NetWeaver Technical Blueprint

Standard Bank BADI Framework Quick Reference Guide

Version: 0.01

Date: 08.01.2015

# Source and Destination

|  |  |
| --- | --- |
| **Sender** | |
| From: | EPI-USE Africa (Pty) Ltd |
| Author: | Connie Mulder |
| Capacity: | Consultant |
| Contact Details: | [Connie.mulder@standardbank.co.za](mailto:Connie.mulder@standardbank.co.za) |

# Document History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Transcriber** | **Date** | **Revision** |
| 0.01 | Connie Mulder | 8 January 2015 | Document Creation |

# References

# Reviews

|  |  |
| --- | --- |
| **EPI-USE Review** | |
| **Reviewer name** | **Date** |
| Mvuyisi Scheepers | 12.01.2015 |

# Copyright

Copyright © 2006 SAP AG. All rights reserved.

No part of this documentation may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG.

SAP reserves the right to change the information contained in this document without prior notice.

# Document Approval

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Role** | **Signature** | **Date Approved** |
|  |  |  |  |

# Legal Notice

The information in this document is confidential and may not be disclosed outside of Standard Bank. The document and its contents may not be duplicated, used, or disclosed in whole or in part. Provided that a written contract exists between EPI-USE Africa (Pty) Ltd and Standard Bank, Standard Bank will have the right to duplicate, use, or disclose the information to the extent provided by the contract. EPI-USE Africa (Pty) Ltd retains ownership of this document, unless explicitly stated otherwise in this document or in a written contract.

EPI-USE, the EPI-USE logo, EPI-USE Labs, the EPI-USE Labs logo, Data Sync Manager, Query Manager, Variance Monitor, Object Sync, Client Sync, Data Secure, Tax Analyser, Pay Recon, and Advanced Time Process Manager are either registered trademarks or trademarks of EPI-USE Systems.

Acrobat is a trademark of Adobe Systems Incorporated and may be registered in various jurisdictions. Excel, Windows and Word are trademarks of Microsoft Corporation. The following are trademarks or registered trademarks of SAP AG: ABAP/4, R/2, R/3 and the word SAP. The SAP logo and all other SAP products, services, logos, or brand names that may be included herein are also trademarks or registered trademarks of SAP AG.

All other trademarks mentioned are the property of their respective owners.

Data contained in this document serves information purposes only, unless explicitly mentioned otherwise. National product specifications may vary.

# Table of Contents

[1. Source and Destination 2](#_Toc408815985)

[2. Document History 2](#_Toc408815986)

[3. References 2](#_Toc408815987)

[4. Reviews 2](#_Toc408815988)

[5. Copyright 2](#_Toc408815989)

[6. Document Approval 2](#_Toc408815990)

[7. Legal Notice 3](#_Toc408815991)

[8. Table of Contents 4](#_Toc408815992)

[9. Introduction 5](#_Toc408815993)

[9.1 Aim 5](#_Toc408815994)

[10. Scenario 1 – New business logic 6](#_Toc408815995)

[10.1 Global 6](#_Toc408815996)

[10.2 Local 9](#_Toc408815997)

[10.3 Group Logic 13](#_Toc408815998)

[11. Scenario 2 – change existing logic 20](#_Toc408815999)

[11.1 Global 20](#_Toc408816000)

[11.2 Local 23](#_Toc408816001)

[11.3 Group Logic 26](#_Toc408816002)

[12. ZXPADU migration 27](#_Toc408816003)

# Introduction

The BADI Framework is meant to replace ZXPADU as a repository for Business Logic, this enables business logic to execute whenever Infotypes are accessed, from within the SAP Backend as well as from external systems.

The Quick Guide will deal with the most common scenarios a developer faces when having to implement Business Logic, and explain how the BADI-framework will handle these scenarios.

## Aim

Through use of filters on Infotype and Country, logic can be separated by country and executed specifically for each country. Because each country-specific BADI has its own class – maintenance can be performed on several pieces of logic simultaneously without locking one another, as is the case with ZXPADU.

The BADI framework is also connected to the portal, and the Decoupled Infotype Framework, enabling business logic to execute when accessing infotypes from the portal, as well as other external systems.

Ultimately the goal is to separate Business and Screen logic, and allow independent maintenance of either.

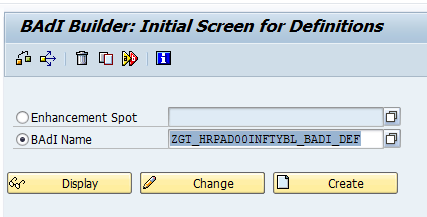
# Scenario 1 – New business logic

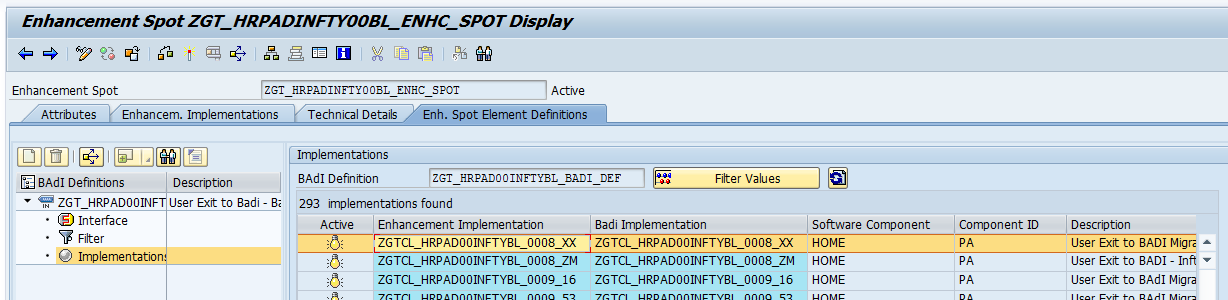
Summary: New business logic on Infotype 0008 has been identified. This logic needs to added to the BADI Framework

## Global

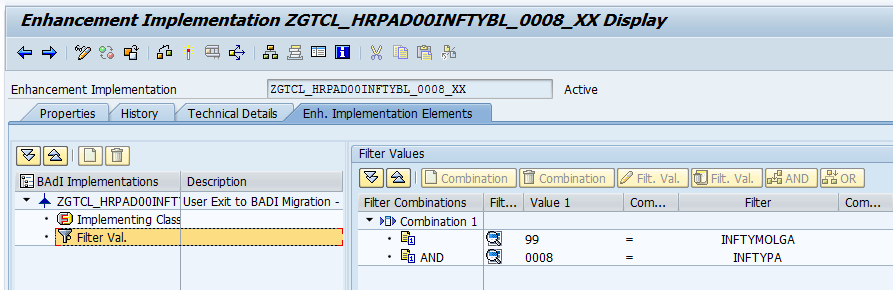
The logic is for all countries, and should execute for all countries.

