

Process Controlled Workflow & Custom BRF Expressions for Complex Approval Processes in SRM 7.0



Applies to:

SAP SRM 7.0

For more information, visit the [Supplier Relationship Management homepage](#).

Summary

Process Controlled Workflow using BRF offers an excellent tool for implementing Workflow processes in SRM 7.0. But sometimes there are situations where standard functionality in BRF and Process Controlled Workflow need enhancements in order to meet customer requirements.

Author: Virender Singh, CPIM CSCP

Company: Intelligroup Inc.

Created on: 5 July 2010

Author Bio

Virender Singh is a SAP SRM 7.0 Certified "*Professional*" consultant with more than 9 years of SAP Consulting Experience working with Leading IT Organization serving global clients. He has extensive experience of a number of implementations, upgrades, rollouts and run/maintain lifecycles of SAP Solutions running on SAP SRM, ECC, SNC, SAP Solution Manager. As a professional member of APICS and APICS Certified CPIM CSCP associate he has adequate knowledge and experience of modern day Supply Chain and Supplier Relationship processes.

Currently in Intelligroup Inc., he is involved in providing innovative solutions to Global clients on SAP SRM and SCM.

Table of Contents

Requirements	3
Requirement 1	3
Requirement 2	3
Approval Process Design using Process Controlled Workflow	3
Process Schema Evaluation	3
Example: Write a Custom BRF Expression for Process Schema Evaluation.....	4
Create a BRF Expression using above Function Module	9
Process Level Evaluation	10
Example: Creating a Custom BRF Expression using Multiple Standard Expressions	12
Example: Creating a Custom BRF Expression using Custom Function Module	16
Create a BRF Expression using above Function Module	21
Disclaimer and Liability Notice.....	23

Requirements

Requirement 1

There are multiple approval hierarchies within an enterprise. Each of these hierarchies has its own approval steps which are dependent on a combination of data in shopping cart items (standard as well as custom fields). Sometimes, it becomes very difficult to distinctly differentiate between various approval hierarchies and the approval processes in each of them.

Requirement 2

SAP has provided so many BRF Expressions which evaluates the data in the shopping cart. These expressions can be used to determine the Process Schema and evaluate Process Levels. But in some cases, the standard expressions are not enough for evaluating the data as per customer's requirements

Approval Process Design using Process Controlled Workflow

Process Schema Evaluation

First of all it is important to know the Approval Processes for a particular business object (SC, Purchase Order etc). We must get answers of following questions for moving further on process schema and evaluations involved for the same. For the rest of this document, we will refer to the workflow for business object for Shopping Cart.

1. How Many different Approval Hierarchies exist in the enterprise?

- a. There could be different approval paths for approving a application document for a business object depending on various factors. Let us say there are N number of different approval paths/hierarchies for SC Approval. Each of these N Paths will represent a process schema. In other words , there will be N number of Process Schemas in the Process configuration for workflow. Each of these N Number of process schemas will have to be configured in the process configurations of SC workflow

2. What is the criteria for determining which Approval Path/Hierarchy to be picked up for a Shopping Cart?

- a. The criteria here could be different product categories, different combinations of product category/purch org elements or data in customer specific fields could also be a driving factor in determining these different approval paths.
- b. **Note:** if there is only one approval path in the enterprise for approving Shopping cart, it is advisable to use standard BRF expressions which will return the schema name as the CONSTANT Value. Result type of such expressions will be Constant. Standard Expression **3C_SC_600_001** is one such example of standard available expression.

From the perspective of the scope of this document, we will create a Custom BRF Expression for Process Schema Evaluation.

Business Object	Event Id	Expression	Expression Type
Shopping Cart (BUS2121)	BRF Event for Evaluating the SC Data and return one unique Process Schema (from N Number of process schemas)	BRF Expression for evaluating the SC Data. This custom Expression could be a combination of various Standard and/or Custom BRF Expressions OR It could also be a Custom BRF Expression evaluating the SC Data	Depending on the nature of evaluation and the availability of suitable standard BRF expressions, it could be 0CF001 or 0FB001

Example: Write a Custom BRF Expression for Process Schema Evaluation

- a. Create a Function Module ZWF_BRF_SCHEMA_EVAL Function Module /SAPSRM/WF_BRF_0EXP000 is used by SAP for lot of expressions in standard delivery. We may reference this FM while creating the custom function module. Import/ Export parameters of this FM will be as follows

