
- **/SAPSRM/BD\_WF\_PO\_RV\_ROLE:** reviewer finding by role for purchase order.
  - Filter DOCUMENT\_TYPE = BUS2201 AND REVIEW\_RUL\_NAME = RV\_ROLE.

## 9.4 Purchase Contract (BUS2000113):

- /SAPSRM/BD\_WF\_CTR\_RV\_EMPLOYEE: reviewer finding via user id for purchase contract.
  - Filter DOCUMENT\_TYPE = BUS2000113 AND REVIEW\_RUL\_NAME = RV\_EMPLOYEE.
- /SAPSRM/BD\_WF\_CTR\_RV\_RLUNIT: find reviewers by reporting line unit for purchase contract.
  - Filter DOCUMENT\_TYPE = BUS2000113 AND REVIEW\_RUL\_NAME = RV\_REPORTING\_LINE\_UNIT.
- /SAPSRM/BD\_WF\_CTR\_RV\_ROLE: reviewer finding by role for purchase contract.
  - Filter DOCUMENT\_TYPE = BUS2000113 AND REVIEW\_RUL\_NAME = RV\_ROLE.

## 9.5 Confirmation (BUS2203):

- /SAPSRM/BD\_WF\_CONF\_RV\_EMPLOYEE: reviewer finding via user id for confirmation.
  - Filter DOCUMENT\_TYPE = BUS2203 AND REVIEW\_RUL\_NAME = RV\_EMPLOYEE.
- /SAPSRM/BD\_WF\_CONF\_RV\_POWNR: get responsible for source document of contract as reviewer
  - Filter DOCUMENT\_TYPE = BUS2203 AND REVIEW\_RUL\_NAME = RV\_SOURCE\_DOCUMENT\_RESPONSIBLES.
- /SAPSRM/BD\_WF\_CONF\_RV\_RLUNIT: find reviewers by reporting line unit for confirmation.
  - Filter DOCUMENT\_TYPE = BUS2303 AND REVIEW\_RUL\_NAME = RV\_REPORTING\_LINE\_UNIT.
- /SAPSRM/BD\_WF\_CONF\_RV\_ROLE: reviewer finding by role for confirmation.
  - Filter DOCUMENT\_TYPE = BUS2203 AND REVIEW\_RUL\_NAME = RV\_ROLE.

## 9.6 Invoice (BUS2205):

- /SAPSRM/BD\_WF\_INV\_RV\_EMPLOYEE: reviewer finding via user id for invoice.
  - Filter DOCUMENT\_TYPE = BUS2205 AND REVIEW\_RUL\_NAME = RV\_EMPLOYEE.
- /SAPSRM/BD\_WF\_INV\_RV\_POWNR: get responsible for source document of contract as reviewer
  - Filter DOCUMENT\_TYPE = BUS2205 AND REVIEW\_RUL\_NAME = RV\_SOURCE\_DOCUMENT\_RESPONSIBLES.
- /SAPSRM/BD\_WF\_INV\_RV\_RLUNIT: find reviewers by reporting line unit for invoice.
  - Filter DOCUMENT\_TYPE = BUS2305 AND REVIEW\_RUL\_NAME = RV\_REPORTING\_LINE\_UNIT.
- /SAPSRM/BD\_WF\_INV\_RV\_ROLE: reviewer finding by role for invoice.
  - Filter DOCUMENT\_TYPE = BUS2205 AND REVIEW\_RUL\_NAME = RV\_ROLE.

## 9.7 Bid Invitation (BUS2200):

- /SAPSRM/BD\_WF\_RFQ\_RV\_EMPLOYEE: reviewer finding via user id for bid invitation.
  - Filter DOCUMENT\_TYPE = BUS2200 AND REVIEW\_RUL\_NAME = RV\_EMPLOYEE.
- /SAPSRM/BD\_WF\_RFQ\_RV\_RLUNIT: find reviewers by reporting line unit for bid invitation.

- Filter DOCUMENT\_TYPE = BUS2200 AND REVIEW\_RUL\_NAME = RV\_REPORTING\_LINE\_UNIT.
- /SAPSRM/BD\_WF\_RFQ\_RV\_ROLE: reviewer finding by role for bid invitation.
  - Filter DOCUMENT\_TYPE = BUS2200 AND REVIEW\_RUL\_NAME = RV\_ROLE.

## 9.8 Supplier Quotation (BUS2202):

- /SAPSRM/BD\_WF\_QTE\_RV\_EMPLOYEE: reviewer finding via user id for quotation.
  - Filter DOCUMENT\_TYPE = BUS2202 AND REVIEW\_RUL\_NAME = RV\_EMPLOYEE.
- /SAPSRM/BD\_WF\_QTE\_RV\_RLUNIT: find reviewers by reporting line unit for quotation.
  - Filter DOCUMENT\_TYPE = BUS2202 AND REVIEW\_RUL\_NAME = RV\_REPORTING\_LINE\_UNIT.
- /SAPSRM/BD\_WF\_QTE\_RV\_ROLE: reviewer finding by role for quotation.
  - Filter DOCUMENT\_TYPE = BUS2202 AND REVIEW\_RUL\_NAME = RV\_ROLE.

