

How to reuse BRFplus Functions Similar to R/3 Function Modules using BRF+ Expression Type “Function Call”



Applies to:

Tax and Revenue Management.

Summary

During the building process of BRF+ Rules you might realized that there are rules that will need to be used by more than one process or tax type and you will like to reuse them. This cookbook describes how this can be done and also gives you some examples about it.

Author: Diego Gaudenzi

Company: SAP America

Created on: 12 October 2011

Author Bio



Diego Gaudenzi has worked at SAP since January 2001. He joined the IBU Public Sector and worked as a software developer for 8 years and started his current position as Solution manager for Tax and Revenue Management in 2009.

Table of Contents

Applies to:	1
Summary.....	1
Author Bio	1
Table of Contents	2
Introduction	3
Steps to create a BRF+ Application, Function, Ruleset and Rules to be reused	3
Create BRF+ Application	3
Create BRF+ Function	5
Create BRF+ RuleSet.....	8
Create BRF+ Rules.....	9
Save and activate all the BRF+ Objects you have created	9
Step by Step to reuse the BRF+ Functions	10
Step 1: Create an Expression type Function Call.....	10
Step 2: Select the BRF+ Function you want to reuse:.....	11
Step 3: Maintain the Mapping Context	12
Examples of Functions to be reused	13
Related Content.....	14
Copyright.....	15

Introduction

Since the introduction of BRF+ for Tax and Revenue Management in Enhancement Package 5 one common question I have heard several times is "How do I reuse rules across BRF+ Applications?" Which translates to "how do I reuse rules across TRM Processes like Tax registration, Tax Refund and Tax Processing" which also translates to "how can rules across Tax Types be reused?" since normally different tax types will be processed by different BRF+ Applications.

This concern regarding to reusing rules is a very valid one especially for those folks that have worked and implemented TRM using BRF where reusability was very limited and almost not available.

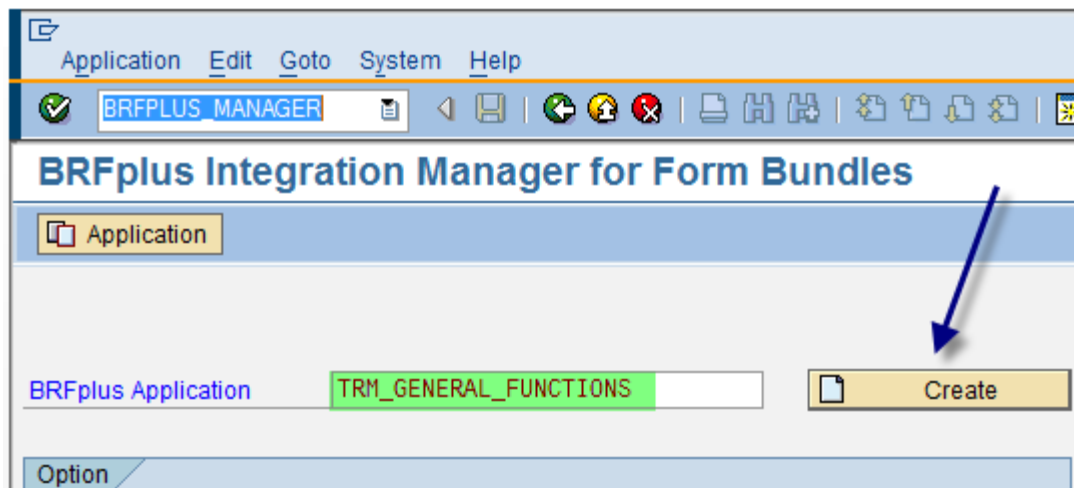
So, what I want to show here is a very easy way to reuse rules within BRF+ Functions/Rulesets following the same philosophy ABAP developers have followed for years when they build and use Function Modules in the backend system.

