



EXTERNAL

## **How to... Use BPC Business Process Flows to Control a Planning Process using SAP Analytics Cloud Applications with BPC Live Connections**

Prerequisites: SAC QRC1.2020; BPC 10.1, 11.0, or 11.1

# TABLE OF CONTENTS

- SCENARIO ..... 3
- GENERAL DESCRIPTION OF THE SOLUTION..... 3
- STEP BY STEP SOLUTION ..... 3
- Create an SAC Application using the BPC live connection ..... 3
- Create a Launchpad ..... 7

SAP Analytics Cloud can use planning models built in SAP Business Planning and Consolidation (Embedded) via the BPC live connection and can be used to build up sophisticated Web based planning applications using BPC data and features. In this paper we show how we can use Business Process Flows to orchestrate a planning process using such SAC based planning applications. We show how the context of a BPF step can be handed over to the SAC application and how the SAC application can pass this context to parameterize used queries.

## SCENARIO

In SAP Business Planning and Consolidation Embedded version (version 10.1, 11, and 11.1) it is possible to call various applications from a Business Process Flow (BPF). Calling SAP Analysis for Office applications is very easy as they are standard targets in BPF. But what about using a Web based front end?

In an old how to paper we have described how this can be done using Design Studio Applications as targets. But with our new BPC live connection in SAC we have also the option to build applications in SAC using BPC data and features. In BPF any external URL and thus specifically the URL of an SAC application can be used as a link in a BPF step. But obviously the called application should receive the context of the BPF step and should use this context to restrict the queries used in the applications.

## GENERAL DESCRIPTION OF THE SOLUTION

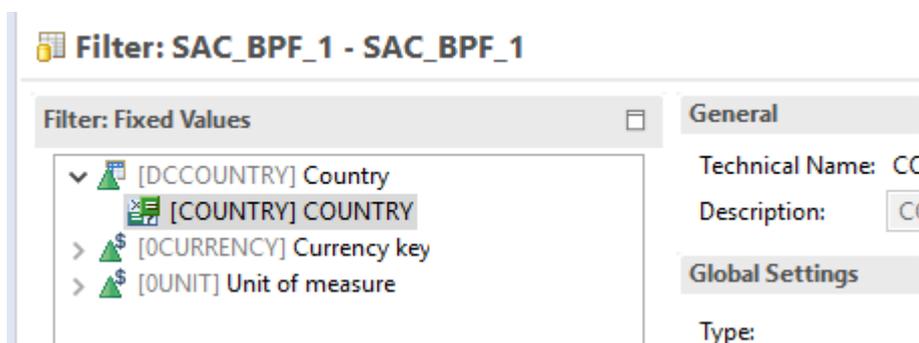
In our example we base BPF on the driving dimension 'Country'. We will call an SAC based application that is used for revenue planning and that uses a query that is to be restricted to the single country from the context. We transmit the value for the country from the BPF context to the SAC application using a URL parameter. The SAC application receives this value and sets a BW variable in the fixed filter of the query. Please note that this technique works for any dimension/characteristic in the context of the BPF.

The solution requires SAC QRC1.2020 or higher.