Parameter Name	Typing	Associated Type	Default value
IT_EXPRESSIONS	TYPE	SBRF260A_T	
IO_EVENT	TYPE REF TO	IF_EVENT_BRF	
IO_EXPRESSION	TYPE REF TO	IF_EXPRESSION_BRF	

Parameter Name	Typing	Associated Type	Pass Val.	Short text
EV_VALUE	TYPE	BRF_RESULT_VALUE	<input type="checkbox"/>	
EV_TYPE	TYPE	BRF_RESULT_TYPE	<input type="checkbox"/>	
EV_LENGTH	TYPE	BRF_RESULT_LENGTH	<input type="checkbox"/>	
EV_CURRENCY	TYPE	BRF_CURRENCY	<input type="checkbox"/>	
EV_OUTPUT_LENGTH	TYPE	BRF_RESULT_OUTPUT_LEN	<input type="checkbox"/>	
EV_DECIMALS	TYPE	BRF_RESULT_DECIMALS	<input type="checkbox"/>	
EV_DATA_MISSING	TYPE	BRF_DATA_MISSING	<input type="checkbox"/>	

- b. Source Code for the Function Module for BRF Expression will be on the lines of this sample code:

```

FUNCTION ZWF_BRF_SCHEMA_EVAL .
* " -----
* " "Local Interface:
* "   IMPORTING
* "     REFERENCE(IT_EXPRESSIONS) TYPE  SBRF260A_T
* "     REFERENCE(IO_EVENT) TYPE REF TO  IF_EVENT_BRF
* "     REFERENCE(IO_EXPRESSION) TYPE REF TO  IF_EXPRESSION_BRF
* "   EXPORTING
* "     REFERENCE(EV_VALUE) TYPE  BRF_RESULT_VALUE
* "     REFERENCE(EV_TYPE) TYPE  BRF_RESULT_TYPE
* "     REFERENCE(EV_LENGTH) TYPE  BRF_RESULT_LENGTH
* "     REFERENCE(EV_CURRENCY) TYPE  BRF_CURRENCY
* "     REFERENCE(EV_OUTPUT_LENGTH) TYPE  BRF_RESULT_OUTPUT_LENGTH
* "     REFERENCE(EV_DECIMALS) TYPE  BRF_RESULT_DECIMALS
* "     REFERENCE(EV_DATA_MISSING) TYPE  BRF_DATA_MISSING
* " -----

DATA o1_wf_brf_event TYPE REF TO /sapsrm/cl_wf_brf_event.
DATA o1_context_provider TYPE REF TO /sapsrm/cl_wf_context_provider.
DATA o1_pdo_sc TYPE REF TO /sapsrm/if_pdo_bo_sc.
DATA o1_wf_pdo TYPE REF TO /sapsrm/if_wf_pdo.

DATA v1_document_guid TYPE /sapsrm/wf_document_guid.
DATA v1_document_type TYPE /sapsrm/wf_document_type.

DATA il_document_owner TYPE /sapsrm/t_wf_agent_id.
DATA il_header_guid TYPE bbpt_guid.
DATA il_item_guid TYPE /sapsrm/t_pdo_hier_guid_list.
DATA il_item TYPE bbpt_pd_sc_item_d.
DATA il_item_i TYPE bbpt_pd_sc_item_d.

DATA w1_header TYPE bbp_pds_sc_header_d.
DATA w1_header_guid TYPE bbp_guid_tab.
DATA w1_item LIKE LINE OF il_item.

DATA r1_item TYPE REF TO bbp_pds_sc_item_d.
DATA r1_item_guid TYPE REF TO /sapsrm/s_pdo_hier_guid_list.
DATA r1_document_owner TYPE REF TO /sapsrm/wf_agent_id.

DATA v1_brf_expression TYPE brf_expression.

