SAP Workflow: A Comprehensive Guide to Troubleshooting and Debugging

by Thomas Nittmann, Dataxstream, LLC

Table of Contents

Introduction

This article focuses on practical techniques to assist in the debugging and resolving of workflow issues during the development and production support phases of SAP® implementations. The SAP workflow application incorporates the use of several components: a graphical editing tool (Workflow Builder), some object-oriented concepts (using the Business Object Repository), ABAP, and a hook/trigger mechanism into the SAP application modules (FI, CO, MM, etc). As business requirements are mapped into a workflow process definition and prototyping starts, debugging becomes an essential tool to expedite the implementation of workflow processes.

In this white paper, we will concentrate on the use of debugging techniques to resolve real-world workflow implementation and production support issues. Before we begin, let’s list some preliminary assumptions and guidelines:

- The workflow engine has already been enabled.

- We are working with SAP Release 4.6C. There are significant changes and enhancement in release 6.1/6.2 SAP workflow technology. SAP Workflow is part of the Basis system, and future enhancements are incorporated into the Web Application Server (WAS). The next version of R/3 functional applications is called R/3 4.7, and is based on the 6.x version of the WAS.

- The user should be familiar with basic workflow configuration and development, e.g., that workflow type ‘TS’ is a single step task and type ‘WS’ is a multi-step task.

Since workflow has many different elements, starting the debugger is not as straight forward as with a stand-alone ABAP program. To illustrate the best way to step in and resolve issues by debugging, we will start by looking at a copy of the standard SAP workflow template for the overall release of MM purchase requisitions.

The debugging principles applied may be used with any workflow templates (customer defined or standard). A custom workflow template (WS) is being used to illustrate the use of debugging concepts on background tasks. A new custom single step task (TS) has been created for notifying the creator of the purchase requisition release, and replaces the **Dialog Foreground Task** that requests the creator to manually confirm a work item.

Primary Components

Each workflow consists of several components that are best described by answering the following questions:

- When should the workflow start? The answer to this question assists in determining the **event triggering** mechanism that starts the workflow.

- What steps need to occur, i.e., what **tasks** (transactions and/or background items need to be performed) need to occur?

- Who performs the tasks, i.e., the **agent resolution**?

As we proceed, we will review problem solving and debugging techniques used in each of these components.

Base Line Configuration

First, let’s break down the CUSTOM release workflow into its primary multi-step and single-step tasks.

- Custom Workflow Template ***WS99500034*** (zwf_req_rel) is the multi-step task for the overall purchase requisition and is a copy of the standard template WS20000077. It contains several single-step (TS) tasks.

- Single-step task ***TS20000159*** (mm_req_rel_c) is a **dialog foreground** task for the overall release of the workflow. Method BUS2105-SINGLERELEASE.

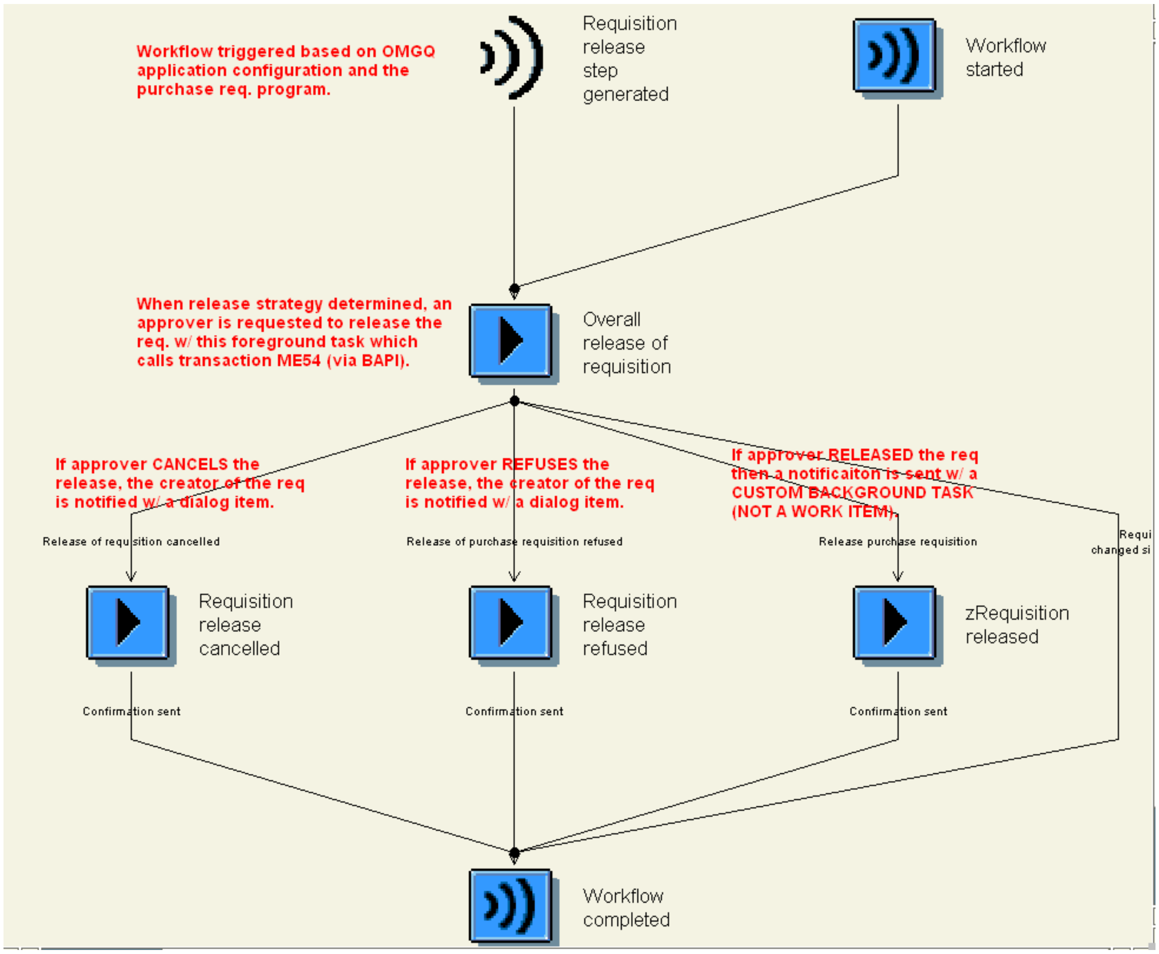
- Single-step task ***TS20000163*** (mm_req_res_c) is a dialog foreground task that requests the creator of the requisition to confirm that the Requisition Release was cancelled. Method BUS2105- INFORELEASERESET.

- Single-step task ***TS20000161*** (mm_req_rej_c) is a dialog foreground task that requests the creator of the requisition to confirm that the Requisition Release was refused. Method BUS2105-INFORELEASEREJECTED.

- Single-step task ***TS99500020*** (zmm_req_ok) is a **custom background** task that replaces the standard task TS20000162 (Requisition Released). The purpose of having a background task is to eliminate the additional work that would be required by confirming a **foreground dialog** work item, i.e., we only want to send a **notification** and not a **work item** to the creator of the requisition. We enable a notification with the custom background task and method ZBUS2105-ZNOTIFYREQRELEASE.

Next, review the requisition release Workflow Builder Process Flow:

> *Note: to display the delivered SAP Workflow, use transaction PFTC > enter ‘Workflow Template’ for Task Type > enter ‘20000077’ for task > click on display > click on workflow builder.*


Requisition release step generated
Workflow started
Overall release of requisition
Release of requisition cancelled
Release of purchase requisition refused
Release purchase requisition
Requi changed si
Requisition release cancelled
Requisition release refused
zRequisition released
Confirmation sent
Confirmation sent
Confirmation sent
Workflow completed

Workflow triggered based on OMGQ application configuration and the purchase req. program.

When release strategy determined, an approver is requested to release the req. w/ this foreground task which calls transaction ME54 (via BAPI).

If approver CANCELS the release, the creator of the req is notified w/ a dialog item.

If approver REFUSES the release, the creator of the req is notified w/ a dialog item.

If approver RELEASED the req then a notificaiton is sent w/ a CUSTOM BACKGROUND TASK (NOT A WORK ITEM).

Figure 1: Workflow Builder Process Flow
(Don't worry if you can't read the smallest print on this screen shot - it's hard to read the smallest print on the process flow even inside SAP. What's important is the overall process flow).

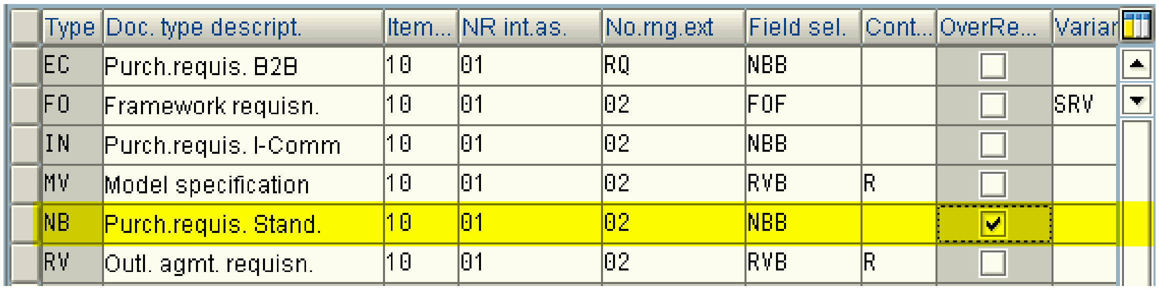
The Requisition Overall Release Workflow is pre-delivered by SAP, and even though we created a custom version of the workflow, there is some baseline configuration required to activate it. Many (but not all) pre-defined workflows require some sort of IMG configuration. Let's review the application configuration:

> *Note: to find the on-line R/3 documentation for workflow, navigate to: Basis Components > Business Management (BC-BMT) > SAP Business Workflow (BC-BMT-WFM) > Reference Documentation > BC – Workflow Scenarios in Applications (BC-BMT-WFM) > MM – Materials Management: Workflow Scenarios*

Use transaction OMEB to activate the Overall Release for the Purchase Requisition document type.

> *Note: SAP initially delivered line item release strategies and later enhanced the application with an overall release. The Purchase*

*Requisition application is built on one table (EBAN) and is **not** separated into header and line item tables like most other applications.*

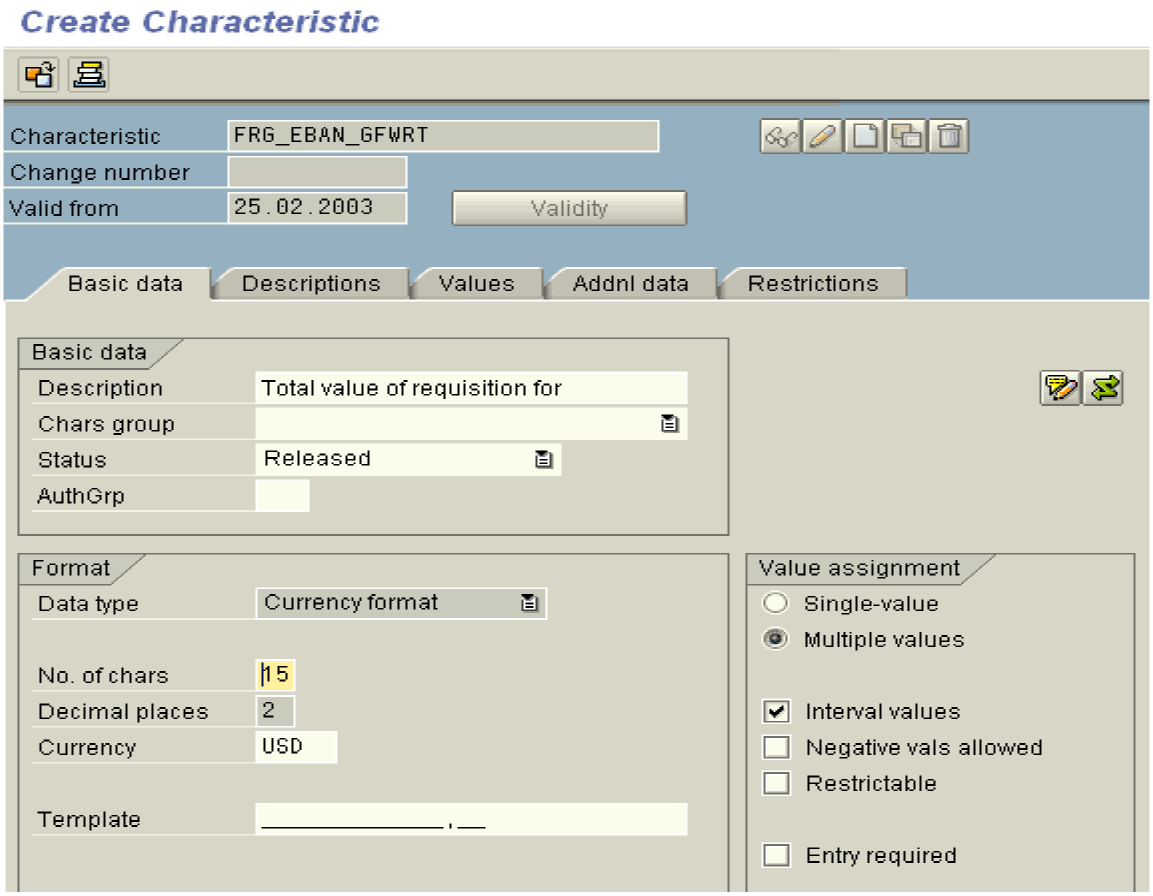


Type	Doc. type descript.	Item...	NR int.as.	No.rng.ext	Field sel.	Cont...	OverRe...	Variar
EC	Purch.requis. B2B	10	01	RQ	NBB		□	
FO	Framework requisn.	10	01	02	FOF		□	SRV
IN	Purch.requis. I-Comm	10	01	02	NBB		□	
MV	Model specification	10	01	02	RVB	R	□	
NB	Purch.requis. Stand.	10	01	02	NBB		☑	
RV	Outl. agmt. requisn.	10	01	02	RVB	R	□	

Figure 2: Requisition Workflow Types

Classification (area menu transaction CL00) is required to set up class and characteristic values used to determine the release strategy.

Create Characteristic Values (CT01)



Create Characteristic

Characteristic	FRG_EBAN_GFWRT
Change number	
Valid from	25.02.2003

Basic data

Description	Total value of requisition for
Chars group	
Status	Released
AuthGrp	

Format

Data type	Currency format
No. of chars	15
Decimal places	2
Currency	USD
Template	______________,__

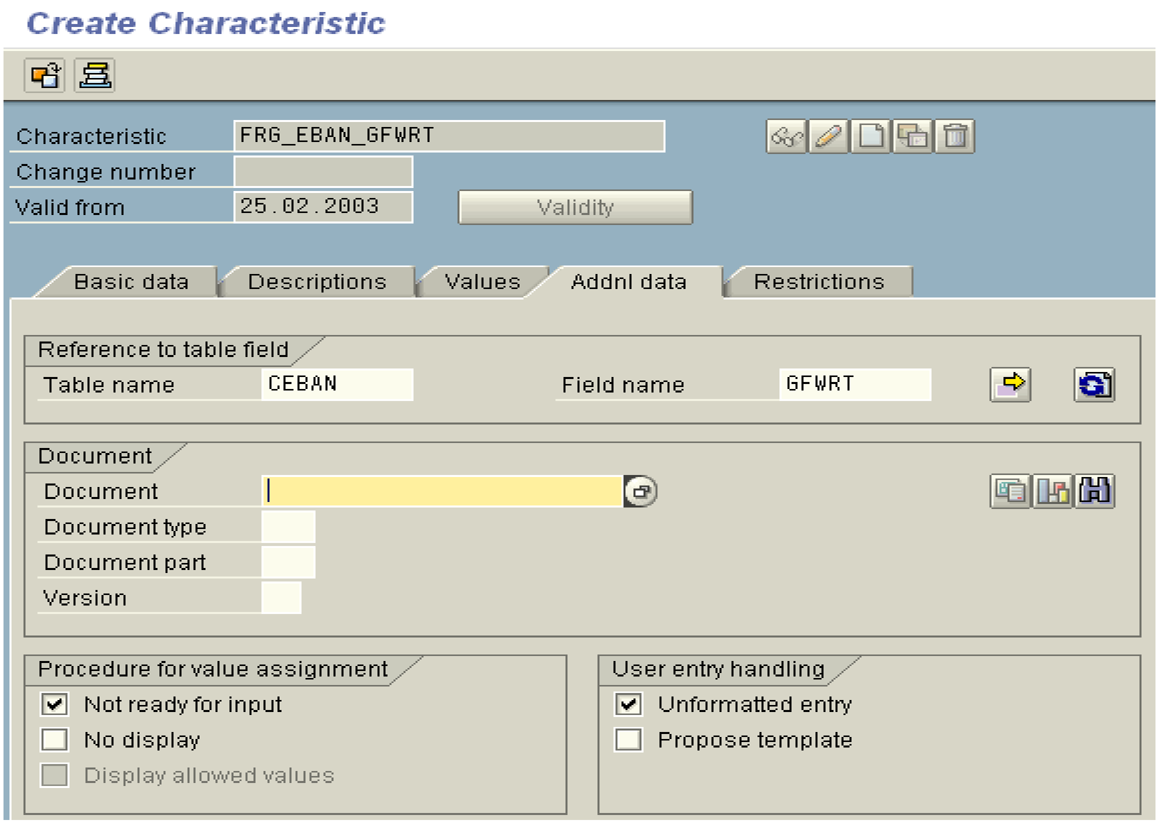
Value assignment

○ Single-value
● Multiple values

☑ Interval values
□ Negative vals allowed
□ Restrictable

□ Entry required

Figure 3: Create Characteristic screen displaying Basic data tab



Create Characteristic

Characteristic	FRG_EBAN_GFWRT
Change number	
Valid from	25.02.2003

Reference to table field

Table name	CEBAN	Field name	GFWRT

Document

Document	
Document type	
Document part	
Version	

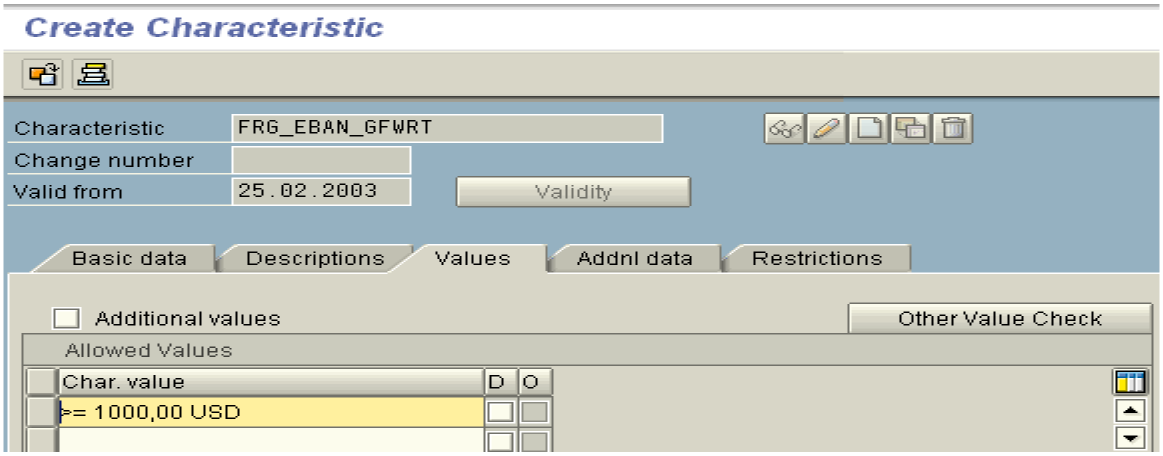
Procedure for value assignment

☑ Not ready for input
□ No display
□ Display allowed values

User entry handling

☑ Unformatted entry
□ Propose template

Figure 4: Create Characteristic screen displaying Addnl Data tab



Create Characteristic

Characteristic	FRG_EBAN_GFWRT
Change number	
Valid from	25.02.2003

□ Additional values

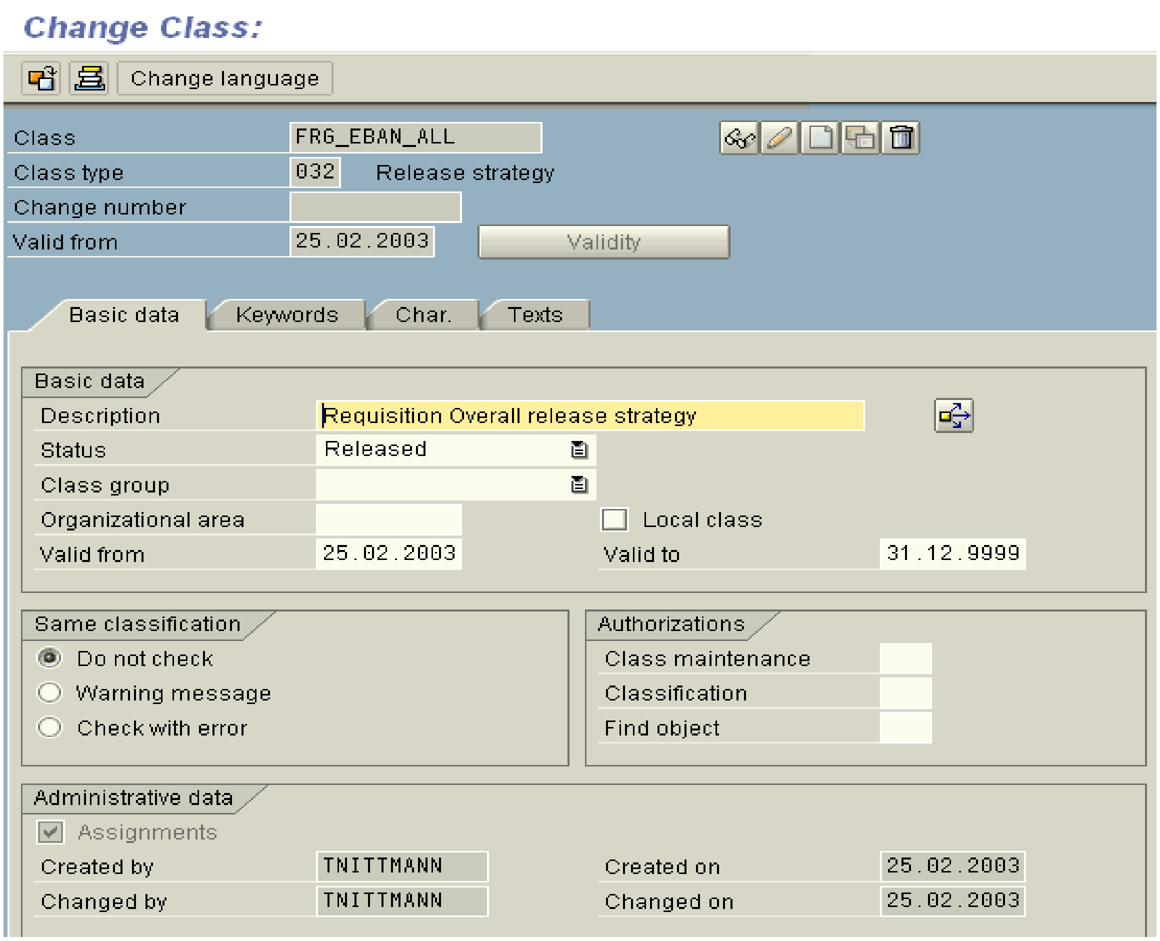
Allowed Values

Char. value	D	O
>= 1000,00 USD	□	□
	□	□

Figure 5: Create Characteristic screen displaying Values tab

Set up an overall release class (CL01), assign it to class type ‘032’, and assign a Characteristic Value to it.

Class Basic Data



Change Class:

Change language

Class	FRG_EBAN_ALL	
Class type	032	Release strategy
Change number		
Valid from	25.02.2003	Validity

Basic data | Keywords | Char. | Texts

Basic data

Description	Requisition Overall release strategy		
Status	Released		
Class group			
Organizational area		□ Local class	
Valid from	25.02.2003	Valid to	31.12.9999

Same classification

- ● Do not check
- ○ Warning message
- ○ Check with error

Authorizations

Class maintenance	
Classification	
Find object	

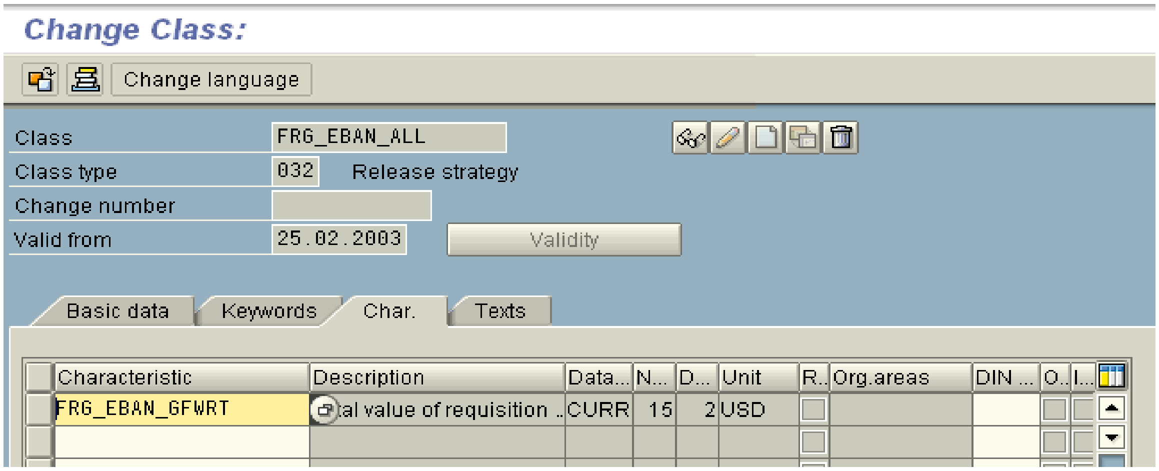
Administrative data

☑ Assignments

Created by	TNITTMANN	Created on	25.02.2003
Changed by	TNITTMANN	Changed on	25.02.2003

Figure 6: Change Class screen displaying Basic data tab

Class Assignment to Characteristic Values



Change Class:

Change language

Class	FRG_EBAN_ALL	
Class type	032	Release strategy
Change number		
Valid from	25.02.2003	Validity

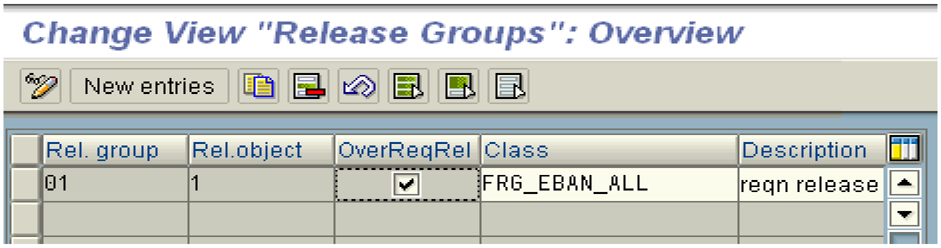
Basic data | Keywords | Char. | Texts

Characteristic	Description	Data...	N...	D...	Unit	R..	Org.areas	DIN ...	O..	I...
FRG_EBAN_GFWRT	al value of requisition ..	CURR	15	2	USD	□			□	□
						□			□	□

Figure 7: Change Class screen displaying Characteristic tab

3. Transaction OMGQ is required to activate the event trigger mechanism from the MM requisition application.

Release Group:

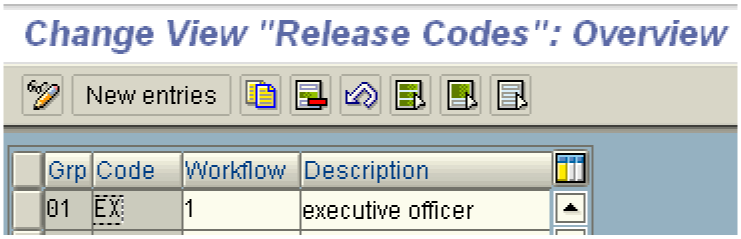


Change View "Release Groups": Overview

Rel. group	Rel.object	OverReqRel	Class	Description
01	1	☑	FRG_EBAN_ALL	reqn release

Figure 8: Change View “Release Groups”: Overview screen

Release code:

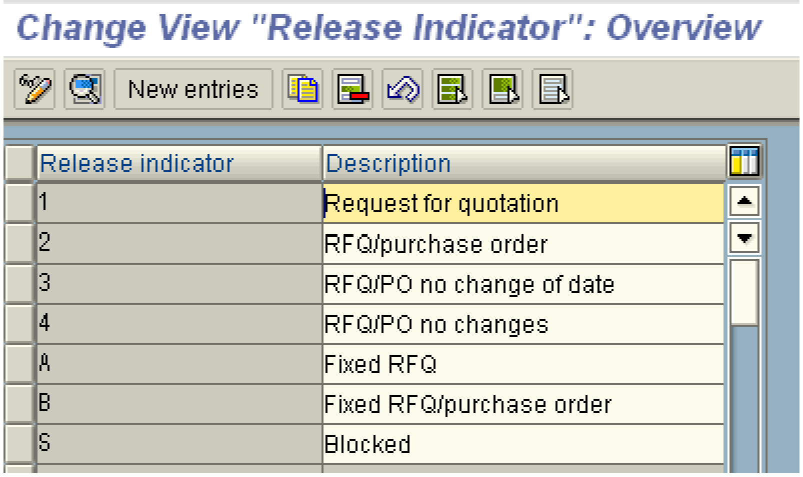


Change View "Release Codes": Overview

Grp	Code	Workflow	Description
01	EX	1	executive officer

Figure 9: Change View “Release Codes”: Overview screen

Release Indicators:

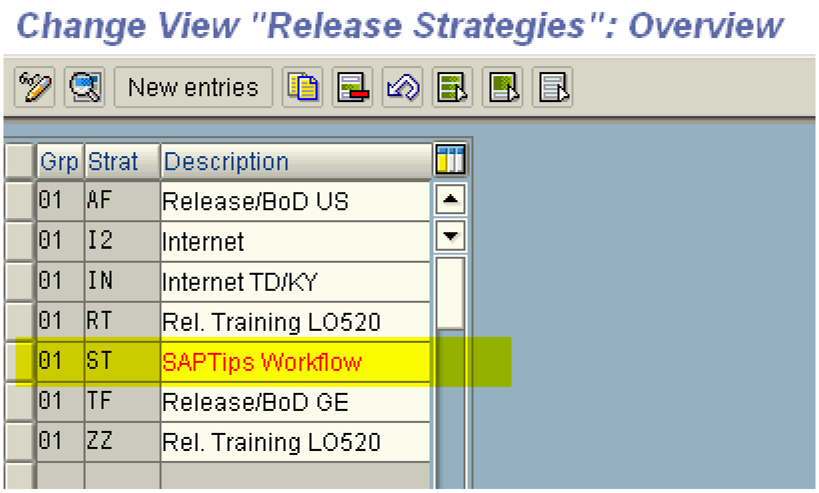


Change View "Release Indicator": Overview

Release indicator	Description
1	Request for quotation
2	RFQ/purchase order
3	RFQ/PO no change of date
4	RFQ/PO no changes
A	Fixed RFQ
B	Fixed RFQ/purchase order
S	Blocked