The steps to implement New Global Logic is as follows

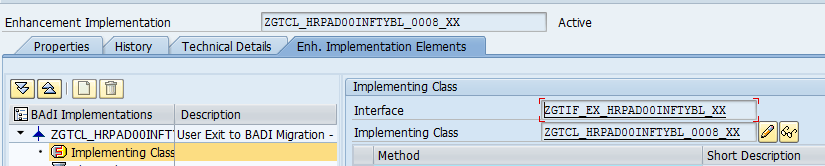
* Go to SE18 and enter BADI Definition ZGT\_HRPAD00INFTYBL\_BADI\_DEF  
  
* In Change mode create a new BADI Implementation for Global Logic on this Infotype with the following name:  
  ZGTCL\_HRPAD00INFTYBL\_0008\_XX
* Naming Convention Explained:
  + ZGTCL - Because this is a custom implementation in the GT space, and it will be a class
  + HRPAD00INFTYBL – This is the SAP-standard BADI that we have used as template
  + 0008 – The infotype you’re creating the Global Logic for
  + XX – The Country Code (MOLGA) you’re creating the logic for – because it is global, it will be XX



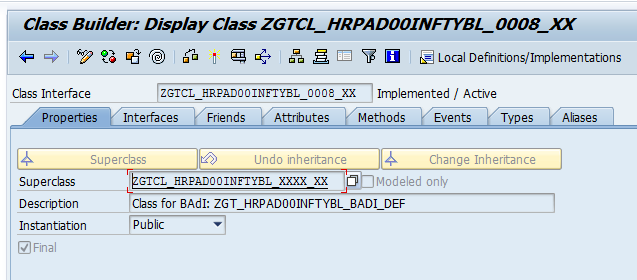
* Set the filter values of this BADI Implementation to the following
  + INFTYMOLGA = 99 (All Countries)
  + INFTYPA = 0008 (Infotype Number)



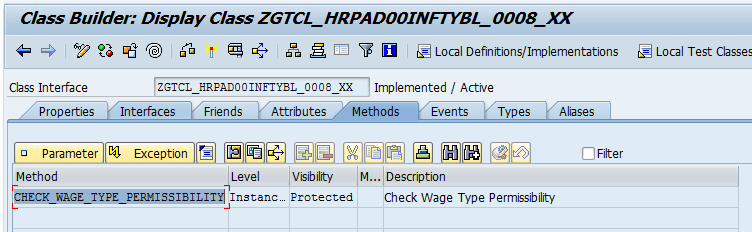
* Create the Implementing Class for this BADI-Implementation with the exact same name as the Implementation, making sure to use Interface ZGTIF\_EX\_HRPAD00INFTYBL\_XX.



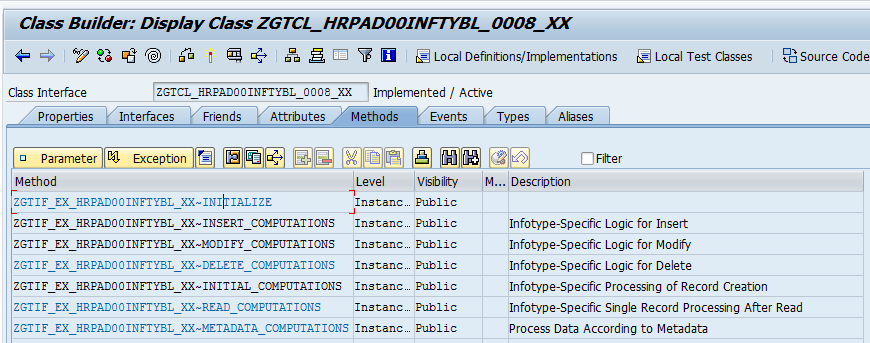
* In the class – set the SuperClass as ZGTCL\_HRPAD00INFTYBL\_XXXX\_XX. This is the Fallback class of the BADI, and has empty implementations for all Interface Methods. This will prevent Short-Dumps.

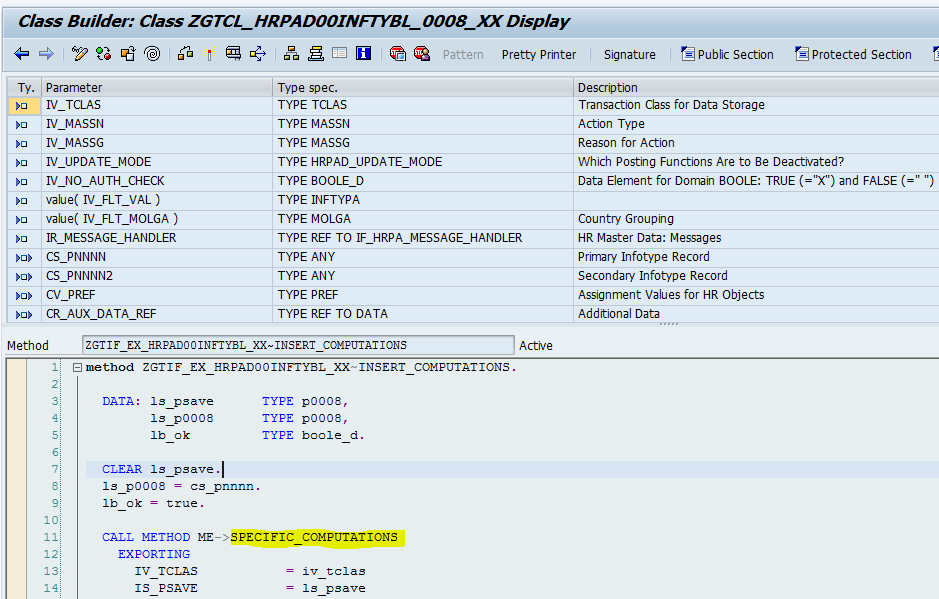


* Implement a method that executes your logic as a protected instance method of the class. You have access to several default values as inherited from the SuperClass.