Steps to create a BRF+ Application, Function, Ruleset and Rules to be reused

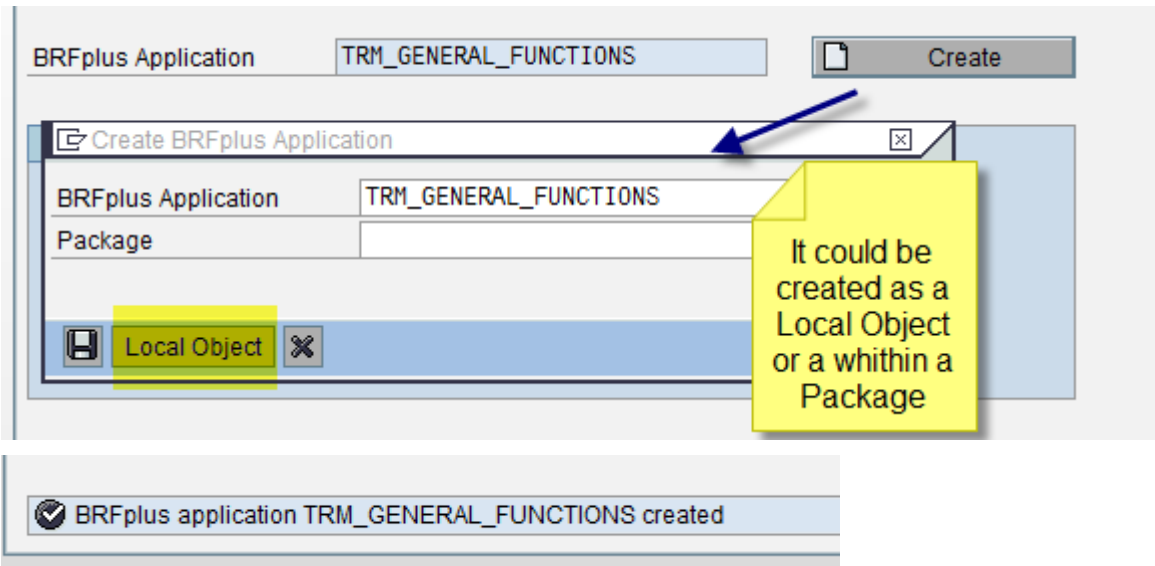
Create BRF+ Application

In R/3 when a new Function Module is created, it has to be assigned to a Function Group. Translating that to the BRF+ world, we need to create a BRF+ Application (e.g. TRM_GENERAL_FUNCTIONS) where we will accommodate the BRF+ Functions. This Application can be created using T-code BRFPLUS_MANAGER or from the BRF+ Workbench (T-code BRF+ or BRFplus).

From BRFPLUS_MANAGER:



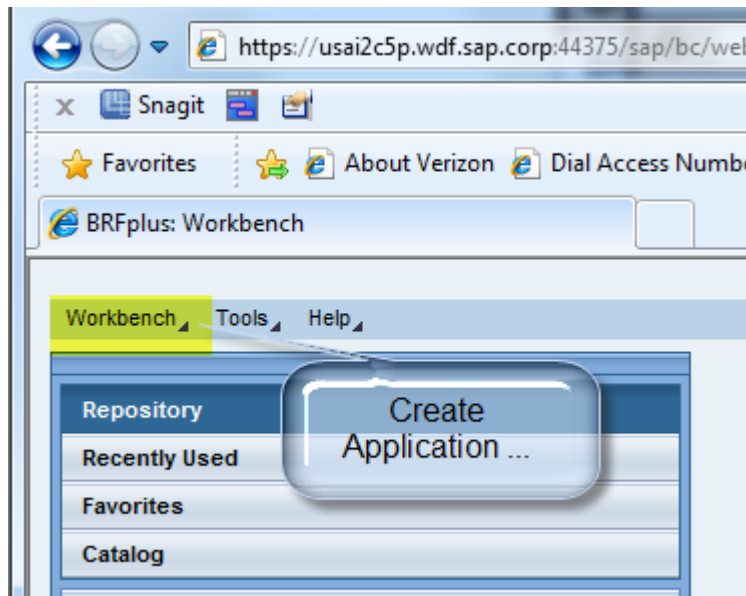
Assign a Package or create it as a Local Object:



From BRF+ Workbench:

Note: You can start the BRFplus workbench either in the backend system with transaction BRFplus or in the web browser with the BRFplus workbench URL that is specific for your system landscape.

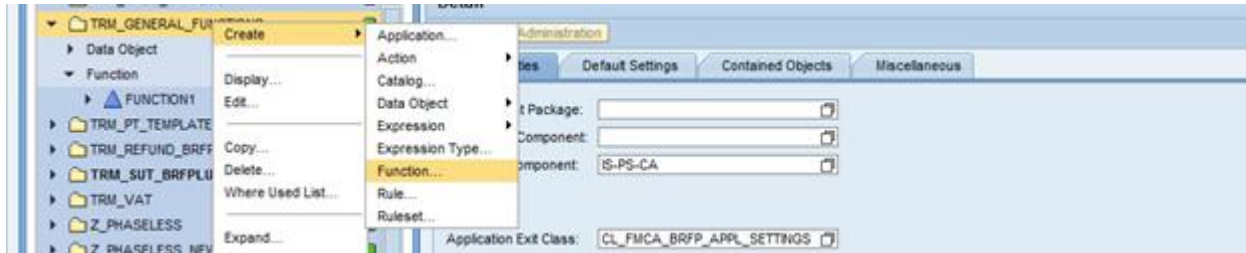
Prerequisite is that the role BRFplus Administrator (SAP_BC_FDT_ADMINISTRATOR) has been added to your user profile.