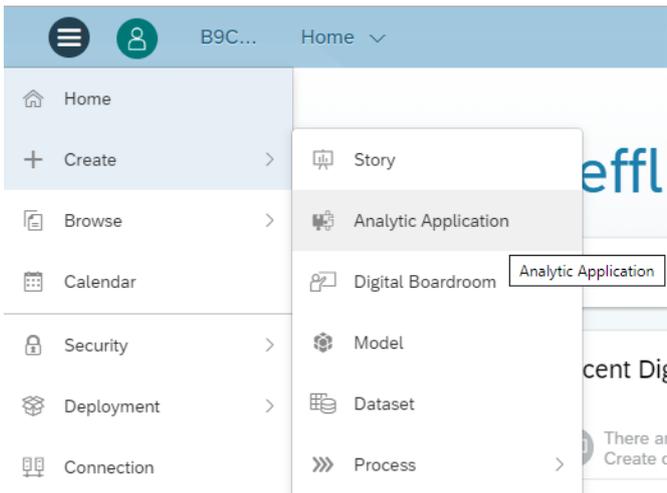
## STEP BY STEP SOLUTION

### Create an SAC Application using the BPC live connection

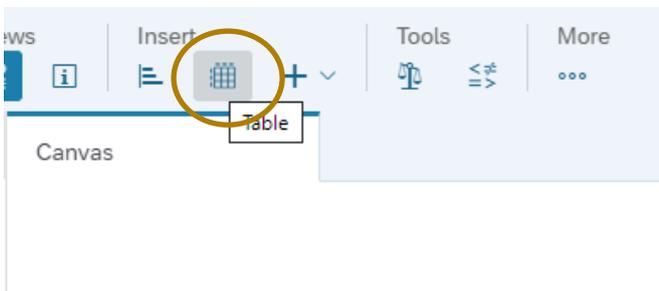
We will not describe every step in detail as they are well described in the corresponding documentation. First create an input enabled query that uses the dimension for country in the fixed filter and uses a variable for the country (input enabled, single value). We simply call the variable 'Country'.



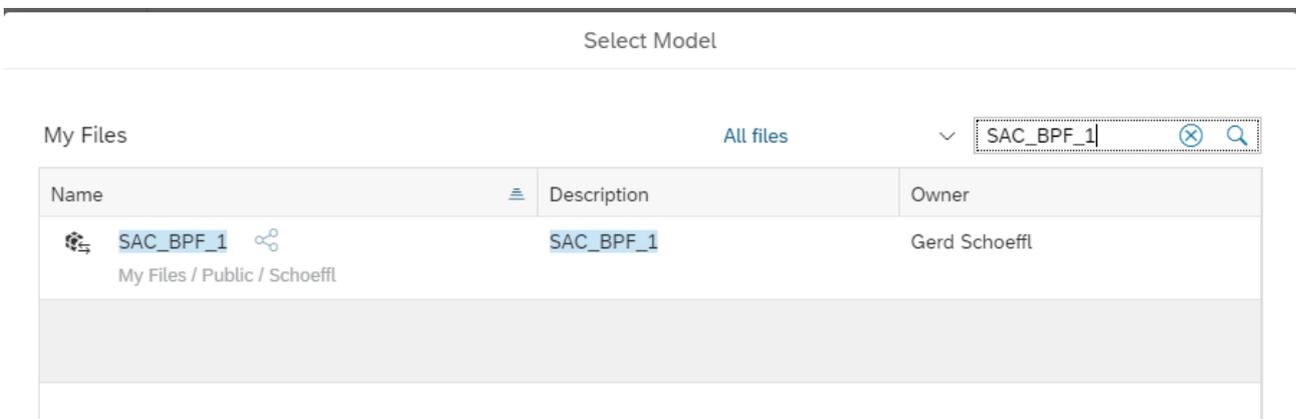
Now go to SAC and create a BPC live model using the query you have just created. After that create a new Analytic Application.



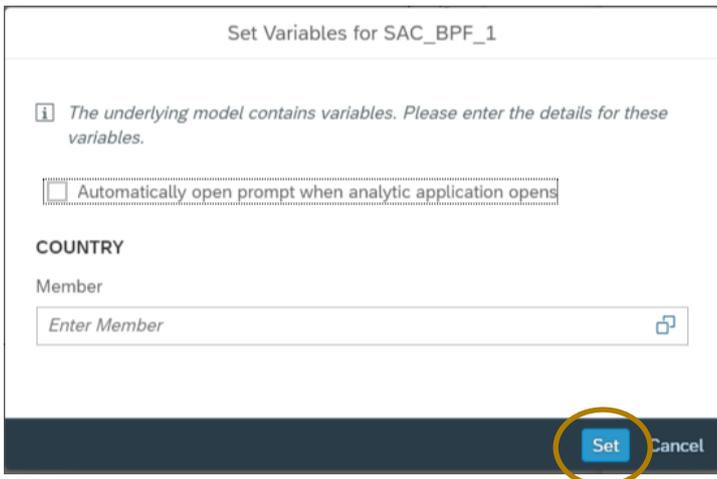
Insert a table in your application.