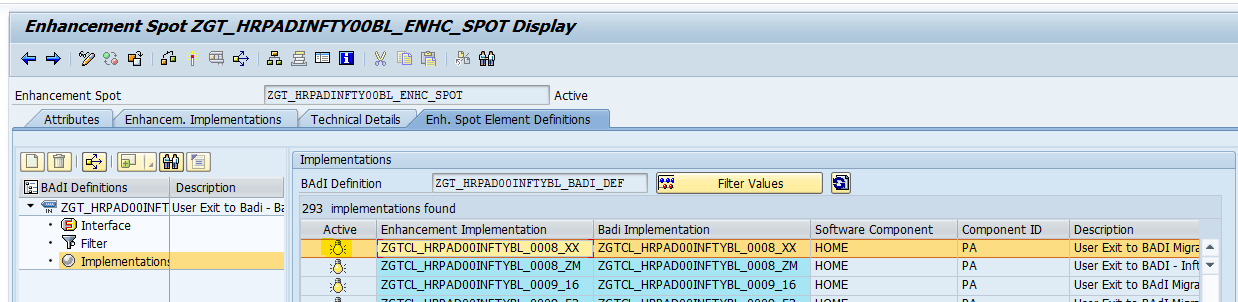


* Include a call to this method in the relevant “Computations” method e.g. Insert Computations for logic that should execute on new record creation. If needed – redefine these “Computations” methods to call your new instance method. Take note – there is a custom method “Specific Computations” that is usually used as a grouping method for logic that should execute on both Insert and Modify.





* Activate the Class and then activate the BADI Implementation.

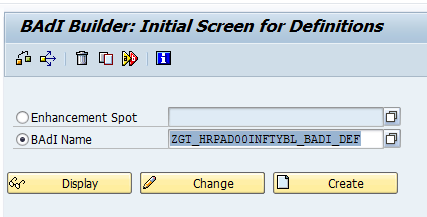


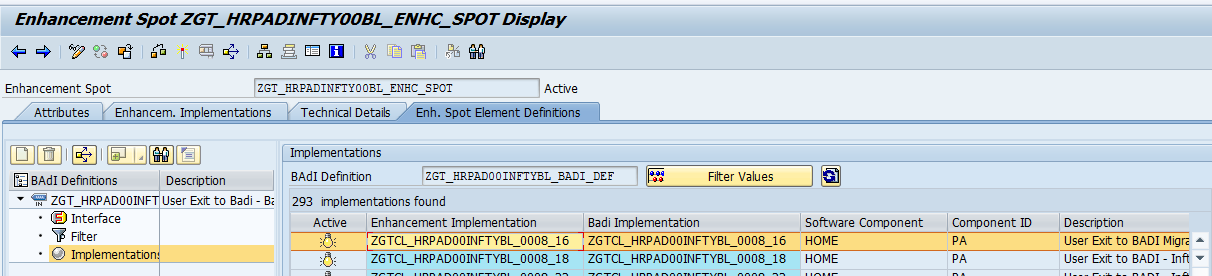
* Your logic will now be called when Infotype 0008 is processed for all countries. This will happen before Country Specific logic gets executed.

## Local

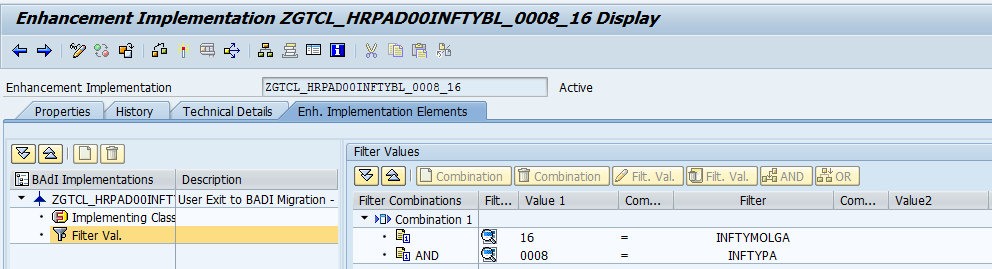
The logic is for only one country, in this case South Africa. This logic should only execute for South Africa.

The steps to implement New Local Logic is as follows

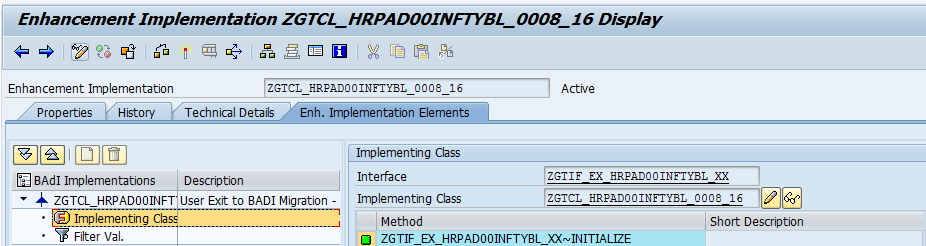
* Go to SE18 and enter BADI Definition ZGT\_HRPAD00INFTYBL\_BADI\_DEF  
  
* In Change mode create a new BADI Implementation for Local Logic on this Infotype with the following name:  
  ZGTCL\_HRPAD00INFTYBL\_0008\_16
* Naming Convention Explained:
  + ZGTCL - Because this is a custom implementation in the GT space, and it will be a class
  + HRPAD00INFTYBL – This is the SAP-standard BADI that we have used as template
  + 0008 – The infotype you’re creating the Local Logic for
  + 16 – The Country Code (MOLGA) you’re creating the logic for – because it is South Africa, it will be 16



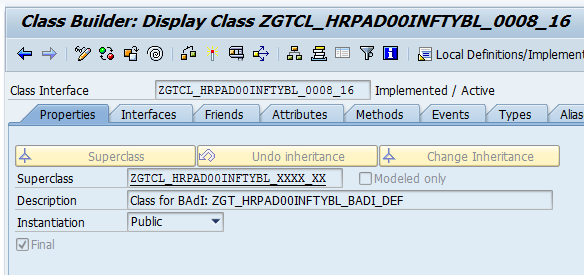
* Set the filter values of this BADI Implementation to the following
  + INFTYMOLGA = 16 (South Africa)
  + INFTYPA = 0008 (Infotype Number)



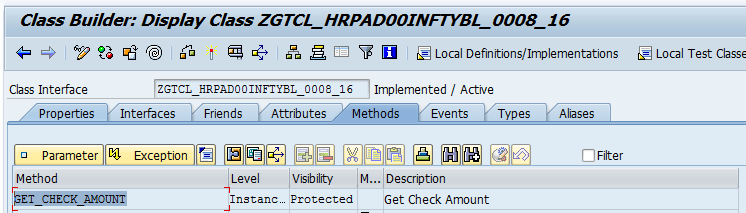
* Create the Implementing Class for this BADI-Implementation with the exact same name as the Implementation, making sure to use Interface ZGTIF\_EX\_HRPAD00INFTYBL\_XX.