Figure 10: Change View “Release Indicator”: Overview screen

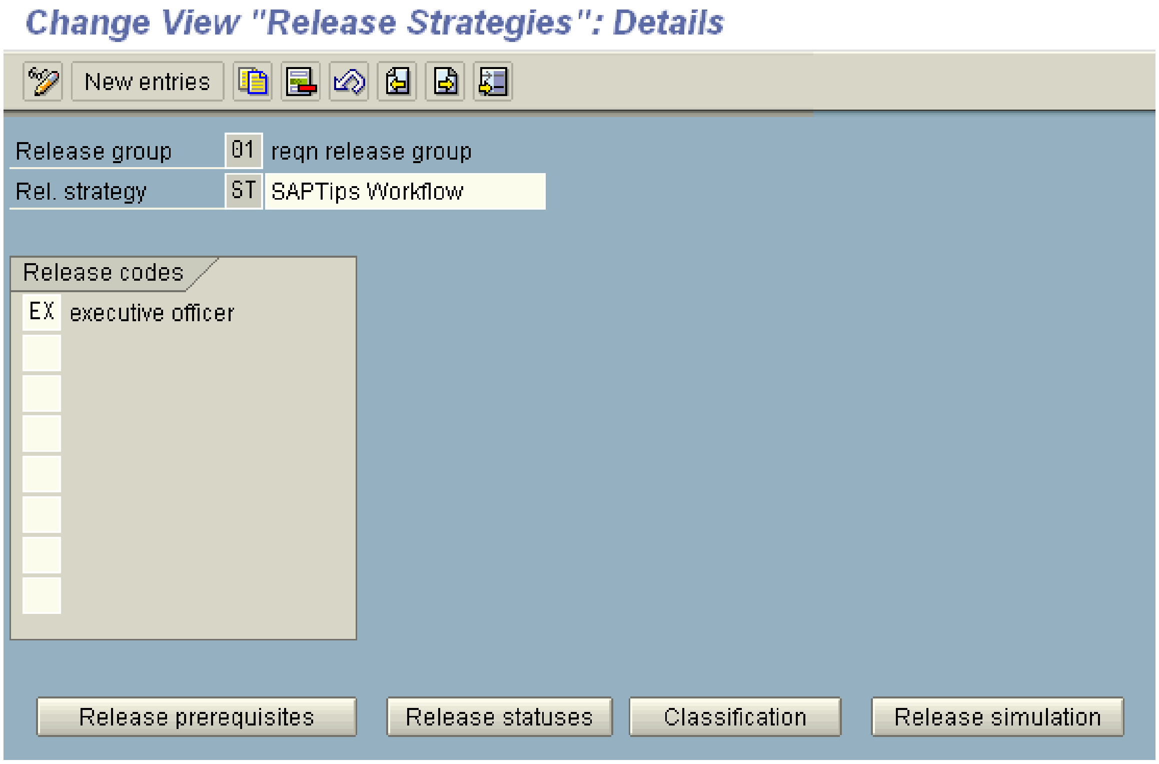
Release Strategy



Change View "Release Strategies": Overview

Grp	Strat	Description
01	AF	Release/BoD US
01	I2	Internet
01	IN	Internet TD/KY
01	RT	Rel. Training LO520
01	ST	SAPTips Workflow
01	TF	Release/BoD GE
01	ZZ	Rel. Training LO520

Figure 11: Change View "Release Strategies": Overview screen



Change View "Release Strategies": Details

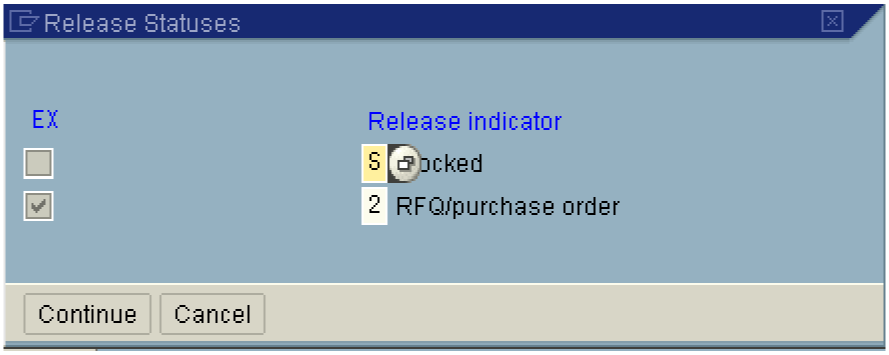
Release group 01 reqn release group

Rel. strategy ST SAPTips Workflow

Release codes

EX executive officer

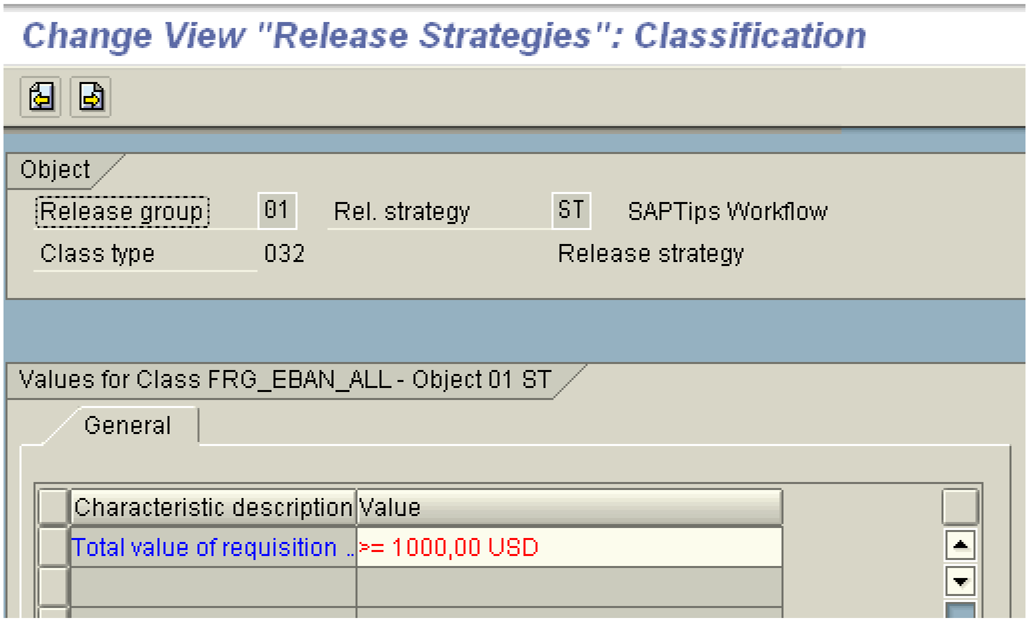
Figure 12: Change View "Release Strategies": Details screen



Release Statuses

EX	Release indicator
□	S ocked
☑	2 RFQ/purchase order

Figure 13: Release Statuses dialog box



Change View "Release Statuses": Classification

Object

Release group 01 Rel. strategy ST SAPTips Workflow

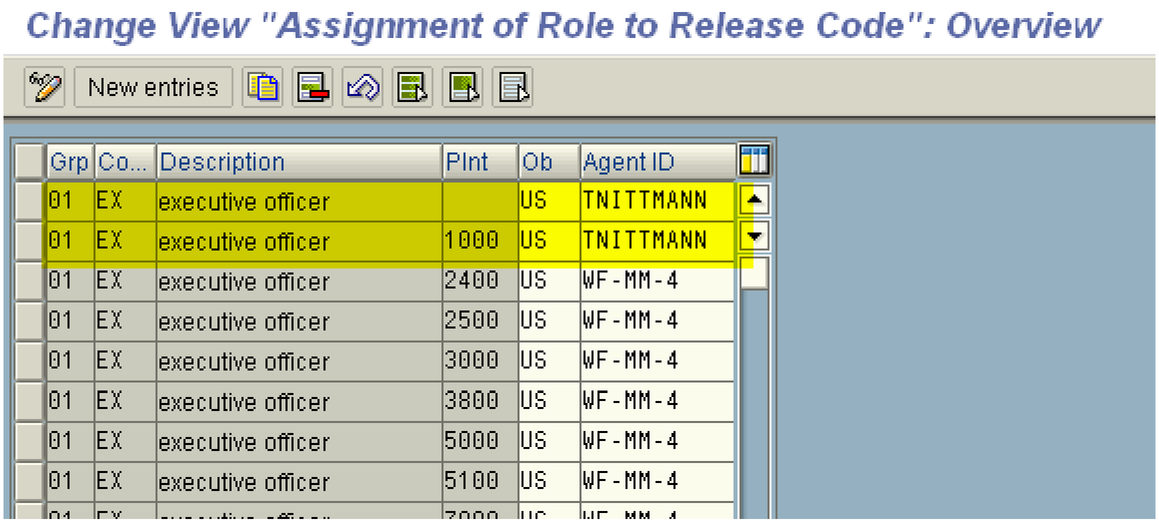
Class type 032 Release strategy

Values for Class FRG_EBAN_ALL - Object 01 ST

Characteristic description	Value
Total value of requisition ..	>= 1000,00 USD

Figure 14: Change View “Release Strategies”: Classification screen

Workflow



Change View "Assignment of Role to Release Code": Overview

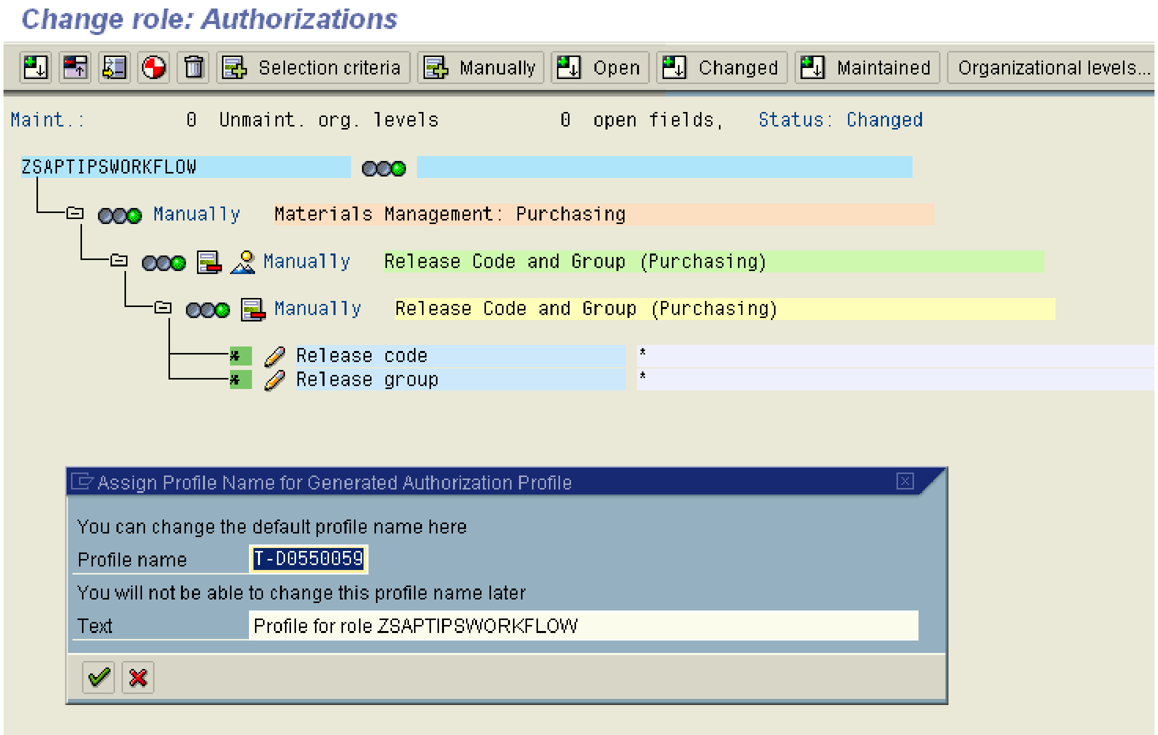
Grp	Co...	Description	Plnt	Ob	Agent ID
01	EX	executive officer		US	TNITTMANN
01	EX	executive officer	1000	US	TNITTMANN
01	EX	executive officer	2400	US	WF-MM-4
01	EX	executive officer	2500	US	WF-MM-4
01	EX	executive officer	3000	US	WF-MM-4
01	EX	executive officer	3800	US	WF-MM-4
01	EX	executive officer	5000	US	WF-MM-4
01	EX	executive officer	5100	US	WF-MM-4
01	EX	executive officer	7000	US	WF-MM-4

Figure 15: Change View “Assignment of Role to Release Code”: Overview screen

> *Note: we are currently using the default agent resolution provided by SAP release strategy configuration (OMGQ). We will discuss the use of plant in the agent resolution process below.*

4. Generate Class Statistics via transaction CLST (class type ‘032’). This optimizes the SAP search process for all characteristics created for the classification.

5. Any user that releases requisitions must be assigned ‘M_EINK_FRGAL’ authorization (PFCG/SU01).



Maint.: 0 Unmaint. org. levels 0 open fields, Status: Changed

ZSAPTIPSWORKFLOW
- Manually Materials Management: Purchasing
  - Manually Release Code and Group (Purchasing)
    - Manually Release Code and Group (Purchasing)
      - Release code *
      - Release group *

Assign Profile Name for Generated Authorization Profile

You can change the default profile name here

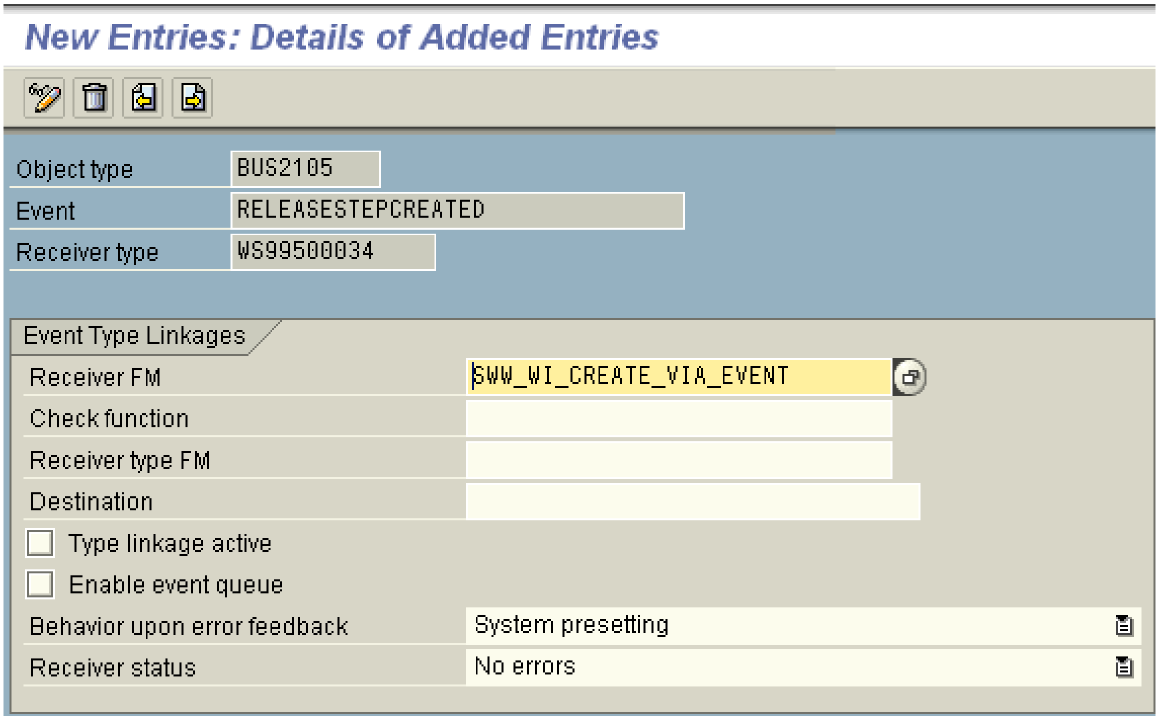
Profile name T-D0550059

You will not be able to change this profile name later

Text Profile for role ZSAPTIPSWORKFLOW

Figure 16: Change role: Authorizations screen

6. Lastly, make sure the event linkage is set up via SWE2.



Object type	BUS2105
Event	RELEASESTEPCREATED
Receiver type	WS99500034

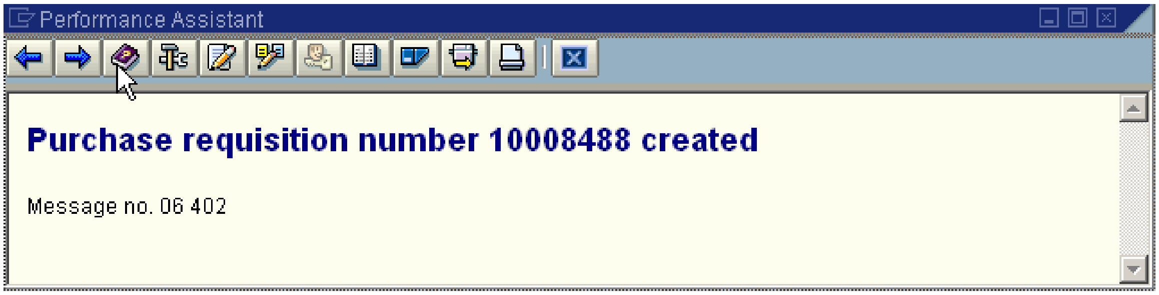
Event Type Linkages

Receiver FM	SWW_WI_CREATE_VIA_EVENT
Check function	
Receiver type FM	
Destination	
□ Type linkage active	
□ Enable event queue	
Behavior upon error feedback	System presetting
Receiver status	No errors

Figure 17: New Entries: Details of Added Entries screen

Event Debugging

As you can see, there was a significant amount of work required to set up a very simple example of the release workflow for purchase requisitions, however, we are now ready to go! Workflows are primarily triggered based on events generated by applications. Once the baseline configuration is complete, we would expect the purchase requisition application to trigger events that will start workflow when we create purchase requisitions that meet our criteria, e.g., any purchase requisition w/ document type 'NB' and a total value over USD1000.00 (based on our OMGQ configuration). Let's create the requisition. The requisition release workflow uses trigger events coded in the original main MM purchasing program SAPMM06B. Most often, issues arise with the parameters that the developer expects to be passed into the workflow container from the event (i.e., the event/workflow template binding condition). To expedite resolution to event/workflow binding issues, we will review some helpful debugging and troubleshooting tips:

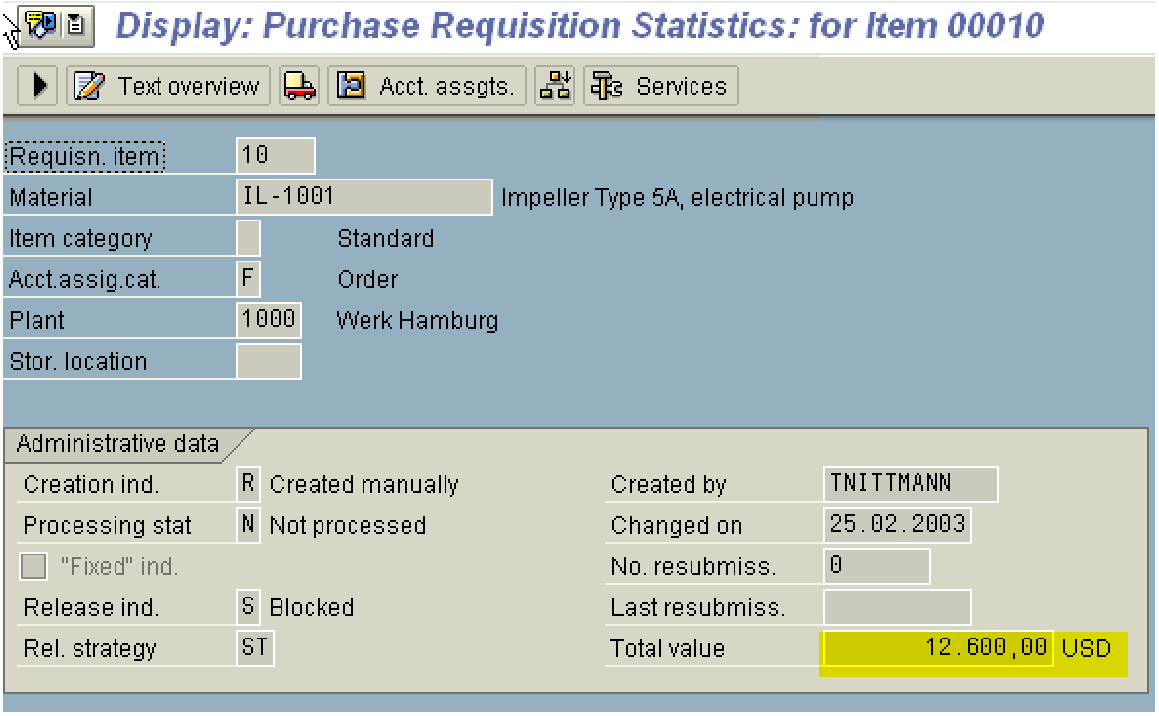


Purchase requisition number 10008488 created

Message no. 06 402

Figure 18: Performance Assistant screen

The total value of requisition '10008488' is USD12,600.00, so it requires the release strategy we setup in OMGQ. So, in theory, we should have a workflow.

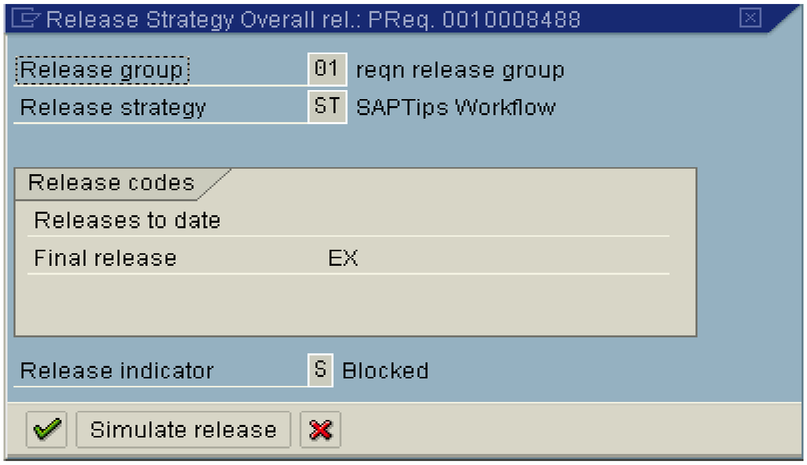


Display: Purchase Requisition Statistics: for Item 00010

Requisn. item	10	
Material	IL-1001	Impeller Type 5A, electrical pump
Item category		Standard
Acct.assig.cat.	F	Order
Plant	1000	Werk Hamburg
Stor. location		

Creation ind.	R	Created manually	Created by	TNITTMANN
Processing stat	N	Not processed	Changed on	25.02.2003
□ "Fixed" ind.			No. resubmiss.	0
Release ind.	S	Blocked	Last resubmiss.	
Rel. strategy	ST		Total value	12.600,00 USD

Figure 19: Display: Purchase Requisition Statistics screen displaying details on selected item


Release Strategy Overall rel.: PReq. 0010008488

Release group	01	reqn release group
Release strategy	ST	SAPTips Workflow

Release codes

Releases to date	
Final release	EX

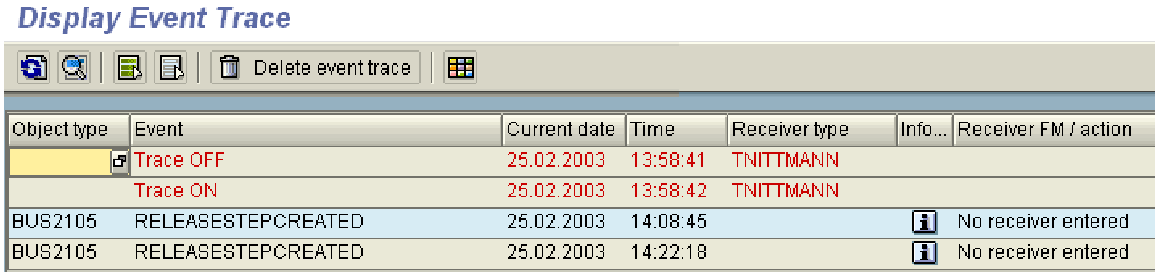
Release indicator	S	Blocked

Figure 20: Release Strategy Overall rel.: screen

Problem1: Event Linkage:

But when we go to transaction ‘SWI1,’ we notice that no workflows exist at all! Before we start going through the purchase requisition program in debug mode, we should start by utilizing some standard SAP utility transactions:

a. Turn on the event log with transaction SWELS.
b. Re-create another purchase requisition by copying the one we just created.
c. Display the event trace w/ transaction SWEL.


Display Event Trace

Object type	Event	Current date	Time	Receiver type	Info...	Receiver FM / action
	Trace OFF	25.02.2003	13:58:41	TNITTMANN		
	Trace ON	25.02.2003	13:58:42	TNITTMANN		
BUS2105	RELEASESTEPCREATED	25.02.2003	14:08:45		ℹ	No receiver entered
BUS2105	RELEASESTEPCREATED	25.02.2003	14:22:18		ℹ	No receiver entered

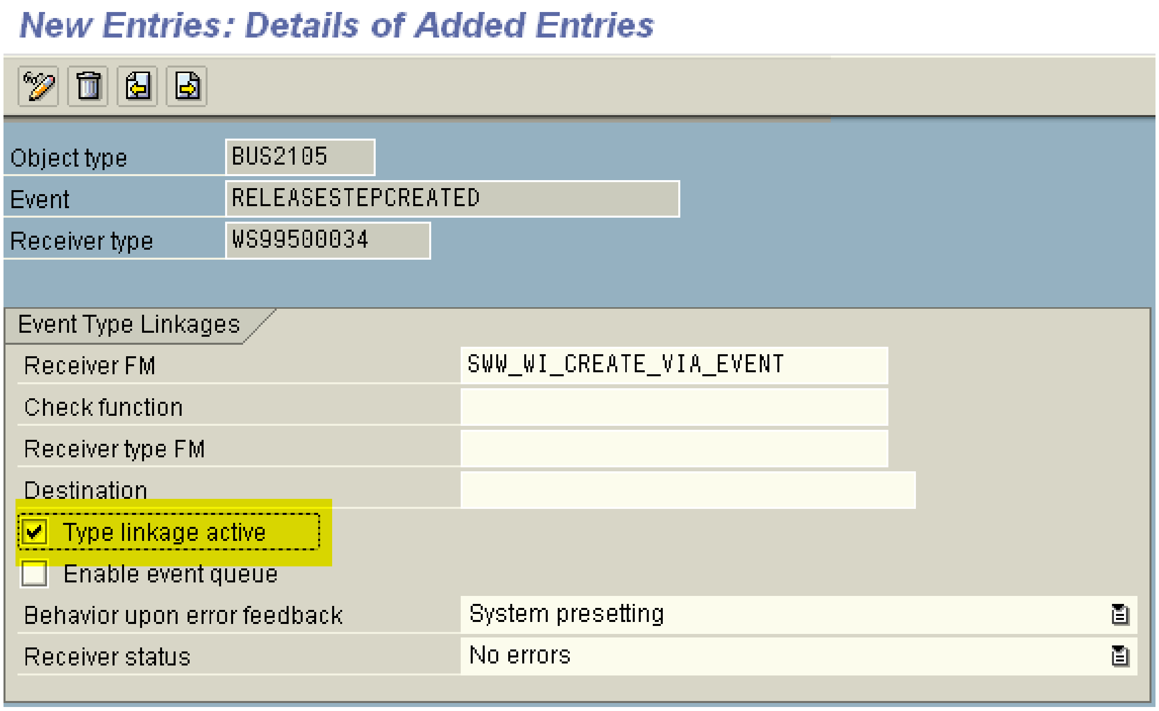
Figure 21: Display Event Trace screen

> *Hint: In previous SAP releases, a workflow development area menu was available to easily access the most common WF transactions. If you are more comfortable with the older area menu, you can escape from the ‘Easy Menu’ by entering ‘/n’ in the transaction window and then entering transaction ‘SWLD’ to get to the old area menu.*

Resolution to Problem 1:

The event trace shows no event receiver was found. This error is typical if the event linkage is not set up or not active in transaction SWE2. By reviewing SWE2, we notice that the type linkage was not set to 'active.' To resolve the issue, activate the event linkage and be sure to save the entries. To continue the exercise, we can either create a new requisition to ensure workflow starts, or manually start workflow for the existing requisitions.

Tip: Ideally, in a production environment, if workflow didn't start when it should have, we would want to manually start it. You can manually create the workflow by using either transaction SWUE (create event) or SWUS (start workflow). However, it's best to use transaction SWI1 and find a similar workflow that has been generated to get the correct syntax and values for the container elements.



New Entries: Details of Added Entries

Object type: BUS2105
Event: RELEASESTEPCREATED
Receiver type: WS99500034

Event Type Linkages

Receiver FM: SWW_WI_CREATE_VIA_EVENT
Check function:
Receiver type FM:
Destination:
☑ Type linkage active
☐ Enable event queue
Behavior upon error feedback: System presetting
Receiver status: No errors

Figure 22: New Entries: Details of Added Entries screen

Other items to review if the issue was not resolved:

- SWU3: Check the Workflow Engine – make sure all items are configured properly (all lights are green).
- SWUB (part of SWU3): Validate that the workflow RFC destination is configured completely. First synchronize the passwords and then test the RFC connection with the caliper icon (function key F5).
- SM58: workflow is based on tRFCs. The tRFC errors are logged and can be reviewed using transaction SM58.
- SM21: Check the system log for workflow specific errors.
- ST22: Check for workflow specific short dumps.

Problem 2: Event Parameters:

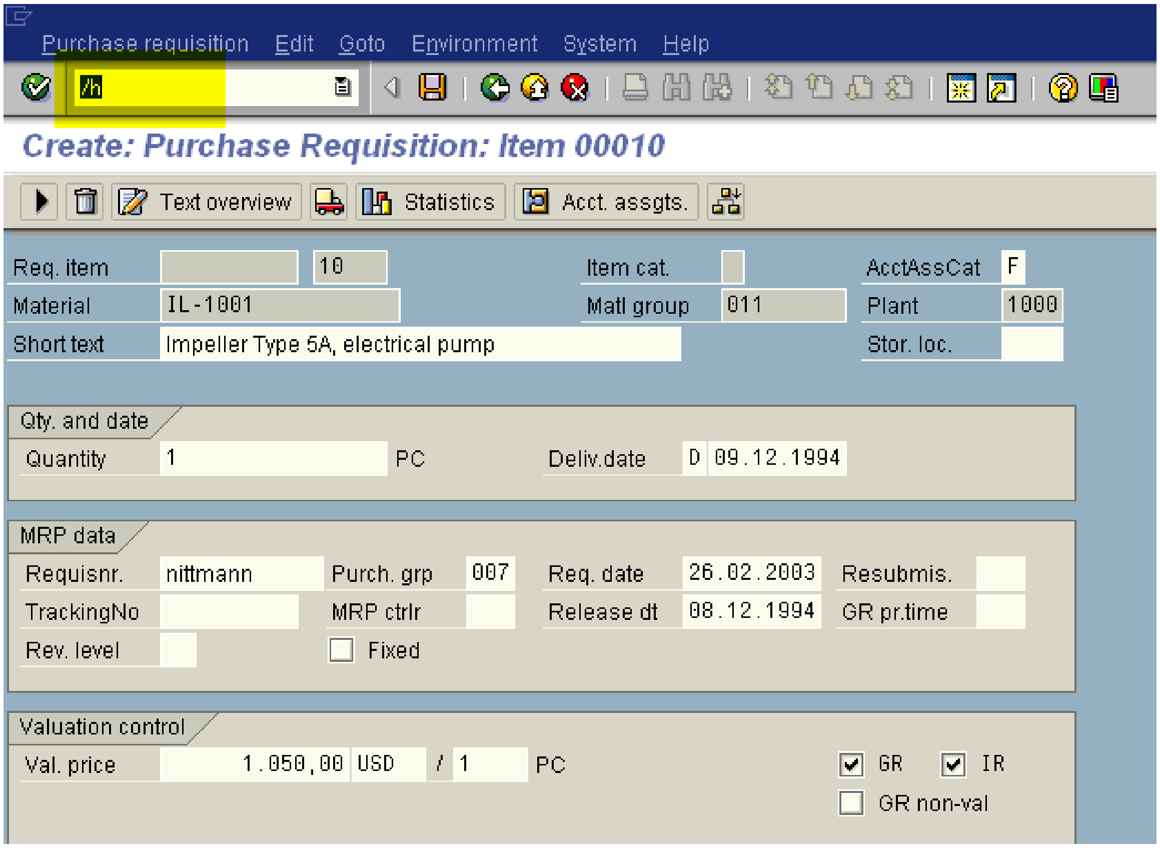
Often, customers identify additional event parameters that are required in the workflow but are not being passed from the event trigger. In addition, you may expect values that you are not seeing in the workflow, so you want to review the code to determine what parameters are actually getting passed from the application to the workflow.