```

```
DATA v1_preval          TYPE char30.
```

```

=====
* Preset return parameters
=====
  ev_data_missing = 'X'.
  ev_value = 0.          # ev_value depends on the result type.
                        # E.g. for Boolean type result it will be abap_true or abap_false.
  ev_type = 'C'.        # ev_type is the result type of the BRF
                        # Expression. It could be Character, Boolean, constant etc.
  ev_length = 32.
  CLEAR ev_currency.
  ev_output_length = 32.
  ev_decimals = 0.
=====
* Get purchasing document
=====
* get event object
  ol_wf_brf_event ?= io_event.

* get context container from BRF event
  ol_context_provider = ol_wf_brf_event->get_context_provider( ).
* get document
  CALL METHOD ol_context_provider->get_document
    IMPORTING
      ev_document_guid = v1_document_guid
      ev_document_type = v1_document_type.
* get instance
  ol_wf_pdo ?= /sapsrm/cl_wf_pdo_impl_factory=>get_instance(
    iv_document_guid = v1_document_guid
    iv_document_type = v1_document_type
  ).
=====
* Get Document Owner
=====
  il_document_owner = ol_wf_pdo->get_document_owner( ).
  READ TABLE il_document_owner REFERENCE INTO r1_document_owner INDEX 1.
  ASSERT sy-subrc = 0.
=====
* Specialize to Shopping Cart
=====

```

```

IF ol_wf_pdo->get_document_type( ) NE
/sapsrm/if_pdo_obj_types_c=>gc_pdo_shop.
    ev_data_missing = 'X'.
    EXIT.
ENDIF.
TRY.
    ol_pdo_sc ?= ol_wf_pdo->get_pdo( ).
    CATCH /sapsrm/cx_pdo_error
        /sapsrm/cx_pdo_abort.
    ENDTRY.
*=====
* Get Shopping Cart Details for Scheme Evaluation
*=====
TRY.
    CALL METHOD ol_pdo_sc->get_header_detail
        IMPORTING
            es_header = wl_header.
*    sl_sc_total_value-value = wl_header-total_value.
*    sl_sc_total_value-currency = wl_header-currency.
    wl_header_guid-guid = wl_header-guid.
    APPEND wl_header_guid TO il_header_guid.
    CALL METHOD ol_pdo_sc->/sapsrm/if_pdo_base~get_item_list
        EXPORTING
            it_parent_guid = il_header_guid
        IMPORTING
            et_item_guid = il_item_guid.
    CATCH /sapsrm/cx_pdo_wrong_bus_type

        /sapsrm/cx_pdo_pd_read_error
        /sapsrm/cx_pdo_lock_failed
        /sapsrm/cx_pdo_no_authorizatio
        /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_missing = 'X'.
        EXIT.
    ENDTRY.

    CLEAR il_item.
    LOOP AT il_item_guid REFERENCE INTO rl_item_guid WHERE leaf = abap_true.

```

```

TRY.
  CALL METHOD ol_pdo_sc->get_item_detail
    EXPORTING
      iv_item_guid = rl_item_guid->guid
    IMPORTING
      et_item      = il_item_i.
  APPEND LINES OF il_item_i TO il_item.
CATCH /sapsrm/cx_pdo_no_authorizatio
/sapsrm/cx_pdo_abort .
  ev_data_missing = 'X'.
EXIT.
ENDTRY.
ENDLOOP.
***** CUSTOMER
SPECIFIC PROCESSING LOGIC *****
*****
*****
Internal table il_item has got the Shopping Cart Item data. This data can
now be evaluated as per customer specific logic in this section. Write
processing logic in this loop based on specific customer requirements.
Return appropriate value of process schema to vl_preval. This value will be
assigned to ev_value after the end of this customer specific processing
logic.

ev_value = vl_preval.

***** END of
CUSTOMER SPECIFIC PROCESSING*****
*****
*****
CLEAR ev_data_missing.
ENDFUNCTION.

```

Create a BRF Expression using above Function Module

Create a BRF Expression in the customer namespace, choose expression type as OCF001 and group as BRF Group represented in Customer Namespace

Expression	ZEX_PROC_SCHM_EVAL		
Administrative Data			
Short Text	Expression for Process Schema Evaluations		
Group	ZCUSTOMER		
Application Class	SRM_WF	SRM Application Class for BRF	
Expression Type	OCF001	CL_CALL_FUNC_METHOD_BRF	Function Module BAdI as Expression
<input type="checkbox"/> <input type="checkbox"/>			
Result Typing			
Result Type	C	Characters	
Fld/Struct. Length	30	Output Length	30
General Settings			
Context			
<input type="checkbox"/> Application Buffering			
<input type="checkbox"/> Invalidate			
<input type="checkbox"/> Data from Test Envirmt			
Data Access			
Procurement Type	Function Module		
<input type="checkbox"/> Calculation of Parameters in Function Module/BAdI/Method			
Access FM	ZWF_BRF_SCHEMA_EVAL		
Parameter			
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>			
<input type="checkbox"/> Parameter ShortTxt Result Type Outp.Len...			