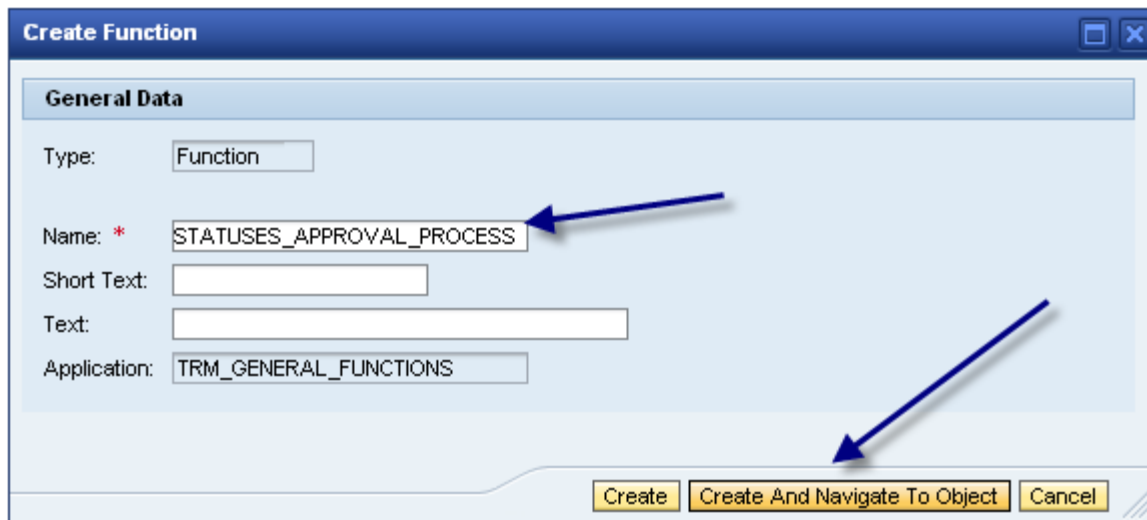
Create BRF+ Function

After the Application is created the BRF+ Functions can be created. All the BRF+ Functions that will be reused can be created under the BRF+ Application created above. Create the BRF+ Function using the BRF+ Workbench. By now you should have already the interface of your function module well defines (e.g. Signature Context/Inputs and Result Data Object/output including the data types to be used):

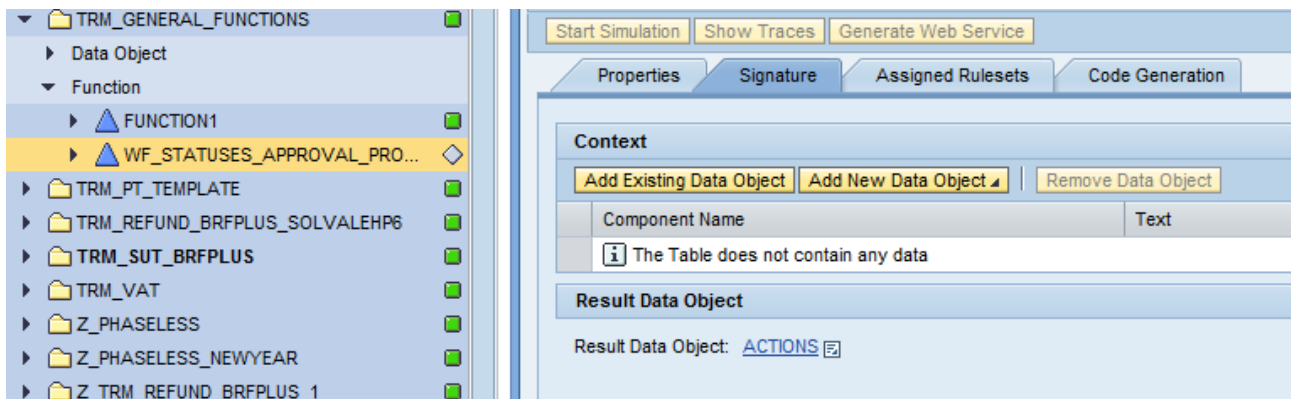
Mouse right click on the Application name to create the new Function:



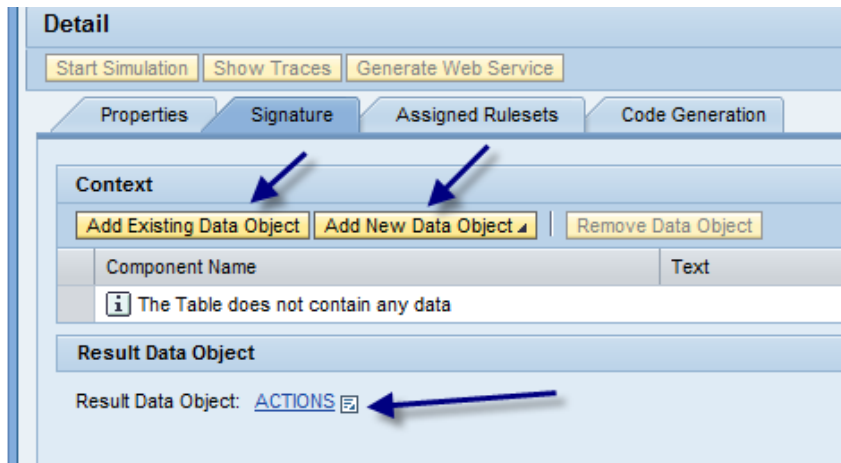
Enter a Name and Click "Create And Navigate To Object":



Click on the Signature Tab where the Function Interface is defined:

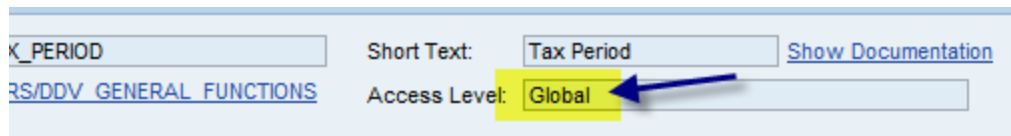


Now for the Context (e.g. Input parameters) you can Add new Data Objects or if you already have created them just add them up. For the Result Data Object you can select or create it:

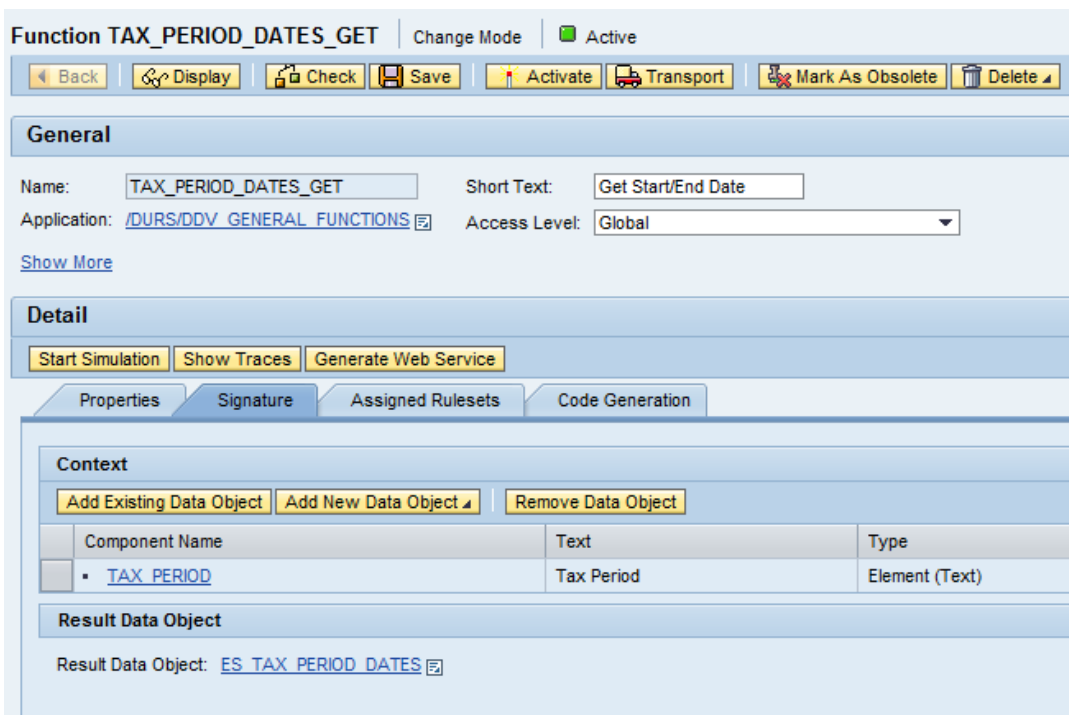


Keep in mind that the Data Objects assigned to the Context and Result Data Object can be Elements, Structures or Tables.