In this specific example, SAP provided the ‘hooks’ (trigger mechanism) to start the workflow directly in the MM Purchase Requisition application. Workflow starts because we configured the OMGQ transaction to activate workflow – the MM Purchase Requisition program checks the OMGQ to determine if WF should be started. Several options exist to view the event parameters being passed:

Resolution A to Problem 2:

Debug the MM Purchase Requisition application and look for the Event Create function modules. Events are created using several possible function modules: SWE_EVENT_CREATE, SWE_EVENT_CREATE_IN_UPD_TASK, and SWE_EVENT_CREATE_FOR_UPDATE_TASK.

Note: Before we begin with this approach, it is important to mention that this particular application calls the workflow routines from a function module ‘in update task’, so we cannot simply start debugging and force a break-point at ‘SWE_EVENT_CREATE. This option is the most work intensive, but we will review it to illustrate the process. We will proceed by using transaction ME51 to create a new purchase requisition (w/ similar characteristics as before so the release strategy is evoked). Before saving the requisition, we switch the debug mode on by entering ‘/h’ in the transaction window.



/h

Create: Purchase Requisition: Item 00010

Req. item: 10
Item cat.:
AcctAssCat: F
Material: IL-1001
Matl group: 011
Plant: 1000
Short text: Impeller Type 5A, electrical pump
Stor. loc.:

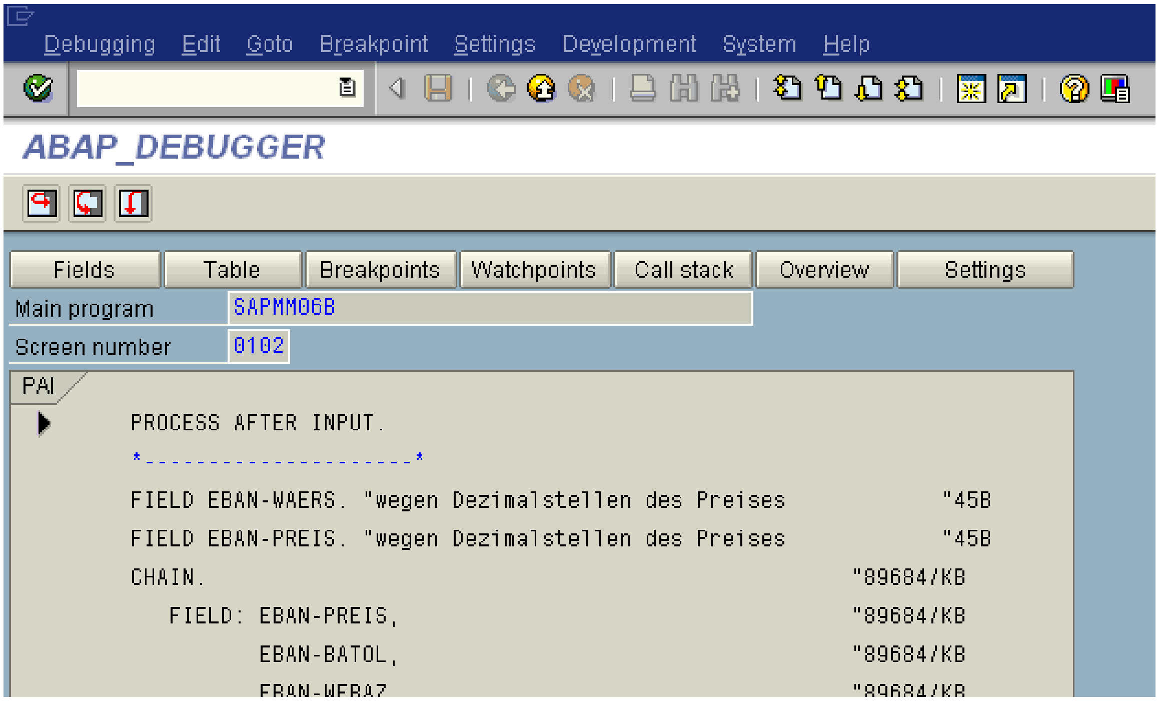
Qty. and date
Quantity: 1 PC
Deliv.date: D 09.12.1994

MRP data
Requisnr.: nittmann
Purch. grp: 007
Req. date: 26.02.2003
Resubmis.:
TrackingNo:
MRP ctrlr:
Release dt: 08.12.1994
GR pr.time:
Rev. level:
□ Fixed

Valuation control
Val. price: 1.050,00 USD / 1 PC
☑ GR ☑ IR
□ GR non-val

Figure 23: Create: Purchase Requisition screen

Now we can save the requisition and enter the primary MM program for purchase requisitions.



Main program SAPMM06B

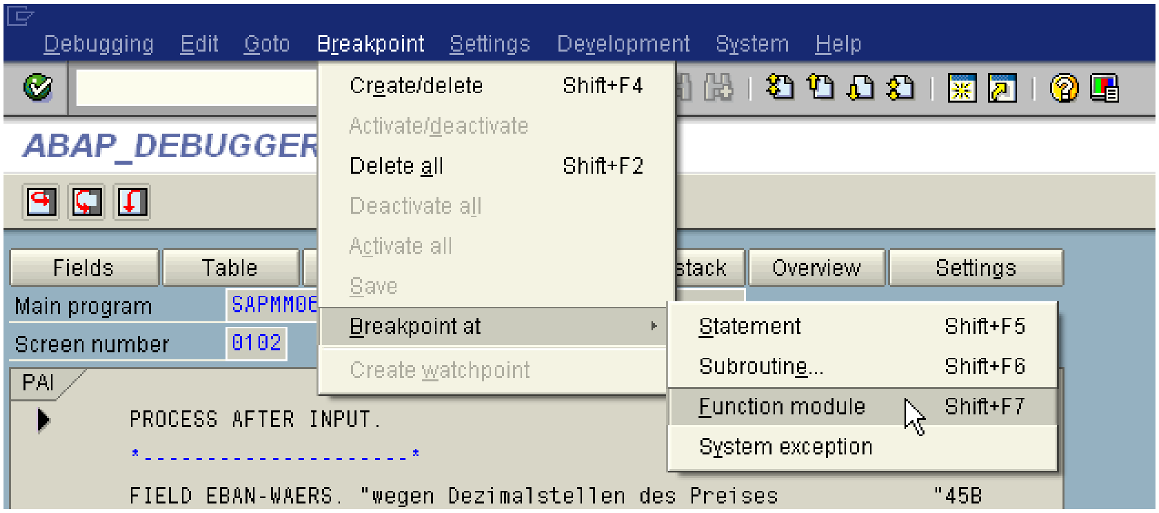
Screen number 0102

```
PROCESS AFTER INPUT.
*---------------------*
FIELD EBAN-WAERS. "wegen Dezimalstellen des Preises            "45B
FIELD EBAN-PREIS. "wegen Dezimalstellen des Preises            "45B
CHAIN.                                                      "89684/KB
   FIELD: EBAN-PREIS,                                       "89684/KB
          EBAN-BATOL,                                       "89684/KB
          EBAN-WEBAZ                                        "89684/KB
```

Figure 24: ABAP_Debugger screen

Next, the fun part begins. Initially, I would suggest to try setting a break point at 'SWE_EVENT_CREATE' and/or go to transaction SE38 or SE80, and enter program SAPMM06B and search for 'SWE_EVENT_CREATE', 'EVENT', 'RELEASE', and/or 'Workflow' to determine where the workflow event is triggered.

Searching by function module Breakpoint.

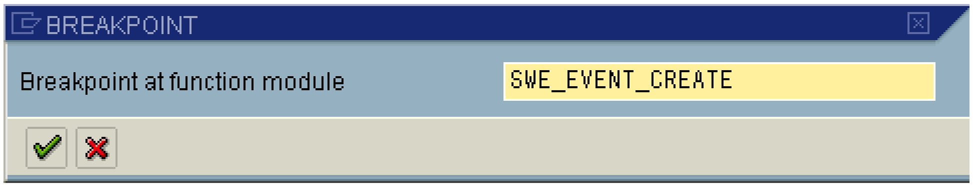


Main program SAPMM06

Screen number 0102

```
PROCESS AFTER INPUT.
*---------------------*
FIELD EBAN-WAERS. "wegen Dezimalstellen des Preises            "45B
```

Figure 25: ABAP_Debugger screen with Breakpoint pull down menu



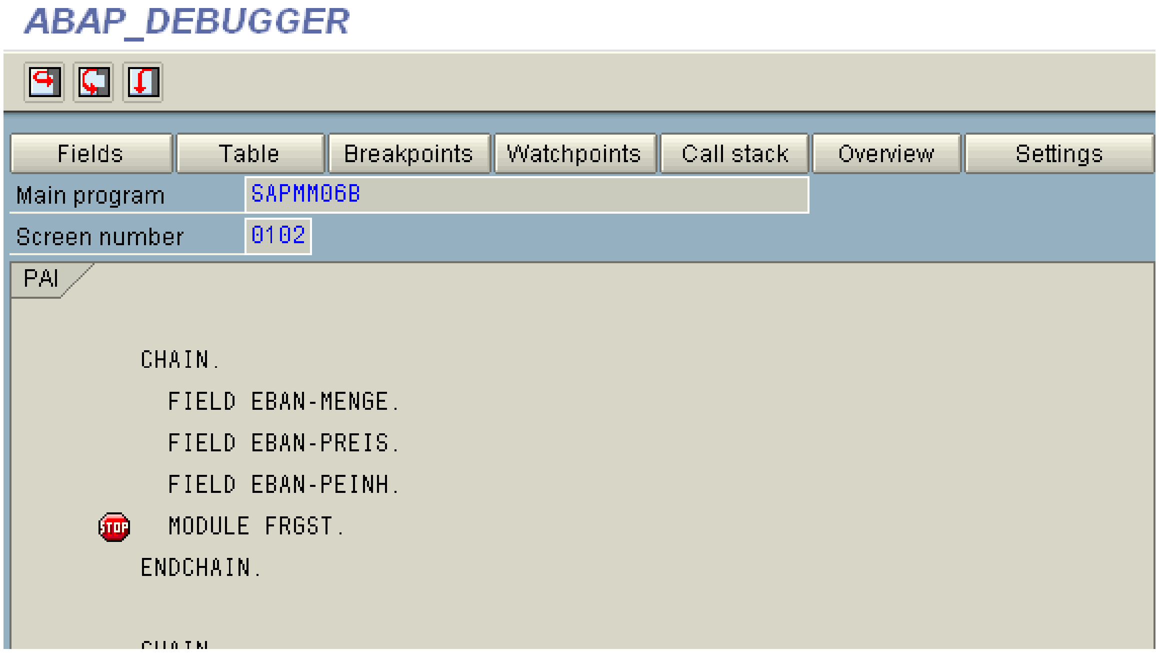
Breakpoint at function module: SWE_EVENT_CREATE

Figure 26: BREAKPOINT dialog box

By performing this once, we notice that the breakpoint is not reached. Another purchase requisition must be created and we enter the debug mode again by entering '/h' in the transaction window.

> *Hint: Most of the SAP code has its roots from the German language. The abbreviation 'FRG' seen throughout the code (and class system) is short for 'Freigabe Strategie' or 'Release Strategy'. To find out where workflow is triggered, search the main program 'SAPMM06B' for the module that begins with 'FRG'.*

In debug mode, we page down and review all the modules and pinpoint 'MODULE FRGST'. Place a breakpoint here and then continue to the breakpoint and step into the module.



Main program: SAPMM06B

Screen number: 0102

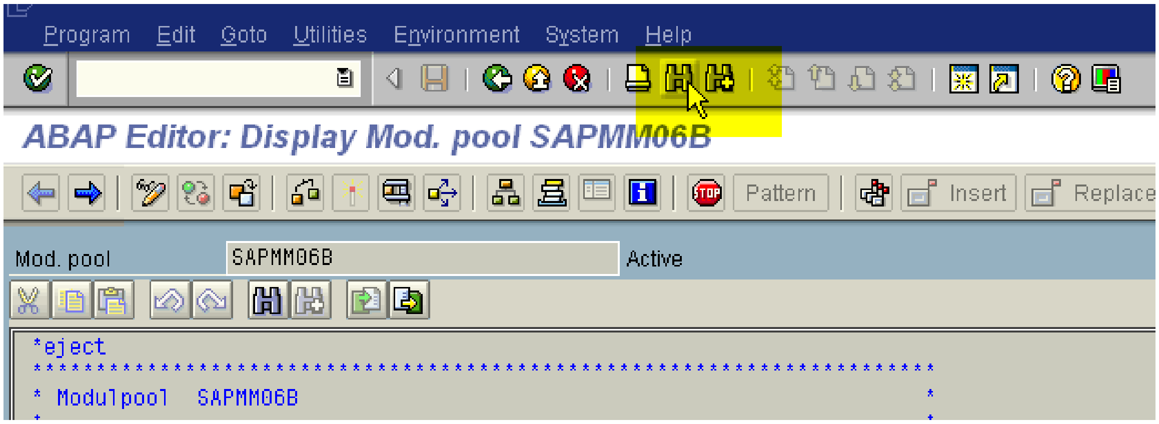
```
CHAIN.
  FIELD EBAN-MENGE.
  FIELD EBAN-PREIS.
  FIELD EBAN-PEINH.
  MODULE FRGST.
ENDCHAIN.
```

Figure 27: ABAP_DEGUGGER screen

The FRGST module is only used to determine the release strategy. Workflow is not started in this module so we must continue.

> *Note: the function modules ME_REL_STRATEGY_REQUISITION and ME_REL_GENERAL_STRATEGY_EBAN used in module FRGST can be used to troubleshoot potential issues with the release strategy determination for Purchase Requisitions.*

Page through the code and try to determine where the workflow is triggered. Let's search the primary purchase requisition program 'SAPMM06B' using search criteria 'event'.

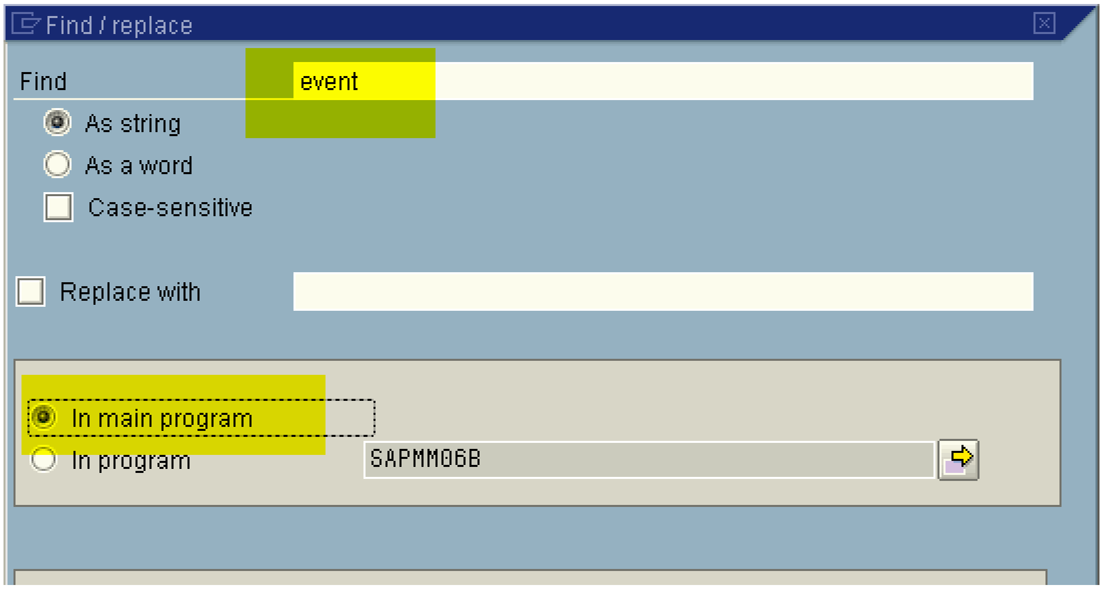


ABAP Editor: Display Mod. pool SAPMM06B

Mod. pool SAPMM06B Active

```
*eject
************************************************************************
* Modulpool  SAPMM06B                                                  *
```

Figure 28: ABAPEditor: Display screen



Find / replace

Find: event

● As string

○ As a word

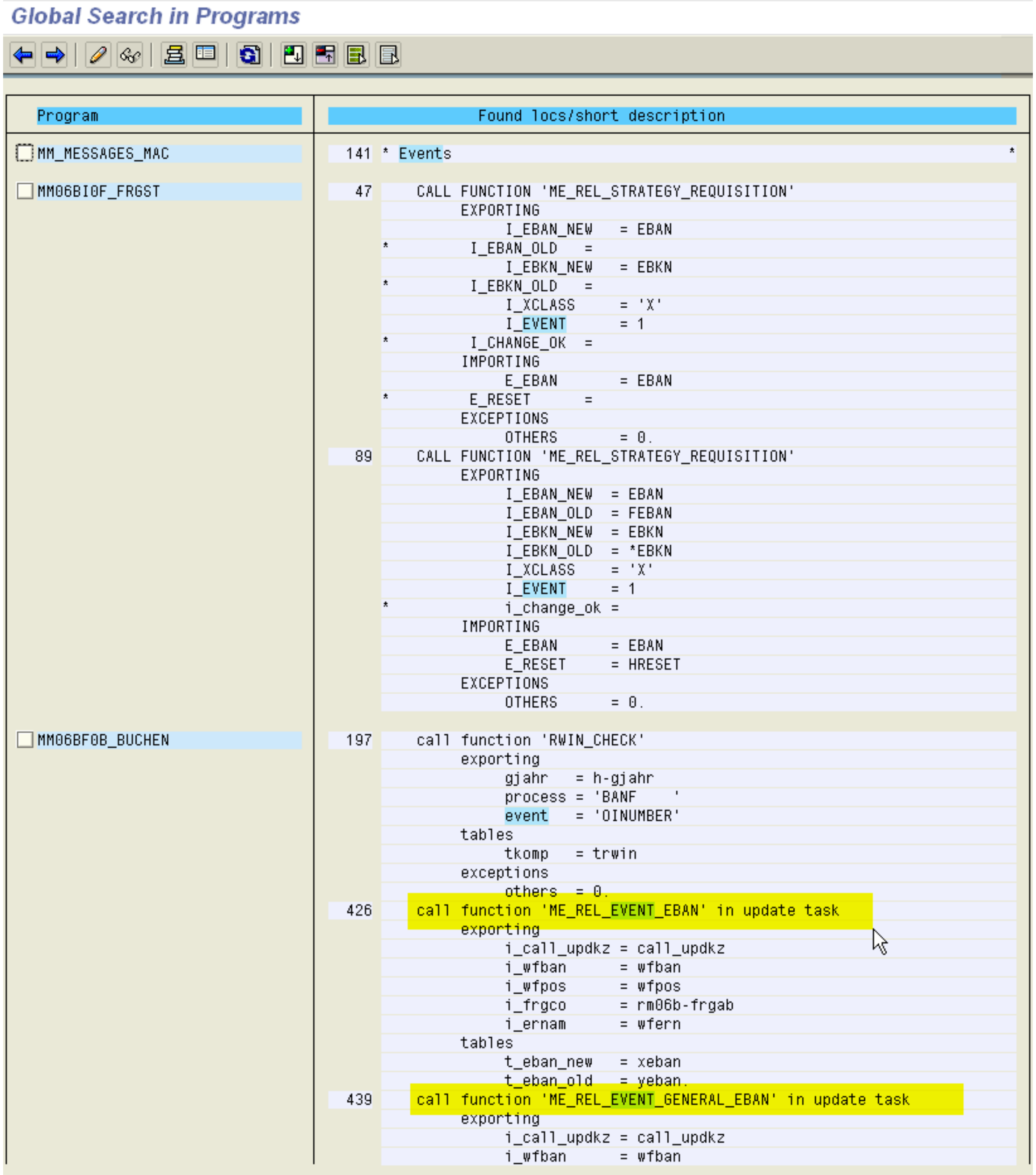
□ Case-sensitive

□ Replace with

● In main program

○ In program SAPMM06B

Figure 29: Find/replace dialog box



Global Search in Programs

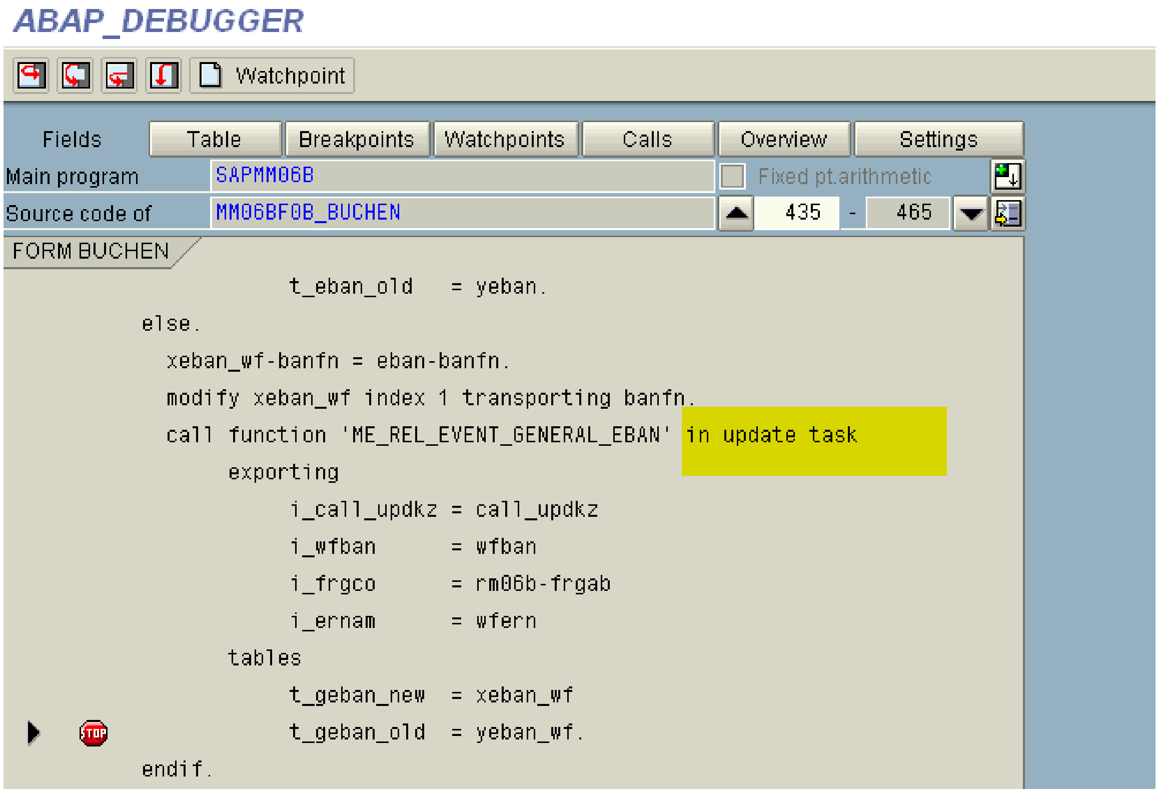
```
Program              Found locs/short description

□ MM_MESSAGES_MAC     141 * Events                                                     *

□ MM06BI0F_FRGST       47     CALL FUNCTION 'ME_REL_STRATEGY_REQUISITION'
                                   EXPORTING
                                        I_EBAN_NEW    = EBAN
                          *         I_EBAN_OLD    =
                                        I_EBKN_NEW    = EBKN
                          *         I_EBKN_OLD    =
                                        I_XCLASS      = 'X'
                                        I_EVENT       = 1
                          *         I_CHANGE_OK   =
                                   IMPORTING
                                        E_EBAN        = EBAN
                          *         E_RESET       =
                                   EXCEPTIONS
                                        OTHERS        = 0.
                       89     CALL FUNCTION 'ME_REL_STRATEGY_REQUISITION'
                                   EXPORTING
                                        I_EBAN_NEW   = EBAN
                                        I_EBAN_OLD   = FEBAN
                                        I_EBKN_NEW   = EBKN
                                        I_EBKN_OLD   = *EBKN
                                        I_XCLASS     = 'X'
                                        I_EVENT      = 1
                          *             i_change_ok =
                                   IMPORTING
                                        E_EBAN       = EBAN
                                        E_RESET      = HRESET
                                   EXCEPTIONS
                                        OTHERS       = 0.

□ MM06BF0B_BUCHEN     197     call function 'RWIN_CHECK'
                                   exporting
                                        gjahr    = h-gjahr
                                        process = 'BANF     '
                                        event    = 'OINUMBER'
                                   tables
                                        tkomp    = trwin
                                   exceptions
                                        others   = 0.
                      426     call function 'ME_REL_EVENT_EBAN' in update task
                                   exporting
                                        i_call_updkz = call_updkz
                                        i_wfban      = wfban
                                        i_wfpos      = wfpos
                                        i_frgco      = rm06b-frgab
                                        i_ernam      = wfern
                                   tables
                                        t_eban_new   = xeban
                                        t_eban_old   = yeban.
                      439     call function 'ME_REL_EVENT_GENERAL_EBAN' in update task
                                   exporting
                                        i_call_updkz = call_updkz
                                        i_wfban      = wfban
```

Figure 30: Global Search in Programs screen

Page through the search criteria and try to select possible areas (and set breakpoints) where the event may be called. After a few attempts, you should be able to determine function module 'ME_REL_EVENT_GENERAL_EBAN' calls the event trigger 'SWE_EVENT_CREATE'.

> *Note: In this example, function module 'ME_REL_EVENT_EBAN' is used for individual line item release. We are using overall line item release, i.e., we want approval based on the total value of all line items.*

Notice that 'ME_REL_EVENT_GENERAL_EBAN' is called ‘in update task’. The ‘in update task’ is used to bundle many database changes into a single logical unit of work. All function modules belonging to the purchase requisition transaction are assigned an update key. When a commit work is performed, the update task reads the queue and processes all requests with the update key. We can set a breakpoint here but won’t be able to debug unless we activate ‘UPDATE DEBUGGING’.



Main program: SAPMM06B

Source code of: MM06BF0B_BUCHEN 435 - 465

FORM BUCHEN

```
                  t_eban_old   = yeban.
         else.
           xeban_wf-banfn = eban-banfn.
           modify xeban_wf index 1 transporting banfn.
           call function 'ME_REL_EVENT_GENERAL_EBAN' in update task
               exporting
                   i_call_updkz = call_updkz
                   i_wfban      = wfban
                   i_frgco      = rm06b-frgab
                   i_ernam      = wfern
               tables
                   t_geban_new  = xeban_wf
                   t_geban_old  = yeban_wf.
         endif.
```

Figure 31: ABAP_DEBUGGER screen

Since the ME_REL_EVENT_GENERAL_EBAN function module is being called ‘in update task,’ we must activate update debugging.

Note: for documentation on UPDATE DEBUGGING, see the R/3 online documentation:
ABAP Run time Tools >
Settings and Warnings (search for update debug)
Settings > Update Mode (Use update debugging in debug mode)

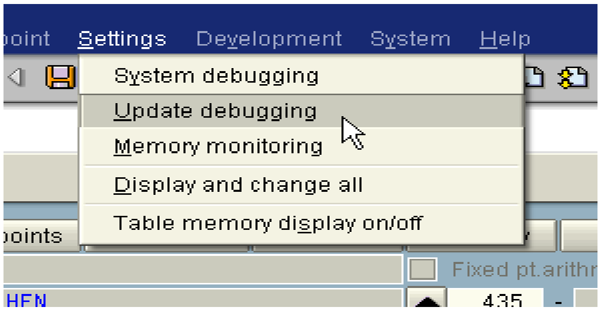
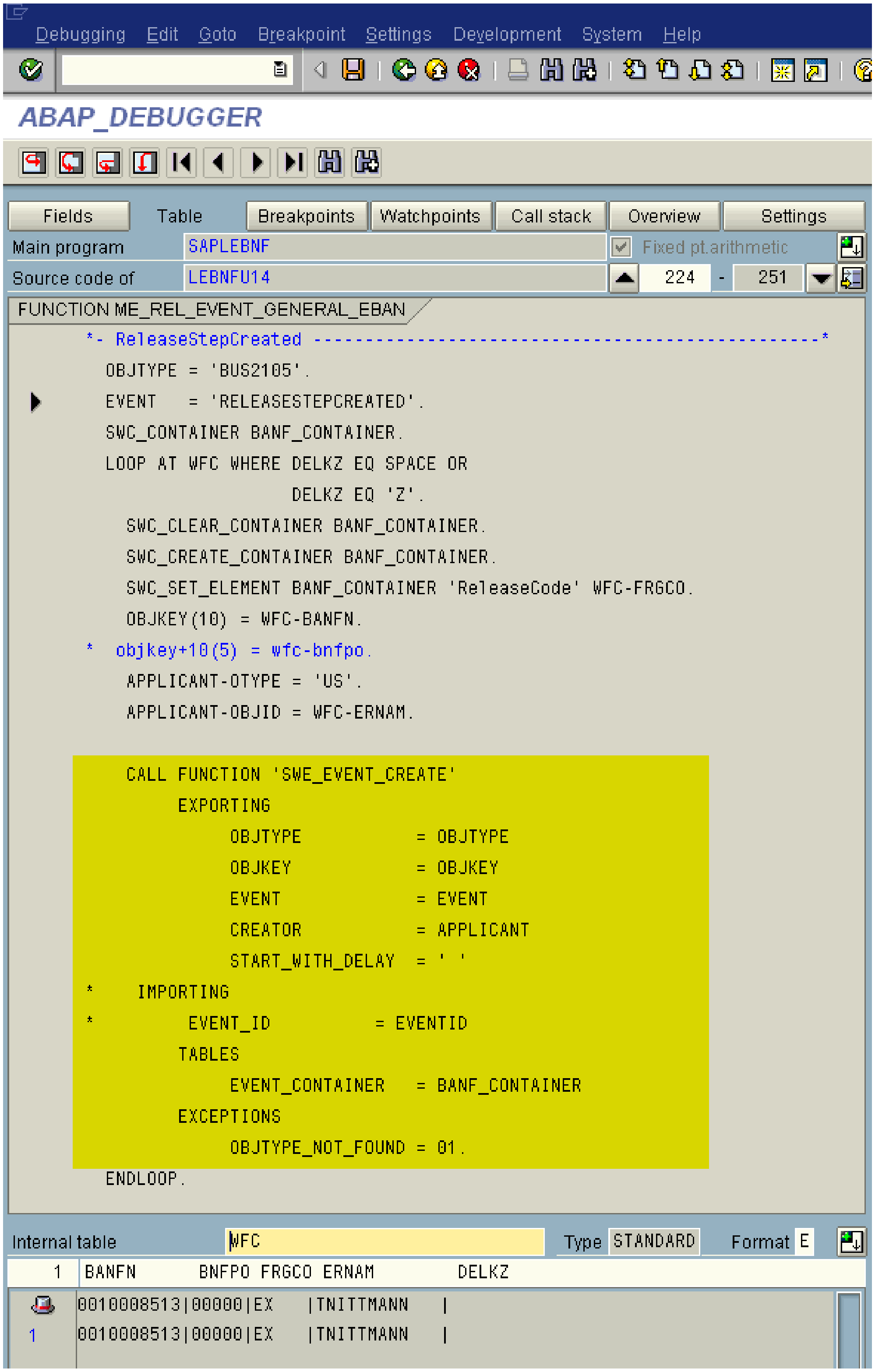


Figure 32: Settings pull down menu

After the purchase requisition passes all pre-processing, the update task function modules are executed. Since we enabled 'Update Debugging,' a new session will open with the 'Update Task' function modules. In the next print screen we finally get to the 'SWE_EVENT_CREATE' function module, which is called from 'ME_REL_EVENT_GENERAL_EBAN'.