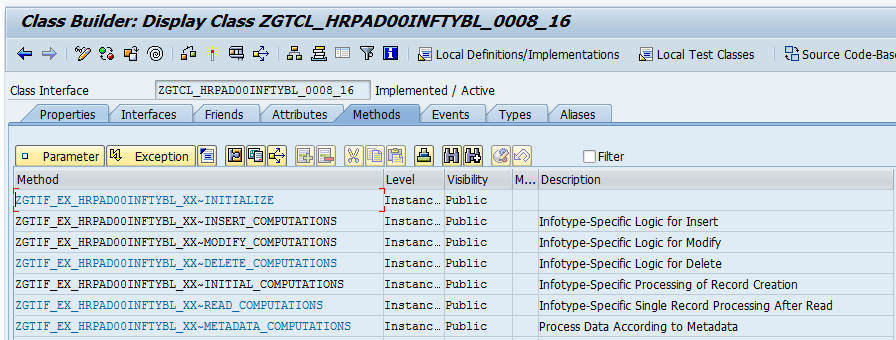
* In the class – set the Superclass as ZGTCL\_HRPAD00INFTYBL\_XXXX\_XX. This is the Fallback class of the BADI, and has empty implementations for all Interface Methods. This will prevent Short-Dumps.

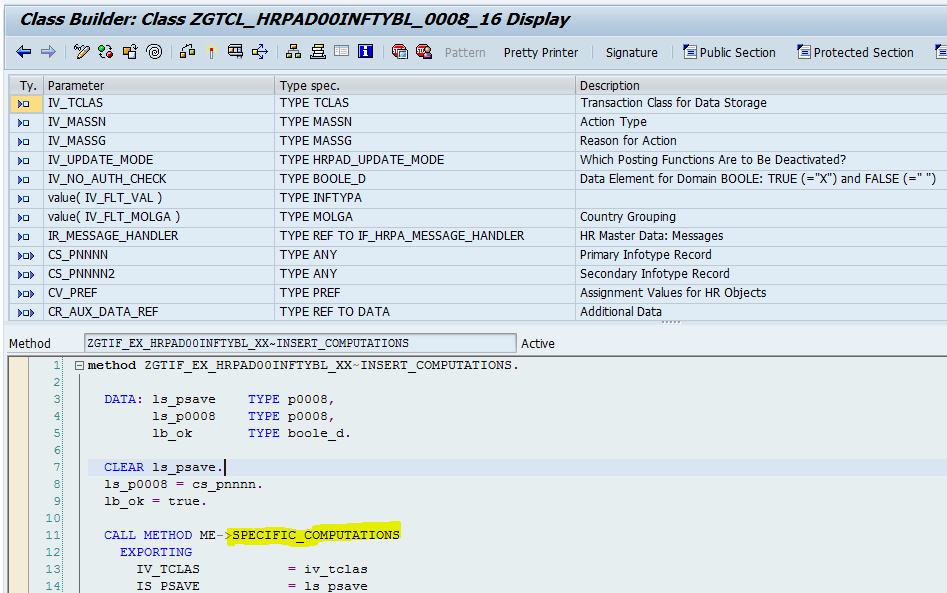


* Implement a method that executes your logic as a protected instance method of the class. You have access to several default values as inherited from the SuperClass.

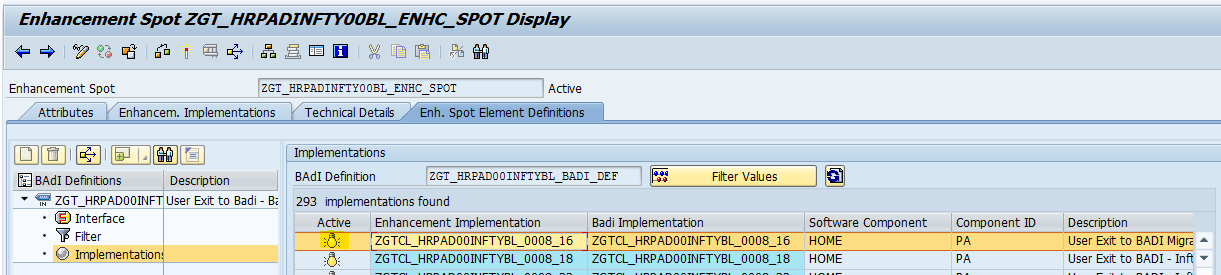


* Include a call to this method in the relevant “Computations” method e.g. Insert Computations for logic that should execute on new record creation. If needed – redefine these “Computations” methods to call your new instance method. Take note – there is a custom method “Specific Computations” that is usually used as a grouping method for logic that should execute on both Insert and Modify.





* Activate the Class and then activate the BADI Implementation.



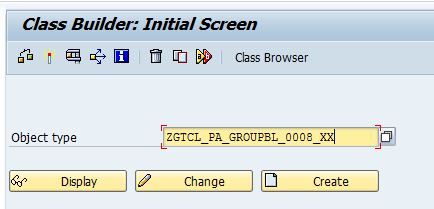
* Your logic will now be called when Infotype 0008 is processed for South Africa. This will happen after Global Logic for this Infotype has been processed.

## Group Logic

The logic is for more than one country, but not all countries, in this case Namibia and Botswana. We want to have this logic execute for both countries, but we don’t want to maintain it in two places.