Select your BPC live model.

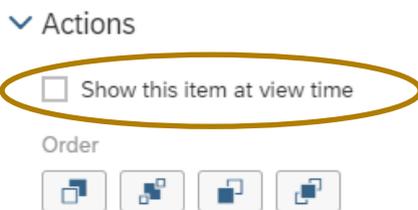


Do not prompt the user for a variable value upon opening. Just press 'Set'.

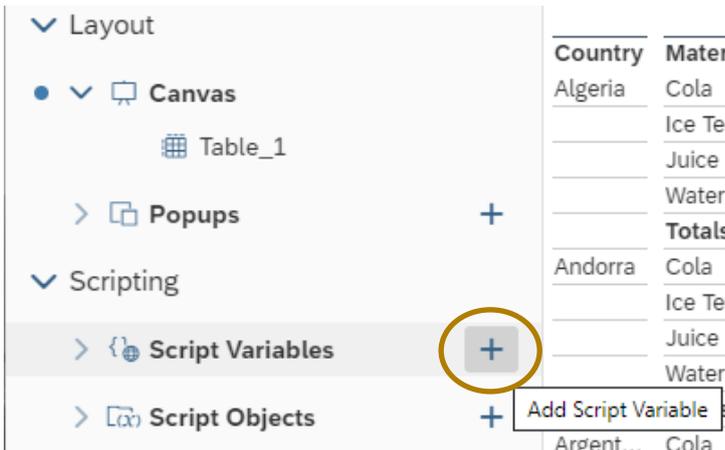


We want to make sure that we do not see the table in the application BEFORE the variable has been set. Thus, we hide the table at start time and display it once the variable is set.

Go to the 'Designer' and change to the 'Styling' tab. Make sure the table is selected. Scroll down and uncheck the flag 'Show this item at view time'.



Now create a script variable that can be exposed as a URL parameter.



Enter a name and a description for the variable. We use the name 'COUNTRY', just as the name of the BW variable. Make sure you check the flag 'Expose variable via URL parameter' as otherwise it cannot be set via an URL parameter.



The first statement sets the variable. The second statement switches the table visible after the variable has been set.

If you have used a different name for the BW variable, then you must change the first parameter in the function accordingly. If you have used another name for the script variable (and thus the URL parameter) you must adapt the second parameter of the function.

You can now save your application. Then click on 'Run Analytic Application' to execute your application and copy the link to that application from the URL. The URL should be similar to

[https://master-planning.master.canary.eu10.projectorca.cloud/sap/fpa/ui/#;mode=present;view\\_id=appBuilding;appld=289E62EAA8C934D647DCB7EB7B531EC6](https://master-planning.master.canary.eu10.projectorca.cloud/sap/fpa/ui/#;mode=present;view_id=appBuilding;appld=289E62EAA8C934D647DCB7EB7B531EC6)

## Create a Launchpad

In order to launch our SAC application from BPF we need a launcher that does the parameter mapping from the BPF context to the SAC URL parameter.

Please have a look at the how to paper [How to...Call a Design Studio Application from a SAP BO Planning and Consolidation 10.1 Business Process Flow](#).