Main program SAPLEBNF ☑ Fixed pt.arithmetic

Source code of LEBNFU14 224 - 251

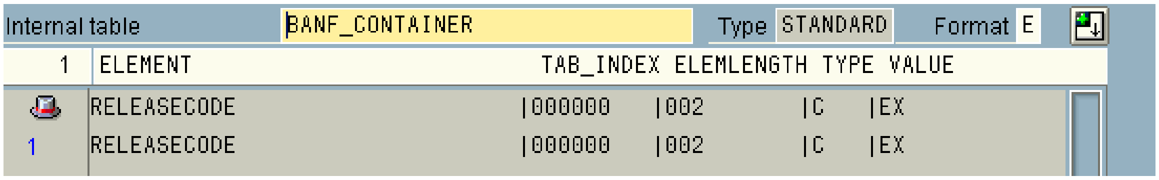
FUNCTION ME_REL_EVENT_GENERAL_EBAN

```
*- ReleaseStepCreated -------------------------------------------------*
     OBJTYPE = 'BUS2105'.
     EVENT   = 'RELEASESTEPCREATED'.
     SWC_CONTAINER BANF_CONTAINER.
     LOOP AT WFC WHERE DELKZ EQ SPACE OR
                       DELKZ EQ 'Z'.
       SWC_CLEAR_CONTAINER BANF_CONTAINER.
       SWC_CREATE_CONTAINER BANF_CONTAINER.
       SWC_SET_ELEMENT BANF_CONTAINER 'ReleaseCode' WFC-FRGCO.
       OBJKEY(10) = WFC-BANFN.
*  objkey+10(5) = wfc-bnfpo.
       APPLICANT-OTYPE = 'US'.
       APPLICANT-OBJID = WFC-ERNAM.

       CALL FUNCTION 'SWE_EVENT_CREATE'
            EXPORTING
                 OBJTYPE           = OBJTYPE
                 OBJKEY            = OBJKEY
                 EVENT             = EVENT
                 CREATOR           = APPLICANT
                 START_WITH_DELAY  = ' '
*           IMPORTING
*                EVENT_ID          = EVENTID
            TABLES
                 EVENT_CONTAINER   = BANF_CONTAINER
            EXCEPTIONS
                 OBJTYPE_NOT_FOUND = 01.
     ENDLOOP.
```

Internal table WFC Type STANDARD Format E

```
    1 BANFN      BNFPO FRGCO ERNAM        DELKZ
      0010008513|00000|EX    |TNITTMANN   |
    1 0010008513|00000|EX    |TNITTMANN   |
```

Figure 33: ABAP_DEBUGGER screen



```
Internal table  BANF_CONTAINER        Type STANDARD   Format E
   1  ELEMENT                     TAB_INDEX ELEMLENGTH TYPE VALUE
      RELEASECODE                |000000    |002        |C   |EX
   1  RELEASECODE                |000000    |002        |C   |EX
```

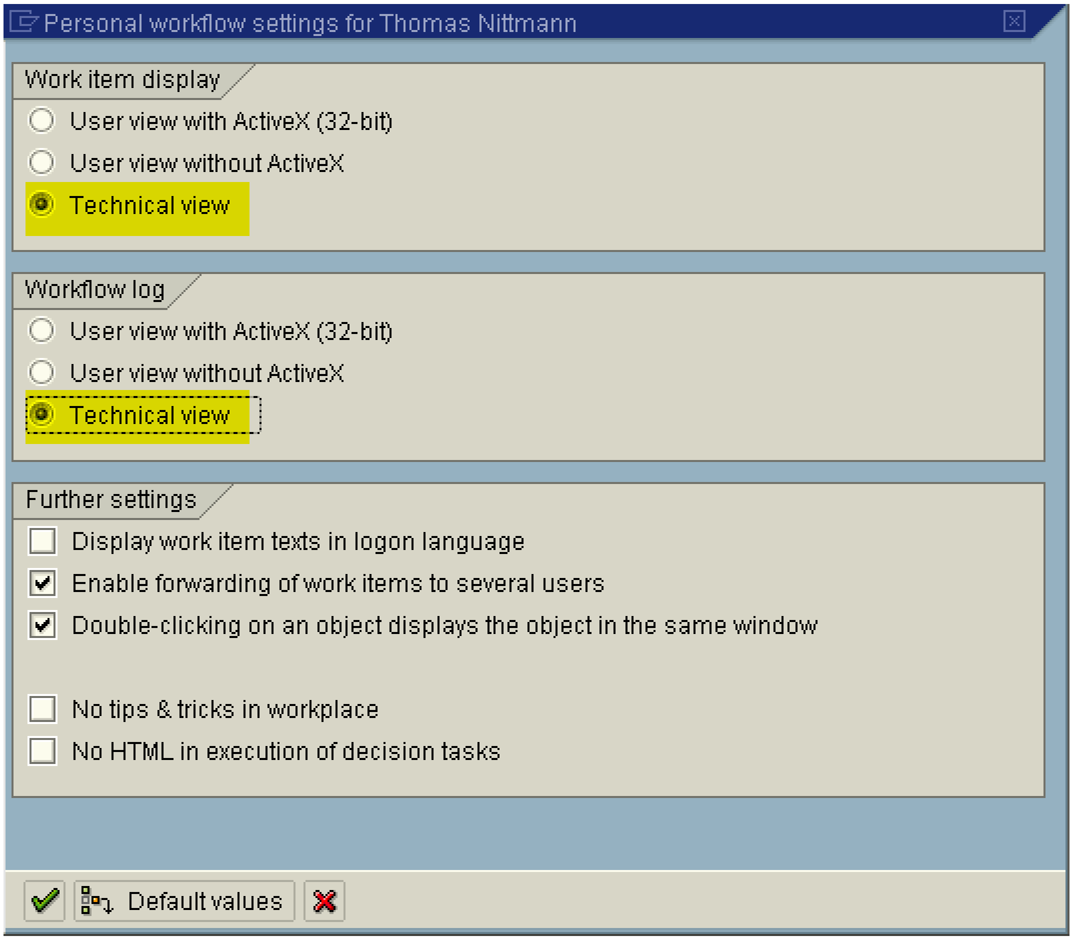
Figure 34: BANF_CONTAINER - RELEASE CODE

At this point, we see that the container element ‘RELEASECODE’ with a value of ‘EX’ is being passed.

Resolution B to Problem 2:

Review the event binding in the work item. The debug option required significant work to determine exactly where the event was being triggered for the purchase requisition release strategy. Sometimes you don’t need to find the exact portion of the code that calls the event and you simply want to review the event container parameters. To do so, use the work item display transaction SWI1.

> *Note: since we are in a relatively technical mode as we analyze workflow issues, it is recommended to switch your default workflow settings from a graphical view to a technical view.*
> *Go to the business workplace (transaction SBWP) >*
> *settings >*
> *workflow settings >*
> *personal settings >*
> *select ‘Technical view’ for ‘Work item display’ and ‘Workflow log’*



Personal workflow settings for Thomas Nittmann

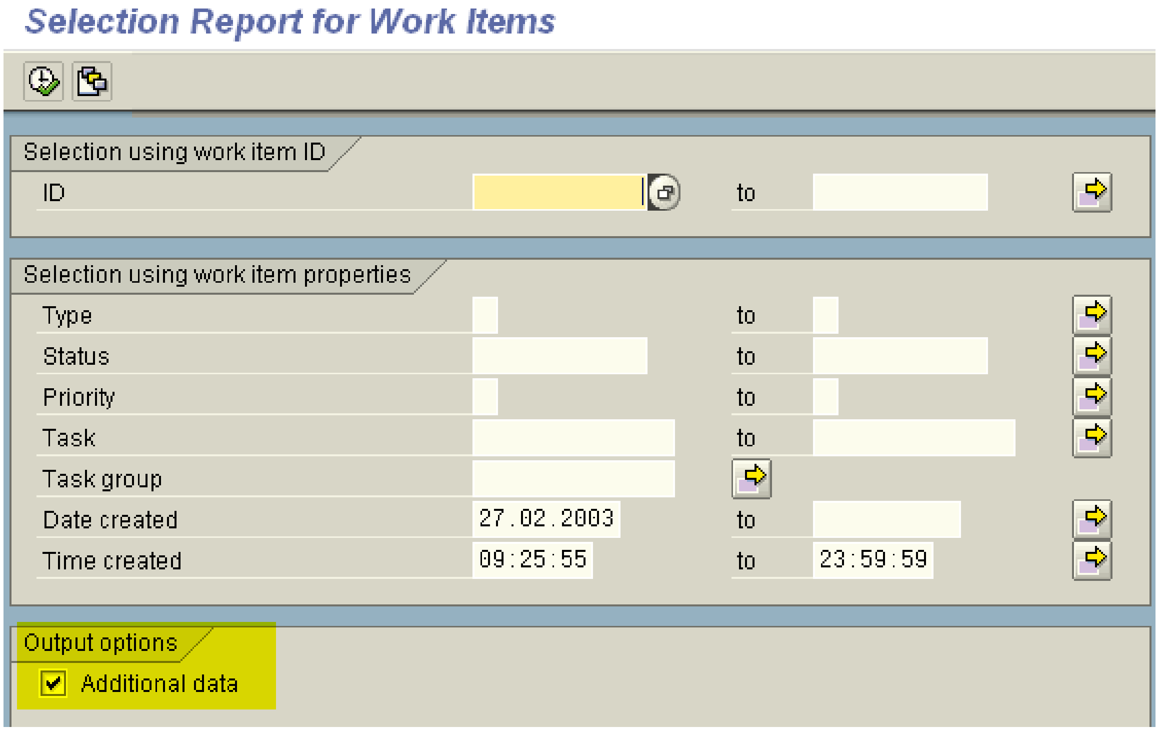
Work item display
○ User view with ActiveX (32-bit)
○ User view without ActiveX
● Technical view

Workflow log
○ User view with ActiveX (32-bit)
○ User view without ActiveX
● Technical view

Further settings
□ Display work item texts in logon language
☑ Enable forwarding of work items to several users
☑ Double-clicking on an object displays the object in the same window
□ No tips & tricks in workplace
□ No HTML in execution of decision tasks

Figure 35: Personal workflow settings for (User Name) screen

Transaction SWI1 Selection Screen (Simply Execute). Consider entering date/time to reduce selection to specific workflow created for the application (purchase requisition) document. Also select 'Additional data' to display the workflow number, agent, and other additional data.



Selection Report for Work Items

Selection using work item ID

ID ____ to ____

Selection using work item properties

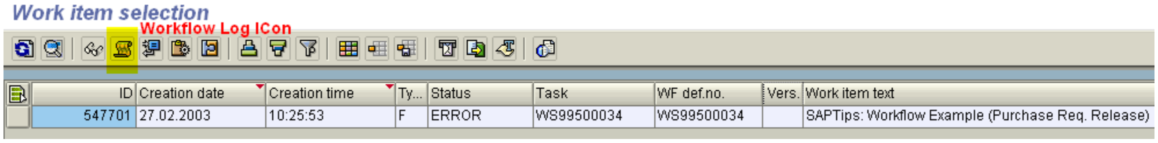
Type		to	
Status		to	
Priority		to	
Task		to	
Task group			
Date created	27.02.2003	to	
Time created	09:25:55	to	23:59:59

Output options

☑ Additional data

Figure 36: Selection Report for Work Items screen

Transaction SWI1 Output.

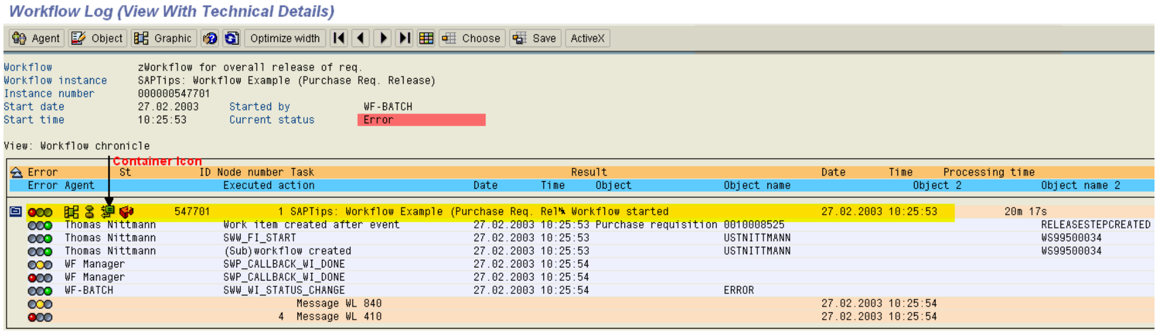


Work item selection

ID	Creation date	Creation time	Ty...	Status	Task	WF def.no.	Vers.	Work item text
547701	27.02.2003	10:25:53	F	ERROR	WS99500034	WS99500034		SAPTips: Workflow Example (Purchase Req. Release)

Figure 37: Work item selection screen with Workflow Log Icon highlighted

Display the workflow log for this item by clicking on the 'Scroll' icon (fourth icon from the left). Be sure to place the cursor on the item you want to display.

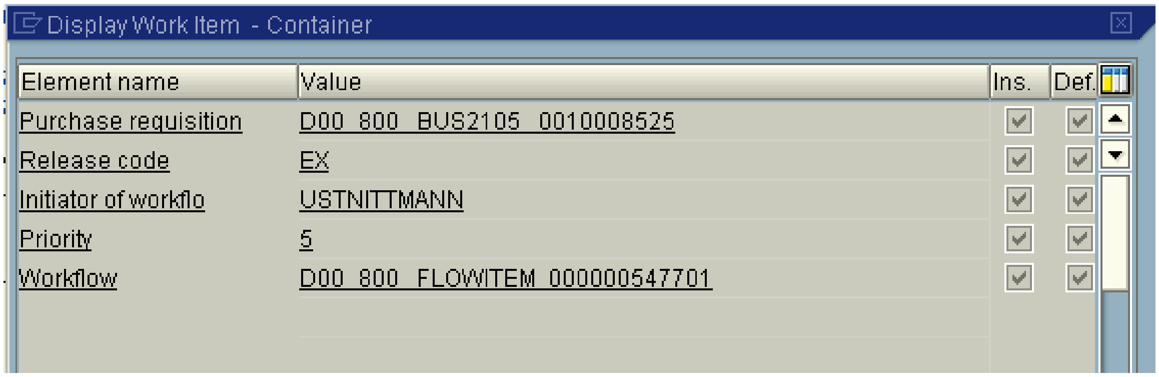


Workflow Log (View With Technical Details)

```
Workflow             zWorkflow for overall release of req.
Workflow instance    SAPTips: Workflow Example (Purchase Req. Release)
Instance number      000000547701
Start date           27.02.2003     Started by        WF-BATCH
Start time           10:25:53       Current status    Error

View: Workflow chronicle

 Error                St          ID Node number Task                        Result                          Date       Time      Processing time
 Error Agent                            Executed action            Date       Time     Object              Object name            Object 2           Object name 2

                      547701         1 SAPTips: Workflow Example (Purchase Req. Rel Workflow started         27.02.2003 10:25:53       20m 17s
       Thomas Nittmann          Work item created after event      27.02.2003 10:25:53 Purchase requisition 0010008525                  RELEASESTEPCREATED
       Thomas Nittmann          SWW_FI_START                       27.02.2003 10:25:53                      USTNITTMANN                 WS99500034
       Thomas Nittmann          (Sub)workflow created              27.02.2003 10:25:53                      USTNITTMANN                 WS99500034
       WF Manager               SWP_CALLBACK_WI_DONE               27.02.2003 10:25:54
       WF Manager               SWP_CALLBACK_WI_DONE               27.02.2003 10:25:54
       WF-BATCH                 SWW_WI_STATUS_CHANGE               27.02.2003 10:25:54                      ERROR
                                          Message WL 840                                                  27.02.2003 10:25:54
                                       4  Message WL 410                                                  27.02.2003 10:25:54
```

Figure 38: Workflow Log (View With Technical Details) screen

We see that our workflow has still errored. The log shows you that the workflow (WS template) started but no single step tasks (TS) have been evoked, so we have an issue before we even get to the first single step task. We will resolve the error in the next couple of sessions, but first let's review the event parameters. Click on the WS container icon (displayed above).



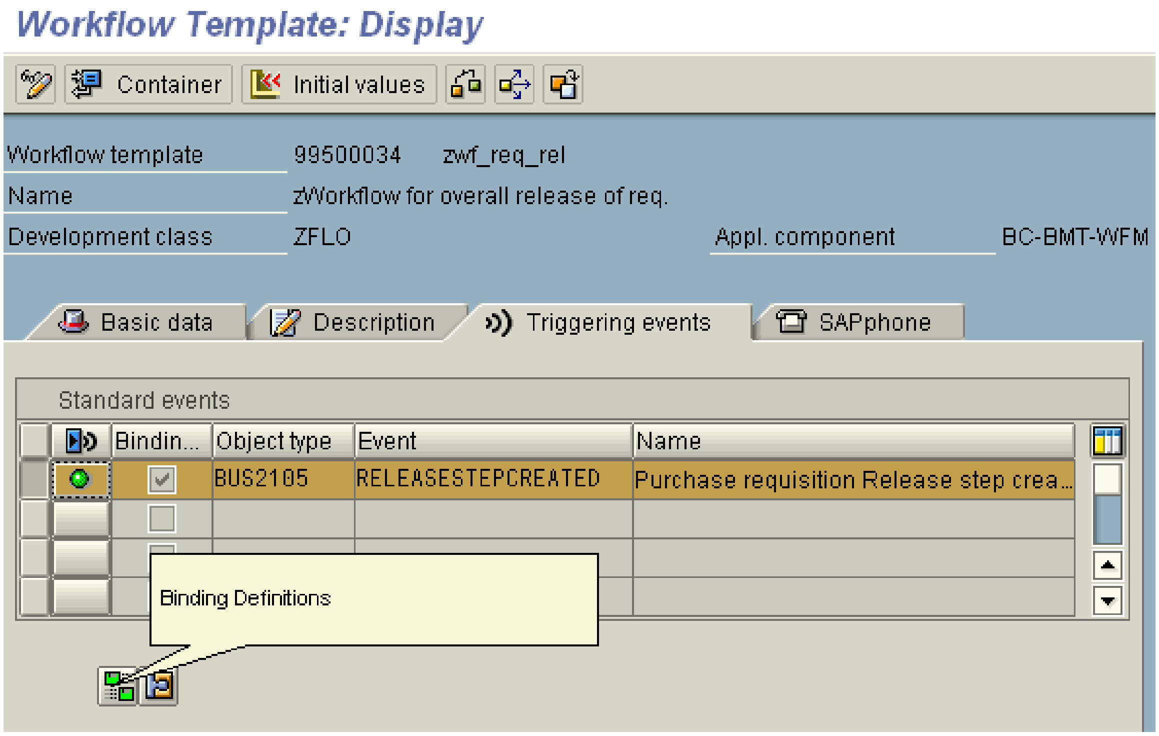
Display Work Item - Container

Element name	Value	Ins.	Def.
Purchase requisition	D00 800 BUS2105 0010008525	☑	☑
Release code	EX	☑	☑
Initiator of workflo	USTNITTMANN	☑	☑
Priority	5	☑	☑
Workflow	D00 800 FLOWITEM 000000547701	☑	☑

Figure 39: Display Work Item - Container screen

The container displays the event parameters passed by the purchase requisition application to the workflow.

> *Note: if you expect additional event parameters to be passed, check the workflow definition via transaction PFTC (enter the workflow template > select 'Triggering events' tab > select binding definition button) to make sure the binding has been set-up correctly.* ***Most often, custom workflow event parameter binding issues are a result of incorrect binding from the event to the workflow task.***

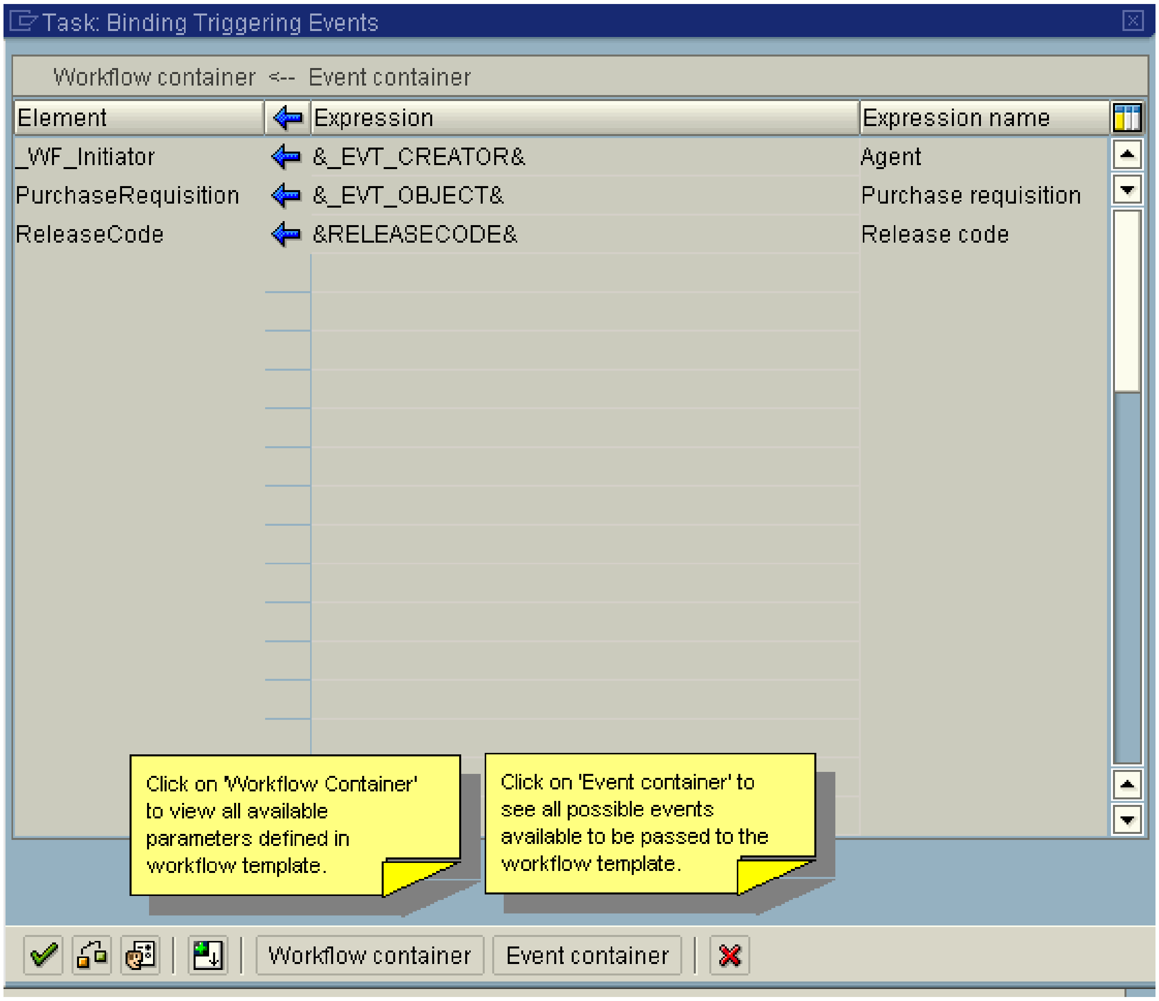


Workflow Template: Display

Workflow template	99500034 zwf_req_rel		
Name	zWorkflow for overall release of req.		
Development class	ZFLO	Appl. component	BC-BMT-WFM

	Bindin...	Object type	Event	Name
●	☑	BUS2105	RELEASESTEPCREATED	Purchase requisition Release step crea...
	□			

Figure 40: Workflow Template: Display screen displaying Triggering events tab.



Task: Binding Triggering Events
Workflow container <-- Event container

Element	⇐	Expression	Expression name
_WF_Initiator	⇐	&_EVT_CREATOR&	Agent
PurchaseRequisition	⇐	&_EVT_OBJECT&	Purchase requisition
ReleaseCode	⇐	&RELEASECODE&	Release code

Click on 'Workflow Container' to view all available parameters defined in workflow template.

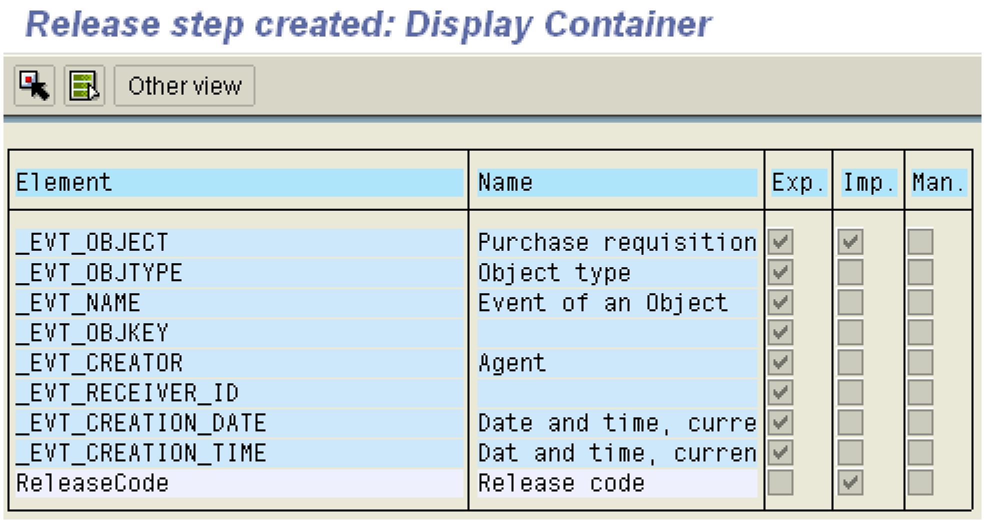
Click on 'Event container' to see all possible events available to be passed to the workflow template.

Workflow container
Event container

Figure 41: Task Binding Triggering Events screen

By clicking on the Event container, we see all the element parameters that the standard SAP Purchase Requisition application intended to pass to the workflow. After debugging the code where the event is triggered and reviewing the event container, we see that it would be nearly impossible to add other event parameters to the workflow without modifying SAP source code OR perhaps looking for a user exit.

> Note: At this point, we have analyzed the triggering mechanism of the purchase requisition release workflow. Since the trigger mechanism in this example is provided by SAP in the primary Purchase Requisition application, there is no ideal way to add additional event parameters. **However, this does not mean we do not have other options. We can create additional (custom) object attributes or methods to pull data into the workflow.**



Release step created: Display Container

Element	Name	Exp.	Imp.	Man.
_EVT_OBJECT	Purchase requisition	☑	☑	□
_EVT_OBJTYPE	Object type	☑	□	□
_EVT_NAME	Event of an Object	☑	□	□
_EVT_OBJKEY		☑	□	□
_EVT_CREATOR	Agent	☑	□	□
_EVT_RECEIVER_ID		☑	□	□
_EVT_CREATION_DATE	Date and time, curre	☑	□	□
_EVT_CREATION_TIME	Dat and time, curren	☑	□	□
ReleaseCode	Release code	□	☑	□

Figure 42: Release step created: Display Container screen

Additional Event Information:

The initial part of any workflow is some sort of triggering mechanism (Event) to start the entire process in motion. Several standard and custom event-triggering options are available to start the workflow:

- The application (original SAP code) triggers workflow based on configuration settings. This is how the purchase requisition release workflow that we have been reviewing is enabled.

- Status Management, primarily used in the PP, PM, and SM modules, triggers the workflow based on either a **system** status or **user** status change. Why isn't status management available in other modules? Because the original developers of the other modules didn't incorporate status management into the applications.

- A custom ABAP report (and/or function module) can be written that calls the standard workflow event creation function modules.

- A user exit (and/or field exit) may be activated and coded to trigger workflow.

- A stand alone 'Start Transaction' can be generated via a standard tool provided by SAP. To start workflow, the end user calls the transaction, fills in the selection screen, and then executes the transaction to trigger the workflow.

- Business Transaction Events (BTEs) are available for FI/CO, SD, MM and other modules. BTEs must be activated via configuration (linking the BTE with the event function module) and/or development, and are standard hooks into the application that SAP provides. When the event is triggered from the application, a function module calls a BOR (Business Object Repository) event. SAP provides a function module for all BTEs that can be linked to an event.