Process Level Evaluation

Once the Schema Evaluation is complete and the above expression returns a Process Schema, system will then evaluate different process steps within this schema. Next question here is “**How do we design the structure of each process schema?**”

Let us solve this puzzle using this matrix:

1. **How many Approval Steps are required in each Process Schema? How many of them are mandatory and how many are optional.**
 - a. Once an approval path is picked up by the system in process schema evaluation, there could be different steps of approval within that path for any shopping cart.
 - b. Few of these steps could be mandatory always , meaning that Shopping Cart will always go to the agents of these approval steps.
 - c. Others could be optional depending on the data in the cart. For example, invoke Buyer approval if there is any free text item in the shopping cart.

2. **Decision about Approval Type**
 - a. Are approvers required to change the SC data during Approval? This decision will decide the Approval Type of the process step.
 - b. **Note:** it is not possible to use Approval type “**Approval with Completion**” after Approval Type “**Approval**”
 - c. **Last Step:** Last approval step should be “**Automatic (System User)**”. This is controlled by standard event **0EV999**, which ensures that the step gets invoked if none of the previous process steps are invoked.

3. **What is the condition for invoking a particular process level?**
 - a. We then need to understand when each of these steps has to be invoked.

4. **Is there any standard BRF expression available to fulfill all or part of the above criteria?**
 - a. There are lots of BRF Expressions delivered by SAP in standard solution. Can these BRF Expressions be used to evaluate the above criteria?
 - b. If pre delivered BRF Expressions can fulfill all or part of the criteria to evaluate the above requirements, then we don't need to write a custom BRF Expression for the same. For example, if the criteria for picking a process level is dependent on the Purchase Organization and value in customer fields in SC data, then we can use the standard expression for determining Purchase organization and write a custom expression for evaluating the data in customer fields.
 - c. Sometimes the criteria for determining a approval path consists of multiple evaluations which could be fulfilled by various Standard BRF Expressions. These BRF expressions can be utilized in a custom BRF expression to meet the requirements.
 - d. For process steps which are always invoked, use the BRF Expression which always returns a TRUE Value. 0C_WF_B_FALSE is one such standard expression.

5. **Agent Determination/Parameter Decision**
 - a. Decide about the BADI Implementation for Agent Determination with or without using the Parameter.

6. **Task**
 - a. Decision about task to be used is very important. Task used depends on the Approval Type picked up for that process step.

7. **Decision Type :** - This selection depends on various factors.
 - a. Are decision sets required for the particular step, decision sets restrictions on selection of decision type.
 - b. Will Approvers take a decision on all items of the SC at item level?

c. Will approvers take a decision on all items of SC at document level?

Pick a relevant decision type based on the answer to the above questions....

Complete the Process Level Matrix after getting the answers to the above questions.

Process Level	When?	Approval Type	Event Id	Expression	Agent Determination	Parameter	Tasks	Decision Type
ZWF_PROC_SCHM1	Always Or Conditional	Approval with Completion OR Approval	Create an Evaluation id for this process step	Evaluation Criteria - Standard or Custom Expression?	BADI Implementation for Agent Determination	Parameter value for BADI Implementation, if Any	Workflow Task e.g. 40007952 for Approval with Completion 40007953 for Approval	1,2 3 or 4
		Approval with Completion OR Approval					Workflow Task e.g. 40007952 for Approval with Completion 40007953 for Approval	1,2 3 or 4
	Always Or Conditional	Approval with Completion OR Approval	Create an Evaluation id for this process step	Evaluation Criteria - Standard or Custom Expression?	BADI Implementation for Agent Determination	Parameter value for BADI Implementation, if Any	Workflow Task e.g. 40007952 for Approval with Completion 40007953 for Approval	1,2 3 or 4
		Approval with Completion OR Approval					Workflow Task e.g. 40007952 for Approval with Completion 40007953 for Approval	1,2 3 or 4
	Always Or Conditional	Approval with Completion OR Approval	Create an Evaluation id for this process step	Evaluation Criteria - Standard or Custom Expression?	BADI Implementation for Agent Determination	Parameter value for BADI Implementation, if Any	Workflow Task e.g. 40007952 for Approval with Completion 40007953 for Approval	1,2 3 or 4
		Automatic (System User)					0EV999	Evaluation Criteria - Standard or Custom Expression?

