HOW TO CREATE A MAI TEXT BADI IMPLEMENTATION

Description:
With Solution Manager ST7.1 SP12 it is possible to use the Monitoring and Alerting Infrastructure (MAI) for displaying the BPOps alerts in a combined alert inbox together with the alerts resulting from the AppOps area (e.g. Technical Monitoring) Similar to the classical BPMon it is also possible to customize the notification (or incident) text, subject, attachments and recipients/processors via a BAdI implementation.
This document describes how to create the BAdI implementation and shows which parameters of the notification (or incident) can be customized.

Implementation
Step 1 – Create a MAI BAdI Implementation:
Start transaction SE18 in the Solution manager system, open Enhancement Spot ALERT_REACTION.
By right-clicking on the BAdI Definition BADI_DYN_INCI_NOTIF_MULTIPLE, a new implementation can be created.
The BAdI implementation needs to be embedded in an enhancement implementation. In the pop-up select an existing or create a new enhancement implementation:

If you create a new enhancement implementation, choose a unique name (here: e.g. ZHK_TEST) and a short text:

The field ‘Composite Enhancement Implementation’ can be left empty.

In the next popup, you need to create the unique BAdI implementation:

Please note: The BAdI implementation and the implementing class are customer objects which need to be created in a customer package. Choose unique names for implementation and class and a descriptive short text.
Open the new BAdI implementation by double-clicking on *Implementations* on the left and selecting the implementation in the appearing list on the main view (ZH_K_TEST_TEXT_BADI in this example).

Open on the left the chosen BAdI implementation by selecting the right-pointing arrow.

When you double-click on *Implementations* you can see the two methods which can be implemented:

- **IF_ALERT_NOTIF_MULTI_CONF~GET_NOTIF_CONF_FOR_ALERT** for changing the email content and recipients and
- **IF_ALERT_NOTIF_MULTI_CONF~GET_INCI_CONF_FOR_ALERT** for changing incident texts and processor.
Step 2 – Implement the Corresponding Method
You need to implement the corresponding method
\( \text{IF\_ALERT\_NOTIF\_MULTI\_CONF} \sim \text{GET\_NOTIF\_CONF\_FOR\_ALERT} \) (changing the email content)
\( \text{IF\_ALERT\_NOTIF\_MULTI\_CONF} \sim \text{GET\_INCI\_CONF\_FOR\_ALERT} \) (changing the incident content)

By double-clicking on the method, an empty method implementation is created.

Insert the favoured coding in the empty method and activate it.

Please see the appendix where the interfaces of both BAdI methods are described in detail. Here you can learn which values and parameters of the original document (email/incident) can be changed or overwritten inside the BAdI implementation.

Step 3 – Add a Filter Value to the MAI BAdI Implementation
Choose node Filter Val. on the left side and create a new filter value by pressing the Create Combination button. Add a value. Activate the Enhancement implementation.

Step 4 – Assign the MAI BAdI Implementation to the Monitoring Object
In a last step, the MAI BAdI implementation needs to be assigned to the selected monitoring object. In the tab Notifications, section Advanced Settings, the filter Mai BAdI implementation which has been created in step 2, can be assigned by using the value help.

Use implementation of BAdI_DYN_INCL_NOTIF_MULTIPLE, with filter-value: HK Test: Safeguarding BPlMon Text BAdI

The custom subject and / or introductory texts could be used without BAdI implementation.

If BAdI implementation was used, however, the subject and / or long text returned by that would take precedence. Along with maintained, would be available inside the BAdI implementation.
Appendix 1 – Interface of IF_ALERT_NOTIF_MULTI_CONF~GET_NOTIF_CONF_FOR_ALERT

Importing parameters:

IO_ALERT_OBJECT

This is the MAI alert object instance containing all information about the corresponding alert group, event and metrics. You can check (or reuse) method cl_bpmon_mai_badi_adapter->map_mai_params_to.bpm on how the alert data can be retrieved from this MAI alert object instance. A good example of how this method is used inside a BAdI implementation can be found in method cl_ags.bpm_nf_subject->if_alert_notif_multi_conf~get_notif_conf_for_alert.

IV_XML

<not used>

IT_STD_NOTIF_CONFIG

<obsolete>

IV_CONFIG_SUBJECT

Subject which is maintained on template or monitor level

IV_INTRO_LONGTEXT

Introductory long text which is maintained on template or monitor level

IV_FILTER_VALUE

BAdI filter value

IV_DEFAULT_SUBJECT

Subject which is built by default at runtime and which will be used if not customized by using the BAdI implementation

IV_DEFAULT_LONG_TEXT

Long text which is built by default at runtime and which will be used if not customized by using the BAdI implementation

Changing parameters:

EP_EMAIL_TEXT_TYPE

Raw text or html

EP_EMAIL_TEXT

Customized email text

EP_SMS_TEXT

Customized SMS text, empty but can be filled for text messages on cell phones

EP_EMAIL_SUBJECT

Customized email subject

ET_ATTACHMENTS

Customized email attachments

EV_SENDER
Customized email sender

EP_APPEND_ATTACHMENT
True = add customized attachments to existing attachments, false = replace existing attachments by customized attachments

ET_EMAIL_IDS
Customized table of email recipients

ET_SMS_NUMBERS
Customized table of SMS receiving phone numbers

EP_OVERWRITE_EMAIL_SMS
True = overwrite configured recipients with customized email IDs from ET_EMAIL_IDS and SMS numbers from ET_SMS_NUMBERS, false = append the customized lists to the configured recipients.

EP_SUPRESS_SMS
True = don’t send SMS, false = follow settings in monitor setup

EP_SUPRESS_EMAIL
True = don’t send email, false = follow settings in monitor setup
Appendix 2 – Interface of IF_ALERT_NOTIF_MULTI_CONF~GET_INCI_CONF_FOR_ALERT

Importing parameters:

IO_ALERT_OBJECT
This is the MAI alert object instance containing all information about the corresponding alert group, event and metrics. You can check (or reuse) method cl_bpmon_mai_badi_adapter->map_mai_params_to.bpm on how the alert data can be retrieved from this MAI alert object instance. A good example of how this method is used inside a BAdI implementation can be found in method cl_ags_bpm_nf_subject->if_alert_notif_multi_conf~get_notif_conf_for_alert.

IV_XML
<not used>

IV_CONFIG_SUBJECT
Subject which is maintained on template or monitor level

IV_INTRO_LONGTEXT
Introductory long text which is maintained on template or monitor level

IV_FILTER_VALUE
BAdI filter value

IV_CONFIG_SUBJECT
Subject which is maintained on template or monitor level

IV_DEFAULT_SHORT_TEXT
Incident short text which is built by default at runtime and which will be used if not customized by using the BAdI implementation

IV_DEFAULT_LONG_TEXT
Incident long text which is built by default at runtime and which will be used if not customized by using the BAdI implementation

Changing parameters:

E_INCIDENT_TEXT
Customized incident long text

E_INCIDENT_SUBJECT
Customized incident subject

E_INCIDENT_COMPONENT
Customized incident component

ET_ATTACHMENT
Customized incident attachments

E_CRM_TRANSACTION_TYPE
Customized CRM transaction type of incident

E_CATEGORYID, E_ASPECTID
Define the customized four-level category of incident

E_PROCESSOR
Customized incident processor as business partner