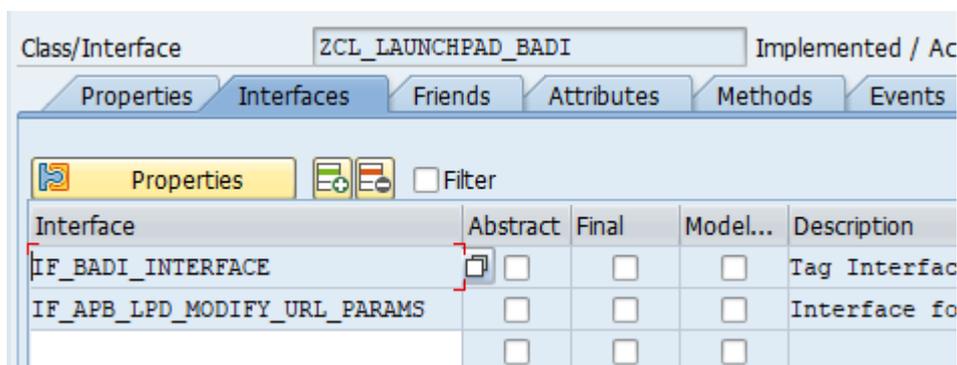
Please follow the instructions there and create a launchpad. You have to deviate from the procedure only slightly:

- Instead of using DS/Design Studio in the name of the launchpad and the role use 'SAC'.
- Instead of pasting the link to the Design Studio Application into the launchpad you have to paste the link to your SAC application (see above).
- Instead of mapping the parameter 'COUNTRY' to the target 'XCOUNTRY' use the target 'p\_COUNTRY'.

## Implement the BAdI for Changing the URL

Now we deviate from our referenced how to paper. As SAC uses slightly different types of URLs than DesignStudio/Lumira we have to implement a BAdI to make sure that the target URL is built up in the correct way.

We first create the necessary class for the implementation. Go to transaction se24 (or transaction se80) and create a new class. In the newly created class add the two interfaces IF\_BADI\_INTERFACE and IF\_APB\_LPD\_MODIFY\_URL\_PARAMS:



Then go to the tab 'Methods' and double click on the method 'IF\_APB\_LPD\_MODIFY\_URL\_PARAMS-MODIFY\_URL\_PARAMETER' (when asked save your changes). Enter the following coding in the implementation:

```

data: ls_parameters type APB_LPD_S_PARAMS,
      l_url type string.

* adapt the role and instance to the names you have used in your launchpad
check ID_Role = 'BPF_SAC' and
      id_instance = 'BPF_SAC'.

read table it_parameters into ls_parameters with key key = 'URL'.

l_url = ls_parameters-value.

loop at it_parameters into ls_parameters.
  check ls_parameters-key(2) = 'p_'.
  concatenate l_url ';' ls_parameters-key '=' ls_parameters-value into l_url.
endloop.

ed_URL_FRAC = l_url.
CD_REPLACE_URL_WITH_URL_FRAC = 'X'.

```

In the check at the beginning of the coding you will have to use the role and instance name of the launchpad you have created above.

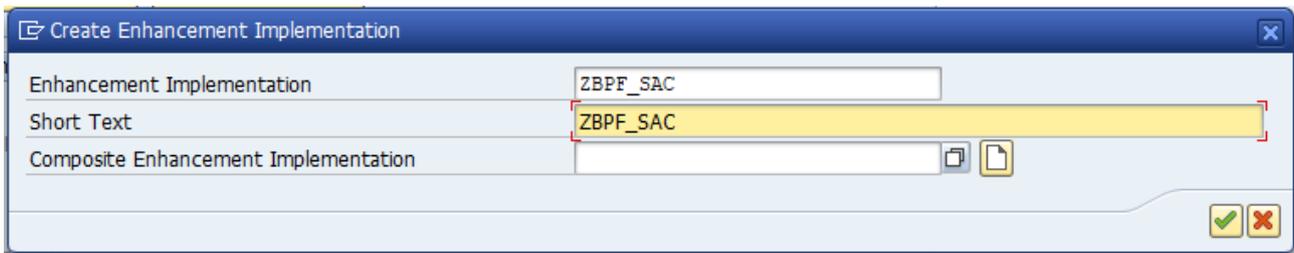
The coding will scan the provided parameter for any parameter that starts with 'p\_' and thus follows the naming convention of the SAC URL parameters. Any such parameter will be added to the URL. As we do not check for the actual names of the variables used this coding should be generic enough for nearly any set up of your parameters.