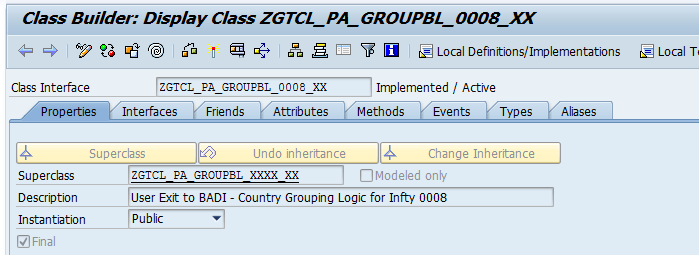
The steps to implement New Group Logic is as follows

* Create a new Group Logic Class in SE24

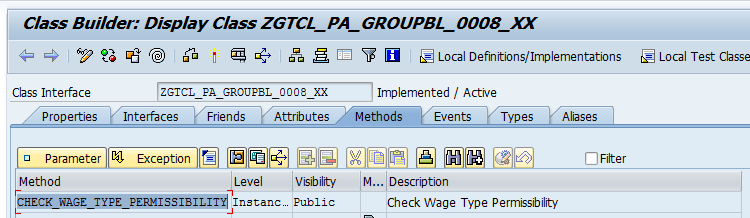


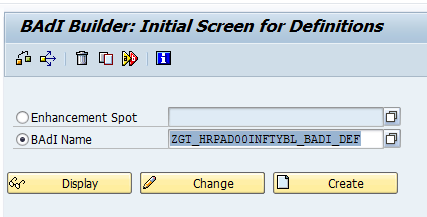
Naming Convention

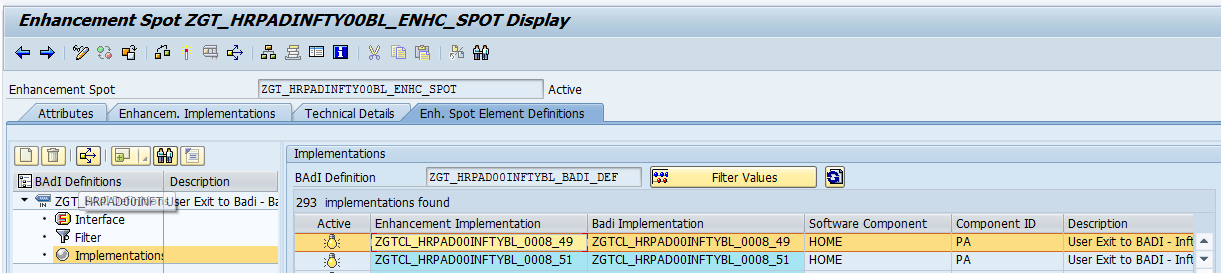
* + ZGTCL - Because this is a custom class in the GT space.
  + PA – Module
  + GROUPBL – This is Group Business Logic
  + 0008 – The infotype you’re creating the Group Logic for
  + XX – The country code you’re creating the Group Logic for, since this is for more than one country – we use the global country code (XX)
* Use ZGTCL\_PA\_GROUPBL\_XXXX\_XX as SuperClass



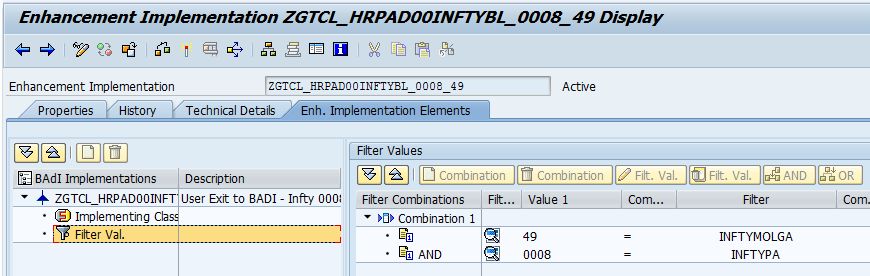
* Implement a method that executes the Group Business Logic as a Public Instance Method



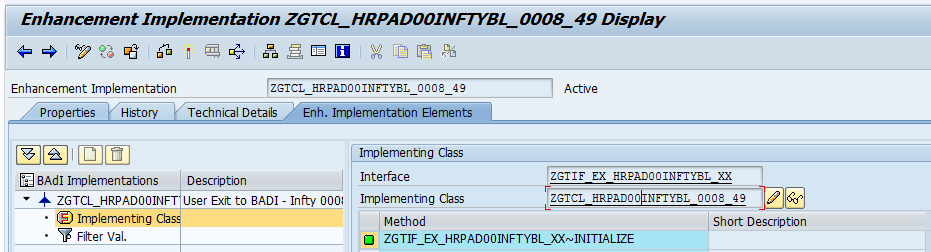
* Save and activate your Group Business Logic class
* Go to SE18 and enter BADI Definition ZGT\_HRPAD00INFTYBL\_BADI\_DEF  
  
* In Change mode create new BADI Implementations for Local Logic on this Infotype with the following name:   
  ZGTCL\_HRPAD00INFTYBL\_0008\_49  
  ZGTCL\_HRPAD00INFTYBL\_0008\_51, continue for as many countries as necessary.
* Naming Convention Explained:
  + ZGTCL - Because this is a custom implementation in the GT space, and it will be a class
  + HRPAD00INFTYBL – This is the SAP-standard BADI that we have used as template
  + 0008 – The infotype you’re creating the Global Logic for
  + 49 – The Country Code (MOLGA) you’re creating the logic for – because it is South Africa, it will be 49 and 50, or as many countries as is needed.



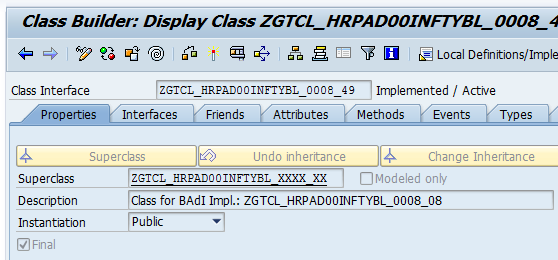
* Set the filter values of these BADI Implementations to the following
  + INFTYMOLGA = 49 or 50 for each implementation as necessary
  + INFTYPA = 0008 (Infotype Number)



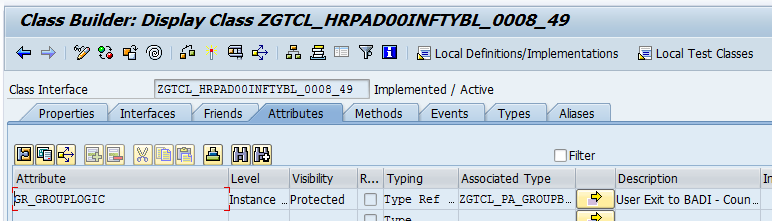
* Create the Implementing Classes for these BADI-Implementations with the exact same name as the Implementations, making sure to use Interface ZGTIF\_EX\_HRPAD00INFTYBL\_XX.