Note: Be sure that every object in the Context and Result Data Object have Access Level equal to **Global**:



Here some screen shots of a sample Function and its signature. On this example Context Element TAX_PERIOD is linked to DDIC Element PERSL_KK and Result Data Object ES_TAX_PERIOD_DATES is a structure that contains two fields:



Element TAX_PERIOD | Change Mode | Active

General

Name: Short Text: [Show D](#)

Application: [/DURS/DDV GENERAL FUNCTIONS](#) Access Level:

[Show More](#)

Detail

Define Data Binding

Select Binding Type:

DDIC Element: Period key

Result Data Object:

Structure ES_TAX_PERIOD_DATES | Change Mode | Active

General

Name: Short Text:

Application: [/DURS/DDV GENERAL FUNCTIONS](#) Access Level:

[Show More](#)

Detail

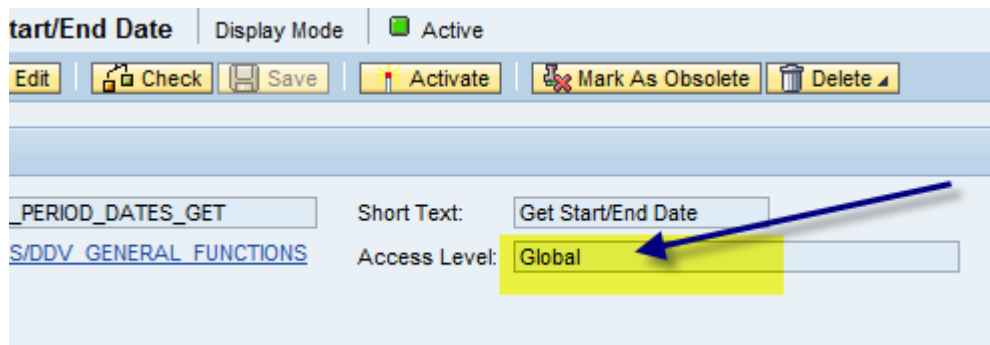
Define Data Binding

Select Bind Type:

Structure Definition

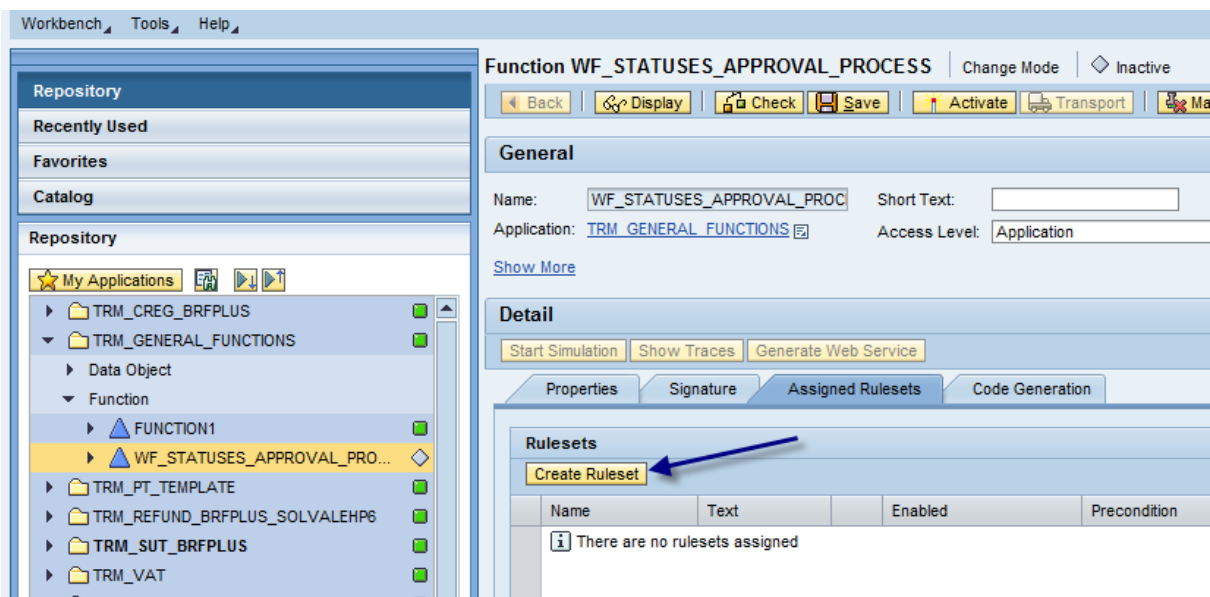
Component Name	Text	Type
▪ ABRZU	Bill Per	Element (Timepoint - Date)
▪ ABRZO	Bill Per	Element (Timepoint - Date)

Finally you need to go, at the Function level, in change mode and set up the Access Level to **Global**:

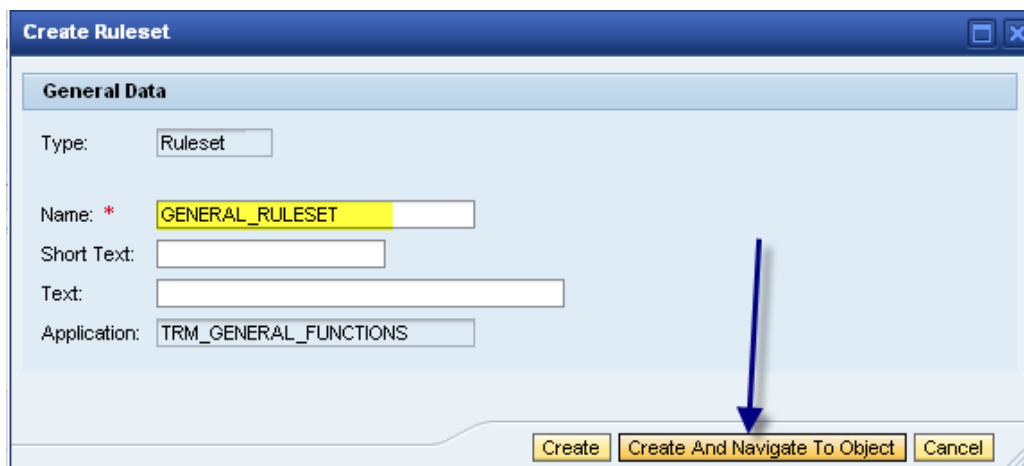


Create BRF+ RuleSet

The next step is to create the Rulesets where the Rules will be located. To create the Ruleset navigate to your Function and under the "Assigned Rulesets" tab click on "Create Ruleset":



Now enter a Ruleset Name and click "Create and Navigate To Object":



Create BRF+ Rules

The next step is to create the rules by clicking "Insert Rule". When building your rules for your BRF+ Function, keep in mind the Signature of your Function, what the input parameters are and what do you have to return as a Result Data Object:

The screenshot shows the SAP BRF+ Rule Editor interface for a rule set named 'GENERAL_RULESET'. The interface is divided into several sections:

- Header:** Shows the rule set name 'GENERAL_RULESET', a 'Change Mode' button, and a status indicator 'Inactive'.
- Toolbar:** Contains buttons for 'Back', 'Display', 'Check', 'Save', 'Activate', 'Transport', 'Mark As Obsolete', and 'Delete'.
- General Section:**
 - Name:** 'GENERAL_RULESET' (text input field)
 - Short Text:** (empty text input field)
 - Application:** 'TRM_GENERAL_FUNCTIONS' (dropdown menu)
 - Access Level:** 'Application' (dropdown menu)
 - [Show More](#) (link)
- Detail Section:**
 - Buttons: 'Hide Ruleset Header', 'Assign Precondition', 'Context Overview'.
 - Status: 'Disable Ruleset' (button), '(Enabled Ruleset)' (checkbox), 'Ruleset contains 0 rule(s) and 0 variable(s)' (text).
 - Trigger: 'Ruleset will be triggered if WF_STATUSES_APPROVAL_PROCESS is processed' (text).
- Rules Section:**
 - Buttons: 'Insert Rule', 'Insert Exit Condition'.
 - A blue arrow points to the 'Insert Rule' button.

Save and activate all the BRF+ Objects you have created

Finally save and activate all the objects within the new Function.

Note: You can create any new Function you will like to reuse under this BRF+ Application.

Step by Step to reuse the BRF+ Functions

On this section I describe the steps needed to reuse BRF+ Functions within your BRF+ Application. The main component to accomplish this is to create a **BRF+ Expression type "Call Function"**.

These steps are based in a real life implementation where a BRF+ Function called `WF_STATUSES_APPROVAL_PROCESS` from application `DDV_GENERAL_FUNCTIONS` is reused in another BRF+ Application by creating Expression "FC_WF_STATUSES":