ZWF_PROC_SCHM2	Always	<i>Approval with Completion</i>	Create an Evaluation id for this process step	Evaluation Criteria - Standard or Custom Expression?	BADI Implementation for Agent Determination	Parameter value for BADI Implementation, if Any	Workflow Task e.g.	1,2 3 or 4
	Or	<i>Approval</i>					40007952 for <i>Approval with Completion</i>	
	Conditional						40007953 for <i>Approval</i>	
Always	<i>Approval with Completion</i>	Create an Evaluation id for this process step	Evaluation Criteria - Standard or Custom Expression?	BADI Implementation for Agent Determination	Parameter value for BADI Implementation, if Any	Workflow Task e.g.	1,2 3 or 4	
Or	<i>Approval</i>					40007952 for <i>Approval with Completion</i>		
Conditional						40007953 for <i>Approval</i>		
Always	<i>Approval with Completion</i>	Create an Evaluation id for this process step	Evaluation Criteria - Standard or Custom Expression?	BADI Implementation for Agent Determination	Parameter value for BADI Implementation, if Any	Workflow Task e.g.	1,2 3 or 4	
Or	<i>Approval</i>					40007952 for <i>Approval with Completion</i>		
Conditional						40007953 for <i>Approval</i>		
		<i>Automatic (System User)</i>	0EV999	Evaluation Criteria - Standard or Custom Expression?	BADI Implementation for Agent Determination	Parameter value for BADI Implementation, if Any	40007988 for <i>Automatic Approval</i>	This step always have decision type
							40007989 for <i>Automatic Rejection</i>	1 – Decision for Entire document

Complete the process schema matrix with Process Level details, after getting the answers to the above questions.

Example: Creating a Custom BRF Expression using Multiple Standard Expressions

Let us understand this using an example where a process level has to be invoked when the Shopping Cart items have any of four specific product categories and gross value of the shopping cart is more than 5000.

There are two standard expressions available for evaluating existence of Product Category and Gross Value of the Shopping Cart. Let us refer to these two expressions in a custom expression to give the result for the required process step.

Step 1: Create custom expression (of type Constant) having Four product categories as constant values. If there are more categories, we can create multiple expressions of this type. Format in constant values will be Product Category\Logical System of product Category. Multiple values can be separated by “;”. Result type of this expression has to be C.

Expression	ZWF_EX_PRCAT_LIST		
Administrative Data			
Short Text	Product Category List		
Group			
Application Class	SRM_WF	SRM Application Class for BRF	
Expression Type	0CN001	CL CONSTANTS SIMPLE BRE	Constant
 			
Result Typing			
Result Type	C	Characters	
Fld/Struct. Length	50	Output Length	0
General Settings			
<input checked="" type="checkbox"/> Application Buffering			
<input type="checkbox"/> Invalidate			
<input type="checkbox"/> Data from Test Envirmnt			
Constant			
Constant	CAT1\LOGSYS;CAT2\LOGSYS;CAT3\LOGSYS;CAT4\LOGSYS		
Description			
        			

Step 2: Create a custom expression for checking whether the product categories in above expressions are available in the shopping cart or not. This expression has to be a Formula based expression and the above expression having the product categories as constants will be assigned to this expression as shown below. Result type of this expression will be Boolean.

Expression

Administrative Data

Short Text

Group

Result Typing

Result Type Boole

Positive Formulatr

Negative Formulatr

Fld/Struct. Lngth Output Length

General Settings

Context

Application Buffering

Invalidate

Data from Test Envirmt

Data Access

Procurement Type

Calculation of Parameters in Function Module/BAdI/Method

Access FM Expression rule adapter

Parameter

Parameter	ShortTxt	Result Type	Outp.Len...
0C_C1_C_FWFSCRLCNTNT	Class: /S	C	28
0C_C2_C_CONTNSPRCAT	Method: C	C	30
ZWF_EX_PRCAT_LIST	Product C	C	50

Step 3: Create a custom expression for checking whether the shopping cart items have any of the above product categories and value of the cart is more than 5000. This expression has to be formula based expression and will return a character value.