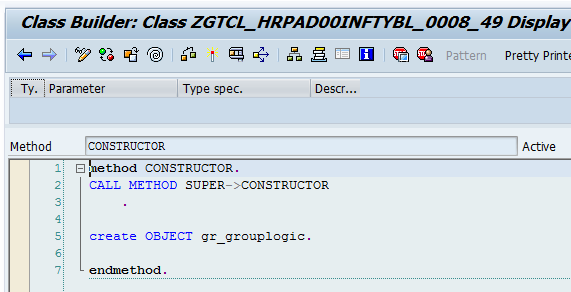
* In the classes – set the SuperClass as ZGTCL\_HRPAD00INFTYBL\_XXXX\_XX. This is the Fallback class of the BADI, and has empty implementations for all Interface Methods. This will prevent Short-Dumps.



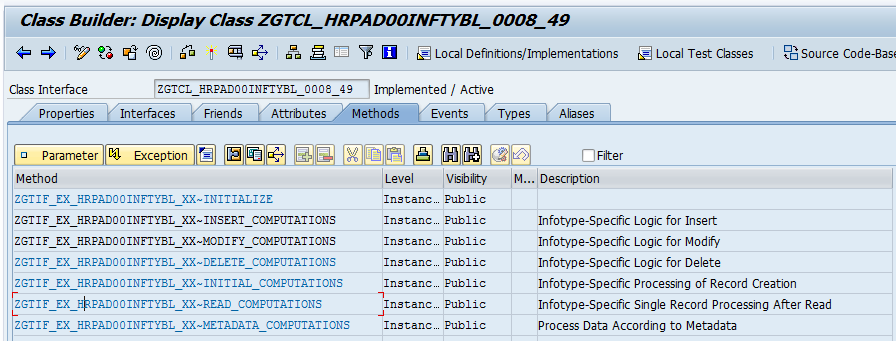
* Add a class attribute that references the Group Logic Class you created earlier.

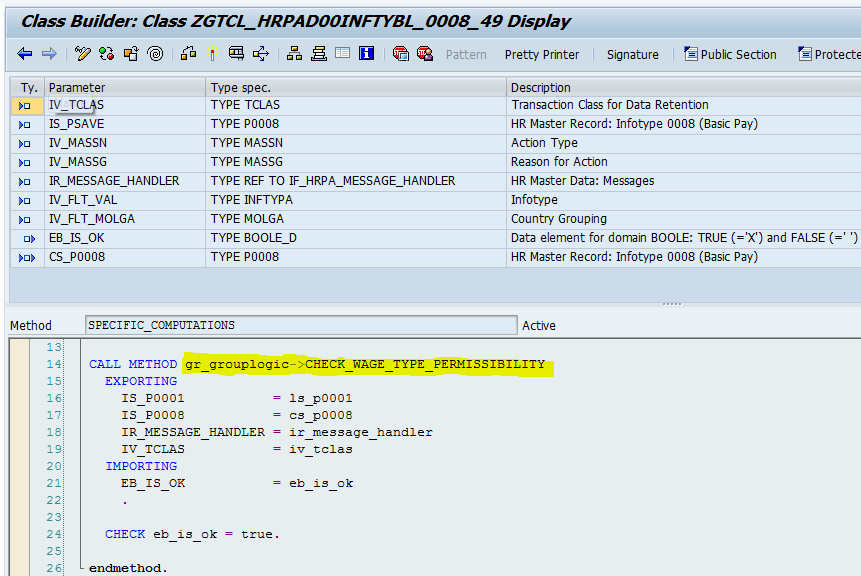


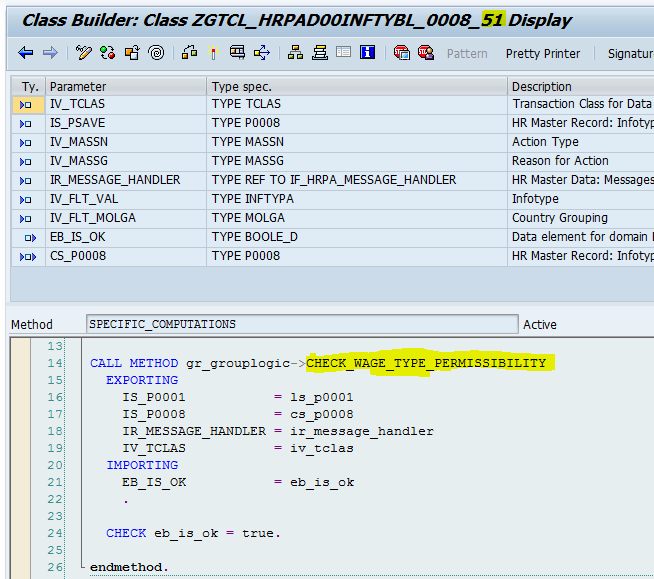
* Redefine the Constructor to instantiate the Group Logic attribute upon Instance Creation



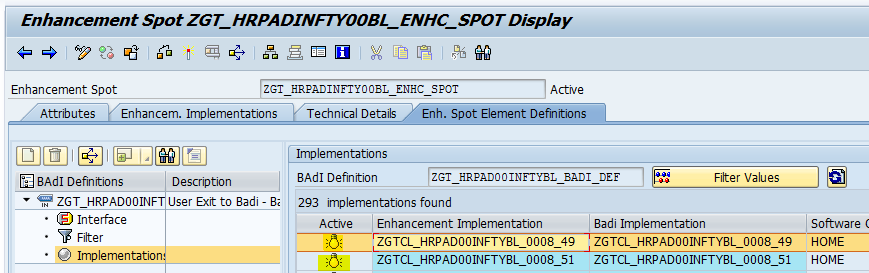
* Include a call to the corresponding method of the group logic class in the relevant “Computations” method e.g. Insert Computations for logic that should execute on new record creation. If needed – redefine these “Computations” methods to call your new Group Logic method. Take note – there is a custom method “Specific Computations” that is usually used as a grouping method for logic that should execute on both Insert and Modify.







* Activate the Class and then activate the BADI Implementations.



* Repeat for each country that needs to call this specific piece of Group Logic
* Your Group Logic will now be called when Infotype 0008 is processed for Namibia and Botswana. This will happen after Global Logic for this Infotype has been processed.