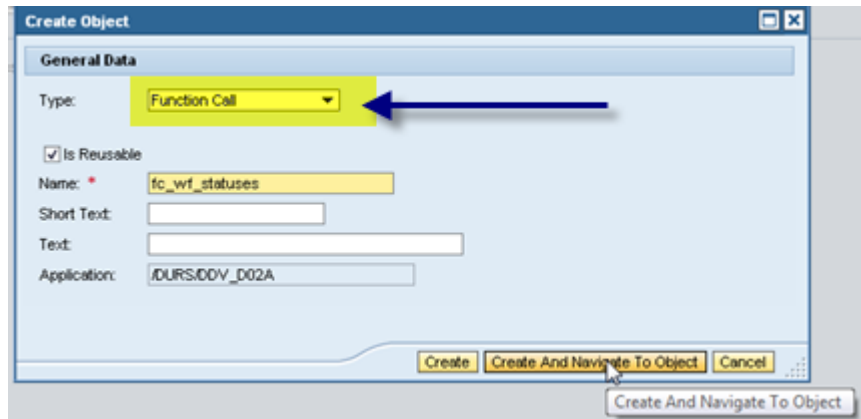
The screenshot shows the configuration for a Function Call named "FC_WF_STATUSES". The interface includes a toolbar with buttons for Back, Display, Check, Save, Activate, and Transport. The "General" section contains fields for Name (FC_WF_STATUSES), Short Text, Application (DURS/DDV_D01A), and Access Level (Application). The "Detail" section shows the Target Function as WF_STATUSES_APPROVAL_PROCESS. The "Mapping" section contains a table with the following data:

Source Context Data Object/Expression	Target Function Context Data Object
FBSTA	IV_FBSTA
LV_FLAG_ERROR	EV_FLAG_ERROR
EVENT	IV_EVENT
SIMULATION	IV_SIMULATION
GV_FLAG_ERROR	IV_ERROR

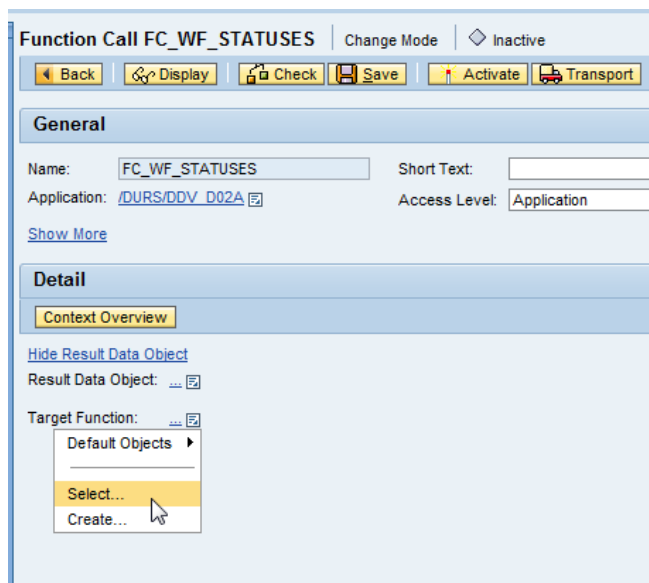
Step 1: Create an Expression type Function Call

Note: A BRF+ Expression can be created from different places on the UI, the screen shots is showing one of these several places which is during the Rule creation:

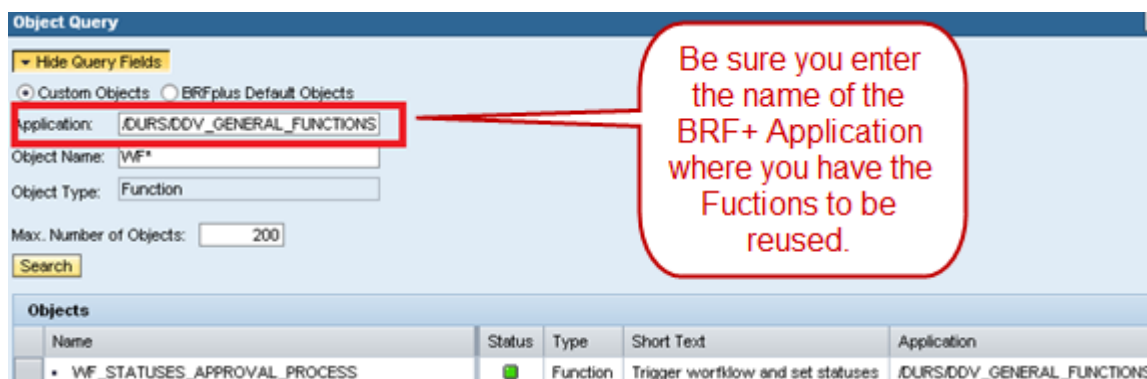
The screenshot shows the configuration for a Rule titled "(3) Rule: No description is available (Enabled Rule)". The rule is valid from 00:00:00 until 00:00:00. The description is "Calling function for changing statuses and triggering wor". The "If" section is empty. The "Then" section is active, and a context menu is open over the "Process Expression" option, showing "Select..." and "Create..." options. The "Create..." option is highlighted by the mouse cursor.



