

CRM Worklist- Secondary Methods Doesn't Work: A Workaround



Applies to:

CRM 6.0 and CRM 7.0. For more information, visit the [Customer Relationship Management homepage](#).

Summary

When you define a secondary method, within a step of workflow definition, its expected that the code that is mentioned under 'Secondary Methods before' gets called synchronously before the corresponding Work-item gets executed, similarly the code that is mentioned under 'Secondary Methods After' gets called after the corresponding work-item gets executed and the code mentioned under the 'Secondary method (amodal Call)' gets called during the execution of corresponding work-item asynchronously. These custom pieces of coding don't get called if the work-items were processed from the CRM work-list anymore. However these pieces of code would still be called if SAP Business Workplace were used to process work-items. This documents aims at ensuring that concept of secondary methods would still be invoked from the CRM work-list. However it must be noted that the measures suggested here would just be considered as workaround instead of concrete solution.

Author: Anand Rangarajan

Company: SAP Global Delivery, Bangalore

Created on: 18th December 2010

Author Bio



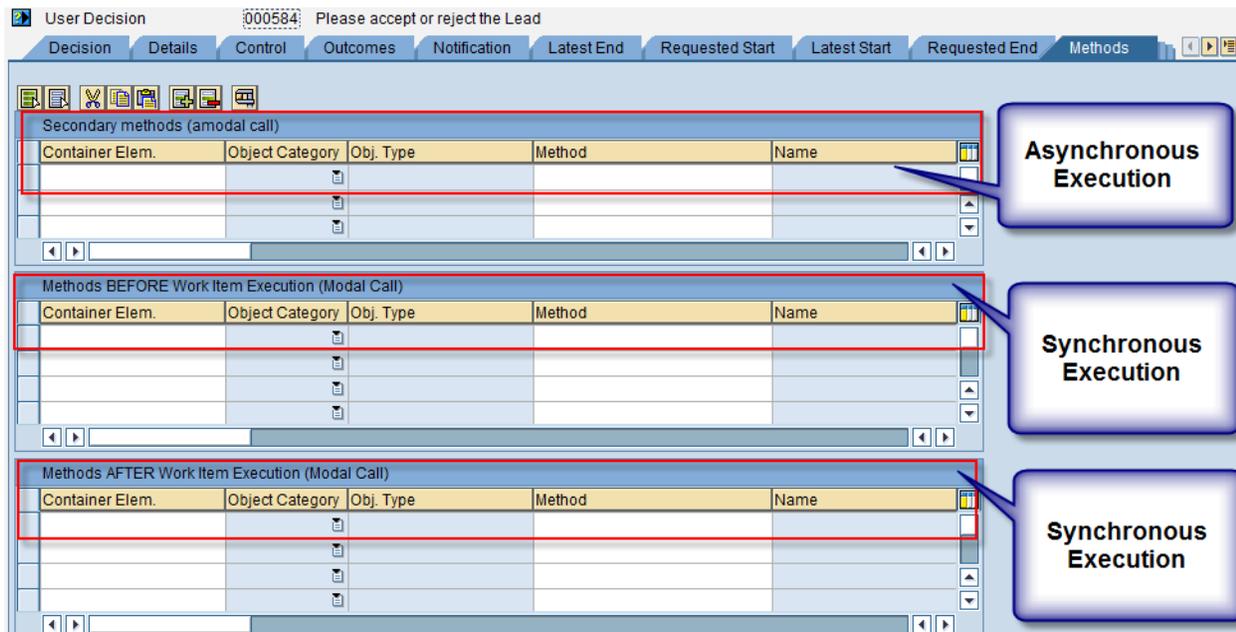
Hi, Let me introduce myself – I am Anand and I am working with SAP Global Delivery, Bangalore. I have been a part of SAP CRM consulting for more than four years now and it has been an amazing journey so far. I have had the privilege of working on multiple SAP products including SAP Solution Manager, SAP CRM 5.0 and SAP CRM 7.0. I started my SAP journey with an ABAP certification in 2006. I have been working on WRICEF components of SAP (Workflows, Reports, Interfaces, Classes, Enhancements, and Forms) for quite some time now. Post my stint with WRICEF, I moved on to working with SAP Solution Manager, and have been responsible for configuring scenarios like Support Desk and Change Request Management. Recently, i was involved in SAP CRM Technical work on SAP CRM 5.0 and CRM 7.0. I have also had the privilege to understand the new SAP CRM 7.0 architecture and thus am aware of SAP CRM WebUI Enhancement framework. I am a certified SAP CRM consultant, certified on SAP CRM 6.0 platform with a good amount of know-how on SAP CRM Marketing. I have extensively worked on SAP CRM Worklist (WebUI) which is similar to SAP Business Workplace in GUI, I have also worked on integrating Social Media(Twitter, Facebook etc..) into SAP CRM 7.0.

Table of Contents

Summary.....	3
Detailed Description	3
Related Content.....	6
Copyright.....	7

Summary

When defining a step within the workflow definition, it's possible for the workflow developer to include secondary methods within this step. This inclusion is done in the below mentioned tab:



When you execute the work-item from the CRM UI Work-list, then the secondary methods are ignored as the architecture and the User interface changed from GUI (SAP Business Workplace) to CRM WEBUI (Work-list). The idea now is to enable the execution of such custom piece of code even from the UI Work-list.