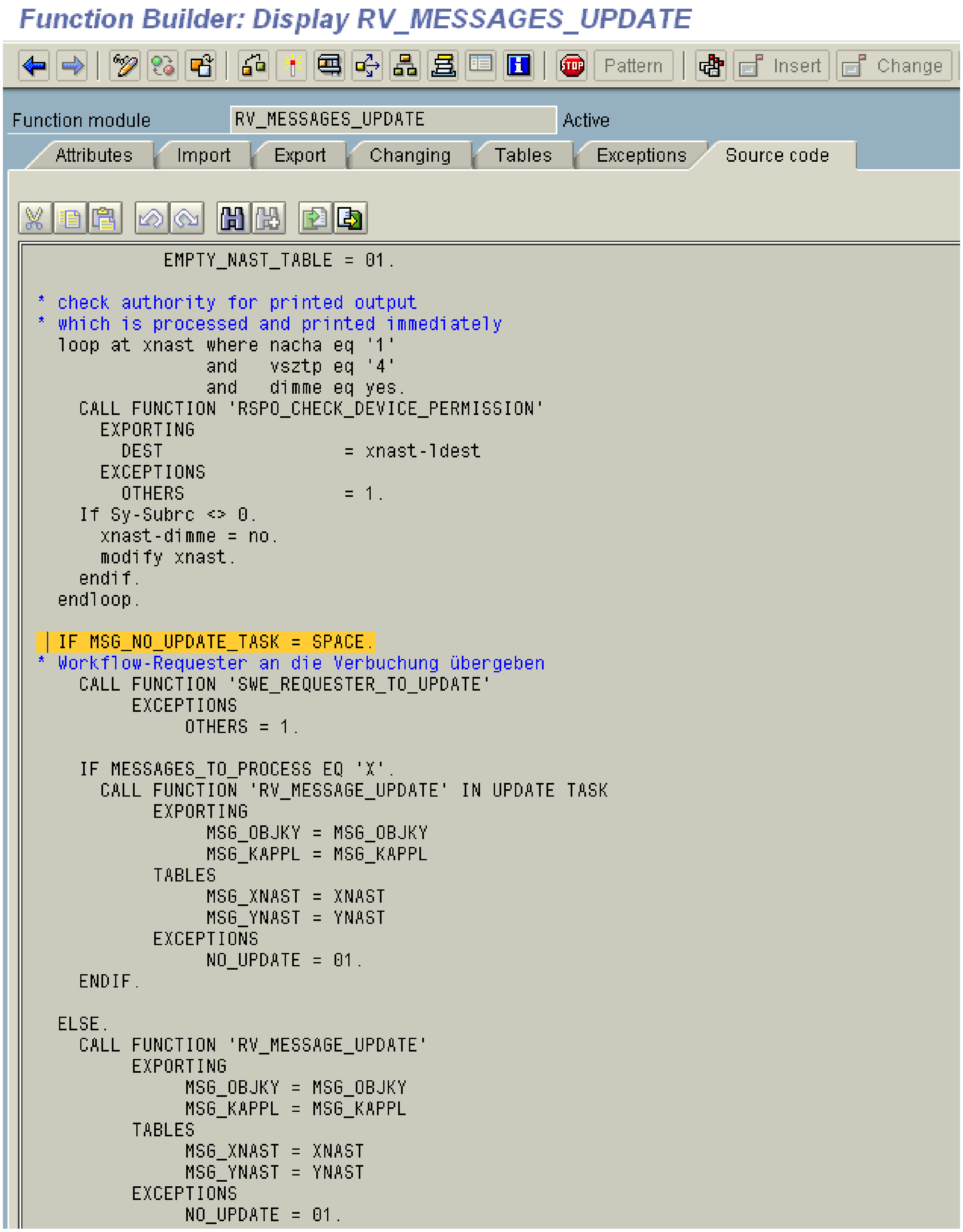
- Change documents. If a change document object is created for the application, you can link any create/change/delete change document objects to a workflow event.

- Message control (output determination) may be used for workflows in a similar fashion to how it's used for printouts, faxes, etc. Message control is used in the MM and SD modules. Since message control is run in the background, it is often difficult to debug the event creation process. We sill show an example of message control in the Purchase **Order** application, because the Purchase **Requisition** application does **not** use message control. Here's a solution to assist in debugging message control (this works for fax, print, workflow, and all output types available in message control):

  Note: Message Control, Output Determination, Output Control are different names that are generically called 'Condition Technique', i.e., don't let SAP confuse you - they all refer to the same process.

  a. Message control (aka output determination) uses function module 'RV_MESSAGES_UPDATE' to generate output **if** the configuration is set up properly.

  b. Set a break point at: 'IF MSG_NO_UPDATE_TASK = SPACE.' statement in the function module 'RV_MESSAGE_UPDATE' (line 95 in R/3 46C).



Function Builder: Display RV_MESSAGES_UPDATE

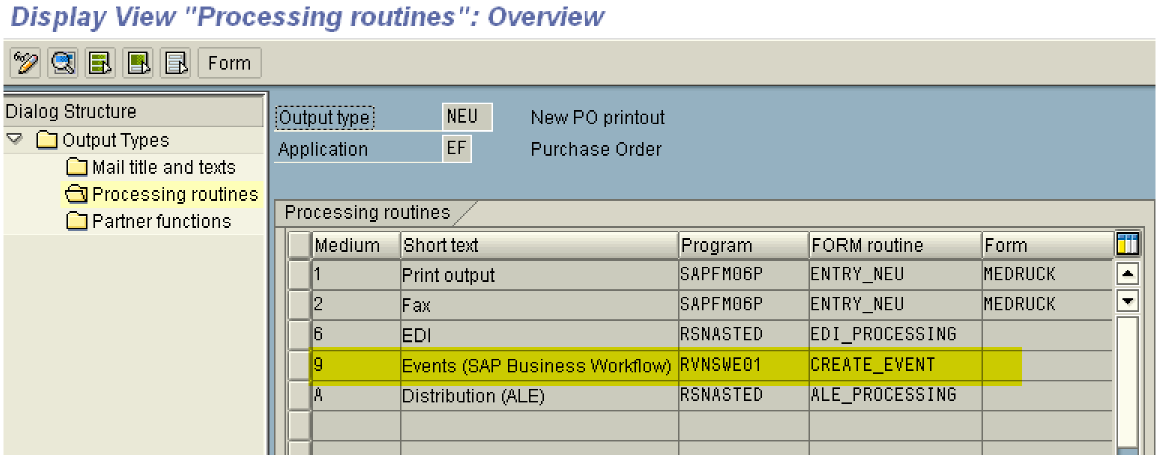
Function module RV_MESSAGES_UPDATE Active

```
            EMPTY_NAST_TABLE = 01.

* check authority for printed output
* which is processed and printed immediately
  loop at xnast where nacha eq '1'
                and   vsztp eq '4'
                and   dimme eq yes.
    CALL FUNCTION 'RSPO_CHECK_DEVICE_PERMISSION'
      EXPORTING
        DEST                     = xnast-ldest
      EXCEPTIONS
        OTHERS                   = 1.
    If Sy-Subrc <> 0.
      xnast-dimme = no.
      modify xnast.
    endif.
  endloop.

  IF MSG_NO_UPDATE_TASK = SPACE.
* Workflow-Requester an die Verbuchung übergeben
    CALL FUNCTION 'SWE_REQUESTER_TO_UPDATE'
         EXCEPTIONS
              OTHERS = 1.

    IF MESSAGES_TO_PROCESS EQ 'X'.
      CALL FUNCTION 'RV_MESSAGE_UPDATE' IN UPDATE TASK
           EXPORTING
                MSG_OBJKY = MSG_OBJKY
                MSG_KAPPL = MSG_KAPPL
           TABLES
                MSG_XNAST = XNAST
                MSG_YNAST = YNAST
           EXCEPTIONS
                NO_UPDATE = 01.
    ENDIF.

  ELSE.
    CALL FUNCTION 'RV_MESSAGE_UPDATE'
         EXPORTING
              MSG_OBJKY = MSG_OBJKY
              MSG_KAPPL = MSG_KAPPL
         TABLES
              MSG_XNAST = XNAST
              MSG_YNAST = YNAST
         EXCEPTIONS
              NO_UPDATE = 01.
```

Figure 43: Function Builder: Display RV_MESSAGES_UPDATE screen displaying Source code tab.

Note: Several steps are required to set up message control for workflow. In this example, Output Medium '9' (Events SAP Business Workflow) was set up for output type 'NEU' (Purchase Orders) via transaction 'M/34'. Output type '9' is used to create an event; the other option is to set up output medium 'T' and trigger a task directly. Program 'RVNSWE01' and form routine 'CREATE_EVENT' are used to generate the event. Custom sub-type ZBUS2012 was created, delegated to ZBUS2012, and custom event 'zMessageWFEvent' was created. Template workflow (WS99500035) was created with a simple task (TS99500021), with a method that has 'break-point' in it. Lastly, a condition record (MN04) was created for document type 'NB' to automatically generate the workflow output when the purchase order is saved.

c. Determine the form routine used by message control to create the workflow event. Message control (output determination) for purchase orders is set up in the IMG under MM > Purchasing > Messages > Output Control. We simply need to review the message type 'NEU' for purchase order output to find the form-driving program and form routine.



Display View "Processing routines": Overview

Output type: NEU — New PO printout
Application: EF — Purchase Order

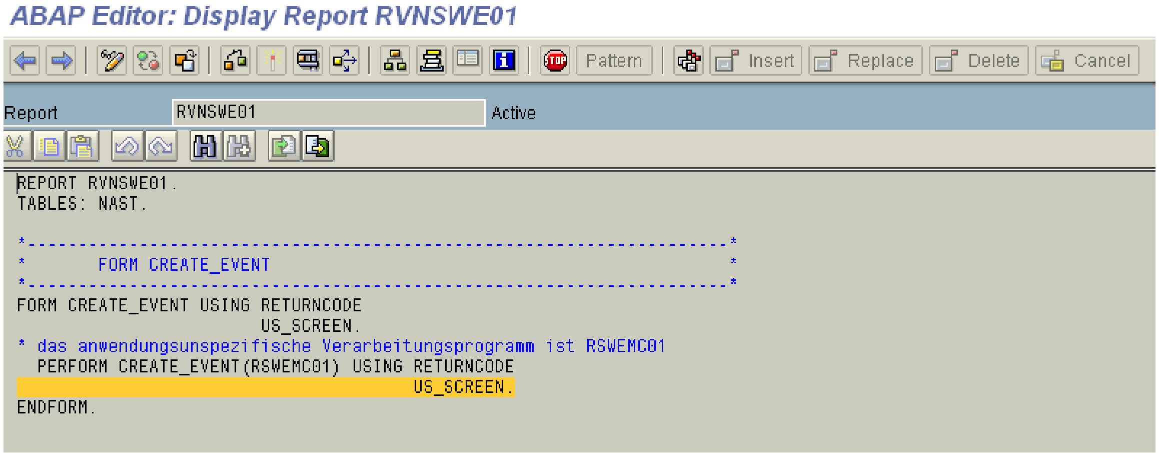
Processing routines

Medium	Short text	Program	FORM routine	Form
1	Print output	SAPFM06P	ENTRY_NEU	MEDRUCK
2	Fax	SAPFM06P	ENTRY_NEU	MEDRUCK
6	EDI	RSNASTED	EDI_PROCESSING	
9	Events (SAP Business Workflow)	RVNSWE01	CREATE_EVENT	
A	Distribution (ALE)	RSNASTED	ALE_PROCESSING	

Figure 44: Display View "Processing routines": Overview screen

Hint: SAP uses the condition technique for pricing, output, account determination, and many other functions. Each application has a different transaction to configure the condition technique. However, you can use transaction VK01 to select any condition technique you would like to maintain.

d. Set a breakpoint in the form routine 'CREATE_EVENT'.



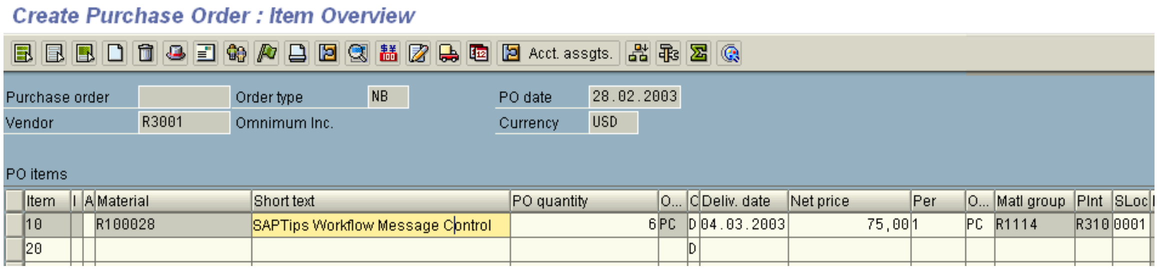
ABAP Editor: Display Report RVNSWE01

Report RVNSWE01 Active

```
REPORT RVNSWE01.
TABLES: NAST.

*---------------------------------------------------------------------*
*       FORM CREATE_EVENT                                             *
*---------------------------------------------------------------------*
FORM CREATE_EVENT USING RETURNCODE
                        US_SCREEN.
* das anwendungsunspezifische Verarbeitungsprogramm ist RSWEMC01
  PERFORM CREATE_EVENT(RSWEMC01) USING RETURNCODE
                                       US_SCREEN.
ENDFORM.
```

Figure 45: ABAP Editor: Display Report screen

e. Go to the application that uses message control and create an application document, e.g., transaction ‘ME21’ for purchase orders (the purchase requisition application does **not** use message control). You need to make certain a condition record (MN04) is set up to automatically propose the workflow output.



Create Purchase Order : Item Overview

Purchase order Order type NB PO date 28.02.2003
Vendor R3001 Omnimum Inc. Currency USD

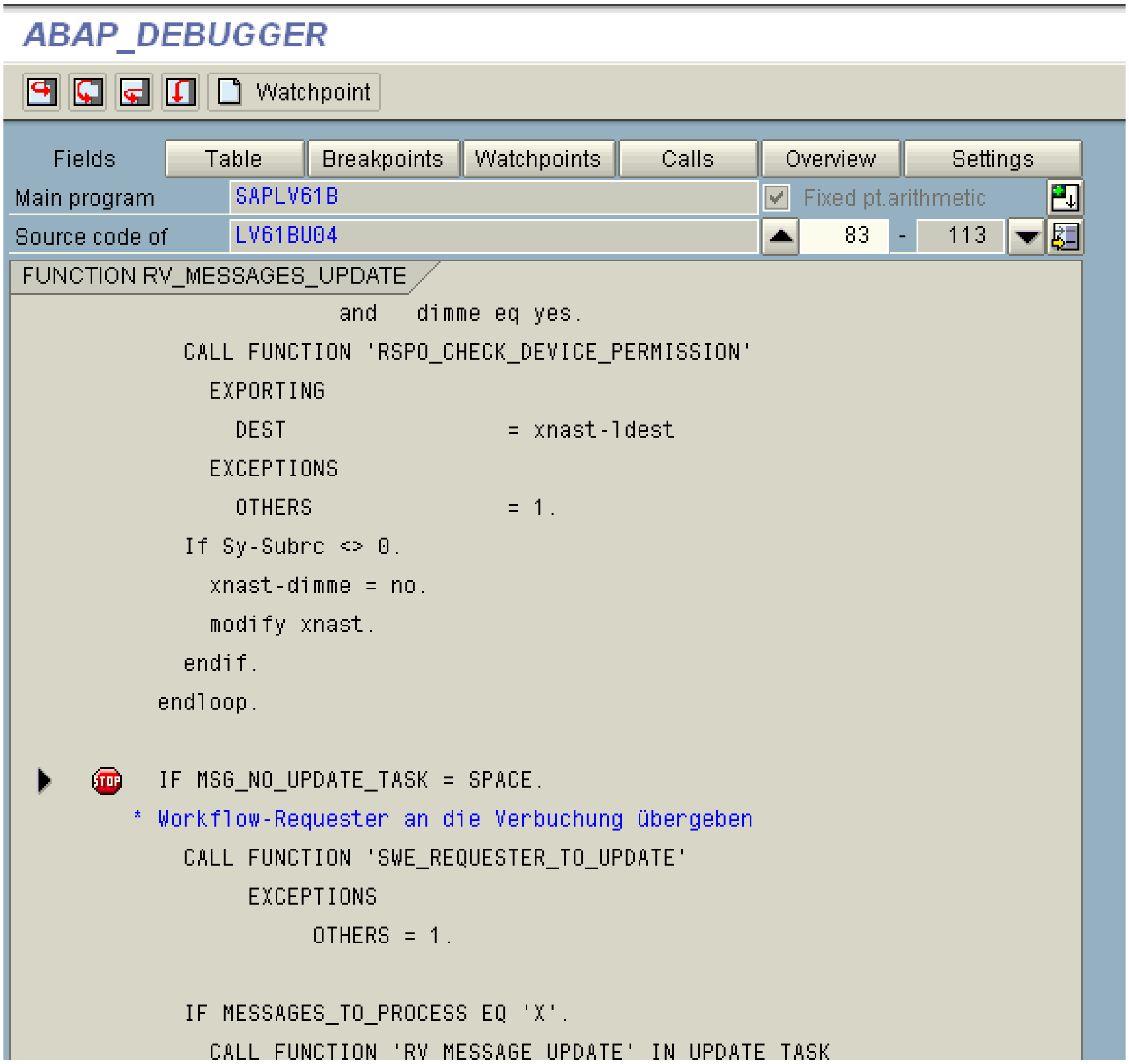
PO items

Item	I	A	Material	Short text	PO quantity	O...	C	Deliv. date	Net price	Per	O...	Matl group	Plnt	SLoc
10			R100028	SAPTips Workflow Message Control	6	PC	D	04.03.2003	75,00	1	PC	R1114	R310	0001
20							D							

Figure 46: Create Purchase Order: Item Overview screen

f. When you save the purchase order, a debug session starts for function module ‘RV_MESSAGES_UPDATE’.

Note: The debug session will start only if message control automatically proposed output. If a debug session did not open, then manually add the output type (Headers > Messages) and save.



Main program SAPLV61B

Source code of LV61BU04

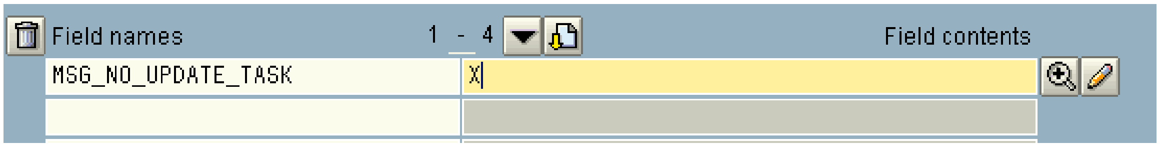
83 - 113

FUNCTION RV_MESSAGES_UPDATE

```
                     and   dimme eq yes.
           CALL FUNCTION 'RSPO_CHECK_DEVICE_PERMISSION'
             EXPORTING
               DEST                    = xnast-ldest
             EXCEPTIONS
               OTHERS                  = 1.
           If Sy-Subrc <> 0.
             xnast-dimme = no.
             modify xnast.
           endif.
         endloop.

         IF MSG_NO_UPDATE_TASK = SPACE.
       * Workflow-Requester an die Verbuchung übergeben
           CALL FUNCTION 'SWE_REQUESTER_TO_UPDATE'
                EXCEPTIONS
                     OTHERS = 1.

           IF MESSAGES_TO_PROCESS EQ 'X'.
             CALL FUNCTION 'RV_MESSAGE_UPDATE' IN UPDATE TASK
```

Figure 47: ABAP_DEBUGGER screen

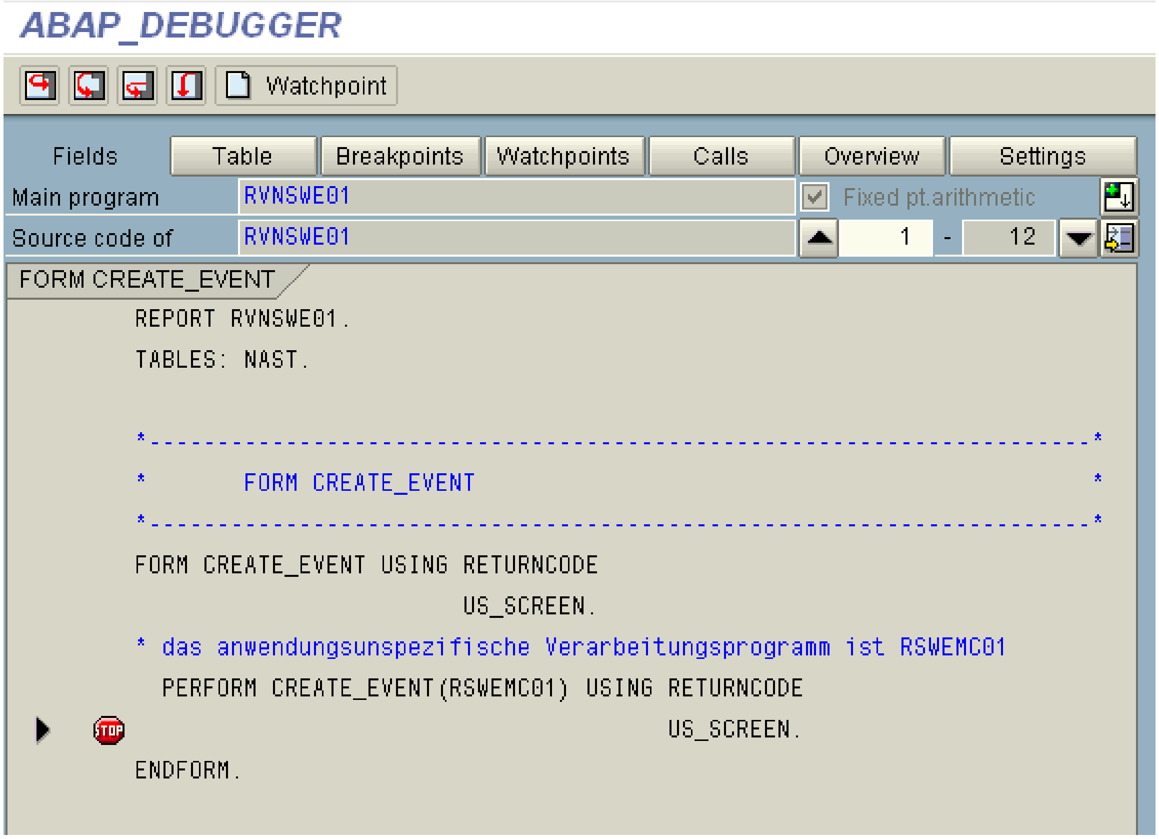
g. Set variable ‘MSG_NO_UPDATE_TASK’ = ‘X’ and continue (be sure to change it by clicking on the pencil).



Field names	Field contents
MSG_NO_UPDATE_TASK	X

Figure 48: Field names screen

h. When you continue, SAP will break at the form routine for the creation of the event.



Main program: RVNSWE01

Source code of: RVNSWE01

☑ Fixed pt.arithmetic

1 - 12

FORM CREATE_EVENT

```
REPORT RVNSWE01.
TABLES: NAST.

*----------------------------------------------------------------------*
*       FORM CREATE_EVENT                                              *
*----------------------------------------------------------------------*
FORM CREATE_EVENT USING RETURNCODE
                        US_SCREEN.
* das anwendungsunspezifische Verarbeitungsprogramm ist RSWEMC01
  PERFORM CREATE_EVENT(RSWEMC01) USING RETURNCODE
                                       US_SCREEN.
ENDFORM.
```

Figure 49: ABAP_DEBUGGER screen

i. Later we will review how to continue debugging at the workflow task level.

Agent Resolution Debugging

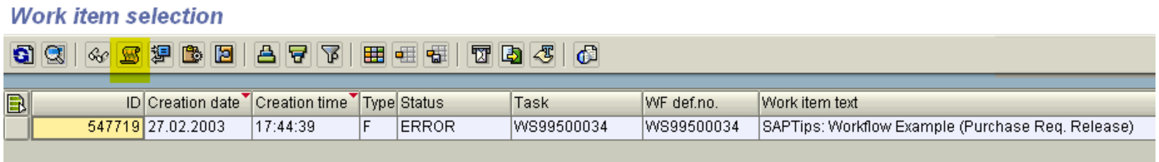
All workflow tasks that generate work items (foreground tasks) need to be assigned to an individual or group of agents.

Problem 3: Agent Resolution:

A workflow work item is created but it is not assigned to an agent.

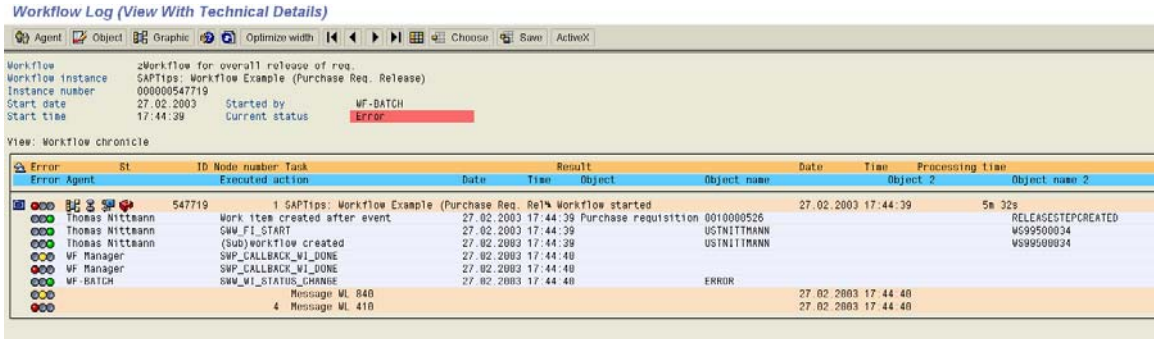
Resolution to Problem 3:

First, review the work item workflow log via transaction SWI1 (click on workflow log 'Scroll' Icon).



ID	Creation date	Creation time	Type	Status	Task	WF def.no.	Work item text
547719	27.02.2003	17:44:39	F	ERROR	WS99500034	WS99500034	SAPTips: Workflow Example (Purchase Req. Release)

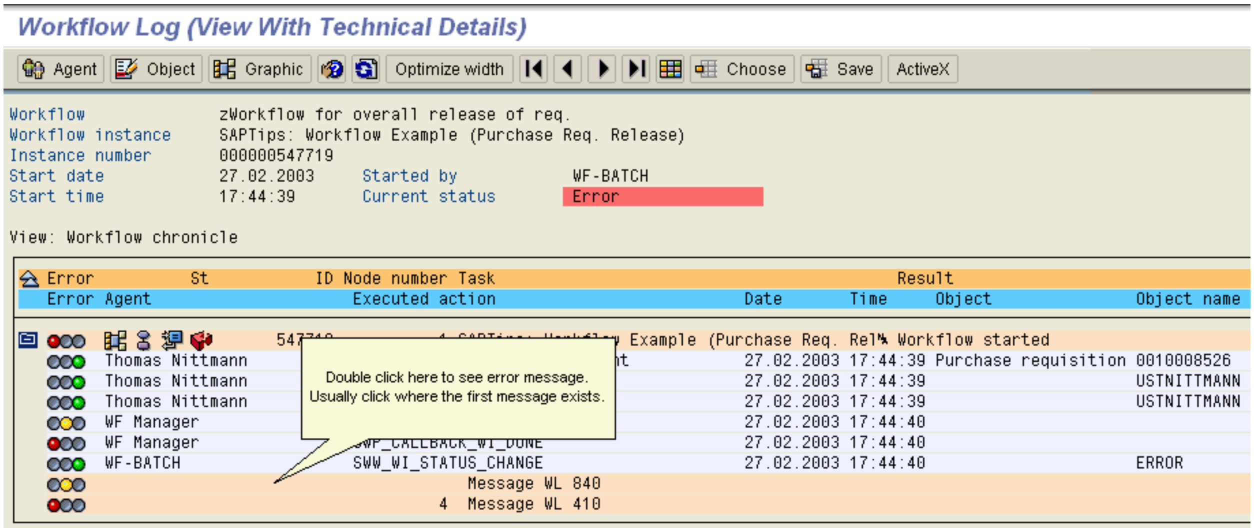
Figure 50: Work item selection screen



```
Workflow            zWorkflow for overall release of req.
Workflow instance   SAPTips: Workflow Example (Purchase Req. Release)
Instance number     000000547719
Start date          27.02.2003    Started by       WF-BATCH
Start time          17:44:39      Current status   Error

View: Workflow chronicle

Error          St          ID Node number Task                                   Result                                Date       Time     Processing time
   Error Agent                      Executed action          Date       Time     Object            Object name              Object 2        Object name 2

                         547719          1 SAPTips: Workflow Example (Purchase Req. Rel Workflow started               27.02.2003 17:44:39        5m 32s
   Thomas Nittmann                  Work item created after event  27.02.2003 17:44:39 Purchase requisition 0010000526                      RELEASESTEPCREATED
   Thomas Nittmann                  SWW_FI_START             27.02.2003 17:44:39                   USTNITTMANN                              WS99500034
   Thomas Nittmann                  (Sub)workflow created    27.02.2003 17:44:39                   USTNITTMANN                              WS99500034
   WF Manager                       SWP_CALLBACK_WI_DONE     27.02.2003 17:44:40
   WF Manager                       SWP_CALLBACK_WI_DONE     27.02.2003 17:44:40
   WF-BATCH                         SWW_WI_STATUS_CHANGE     27.02.2003 17:44:40                   ERROR
                                          Message WL 840                                                               27.02.2003 17:44:40
                                        4 Message WL 410                                                               27.02.2003 17:44:40
```

Figure 51: Workflow Log (View With Technical Details) screen

Review the log and pinpoint where the error occurs. Double click where you see the first 'Message …' occur.

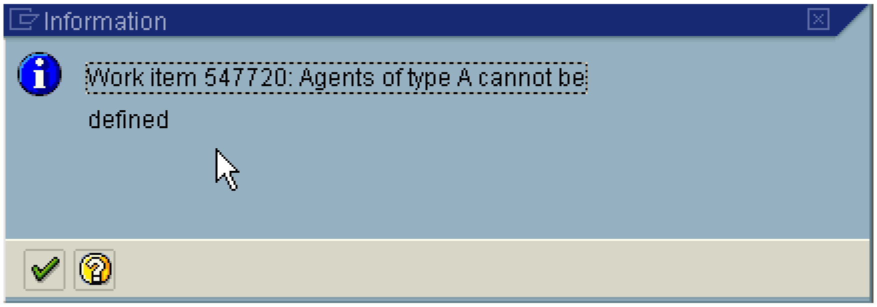


```
Workflow            zWorkflow for overall release of req.
Workflow instance   SAPTips: Workflow Example (Purchase Req. Release)
Instance number     000000547719
Start date          27.02.2003    Started by       WF-BATCH
Start time          17:44:39      Current status   Error

View: Workflow chronicle

Error          St          ID Node number Task                          Result
   Error Agent                      Executed action          Date       Time     Object            Object name

                         547                       y Example (Purchase Req. Rel Workflow started
   Thomas Nittmann                                 t         27.02.2003 17:44:39 Purchase requisition 0010008526
   Thomas Nittmann                                           27.02.2003 17:44:39                   USTNITTMANN
   Thomas Nittmann                                           27.02.2003 17:44:39                   USTNITTMANN
   WF Manager                                                27.02.2003 17:44:40
   WF Manager                       WP_CALLBACK_WI_DONE      27.02.2003 17:44:40
   WF-BATCH                         SWW_WI_STATUS_CHANGE     27.02.2003 17:44:40                   ERROR
                                          Message WL 840
                                        4 Message WL 410
```

Double click here to see error message.
Usually click where the first message exists.