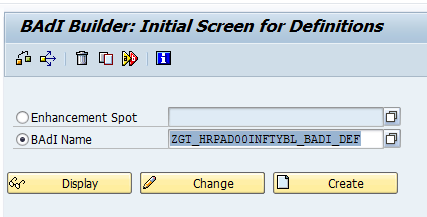
# Scenario 2 – change existing logic

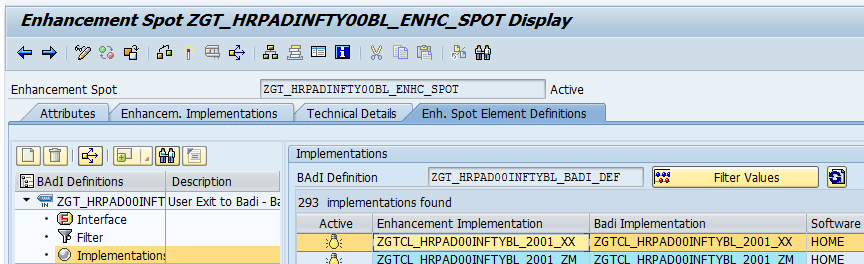
Summary: Business Logic on Infotype 2001 needs to be changed. The logic is currently in the BADI-framework, and needs to be maintained.

## Global

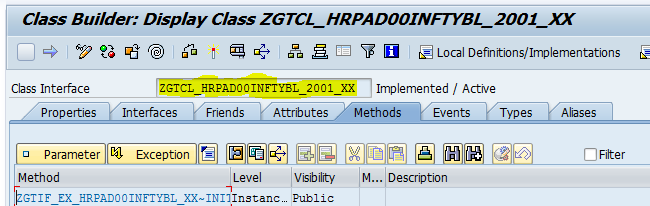
The logic is currently executing for all countries.

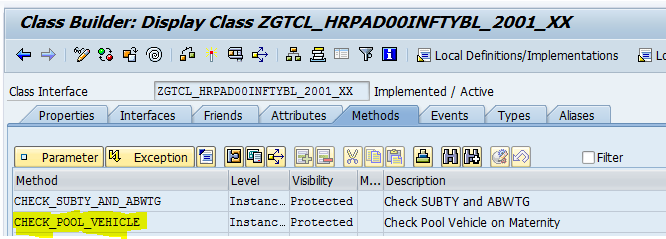
The steps to change Global Logic is as follows

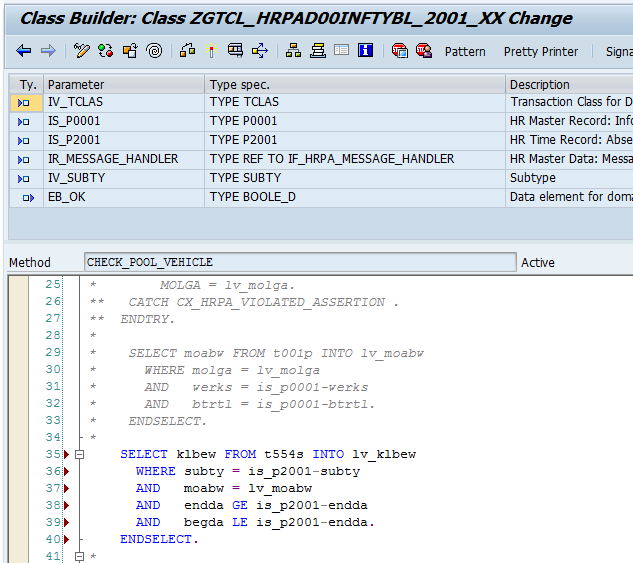
* Go to SE18 and enter BADI Definition ZGT\_HRPAD00INFTYBL\_BADI\_DEF  
  
* In Change mode go to the BADI Implementation for Global Logic on this Infotype with the following name:  
  ZGTCL\_HRPAD00INFTYBL\_2001\_XX



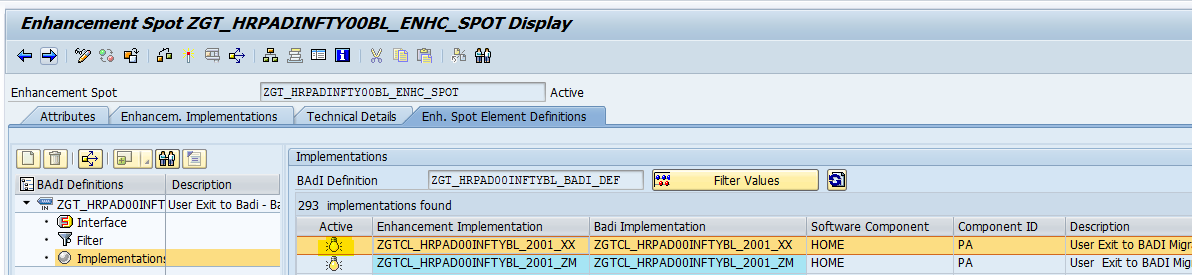
* Go to the Implementing Class for this BADI-Implementation with the exact same name as the Implementation.



* Go to the method that executes your logic at the moment, and change the logic to the desired new logic. In this case, we have to comment out some statements.



* Activate the Class and then activate the BADI Implementation.

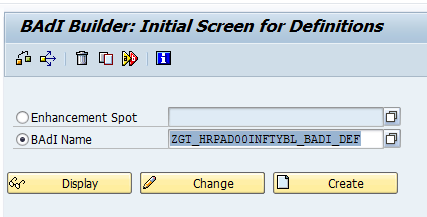


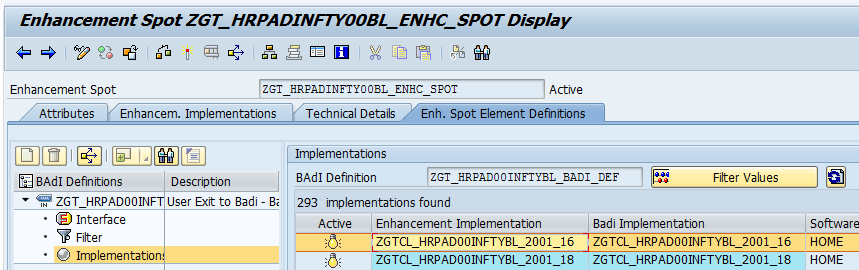
* Your changed logic will now be called when Infotype 2001 is processed for all countries. This will happen before Country Specific logic gets executed.