Detailed Description

The approach to solve this problem differs in type of the step we are dealing with. Ideally for background steps, the concept of secondary method should still hold good as for background steps. However for Dialog steps, a different approach needs to be examined.

When you execute a Dialog step from the CRM work-list, depending on the type of dialog work-item , whether its Decision Work-item or whether its Executable work-item, the component/view which needs to be enhanced is different.

- For Executable Dialog Work-item, the idea lies in enhancing Component GSWIDET/DetVS and also enhance Event handler EH_ONBTN_EXECUTE_CLICK.

Here we must follow the below mentioned steps:

Reference: CL_GSWIDET_DETVS_IMPL-> EH_ONBTN_EXECUTE_CLICK

- 1) Read the Task attributes of the current work-item.

Data Declaration:

* Structures, Tables

DATA: ls_wi_attributes TYPE crmt_ct_inbox_wf_root_attrib,

* References

DATA: lrf_current_wi TYPE REF TO cl_crm_bo1_entity,

lrf_current_wi ?= me->typed_context->wfidetail->collection_wrapper->get_current().

CHECK lrf_current_wi IS BOUND.

* Get Workitem Attributes for Navigation

lrf_current_wi->get_properties(IMPORTING es_attributes = ls_wi_attributes).

Variable ls_wi_attributes-WI_RH_TASK will have the task-id.

- 2) Depending on the Task-id, you can then perform the custom coding before and after Code Block mentioned below. Reference

```
-----here----- <Secondary Methods before Custom Code >-----
-here-----
< Invoke your BOR method here >
'op_wi_execute( iv_data_collection = lrf_nav_descrpt_coll )'.
-----here----- <Secondary Methods after Custom Code >-----
here-----
< Invoke your BOR method here >
```

- For Decision Dialog Work-item, the idea lies in enhancing Component GSWIDET/btnDec and also enhance Event handler EH_ONBTN_DECISION.

Here we must follow the below mentioned steps:

Reference: CL_GSWIDET_BTNDEC_IMPL -> EH_ONBTN_DECISION

- 1) Read the Task attributes of the current work-item.

```
Data Declaration:
* Structures, Tables
DATA: ls_wi_attributes          TYPE crmt_ct_inbox_wf_root_attrib,
* References
DATA: lref_wfi                 TYPE REF TO cl_crm_bol_entity,
lref_wfi ?= me->typed_context->wfidet->collection_wrapper->get_current( ).
```

CHECK lref_wfi IS BOUND.

```
* Get Workitem Attributes for checking the task id
lref_wfi->get_properties( IMPORTING es_attributes = ls_wi_attributes ).
Variable ls_wi_attributes-WI_RH_TASK will have the task-id.
```

- 2) Depending on the Task-id, you can then perform the custom coding before & after Code Block mentioned below.

```
-----here----- <Secondary Methods before Custom Code >-----
-here-----
< Invoke your BOR method here >
TRY.
*   Execute Decision
    lref_wfi->execute(
        iv_method_name = if_crm_wf_gc=>c_wi_meth_decision
        it_param       = lt_method_param ).
CATCH cx_crm_genil_model_error.
*   Shouldn't happen...
    RETURN.
CATCH cx_crm_bol_meth_exec_failed.
*   Coludn't execute Workitem
    RETURN.
ENDTRY.
-----here----- <Secondary Methods after Custom Code >-----
here-----
< Invoke your BOR method here >
```

Please note that this is still a workaround and not the concrete solution in itself. This would work well even if you don't maintain the secondary method under the concerned step as shown below:

The screenshot shows the SAP CRM Worklist interface for a campaign. The main window displays three tables related to secondary methods:

- Secondary methods (amodal call)**: A table with columns: Container Elem., Object Category, Obj. Type, Method, Name.
- Methods BEFORE Work Item Execution (Modal Call)**: A table with columns: Container Elem., Object Category, Obj. Type, Method, Name.
- Methods AFTER Work Item Execution (Modal Call)**: A table with columns: Container Elem., Object Category, Obj. Type, Method, Name.

All three tables are currently empty. A callout box with a blue border and white background points to the 'Methods BEFORE Work Item Execution' table, containing the text: **Nothing maintained here at step Level**.

This needs to be done on the case by case basis and is not a generic solution to the problem.

However it goes to show that by making some UI enhancements, it's still possible to realize secondary methods (before and after). Secondary methods (amodal call), I guess couldn't be practically realized from the CRM work-list.

The whole idea can be utilized for eg if you want to may be give a UI pop-up which is only possible via UI enhancements and not possible with the BOR object method coding. However it must also be noted the custom piece of code should not call up any GUI screen which might result into system dumps.

Possibility of a generic solution:

I guess we can also have a generic solution to this problem provided by can utilize API like 'SWD_GRAPH_IN_SEQUENCE_CONVERT' (for getting all the steps within the workflow templates) and then using another API 'SWD_GET_STEP_PROPERTIES' (for reading the Secondary BOR methods mentioned in the step within an workflow definition although I haven't myself tried this yet.

Related Content

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

Copyright

© Copyright 2010 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.