Figure 52: Workflow Log (View With Technical Details) screen indicating where to click

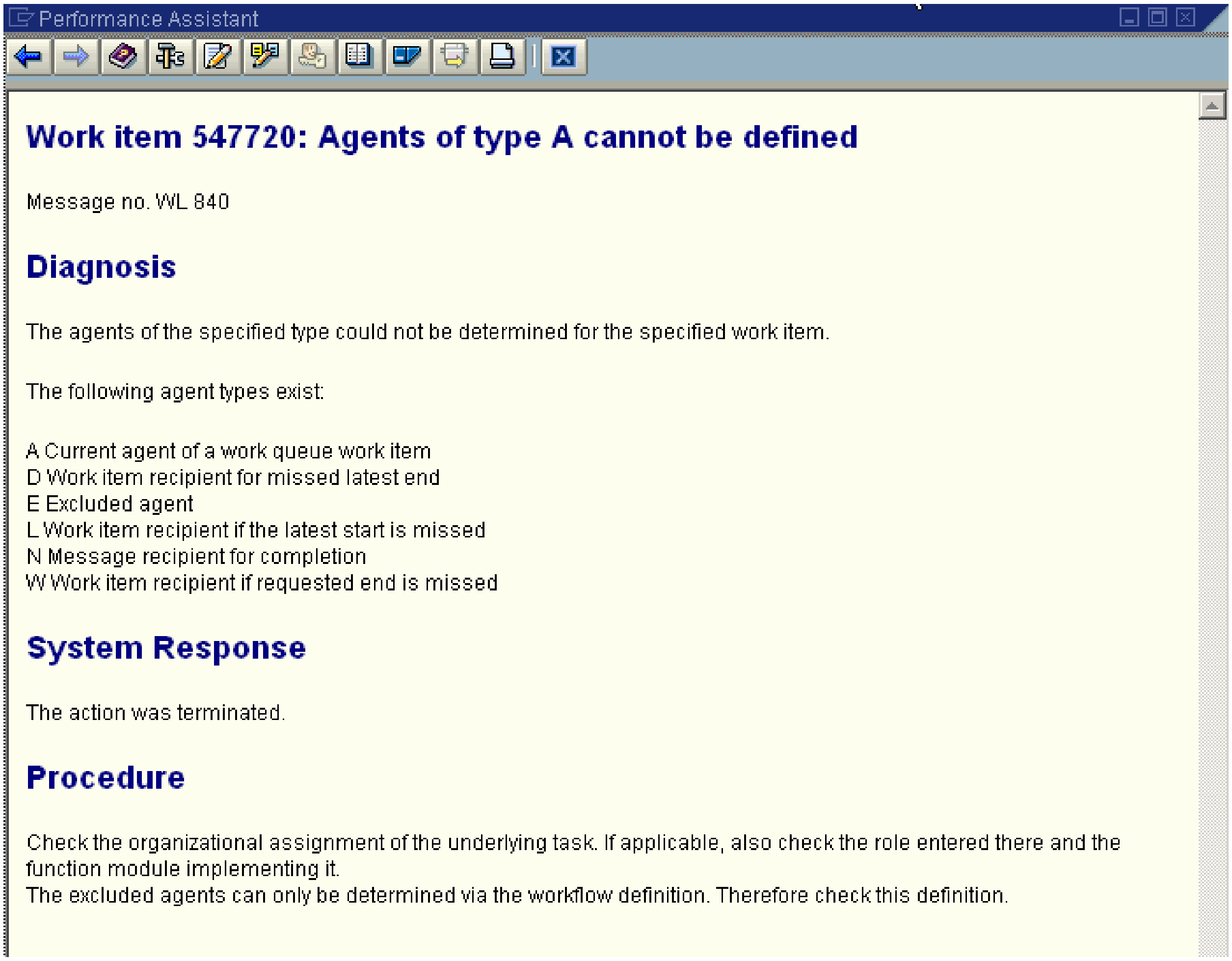
Click the long text ('?' icon)



Work item 547720: Agents of type A cannot be defined

Figure 53: Information dialog box

The long text of the issues describes that an agent could not be determined.



Work item 547720: Agents of type A cannot be defined

Message no. WL 840

Diagnosis

The agents of the specified type could not be determined for the specified work item.

The following agent types exist:

A Current agent of a work queue work item
D Work item recipient for missed latest end
E Excluded agent
L Work item recipient if the latest start is missed
N Message recipient for completion
W Work item recipient if requested end is missed

System Response

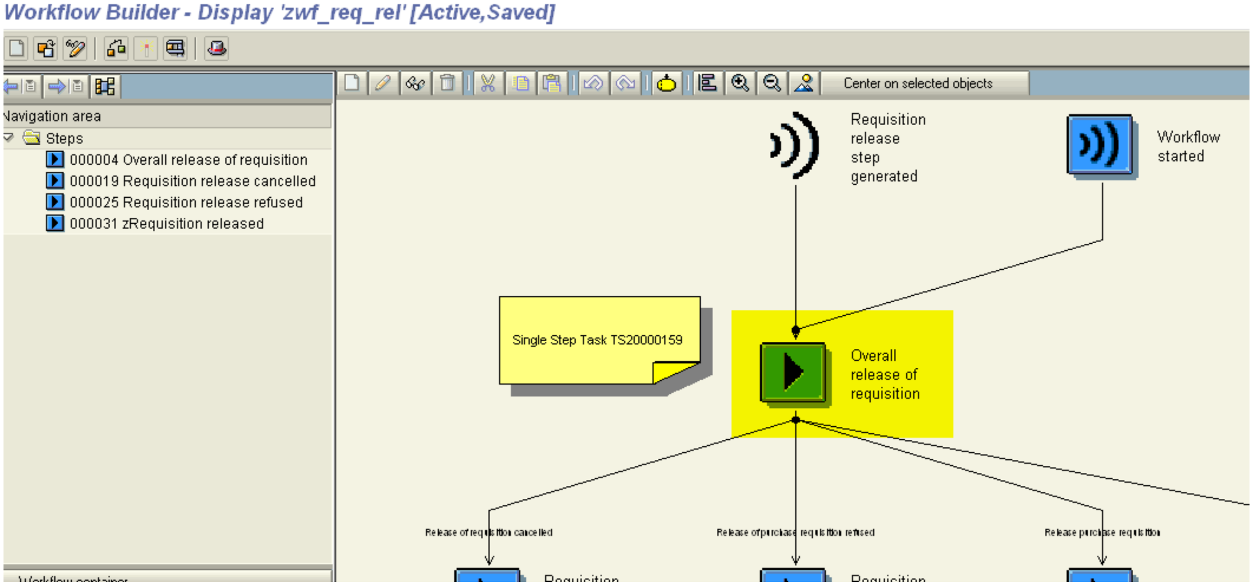
The action was terminated.

Procedure

Check the organizational assignment of the underlying task. If applicable, also check the role entered there and the function module implementing it.
The excluded agents can only be determined via the workflow definition. Therefore check this definition.

Figure 54: Performance Assistance screen

We need to review the standard SAP workflow agent resolutions before we continue. The first single step task (TS) step of the purchase requisition workflow (WS) release strategy is ‘000004’ (TS20000159).



Workflow Builder - Display 'zwf_req_rel' [Active,Saved]

Navigation area

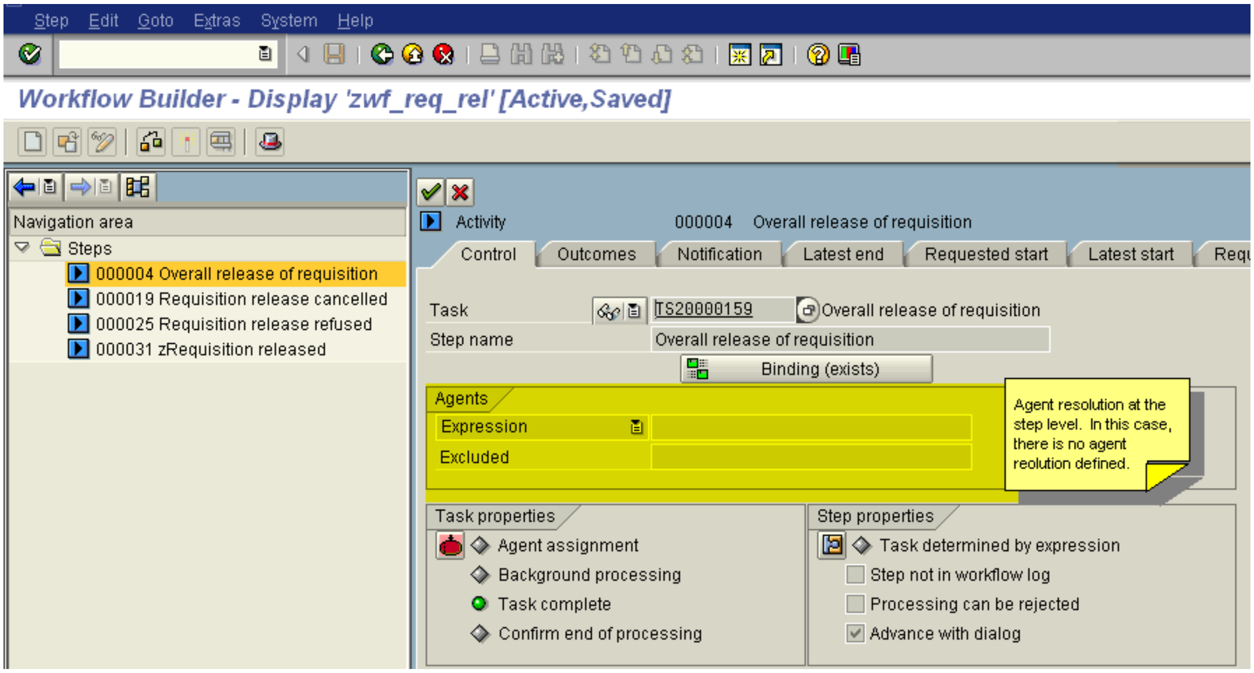
Steps

- 000004 Overall release of requisition
- 000019 Requisition release cancelled
- 000025 Requisition release refused
- 000031 zRequisition released

Center on selected objects
Requisition release step generated
Workflow started
Single Step Task TS20000159
Overall release of requisition
Release of requisition cancelled
Release of purchase requisition refused
Release purchase requisition
Requisition
Requisition
Workflow container

Figure 55: Workflow Builder - Display screen

You can define agent resolution at the step (000004) level. Any agent resolution defined at this level will be superceded if agent resolution is at the single step (TS) task level.



Step Edit Goto Extras System Help

Workflow Builder - Display 'zwf_req_rel' [Active,Saved]

Navigation area

Steps

- 000004 Overall release of requisition
- 000019 Requisition release cancelled
- 000025 Requisition release refused
- 000031 zRequisition released

Activity 000004 Overall release of requisition

Control | Outcomes | Notification | Latest end | Requested start | Latest start | Requ

Task: TS20000159 Overall release of requisition

Step name: Overall release of requisition

Binding (exists)

Agents

Expression:

Excluded:

Agent resolution at the step level. In this case, there is no agent reolution defined.

Task properties

- Agent assignment
- Background processing
- Task complete
- Confirm end of processing

Step properties

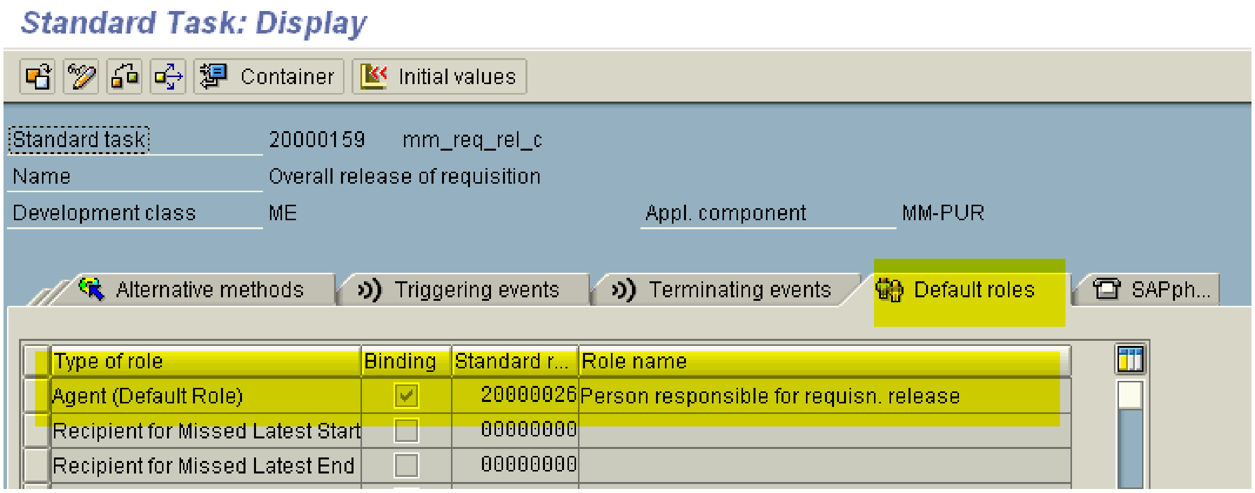
- Task determined by expression
- □ Step not in workflow log
- □ Processing can be rejected
- ☑ Advance with dialog

Figure 56: Workflow Builder - Display screen showing no agent resolution is defined.

Double click on task TS20000159 and select the 'default role' tab. A standard role (AC20000026) is defined. This is used to determine the agent for the overall release. Here's where our problem occurs. We need to figure out why an error is recurring here.

Note: All of the workflow objects usually have prefixes that are most often NOT displayed on screen. Workflow objects are stored in the SAP HR-PD tables with the prefixes. Here's an overview of the most common prefixes:

'TG' – Task Group
'WF' – Custom multi-step task (obsolete)
'WS' – Multi-step workflow task
'TS' – Single-step workflow task
'T' – Customer Single-step task (obsolete)
'AC' – Role
'C' – Job
'O' – Organizational Unit
'S' - Position



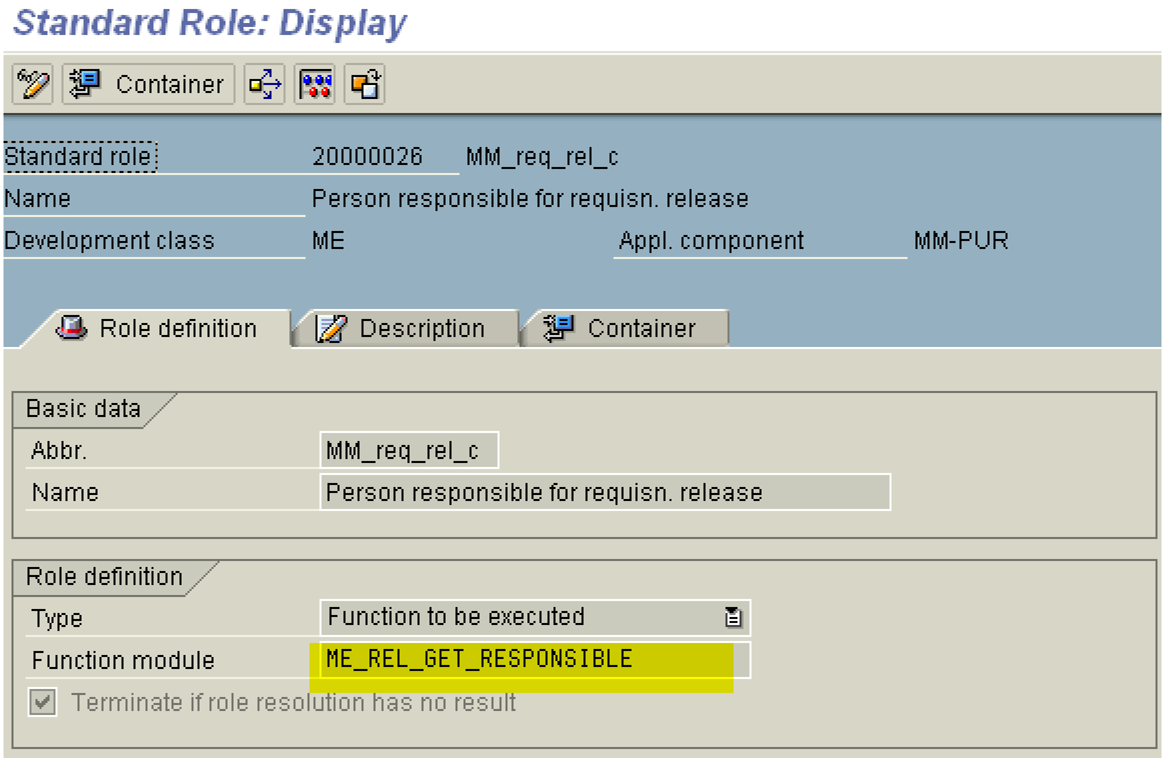
Standard Task: Display

Standard task 20000159 mm_req_rel_c
Name Overall release of requisition
Development class ME
Appl. component MM-PUR

Type of role	Binding	Standard r...	Role name
Agent (Default Role)	☑	20000026	Person responsible for requisn. release
Recipient for Missed Latest Start	□	00000000	
Recipient for Missed Latest End	□	00000000	

Figure 57: Standard Task: Display screen displaying Default roles tab

You can double click on the standard role (AC20000026) to launch the PFAC transaction and review the function module that determines the agent.



Standard Role: Display

Standard role	20000026 MM_req_rel_c		
Name	Person responsible for requisn. release		
Development class	ME	Appl. component	MM-PUR

Basic data

Abbr.	MM_req_rel_c
Name	Person responsible for requisn. release

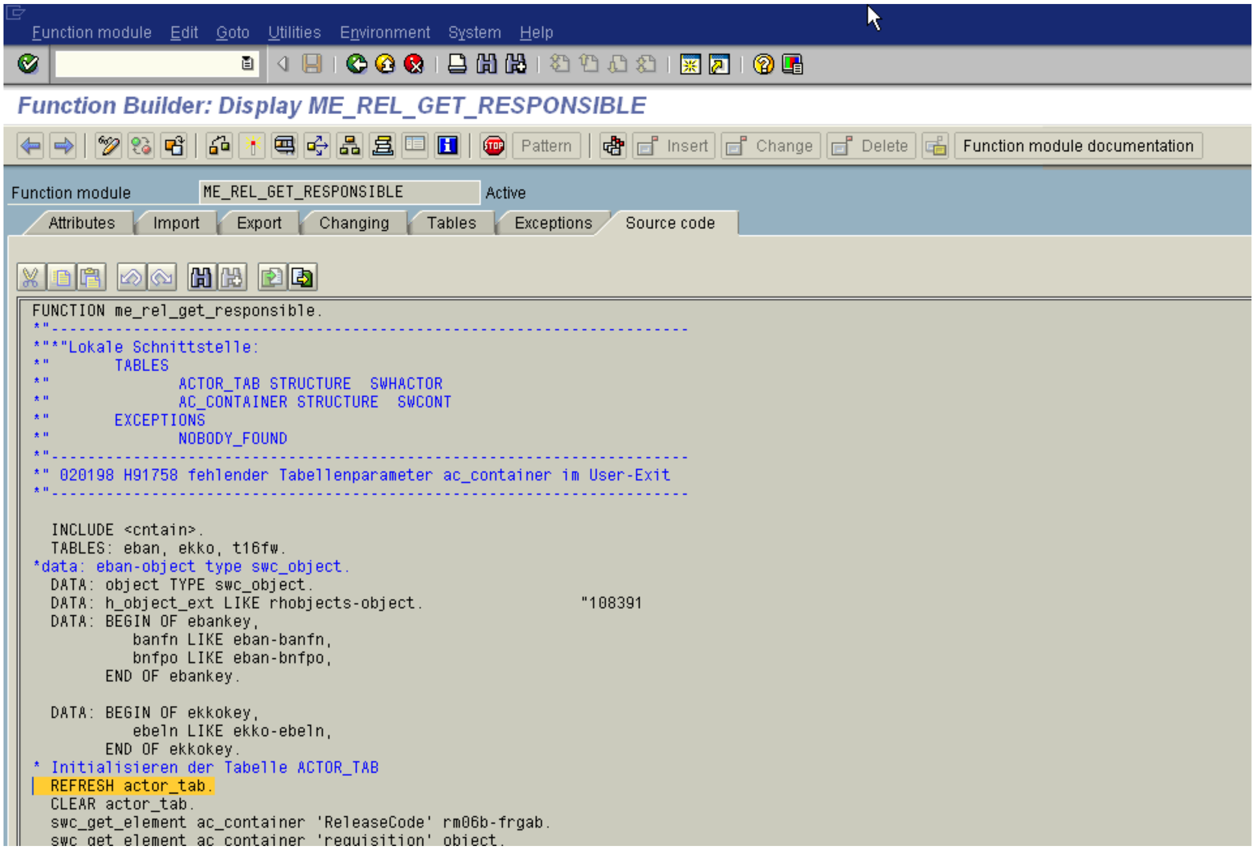
Role definition

Type	Function to be executed
Function module	ME_REL_GET_RESPONSIBLE

☑ Terminate if role resolution has no result

Figure 58: Standard Role: Display screen displaying Role definition tab

Set a SOFT breakpoint in ME_REL_GET_RESPONSIBLE (use transaction SE37).



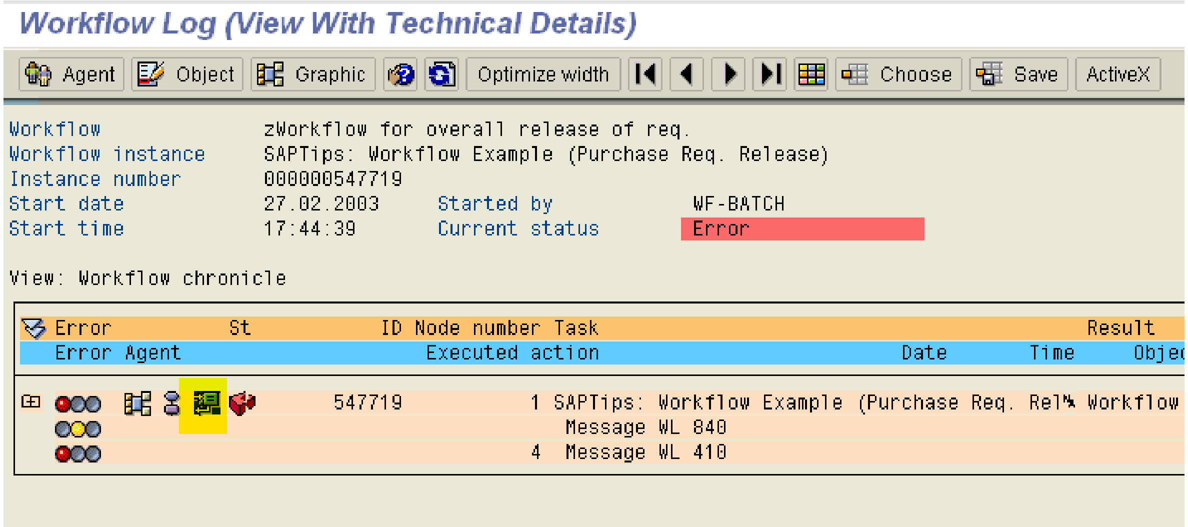
Function Builder: Display ME_REL_GET_RESPONSIBLE

Function module ME_REL_GET_RESPONSIBLE Active

```
FUNCTION me_rel_get_responsible.
*"----------------------------------------------------------------------
*"*"Lokale Schnittstelle:
*"       TABLES
*"              ACTOR_TAB STRUCTURE  SWHACTOR
*"              AC_CONTAINER STRUCTURE  SWCONT
*"       EXCEPTIONS
*"              NOBODY_FOUND
*"----------------------------------------------------------------------
*" 020198 H91758 fehlender Tabellenparameter ac_container im User-Exit
*"----------------------------------------------------------------------

  INCLUDE <cntain>.
  TABLES: eban, ekko, t16fw.
*data: eban-object type swc_object.
  DATA: object TYPE swc_object.
  DATA: h_object_ext LIKE rhobjects-object.                 "108391
  DATA: BEGIN OF ebankey,
          banfn LIKE eban-banfn,
          bnfpo LIKE eban-bnfpo,
        END OF ebankey.

  DATA: BEGIN OF ekkokey,
          ebeln LIKE ekko-ebeln,
        END OF ekkokey.
* Initialisieren der Tabelle ACTOR_TAB
  REFRESH actor_tab.
  CLEAR actor_tab.
  swc_get_element ac_container 'ReleaseCode' rm06b-frgab.
  swc get element ac container 'requisition' object.
```

Figure 59: Function Builder: Display screen displaying Source code tab

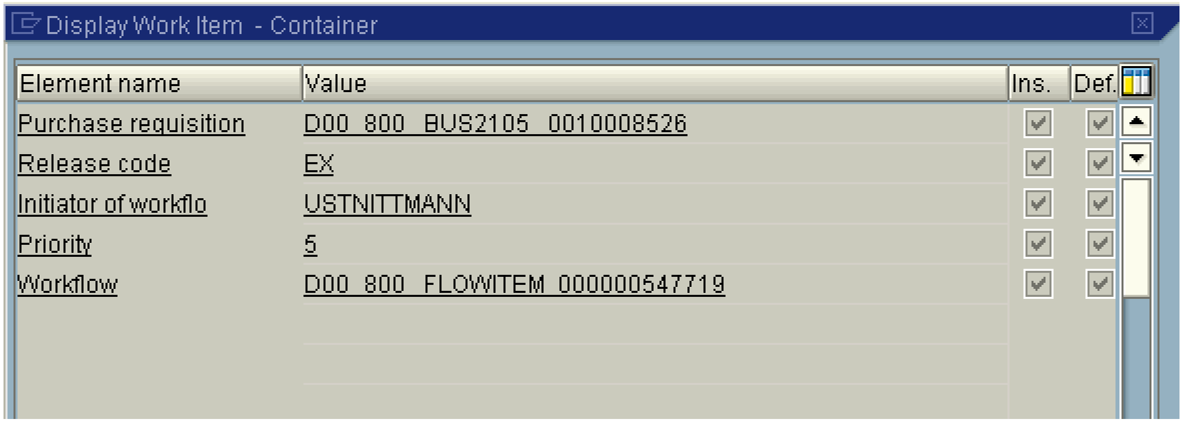
Now, we'll use the workflow test transaction SWUS to execute requisition release workflow WS99500034. We will need the container elements so we should go to SWI1 (and then workflow log > container elements) first and display the errored workflow event parameters.



Workflow Log (View With Technical Details)

```
Workflow             zWorkflow for overall release of req.
Workflow instance    SAPTips: Workflow Example (Purchase Req. Release)
Instance number      000000547719
Start date           27.02.2003     Started by           WF-BATCH
Start time           17:44:39       Current status       Error

View: Workflow chronicle

  Error          St          ID Node number Task                               Result
  Error Agent                   Executed action                     Date       Time      Objec

  ●○○                     547719          1 SAPTips: Workflow Example (Purchase Req. Rel Workflow
  ○●○                                       Message WL 840
  ●○○                                     4 Message WL 410
```

Figure 60: Workflow Log (View With Technical Details) screen

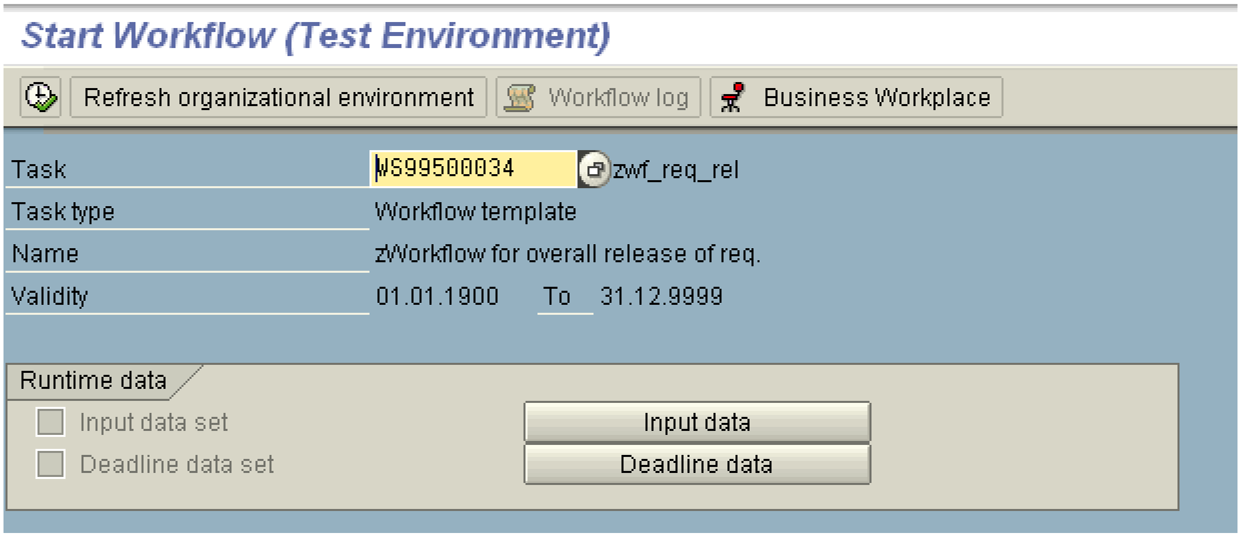


Display Work Item - Container

Element name	Value	Ins.	Def.
Purchase requisition	D00 800 BUS2105 0010008526	☑	☑
Release code	EX	☑	☑
Initiator of workflo	USTNITTMANN	☑	☑
Priority	5	☑	☑
Workflow	D00 800 FLOWITEM 000000547719	☑	☑

Figure 61: Display Work Item - Container screen

Go to SWUS and execute the workflow with the above container elements.



Start Workflow (Test Environment)

Task	WS99500034 zwf_req_rel
Task type	Workflow template
Name	zWorkflow for overall release of req.
Validity	01.01.1900 To 31.12.9999

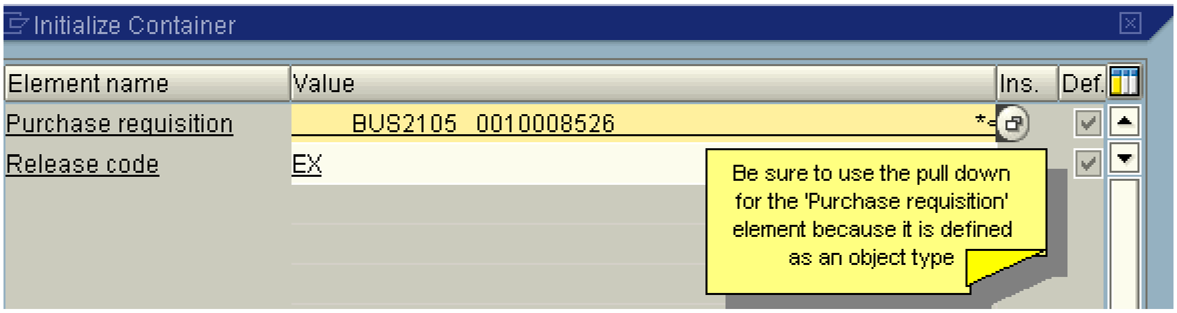
Runtime data

☐ Input data set

☐ Deadline data set

Figure 62: Start Workflow (Test Environment) screen

Select 'Input Data' and enter the container elements



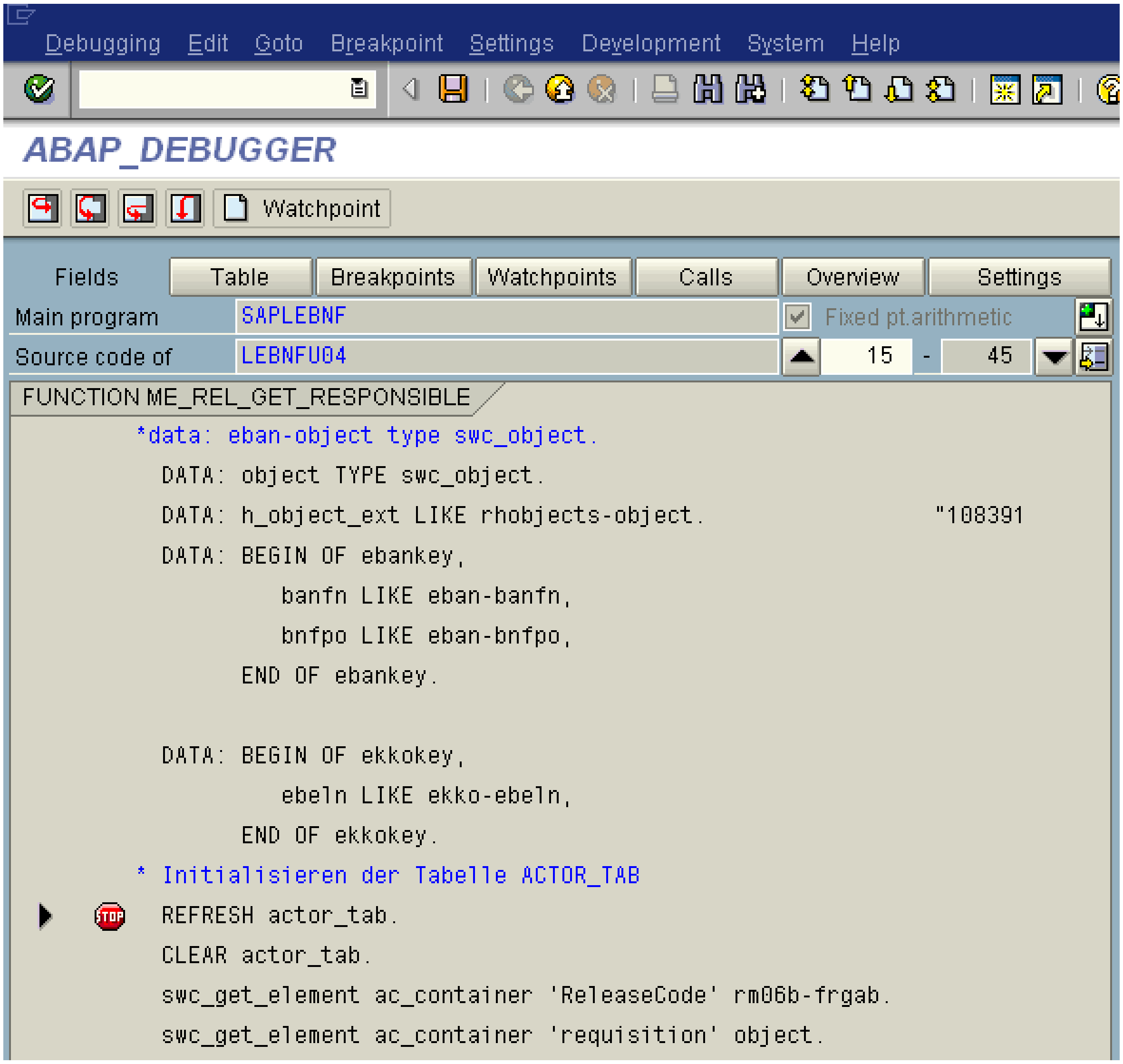
Initialize Container

Element name	Value	Ins.	Def.
Purchase requisition	BUS2105 0010008526 *		☑
Release code	EX		☑

Be sure to use the pull down for the 'Purchase requisition' element because it is defined as an object type

Figure 63: Initialize Container screen

Execute the test workflow. The debug point set up in function module ME_REL_GET_RESPONSIBLE will be evoked and you can start debugging the issue.