This expression will return a value X if both the expression ZWF_EX_PRCAT_CHECK and OV_SC_GROSSTOTLVLU satisfies the criteria. The result will be space if any of the expression is not satisfied.

Expression	ZWF_EX_PRLVL_EVAL		
Administrative Data			
Short Text	Expression for evaluating Process Level 1		
Group			
Application Class	SRM_WF	SRM Application Class for BRF	
Expression Type	0FB001	CL_FORMULA_BRF	SAP Formula Interpreter
 			
Result Typing			
Result Type	C	Characters	
Fld/Struct. Length	1	Output Length	0
General Settings			
<input type="checkbox"/> Application Buffering			
<input type="checkbox"/> Invalidate			
<input type="checkbox"/> Data from Test Envirmt			
Formula			
 FormulaEditor  Techn./Long Text  ID			
<pre>IF (ZWF_EX_PRCAT_CHECK AND Gross Total Value > 5,000, 'X', ' ')</pre>			

Step 4: Create another formula based BRF Expression ZWF_EX_PROCESS_1 with result type as Boolean. The formula in the expression will be ZWF_EX_PRLVL_EVAL = 'X'. If the above expression returns a result 'X' then the expression ZWF_EX_PROCESS_1 will give TRUE as result, if the above expression returns ' ', then the boolean expression ZWF_EX_PROCESS_1 will give FALSE.

This expression will be assigned to the Event of the process step in Process configuration and the process step will be invoked depending on the outcome of the expression ZWF_EX_PROCESS_1

Expression: ZWF_EX_PROCESS_1

Administrative Data

Short Text: Expression for Evaluating Process Step 1

Group: [Empty]

Result Typing

Result Type: B Boole

Positive Formulatr: [Empty]

Negative Formulatr: [Empty]

Fld/Struct. Lngth: 1 Output Length: 1

General Settings

Application Buffering

Invalidate

Data from Test Envirmnt

Formula

FormulaEditor Techn./Long Text ID

ZWF_EX_PROCESS_1 = 'X'

Example: Creating a Custom BRF Expression using Custom Function Module

Custom BRF expression using Custom Function Module for evaluating process levels can be created on the same lines as we created a custom expression for evaluating process level.

- a. Create a Function Module ZWF_BRF_PRLEVEL_1 Function Module /SAPSRM/WF_BRF_0EXP000 is used by SAP for lot of expressions in standard delivery. We may reference this FM while creating the custom function module. Import/ Export parameters of this FM will be as follows

Attributes Import Export Changing Tables Exceptions Source code			
Parameter Name	Typing	Associated Type	Default value
IT_EXPRESSIONS	TYPE	SBRF260A_T	
IO_EVENT	TYPE REF TO	IF_EVENT_BRF	
IO_EXPRESSION	TYPE REF TO	IF_EXPRESSION_BRF	

Attributes	Import	Export	Changing	Tables	Exceptions	Source code
Parameter Name	Typing	Associated Type	Pass Val	Short text		
EV_VALUE	TYPE	BRF_RESULT_VALUE	<input type="checkbox"/>			
EV_TYPE	TYPE	BRF_RESULT_TYPE	<input type="checkbox"/>			
EV_LENGTH	TYPE	BRF_RESULT_LENGTH	<input type="checkbox"/>			
EV_CURRENCY	TYPE	BRF_CURRENCY	<input type="checkbox"/>			
EV_OUTPUT_LENGTH	TYPE	BRF_RESULT_OUTPUT_LEN	<input type="checkbox"/>			
EV_DECIMALS	TYPE	BRF_RESULT_DECIMALS	<input type="checkbox"/>			
EV_DATA_MISSING	TYPE	BRF_DATA_MISSING	<input type="checkbox"/>			