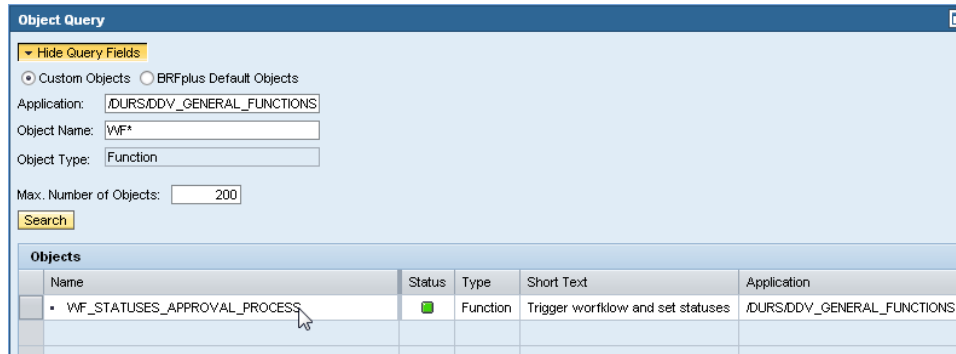
Step 2: Select the BRF+ Function you want to reuse:



Enter the BRF+ Application where you have all your Function to be reused (defaulted is application you are currently working on):



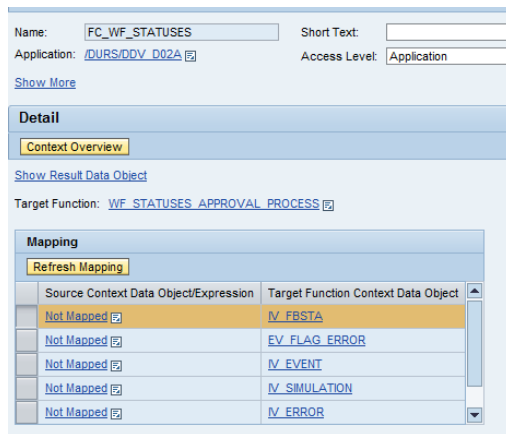
Choose the function you want to reuse:



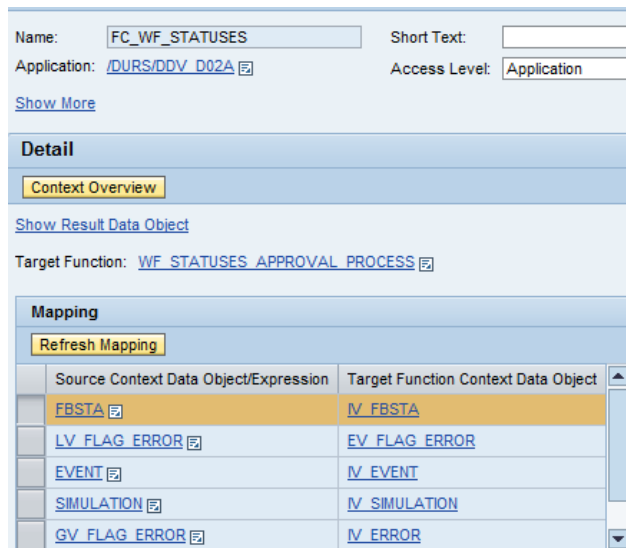
Step 3: Maintain the Mapping Context

Take care of the mapping for all your Input Context data and Result Data Object if you have defined one:

Before mapping:



After mapping:



Examples of Functions to be reused

Here, to give you ideas about what kind of rules are good candidates to be shared, a screen shot of some real life examples of Functions I have seen that are shared among BRF+ Applications. The screen shot does not provide details about these functions but I hope that just by reading the name you can have some good hints:

The screenshot displays the SAP BRF+ configuration interface for the function **TAX_PERIOD_DATES_GET**. The interface is divided into a left-hand repository pane and a right-hand configuration pane.

Left Pane (Repository): Shows a tree view of functions. The function **TAX_PERIOD_DATES_GET** is highlighted under the **Ruleset** category.

Right Pane (Configuration):

- Function Name:** TAX_PERIOD_DATES_GET
- Short Text:** Get Start/End Date
- Application:** DURS/DDV_GENERAL_FUNCTIONS
- Access Level:** Global
- Detail Tab:** Shows a table of rulesets. One ruleset is listed:

Name	Priority	Text
TAX_PERIOD_DATES_GET	00	TAX_PERIOD_DATES_GET

Related Content

[Business Rule Framework plus \(BRFplus\)](#)

Copyright

© Copyright 2011 SAP AG. All rights reserved.

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

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

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

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

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

Oracle is a registered trademark of Oracle Corporation.

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

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

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

Java is a registered trademark of Sun Microsystems, Inc.

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

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects S.A. in the United States and in other countries. Business Objects is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

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