Main program SAPLEBNF ☑ Fixed pt.arithmetic

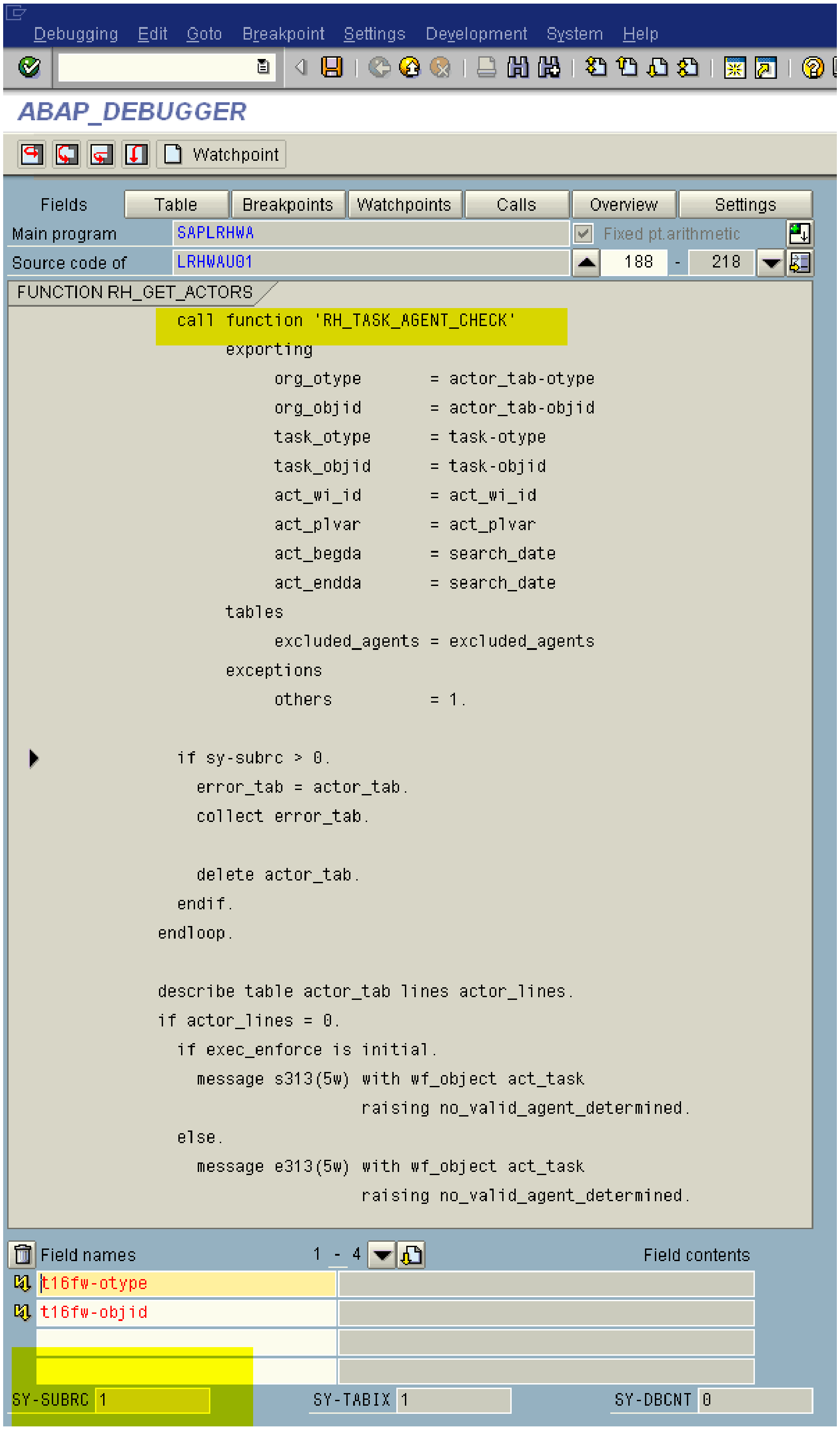
Source code of LEBNFU04 15 - 45

FUNCTION ME_REL_GET_RESPONSIBLE

```
    *data: eban-object type swc_object.
      DATA: object TYPE swc_object.
      DATA: h_object_ext LIKE rhobjects-object.                   "108391
      DATA: BEGIN OF ebankey,
               banfn LIKE eban-banfn,
               bnfpo LIKE eban-bnfpo,
            END OF ebankey.

      DATA: BEGIN OF ekkokey,
               ebeln LIKE ekko-ebeln,
            END OF ekkokey.
    * Initialisieren der Tabelle ACTOR_TAB
▶  STOP  REFRESH actor_tab.
      CLEAR actor_tab.
      swc_get_element ac_container 'ReleaseCode' rm06b-frgab.
      swc_get_element ac_container 'requisition' object.
```

Figure 64: ABAP_DEBUGGER screen

Run through the debugger and you will eventually get to a function module where SYST-SUBRC = 1.



Main program SAPLRHWA

Source code of LRHWAU01 188 - 218

FUNCTION RH_GET_ACTORS

```
    call function 'RH_TASK_AGENT_CHECK'
           exporting
                org_otype       = actor_tab-otype
                org_objid       = actor_tab-objid
                task_otype      = task-otype
                task_objid      = task-objid
                act_wi_id       = act_wi_id
                act_plvar       = act_plvar
                act_begda       = search_date
                act_endda       = search_date
           tables
                excluded_agents = excluded_agents
           exceptions
                others          = 1.

    if sy-subrc > 0.
      error_tab = actor_tab.
      collect error_tab.

      delete actor_tab.
    endif.
  endloop.

  describe table actor_tab lines actor_lines.
  if actor_lines = 0.
    if exec_enforce is initial.
      message s313(5w) with wf_object act_task
                       raising no_valid_agent_determined.
    else.
      message e313(5w) with wf_object act_task
                       raising no_valid_agent_determined.
```

Field names 1 - 4	Field contents
t16fw-otype	
t16fw-objid	

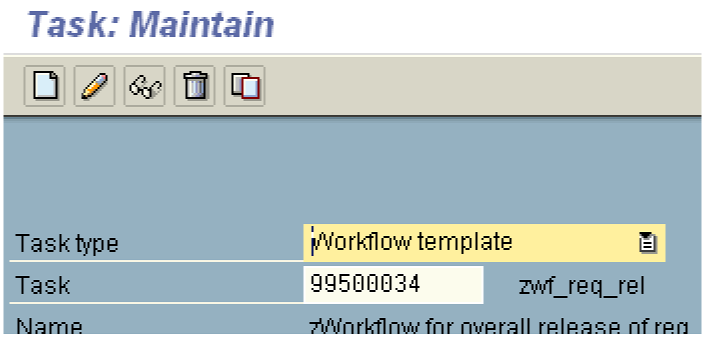
SY-SUBRC 1 SY-TABIX 1 SY-DBCNT 0

Figure 65: ABAP_DEBUGGER screen

The function module 'RH_TASK_AGENT_CHECK' determines if the task is assigned to a user id. It isn't, so we receive the error. Let's briefly discuss task '**attributes**'.

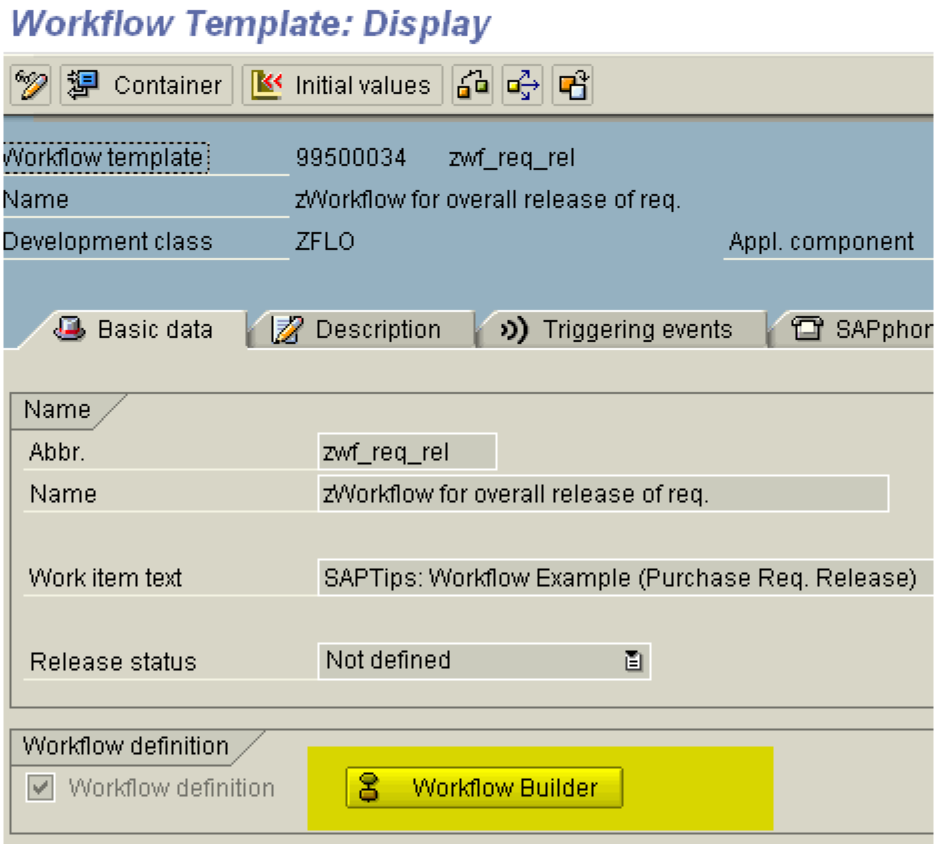
- A task w/ attribute 'General task' indicates any SAP user can execute the task.
- A task w/ attribute 'General forwarding allowed' indicates that the original agent may forward the task to any user (even if they are not assigned to the task).
- A task w/ attribute 'General forwarding not allowed' indicates that the work item can only be forwarded to agents assigned to the task.

To determine the task attributes, go to transaction PFTC > enter the Workflow template > go to display > select Workflow Builder >


Task: Maintain

Task type	Workflow template	
Task	99500034	zwf_req_rel
Name	zWorkflow for overall release of req	

Figure 66: Task: Maintain screen


Workflow Template: Display

Workflow template	99500034	zwf_req_rel
Name	zWorkflow for overall release of req.	
Development class	ZFLO	Appl. component

Name

Abbr.	zwf_req_rel
Name	zWorkflow for overall release of req.
Work item text	SAPTips: Workflow Example (Purchase Req. Release)
Release status	Not defined

Workflow definition

☑ Workflow definition

Figure 67: Workflow Template: Display screen showing Basic data tab

Navigate to the step in question (here step ‘000004’) > double click on the step >

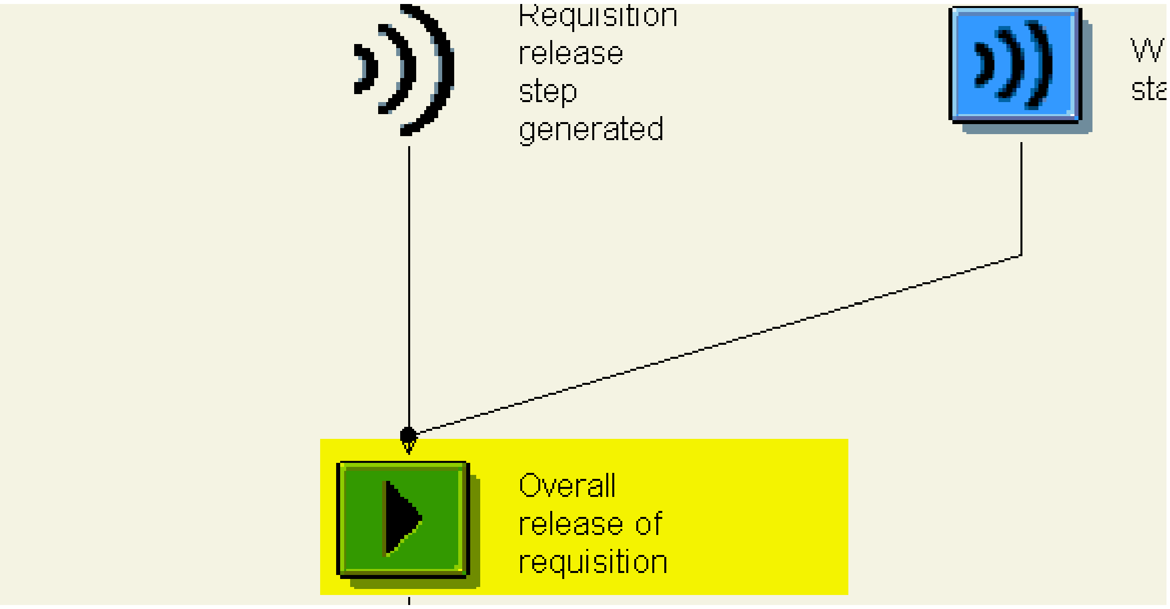
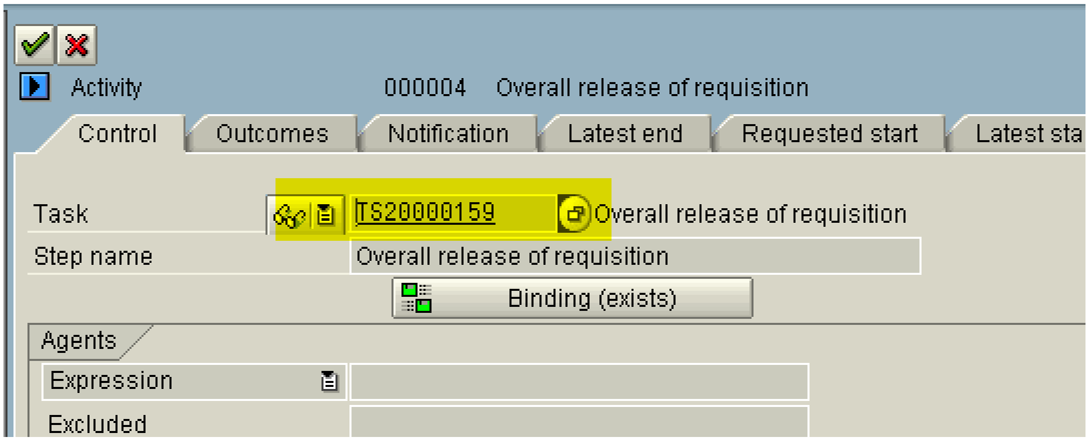

Requisition release step generated
W sta
Overall release of requisition

Figure 68: Workflow Builder Process Flow
Double click on TS20000159 >



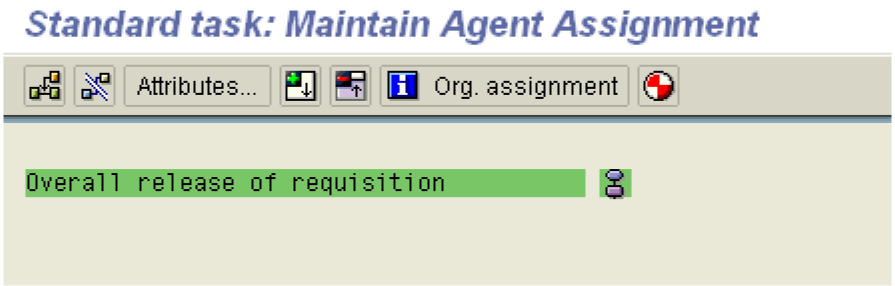
Activity 000004 Overall release of requisition

Task TS20000159 Overall release of requisition

Step name Overall release of requisition

Figure 69: Control tab

From the menu, select Additional Data > Agent Assignment > Maintain >

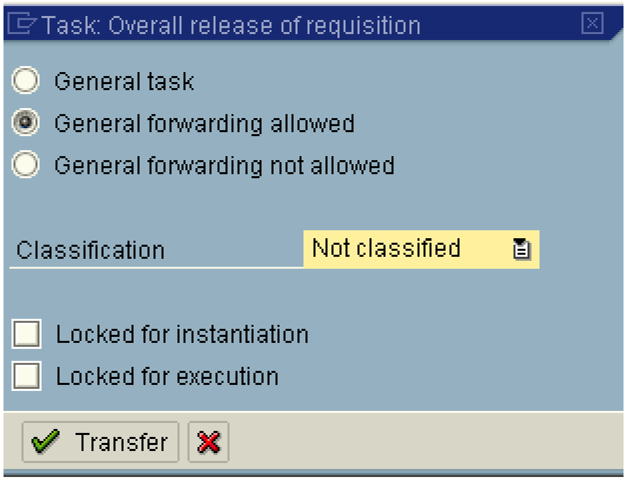


Standard task: Maintain Agent Assignment

Overall release of requisition

Figure 70: Standard task: Maintain Agent Assignment screen

Select the line item and then click attributes >



Task: Overall release of requisition

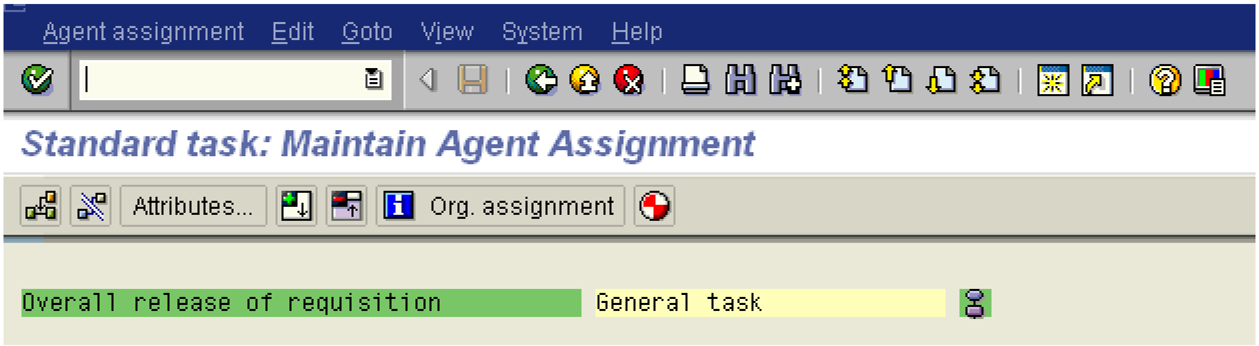
○ General task
● General forwarding allowed
○ General forwarding not allowed

Classification: Not classified

□ Locked for instantiation
□ Locked for execution

Figure 71: Task Overall release of requisition screen

This is our issue. The task is not classified as general. Option one is to create an org. structure, assign a position, assign a user (or person), and then assign the task to the position. Or we can make the task general so it can be executed by anybody (with the expectation that the OMGQ configuration role resolution will be used). Let's change the assignment to 'General task' and save.

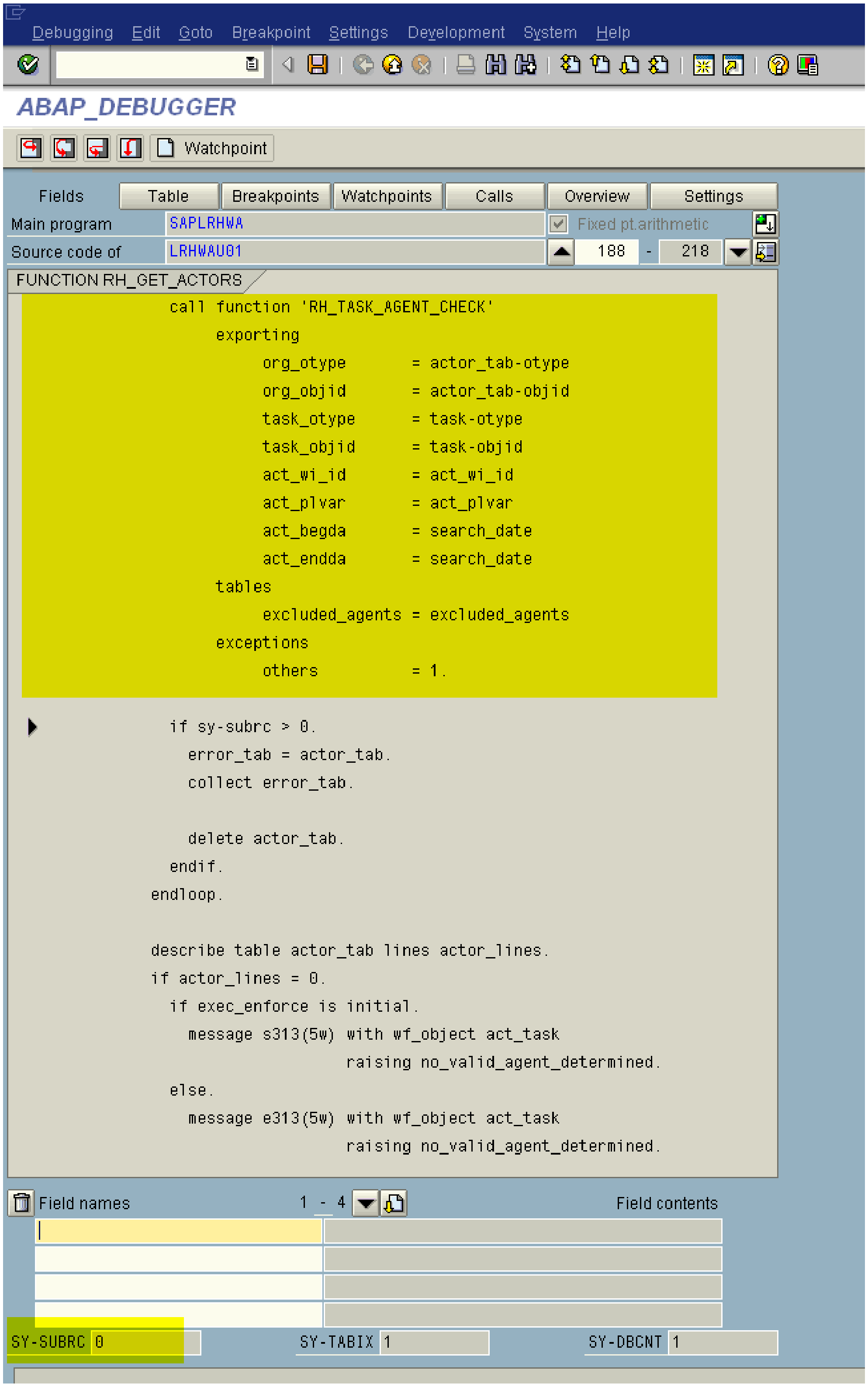


Standard task: Maintain Agent Assignment

Overall release of requisition | General task

Figure 72: Standard task: Maintain Agent Assignment screen

Re-execute the SWUS test transaction. Notice we no longer have an issue w/ RH_TASK_AGENT_CHECK



Main program SAPLRHWA

Source code of LRHWAU01 188 - 218

FUNCTION RH_GET_ACTORS

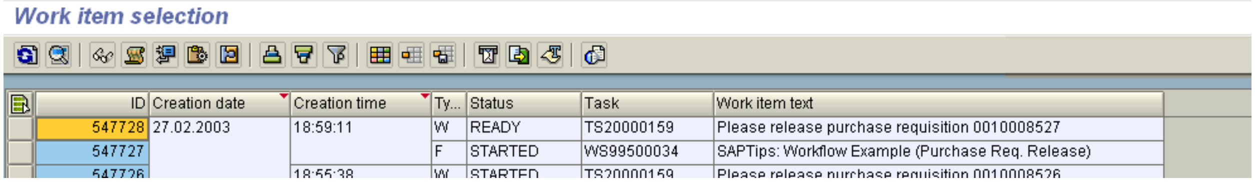
```
          call function 'RH_TASK_AGENT_CHECK'
               exporting
                    org_otype       = actor_tab-otype
                    org_objid       = actor_tab-objid
                    task_otype      = task-otype
                    task_objid      = task-objid
                    act_wi_id       = act_wi_id
                    act_plvar       = act_plvar
                    act_begda       = search_date
                    act_endda       = search_date
               tables
                    excluded_agents = excluded_agents
               exceptions
                    others          = 1.

          if sy-subrc > 0.
            error_tab = actor_tab.
            collect error_tab.

            delete actor_tab.
          endif.
        endloop.

        describe table actor_tab lines actor_lines.
        if actor_lines = 0.
          if exec_enforce is initial.
            message s313(5w) with wf_object act_task
                             raising no_valid_agent_determined.
          else.
            message e313(5w) with wf_object act_task
                             raising no_valid_agent_determined.
```

Field names 1 - 4 Field contents

SY-SUBRC 0 SY-TABIX 1 SY-DBCNT 1

Figure 73: ABAP_DEBUGGER screen

Create another purchase requisition from scratch and validate that the workflow is correctly created (SWI1).

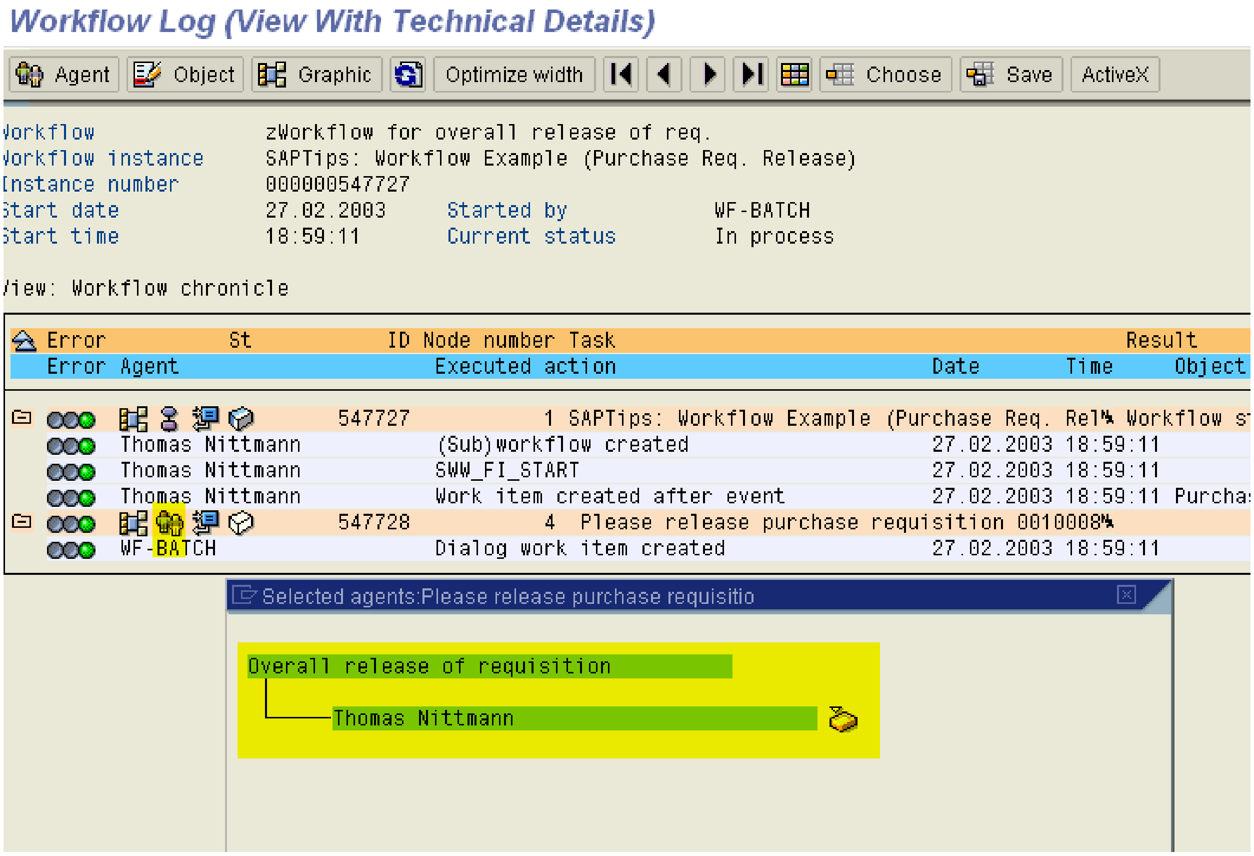


Work item selection

ID	Creation date	Creation time	Ty...	Status	Task	Work item text
547728	27.02.2003	18:59:11	W	READY	TS20000159	Please release purchase requisition 0010008527
547727			F	STARTED	WS99500034	SAPTips: Workflow Example (Purchase Req. Release)
547726		18:55:38	W	STARTED	TS20000159	Please release purchase requisition 0010008526

Figure 74: Work item selection screen

By clicking on the agent icon, you will notice that an agent is now correctly assigned and the workflow is no longer in error.