- b. Source Code for the Function Module for BRF Expression will be on the lines of this sample code:

```
FUNCTION ZWF_BRF_PRLEVEL_1.
```

```

* "-----
* "Local Interface:
* " IMPORTING
* "   REFERENCE(IT_EXPRESSIONS) TYPE SBRF260A_T
* "   REFERENCE(IO_EVENT) TYPE REF TO IF_EVENT_BRF
* "   REFERENCE(IO_EXPRESSION) TYPE REF TO IF_EXPRESSION_BRF
* " EXPORTING
* "   REFERENCE(EV_VALUE) TYPE BRF_RESULT_VALUE
* "   REFERENCE(EV_TYPE) TYPE BRF_RESULT_TYPE
* "   REFERENCE(EV_LENGTH) TYPE BRF_RESULT_LENGTH
* "   REFERENCE(EV_CURRENCY) TYPE BRF_CURRENCY
* "   REFERENCE(EV_OUTPUT_LENGTH) TYPE BRF_RESULT_OUTPUT_LENGTH
* "   REFERENCE(EV_DECIMALS) TYPE BRF_RESULT_DECIMALS
* "   REFERENCE(EV_DATA_MISSING) TYPE BRF_DATA_MISSING
* "-----

DATA o1_wf_brf_event TYPE REF TO /sapsrm/cl_wf_brf_event.
DATA o1_context_provider TYPE REF TO /sapsrm/cl_wf_context_provider.
DATA o1_pdo_sc TYPE REF TO /sapsrm/if_pdo_bo_sc.
DATA o1_wf_pdo TYPE REF TO /sapsrm/if_wf_pdo.

DATA v1_document_guid TYPE /sapsrm/wf_document_guid.
DATA v1_document_type TYPE /sapsrm/wf_document_type.

DATA il_document_owner TYPE /sapsrm/t_wf_agent_id.
DATA il_header_guid TYPE bbpt_guid.
DATA il_item_guid TYPE /sapsrm/t_pdo_hier_guid_list.
DATA il_item TYPE bbpt_pd_sc_item_d.
DATA il_item_i TYPE bbpt_pd_sc_item_d.
```

```
DATA w1_header TYPE bbp_pds_sc_header_d.
```

```
DATA w1_header_guid TYPE bbp_guid_tab.
```

```
DATA w1_item LIKE LINE OF il_item.
```

```
DATA r1_item TYPE REF TO bbp_pds_sc_item_d.
```

```
DATA r1_item_guid TYPE REF TO /sapsrm/s_pdo_hier_guid_list.
```

```
DATA r1_document_owner TYPE REF TO /sapsrm/wf_agent_id.
```

```
DATA v1_brf_expression TYPE brf_expression.
```

```
DATA v1_pr1vlval TYPE char1.
```

```

*=====
* Preset return parameters
*=====
  ev_data_missing = 'X'.
  ev_value = 0.          # ev_value depends on the result type. E.g. for
                        # Boolean type result it will be abap_true or abap_false.
  ev_type = 'C'.        # ev_type is the result type of the BRF
                        # Expression. It could be Character, Boolean, constant etc.
  ev_length = 1.
  CLEAR ev_currency.
  ev_output_length = 32.
  ev_decimals = 0.
*=====
* Get purchasing document
*=====
* get event object
  ol_wf_brf_event ?= io_event.

* get context container from BRF event
  ol_context_provider = ol_wf_brf_event->get_context_provider( ).
* get document
  CALL METHOD ol_context_provider->get_document
    IMPORTING
      ev_document_guid = v1_document_guid
      ev_document_type = v1_document_type.
* get instance
  ol_wf_pdo ?= /sapsrm/cl_wf_pdo_impl_factory=>get_instance(
    iv_document_guid = v1_document_guid
    iv_document_type = v1_document_type
  ).
*=====

```

* Get Document Owner

```

=====
il_document_owner = ol_wf_pdo->get_document_owner( ).
READ TABLE il_document_owner REFERENCE INTO rl_document_owner INDEX 1.
ASSERT sy-subrc = 0.
=====

```

* Specialize to Shopping Cart

```

=====
IF ol_wf_pdo->get_document_type( ) NE
/sapsrm/if_pdo_obj_types_c=>gc_pdo_shop.
  ev_data_missing = 'X'.
  EXIT.
ENDIF.
TRY.
  ol_pdo_sc ?= ol_wf_pdo->get_pdo( ).
  CATCH /sapsrm/cx_pdo_error
    /sapsrm/cx_pdo_abort.
ENDTRY.
=====