You should also create (empty) implementations for the other methods in the class. Then save and activate your class.

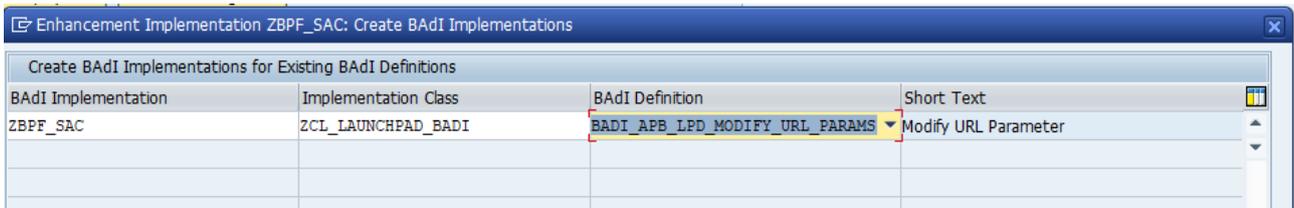
Now go to the transaction se19. Choose 'New BAdI' and enter the name of the enhancement spot (APB\_LAUNCHPAD). Press the button 'Create Impl.'.

The screenshot shows the 'Create Implementation' dialog box. It has two radio buttons: 'New BAdI' (selected) and 'Classic BAdI' (unselected). Under 'New BAdI', there is a text field for 'Enhancement Spot' containing 'APB\_LAUNCHPAD'. Under 'Classic BAdI', there is a text field for 'BAdI Name' which is currently empty. At the bottom left, there is a yellow button labeled 'Create Impl.' with a document icon.

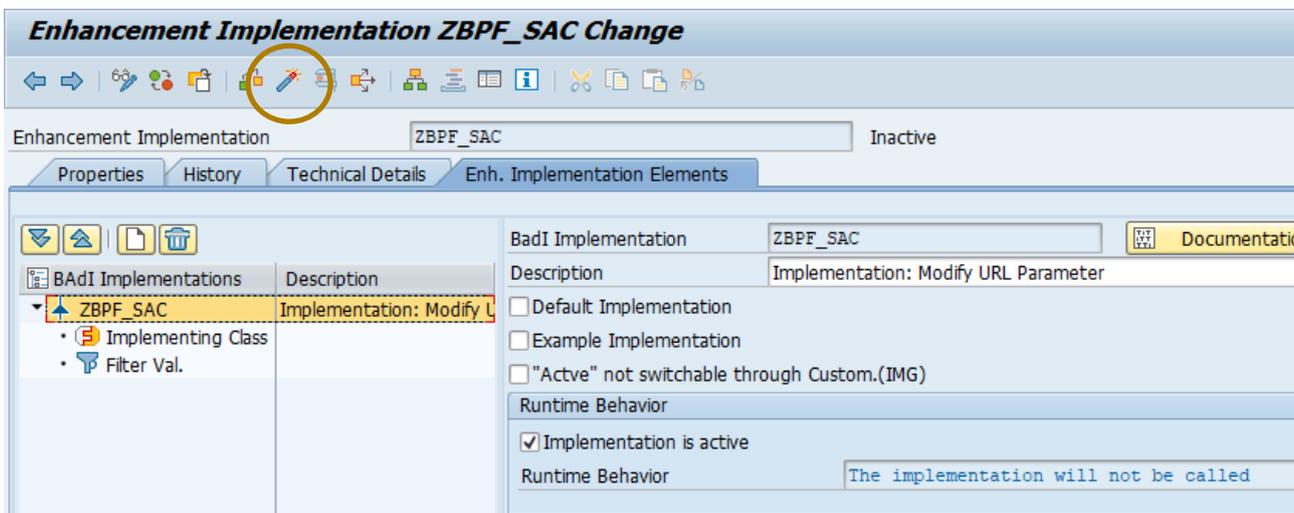
Enter a name for your enhancement implementation and press ok.