## 9.9 Documentation of Implementing Class for RV\_EMPLOYEE

The implementing class is class /SAPSRM/CL\_IM\_WF\_REV\_EMPL. An employee specified in the rule parameter is returned as reviewer.

### Method GET\_REVIEWER\_BY\_DOCUMENT:

For the rule name RV\_EMPLOYEE the process level contains the ID of the employee as rule parameter. For this ID the corresponding GUID of the employee is determined. This employee GUID is then used to get the user name of the employee. The user name of the employee is returned as reviewer.

### Method GET\_REVIEWER\_RULE\_PARAM\_F4:

This method calls a search help to find the business partner numbers for employees. A screen is displayed to search for the business partner ID of employees by several criteria like last name, user name, organization. The search help is BBP\_BUPA\_EMPLOYEE provided by the function module /SAPSRM/F4IF\_SHLP\_EXIT\_BUPA. The business partner ID of the (first) selected employee is returned.

### Method CHECK\_REVIEWER\_RULE\_PARAM:

The provided business partner ID is checked against the organizational management that the business partner is an employee. If the business partner is not an employee or if the field is empty the exception /SAPSRM/CX\_WF\_NOT\_FOUND is raised.

## 9.10 Documentation of Implementing Class for RV\_REPORTING\_LINE\_UNIT

The implementing class is class /SAPSRM/CL\_IM\_WF\_REV\_RLUNIT. All users assigned to the department passed in the rule parameter are returned as reviewers of the document.

### Method GET\_REVIEWER\_BY\_DOCUMENT:

For the rule name RV\_REPORTING\_LINE\_UNIT the process level contains the name of the organizational unit as rule parameter. For this organizational unit (department) all assigned users are determined. The table of the corresponding user names is passed as reviewer list.

Method GET REVIEWER\_RULE\_PARAM\_F4:

This method calls the input help of the organizational management to find departments (organizational units).

Method CHECK\_REVIEWER\_RULE\_PARAM:

It is checked that the provided organizational unit exists. If not the exception /SAPSRM/CX\_WF\_NOT\_FOUND is raised.

## 9.11 Documentation of Implementing Class for RV\_ROLE

The implementing class is class /SAPSRM/CL\_IM\_WF\_REV\_ROLE.

Method GET\_REVIEWER\_BY\_DOCUMENT:

For the rule name RV\_ROLE the process level contains the name of the activity group as rule parameter. This activity group is then used to determine all users which are assigned to this activity group. All user names are returned as reviewers.

Method GET\_REVIEWER\_RULE\_PARAM\_F4:

This method calls a search help to find the activity groups. A screen is displayed to choose an existing activity group by name. The elementary search help is PRGN\_AGR. The selected value is returned.

Method CHECK\_REVIEWER\_RULE\_PARAM:

It is checked that the provided activity group exists. If not the exception /SAPSRM/CX\_WF\_NOT\_FOUND is raised.

## 9.12 Documentation of Implementing Class for RV\_SOURCE\_DOCUMENT\_RESPONSIBLE

The implementing class is class /SAPSRM/CL\_IM\_WF\_REV\_POWNR.

Method GET\_REVIEWER\_BY\_DOCUMENT:

The rule parameter is not used here. The responsible persons are determined from the document attributes. First the document of the current process is determined. For the document the responsible persons are determined. All users found are returned as reviewers.

The determination of the responsible persons for a document depends on the object type:

- Shopping cart: all partners of type REQUESTOR of the document
- Purchase order:
  - User who created the PO determined from doc. header, if not a technical user
  - Else: all partners of type REQUESTOR
- Purchase contract:
  - User who created the PO determined from doc. header, if not a technical user

- Else: current user name if the user is an employee
- Invoice:
  - Partner of type requestor, if exactly one is found
  - Else: creator of the PO (local or back end) determined from the doc. Header
- Confirmation:
  - Creator of confirmation if user is a professional purchaser
  - Else: Partner of type requestor:
    - If exactly one and the user is the creator and belongs to the purchasing organization
    - If not the creator, but an user belonging to the company
    - If not the creator and external, the responsible persons of the source document (purchase order): this is the requestor if the person belongs to the purch. Organization or all purchasers of the relevant purchasing groups in the purchase order
- Bid invitation: Creator of document if the user is an employee and all partners of type requestor.
- Supplier quotation: all partners of type requestor from the parent document (bid invitation)

Method GET\_REVIEWER\_RULE\_PARAM\_F4:

Not implemented.

Method CHECK\_REVIEWER\_RULE\_PARAM:

Not implemented.