```

* Get Shopping Cart Details for Scheme Evaluation

```

=====
TRY.
  CALL METHOD ol_pdo_sc->get_header_detail
    IMPORTING
      es_header = wl_header.
  *   sl_sc_total_value-value = wl_header-total_value.
  *   sl_sc_total_value-currency = wl_header-currency.
  wl_header_guid-guid = wl_header-guid.
  APPEND wl_header_guid TO il_header_guid.
  CALL METHOD ol_pdo_sc->/sapsrm/if_pdo_base-get_item_list
    EXPORTING
      it_parent_guid = il_header_guid
    IMPORTING
      et_item_guid = il_item_guid.
  CATCH /sapsrm/cx_pdo_wrong_bus_type

    /sapsrm/cx_pdo_pd_read_error
    /sapsrm/cx_pdo_lock_failed
    /sapsrm/cx_pdo_no_authorizatio
    /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_missing = 'X'.
  EXIT.
ENDTRY.

CLEAR il_item.
LOOP AT il_item_guid REFERENCE INTO rl_item_guid WHERE leaf = abap_true.
  TRY.
    CALL METHOD ol_pdo_sc->get_item_detail
      EXPORTING
        iv_item_guid = rl_item_guid->guid
      IMPORTING
        et_item      = il_item_i.
    APPEND LINES OF il_item_i TO il_item.
  CATCH /sapsrm/cx_pdo_no_authorizatio
    /sapsrm/cx_pdo_abort .
    ev_data_missing = 'X'.
    EXIT.
  ENDTRY.
ENDLOOP.

```

```

***** CUSTOMER
SPECIFIC PROCESSING LOGIC *****
*****
**

```

Internal table `il_item` has got the Shopping Cart Item data. This data can now be evaluated as per customer specific logic in this section. Write processing logic in this loop based on specific customer requirements. Return X to `vl_preval`. This value will be assigned to `ev_value` after the end of this customer specific processing logic.

```
ev_value = vl_prlvleval.
```

```

***** END of
CUSTOMER SPECIFIC PROCESSING*****
*****
**

```

```
CLEAR ev_data_missing.
```

```
ENDFUNCTION.
```

Create a BRF Expression using above Function Module

Create a BRF Expression in the customer namespace, choose expression type as OCF001 and group as BRF Group represented in Customer Namespace

Expression	ZWF_EX_PROCESS_EVAL		
Administrative Data			
Short Text	Expression for Evaluating SC Data using Customer Logic		
Group	<input type="text"/>		
 			
Result Typing			
Result Type	C	Output Length	<input type="text"/>
Fid/Struct. Lngth	1		
General Settings			
Context	<input type="text"/>		
<input type="checkbox"/> Application Buffering			
<input type="checkbox"/> Invalidate			
<input type="checkbox"/> Data from Test Envirmt			
Data Access			
Procurement Type	Function Module 		
<input type="checkbox"/> Calculation of Parameters in Function Module/BAdI/Method			
Access FM	ZWF_BRF_PRLEVEL_1		
Parameter			
   			
	Parameter	Short Text	ResultType
			Outp.Leng.

In order to decide whether the particular process level should invoke or not, create another custom expression which will read the outcome of above function module based expression ZWF_EX_PROCESS_EVAL and return a Boolean result accordingly. This expression will be assigned to the respective event id of the process level and will control whether the process level will be invoked or not

Expression	ZWF_EX_PROCESS_2		
Administrative Data			
Short Text	Expression for Evaluating Process Level 2		
Group	<input type="text"/>		
<input type="button" value="+"/> <input type="button" value="i"/>			
Result Typing			
Result Type	B	Boole	
Positive Formulatr	<input type="text"/>		
Negative Formulatr	<input type="text"/>		
FId/Struct. Lngth	1	Output Length	1
General Settings			
<input type="checkbox"/> Application Buffering			
<input type="checkbox"/> Invalidate			
<input type="checkbox"/> Data from Test Envirmnt			
Formula			
<input type="button" value="FormulaEditor"/> <input type="button" value="Techn./Long Text"/> <input type="button" value="ID"/>			
ZWF_EX_PROCESS_EVAL = 'X'			

Disclaimer and Liability Notice

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

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

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