Now enter a name for the BAdI implementation, the name of the class you have just created, and select the BAdI definition 'BADI\_APB\_LPD\_MODIFY\_URL\_PARAMS'. Press ok.



Now activate your implementation (and all dependent objects).



The implementation will now be called.

## Build your BPF Template

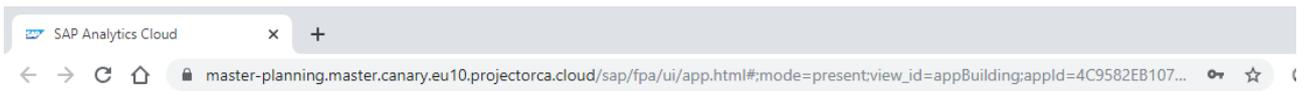
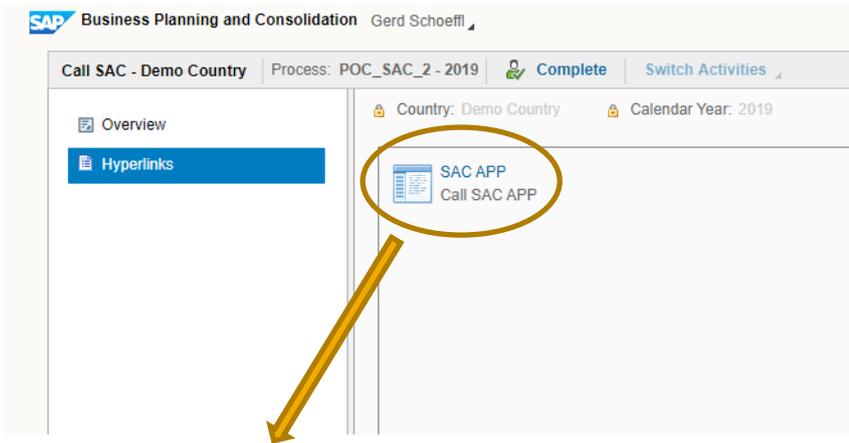
Again, we can follow the steps described in the referenced how to paper. Select the launchpad you have just created and deploy your template.

## Create and Test your Instance

Please follow the corresponding chapter of the referenced how to paper. Create an instance of the BPF and test it.

The called ULR should look like:

[https://master-planning.master.canary.eu10.projectorca.cloud/sap/fpa/ui/app.html#:mode=present;view\\_id=appBuilding;appId=289E62EAA8C934D647DCB7EB7B531EC6;p\\_COUNTRY=AB](https://master-planning.master.canary.eu10.projectorca.cloud/sap/fpa/ui/app.html#:mode=present;view_id=appBuilding;appId=289E62EAA8C934D647DCB7EB7B531EC6;p_COUNTRY=AB)



HBD\_POC  
2 sorts applied | 1

Measures		Actual Revenue 2019	Actual Sales Quantity 2019	Plan Revenue 2020	Plan Sales Quantity 2020
Country	Material Group				
Demo Country	Cola	EUR307,320.00	327,602.000 PC	EUR322,668.00	343,982.000 PC
	Ice Tea	EUR684,534.99	581,560.000 PC	-EUR54,841.90	610,648.000 PC
	Juice	EUR529,399.68	218,518.000 PC	-EUR42,414.97	229,460.000 PC
	Water	EUR317,190.85	327,959.000 PC	-EUR25,411.13	344,351.000 PC
	<b>Totals</b>	<b>EUR1,838,445.52</b>	<b>1,455,639.000 PC</b>	<b>EUR200,000.00</b>	<b>1,528,441.000 PC</b>
<b>Totals</b>		<b>EUR1,838,445.52</b>	<b>1,455,639.000 PC</b>	<b>EUR200,000.00</b>	<b>1,528,441.000 PC</b>

[www.sap.com/contactsap](http://www.sap.com/contactsap)

© 2019 SAP SE or an SAP affiliate company. 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 SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company 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.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies. See [www.sap.com/copyright](http://www.sap.com/copyright) for additional trademark information and notices.

THE BEST RUN