## Local

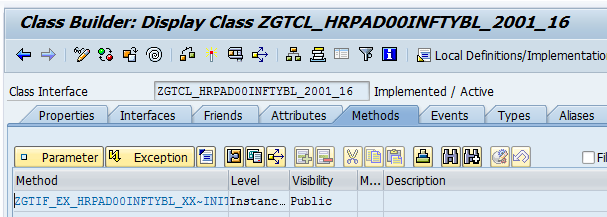
The logic is currently executing for only one country – South Africa

The steps to change Country Specific Logic is as follows:

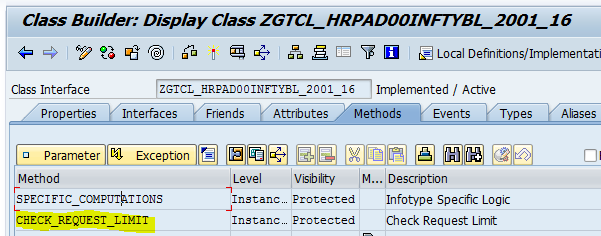
* Go to SE18 and enter BADI Definition ZGT\_HRPAD00INFTYBL\_BADI\_DEF  
  
* In Change mode go to the BADI Implementation for Global Logic on this Infotype with the following name:  
  ZGTCL\_HRPAD00INFTYBL\_2001\_16

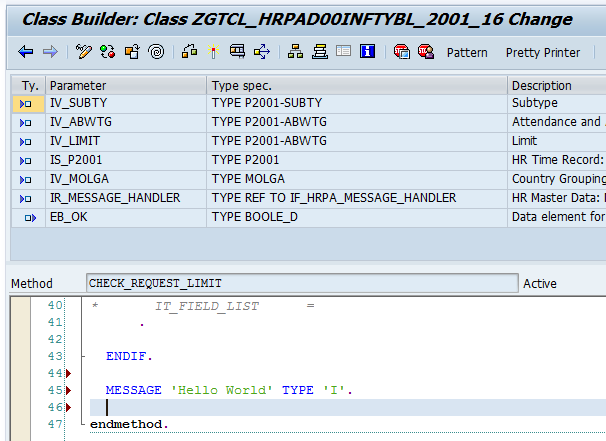


* Go to the Implementing Class for this BADI-Implementation with the exact same name as the Implementation.

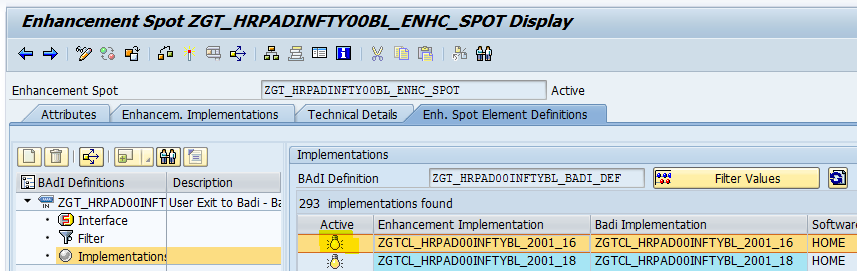


* Go to the method that executes your logic at the moment, and change the logic to the desired new logic. In this case, we have to add an information message.





* Activate the Class and then activate the BADI Implementation.

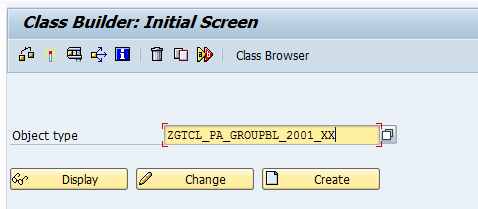


* Your changed logic will now be called when Infotype 2001 is processed for South Africa. This will happen after Global Logic gets executed.

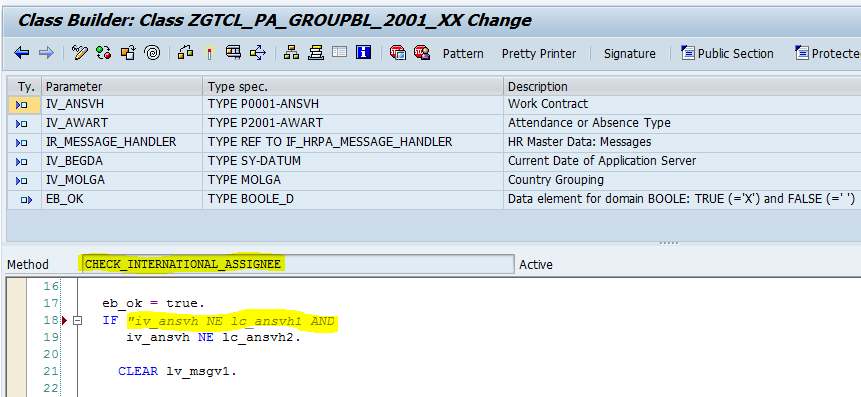
## Group Logic

The logic is currently executing for more than one country, but not all countries. Here this is logic that applies to Namibia, Tanzania and Uganda

* Go to SE24 and change the Group Logic Class



* Go to the method in which the logic is currently executing, and change the logic to the desired outcome. Here we are commenting out some statements.



* Activate the Group Logic Class. Since the method should already be called from each countries’ Local Logic Class, you don’t need to change anything there. If you change the Importing Parameters for the method, then you will have to change the method call in the corresponding Local Logic Classes
* Your changed logic will now be called when Infotype 2001 is processed for Namibia, Tanzania and Uganda. This will happen after Global Logic gets executed.

# ZXPADU migration

Business Logic has been migrated from ZXPADU01 and ZXPADU02 to the BADI-framework.

The goal of the BADI-framework is to separate the Business and Screen logic, as well as create easier maintenance.

For future requests, you would have to identify if you are busy with Business Logic (Rules that apply based on certain criteria being fulfilled, e.g. Not allowing more than 4 months maternity leave) or Screen Logic (Logic that changes the appearance of PA30 based on certain criteria, e.g. hiding certain fields on the infotype screen from certain users.)

Screen Logic still needs to be applied in ZXPADU, or a relevant screen enhancement.

Business Logic needs to be implemented in the BADI-framework in the corresponding BADI-implementation.