Workflow Log (View With Technical Details)

```
Workflow            zWorkflow for overall release of req.
Workflow instance   SAPTips: Workflow Example (Purchase Req. Release)
Instance number     000000547727
Start date          27.02.2003    Started by          WF-BATCH
Start time          18:59:11      Current status      In process

View: Workflow chronicle

Error          St             ID Node number Task                                  Result
Error Agent                      Executed action                          Date       Time     Object

                         547727          1 SAPTips: Workflow Example (Purchase Req. Rel Workflow s
  Thomas Nittmann                (Sub)workflow created                   27.02.2003 18:59:11
  Thomas Nittmann                SWW_FI_START                            27.02.2003 18:59:11
  Thomas Nittmann                Work item created after event           27.02.2003 18:59:11 Purcha
                         547728          4  Please release purchase requisition 0010008
  WF-BATCH                       Dialog work item created                27.02.2003 18:59:11
```

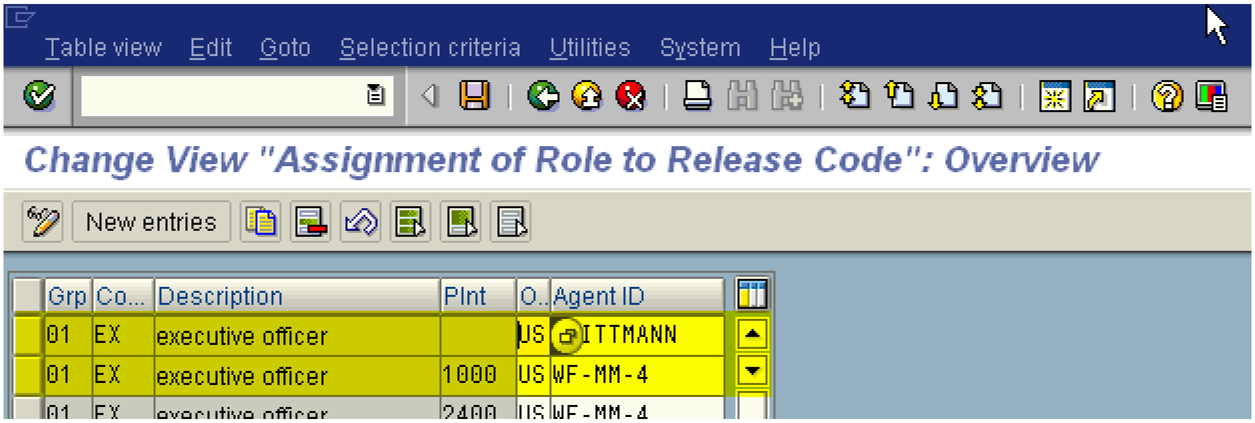
Selected agents:Please release purchase requisitio

```
Overall release of requisition
   └──Thomas Nittmann
```

Figure 75: Workflow Log (View With Technical Details) screen

> *Note: a common migration issue specific to purchase requisition releases occurs when clients transition from individual line item release to overall release. If OMGQ workflow configuration is maintained at the plant level (based on single line item release), the workflow will error for overall release because there is* ***no plant at the header level of the purchase requisition****, i.e., you must remove the plant from OMGQ > Workflow. By using the*

same debugging principles just described, you would be able to determine the error.



Change View "Assignment of Role to Release Code": Overview

Grp	Co...	Description	Plnt	O..	Agent ID
01	EX	executive officer		US	ITTMANN
01	EX	executive officer	1000	US	WF-MM-4
01	EX	executive officer	2400	US	WF-MM-4

Figure 76: Change View “Assignment of Role to Release Code”: Overview screen

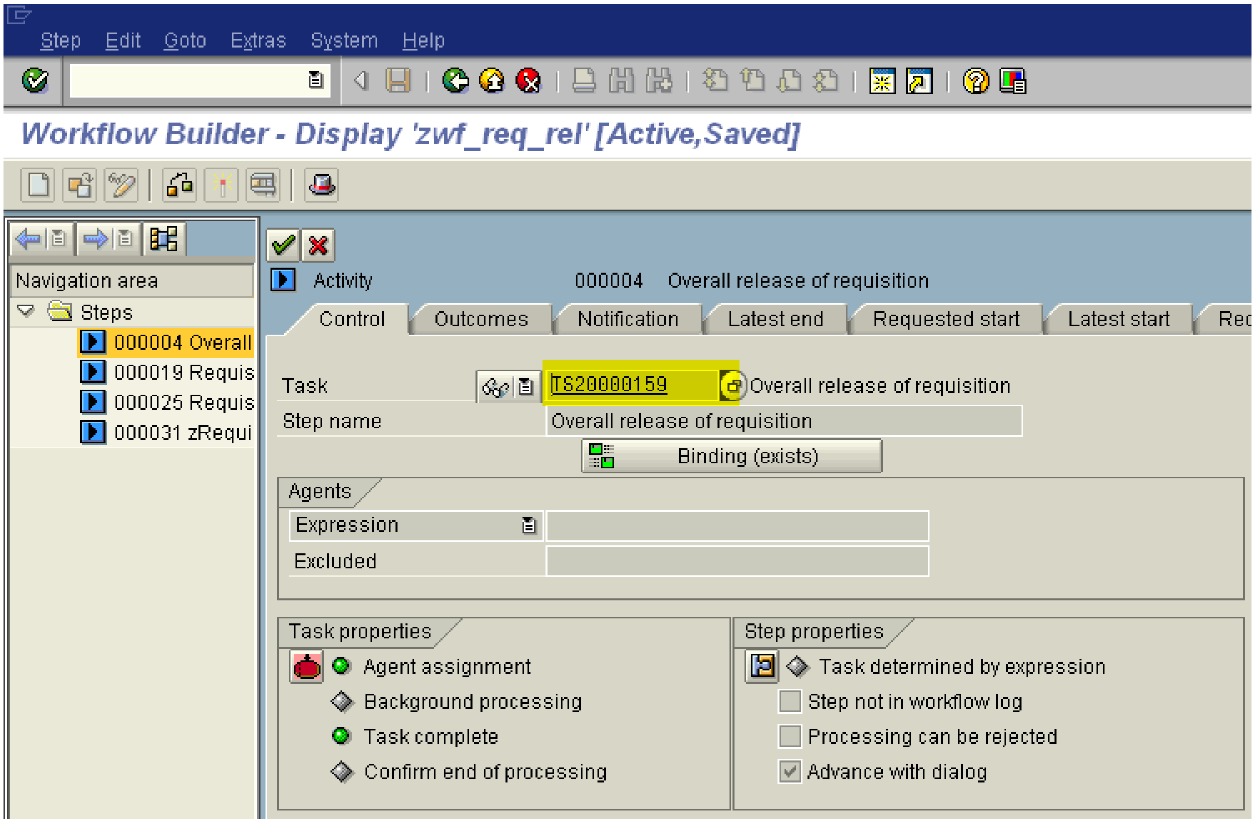
Workflow Task (Method) Debugging

The code for a single step task (TS) is written in methods assigned to objects in the Business Object Repository (BOR). It is often necessary to debug the method code to resolve workflow issues. Ideally, you want to start the debugger for the entire workflow but break at some point in the custom method code. Let’s review the process for debugging **both** foreground (dialog) and background (synchronous) single step tasks.

Foreground (Dialog) Tasks

We will start by dissecting the first foreground task in the custom purchase requisition workflow (WS99500034) we created. The first task is the overall release of the requisition (TS20000159). Use the workflow builder (transaction PFTC > enter ‘Workflow template’ for task type > enter task number ‘99500034’ for Name >select display > click on the ‘Workflow builder’ button) to call up the workflow and double click on the first task (step ‘000004’) in the process flow diagram to launch the workflow template task detail screen.

The goal is to set a break-point in the first dialog task to aid in debugging any issues.



Workflow Builder - Display 'zwf_req_rel' [Active,Saved]

Activity 000004 Overall release of requisition

Task TS20000159 Overall release of requisition

Step name Overall release of requisition

Binding (exists)

Agents

Expression

Excluded

Task properties

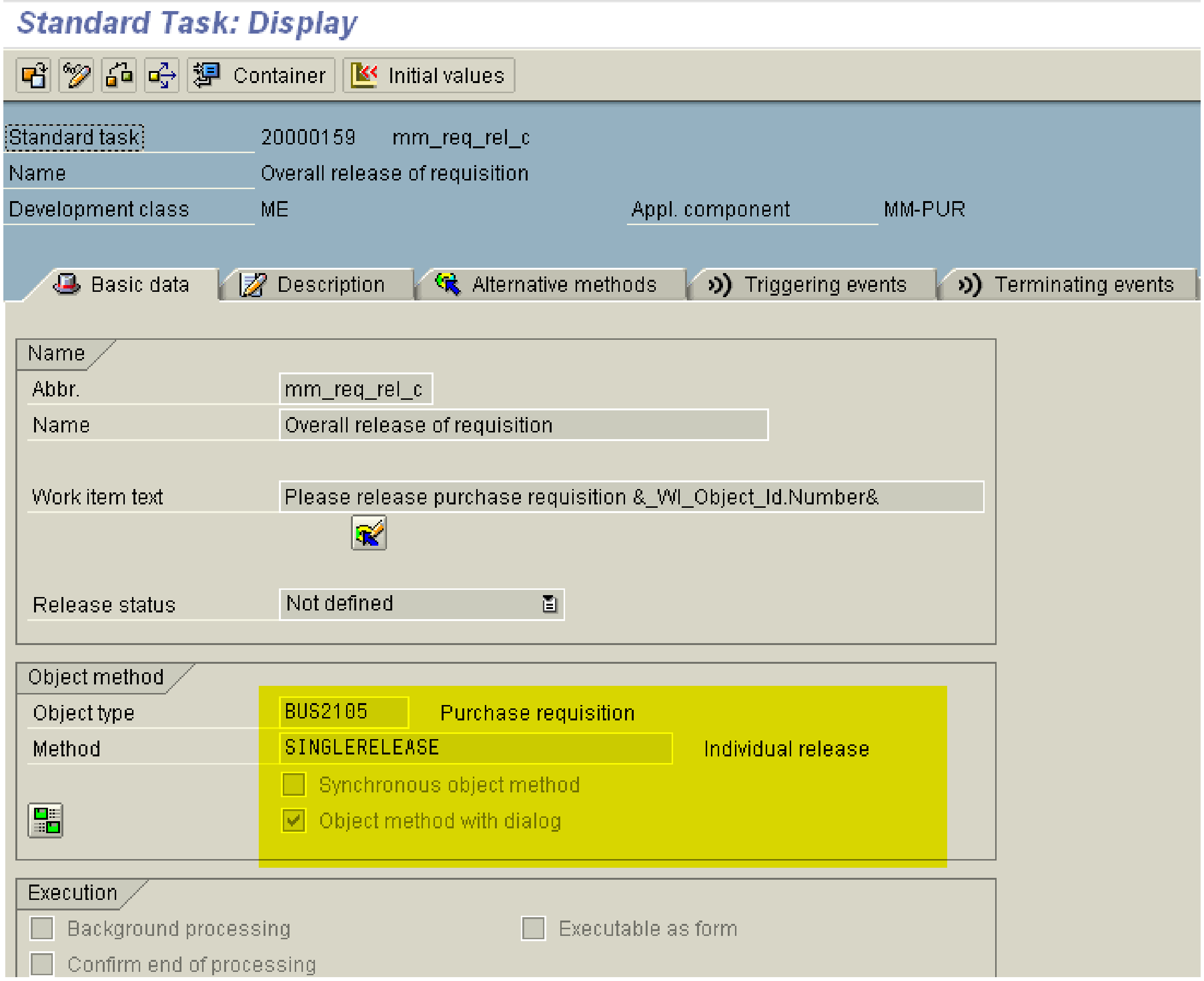
- Agent assignment
- Background processing
- Task complete
- Confirm end of processing

Step properties

- Task determined by expression
- □ Step not in workflow log
- □ Processing can be rejected
- ☑ Advance with dialog

Figure 77: Workflow Builder - Display screen

Next, double click on the task to launch the task details.



Standard Task: Display

Standard task	20000159 mm_req_rel_c		
Name	Overall release of requisition		
Development class	ME	Appl. component	MM-PUR

Name

Abbr.	mm_req_rel_c
Name	Overall release of requisition
Work item text	Please release purchase requisition &_WI_Object_Id.Number&
Release status	Not defined

Object method

Object type	BUS2105	Purchase requisition
Method	SINGLERELEASE	Individual release

□ Synchronous object method

☑ Object method with dialog

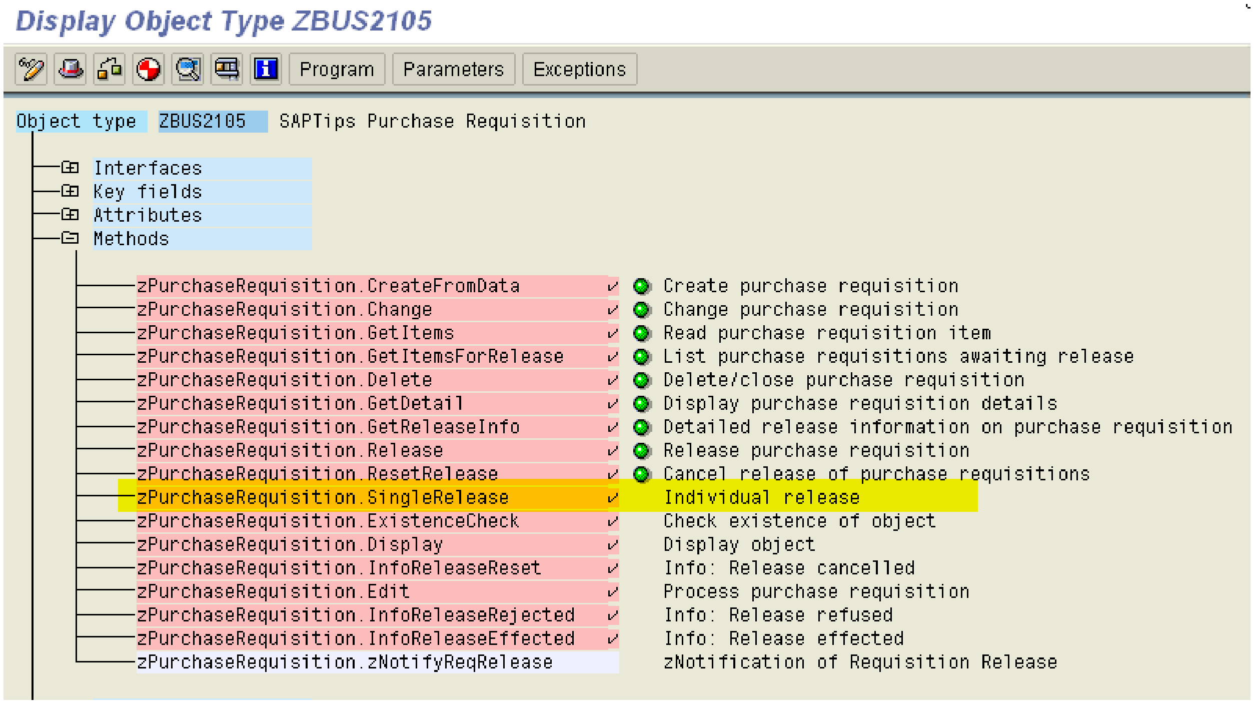
Execution

□ Background processing □ Executable as form

□ Confirm end of processing

Figure 78: Standard Task: Display screen displaying Basic data tab

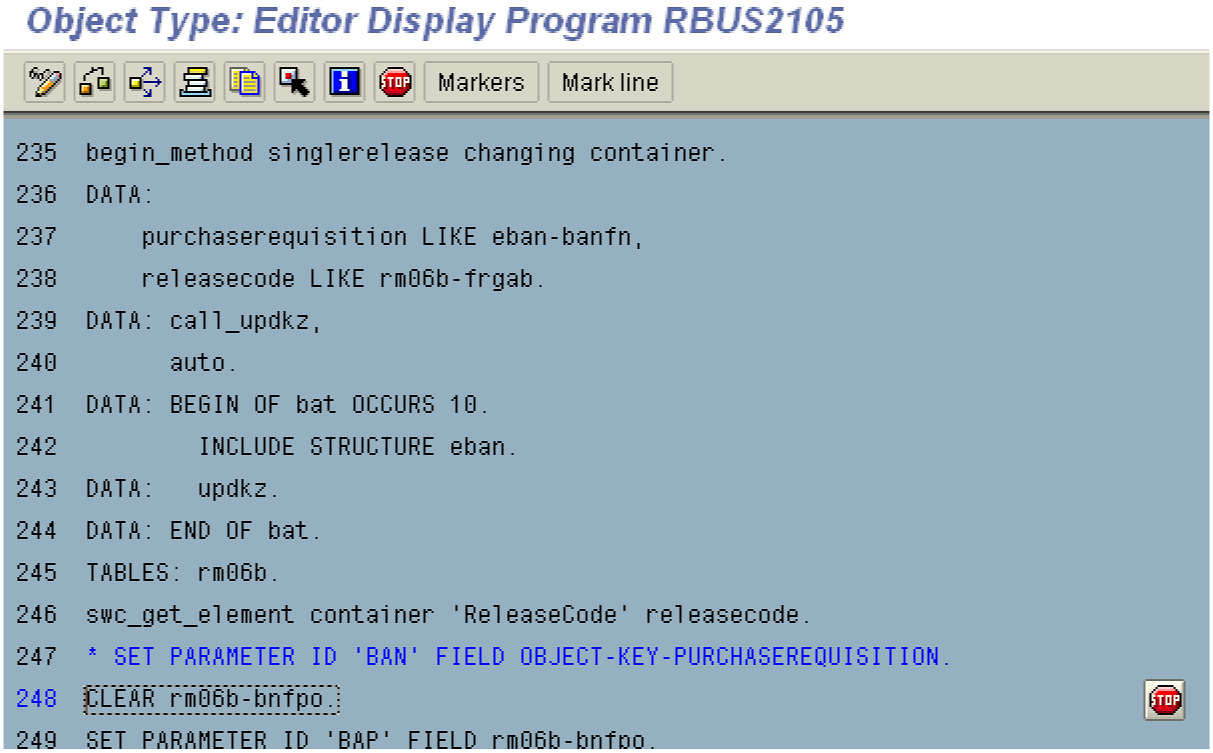
Double click on method ‘SINGLERELEASE’ to launch the BOR definition of the method.



```
Object type  ZBUS2105  SAPTips Purchase Requisition

  Interfaces
  Key fields
  Attributes
  Methods
      zPurchaseRequisition.CreateFromData        ✓ ● Create purchase requisition
      zPurchaseRequisition.Change                ✓ ● Change purchase requisition
      zPurchaseRequisition.GetItems              ✓ ● Read purchase requisition item
      zPurchaseRequisition.GetItemsForRelease    ✓ ● List purchase requisitions awaiting release
      zPurchaseRequisition.Delete                ✓ ● Delete/close purchase requisition
      zPurchaseRequisition.GetDetail             ✓ ● Display purchase requisition details
      zPurchaseRequisition.GetReleaseInfo        ✓ ● Detailed release information on purchase requisition
      zPurchaseRequisition.Release               ✓ ● Release purchase requisition
      zPurchaseRequisition.ResetRelease          ✓ ● Cancel release of purchase requisitions
      zPurchaseRequisition.SingleRelease         ✓   Individual release
      zPurchaseRequisition.ExistenceCheck        ✓   Check existence of object
      zPurchaseRequisition.Display               ✓   Display object
      zPurchaseRequisition.InfoReleaseReset      ✓   Info: Release cancelled
      zPurchaseRequisition.Edit                  ✓   Process purchase requisition
      zPurchaseRequisition.InfoReleaseRejected   ✓   Info: Release refused
      zPurchaseRequisition.InfoReleaseEffected   ✓   Info: Release effected
      zPurchaseRequisition.zNotifyReqRelease         zNotification of Requisition Release
```

Figure 79: Display Object Type screen

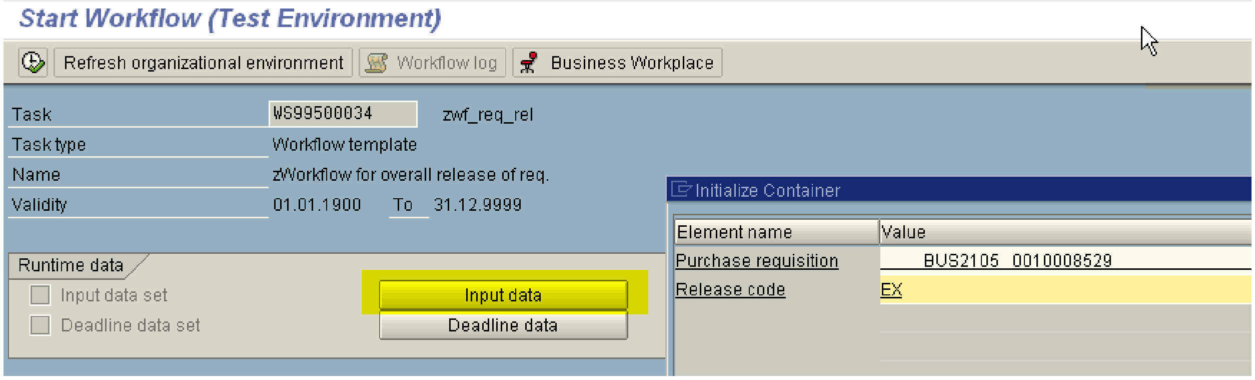
Notice that object 'ZBUS2105' was called - this is fine. Custom sub-type 'ZBUS2105' was created and delegated to super type 'BUS2105'. Place the cursor on the method 'zPurchaseRequisiton.SingleRelease,' and then click the 'Program' button to launch the code for this method. We will set a soft breakpoint.



```
begin_method singlerelease changing container.
DATA:
      purchaserequisition LIKE eban-banfn,
      releasecode LIKE rm06b-frgab.
DATA: call_updkz,
        auto.
DATA: BEGIN OF bat OCCURS 10.
          INCLUDE STRUCTURE eban.
DATA:   updkz.
DATA: END OF bat.
TABLES: rm06b.
swc_get_element container 'ReleaseCode' releasecode.
* SET PARAMETER ID 'BAN' FIELD OBJECT-KEY-PURCHASEREQUISITION.
CLEAR rm06b-bnfpo.
SET PARAMETER ID 'BAP' FIELD rm06b-bnfpo.
```

Figure 80: Object Type: Editor Display Program screen

We cannot simply create a new purchase requisition (ME51) and expect the breakpoint to hit, because workflow is based on the tRFC (transactional RFC) engine; it's an asynchronous RFC (which ensures unique execution). To test, we will use the workflow test transaction 'SWUS' and a former purchase requisition (00100008529) that was already created and has a release strategy.



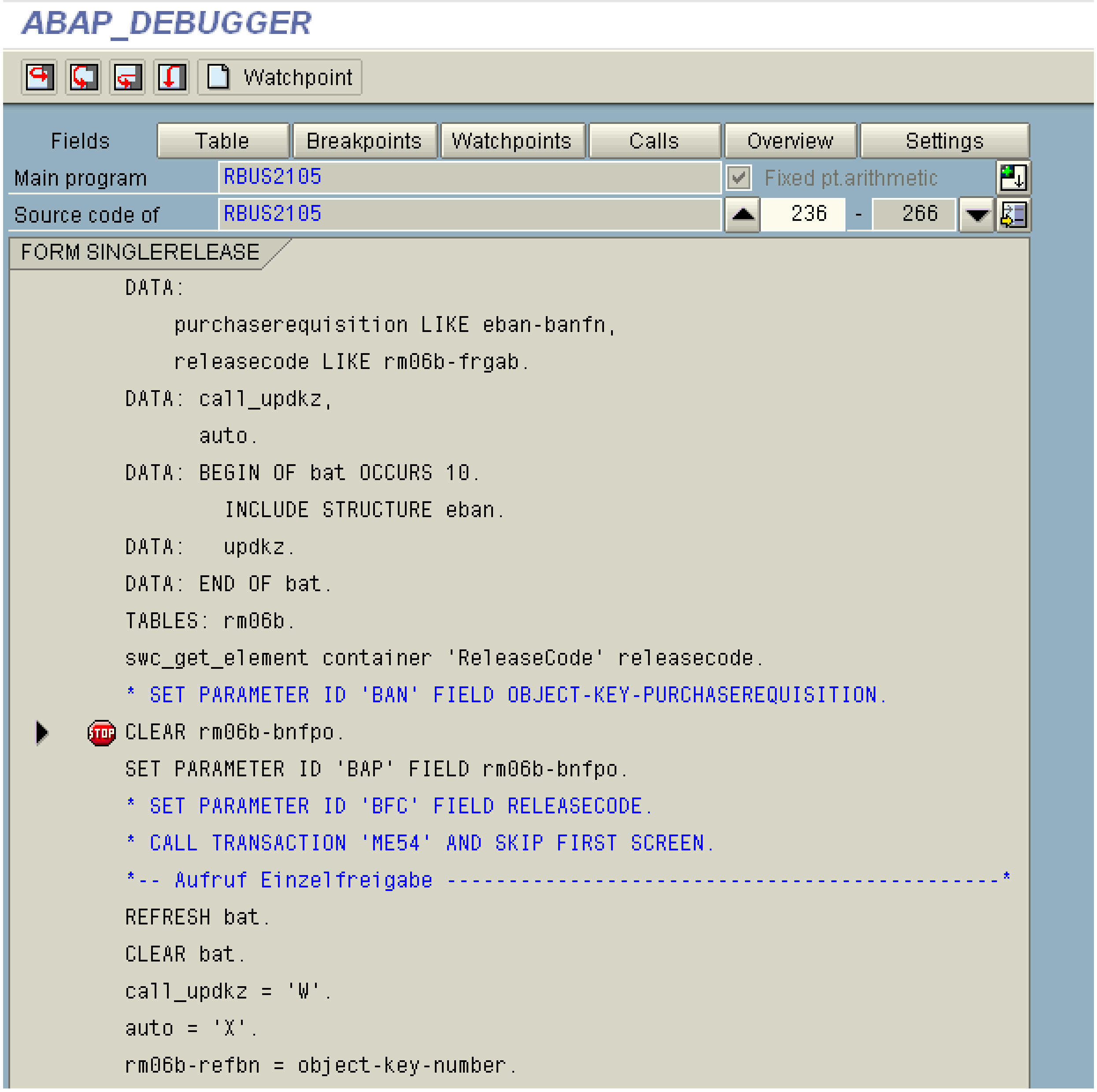
Task	WS99500034 zwf_req_rel
Task type	Workflow template
Name	zWorkflow for overall release of req.
Validity	01.01.1900 To 31.12.9999

□ Input data set
□ Deadline data set

Element name	Value
Purchase requisition	BUS2105 0010008529
Release code	EX

Figure 81: Start Workflow (Test Environment) screen

When we execute the data (after adding the input data), the method code is available for debugging.



Main program RBUS2105

Source code of RBUS2105 236 - 266

FORM SINGLERELEASE

```
DATA:
    purchaserequisition LIKE eban-banfn,
    releasecode LIKE rm06b-frgab.
DATA: call_updkz,
      auto.
DATA: BEGIN OF bat OCCURS 10.
        INCLUDE STRUCTURE eban.
DATA:   updkz.
DATA: END OF bat.
TABLES: rm06b.
swc_get_element container 'ReleaseCode' releasecode.
* SET PARAMETER ID 'BAN' FIELD OBJECT-KEY-PURCHASEREQUISITION.
CLEAR rm06b-bnfpo.
SET PARAMETER ID 'BAP' FIELD rm06b-bnfpo.
* SET PARAMETER ID 'BFC' FIELD RELEASECODE.
* CALL TRANSACTION 'ME54' AND SKIP FIRST SCREEN.
*-- Aufruf Einzelfreigabe ---------------------------------------------*
REFRESH bat.
CLEAR bat.
call_updkz = 'W'.
auto = 'X'.
rm06b-refbn = object-key-number.
```

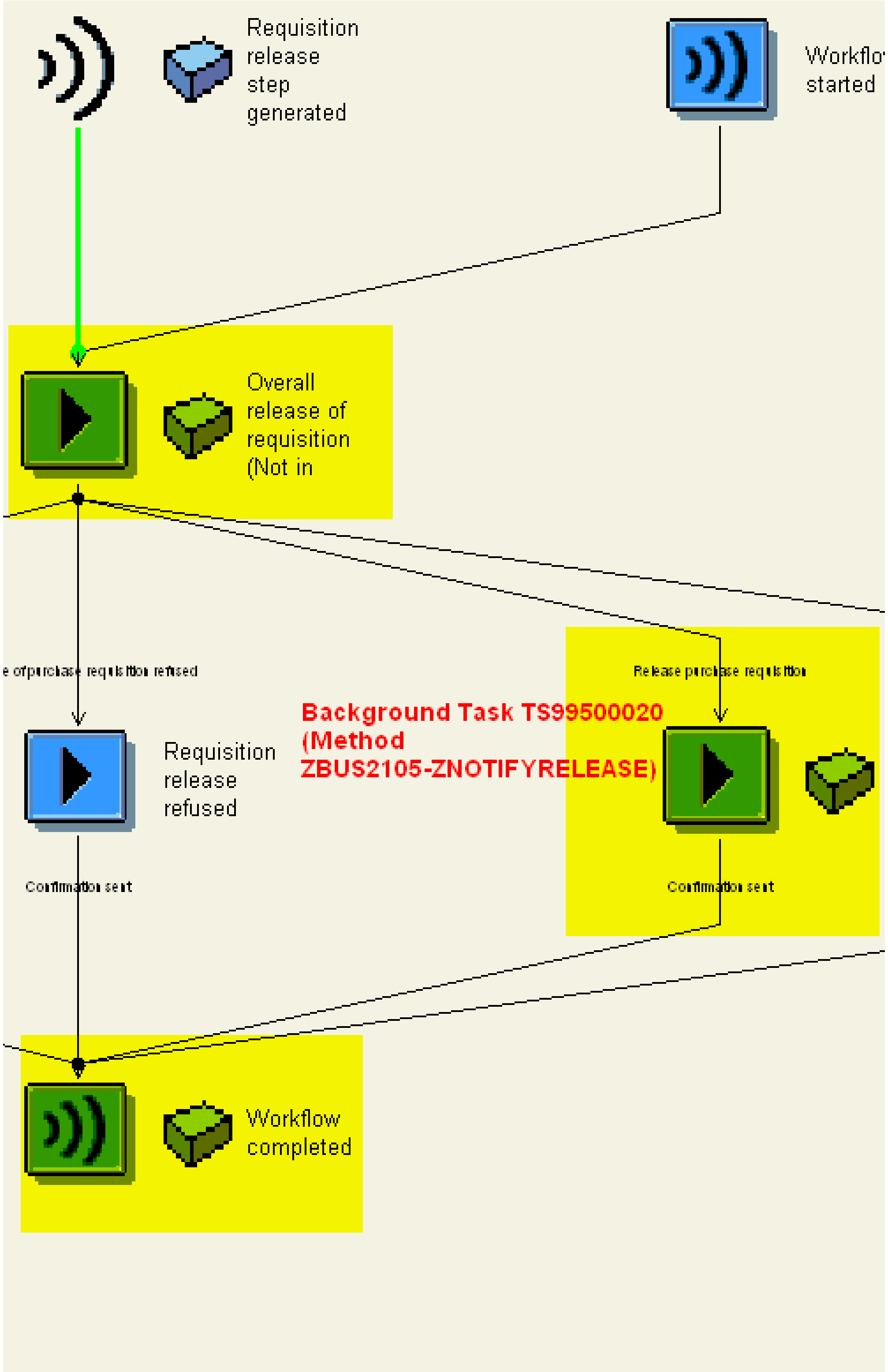
Figure 82: ABAP_DEBUGGER screen

Background (Synchronous) Tasks

Special consideration is required for background tasks, because any breakpoints in the code **do not** cause the task to stop and wait for the user to debug. To illustrate the process of debugging, we will use background task TS99500020 (ZBUS2105-ZNOTIFYREQRELEASE), which sends a notification (not a work item) to a recipient when the requisition is released and has been added to the sample workflow template.

> *Hint: Several options are available for creating notifications via workflow: the SAP-delivered 'SendMail' task a custom method (as illustrated in this example with TS99500020/zmm_req_ok*

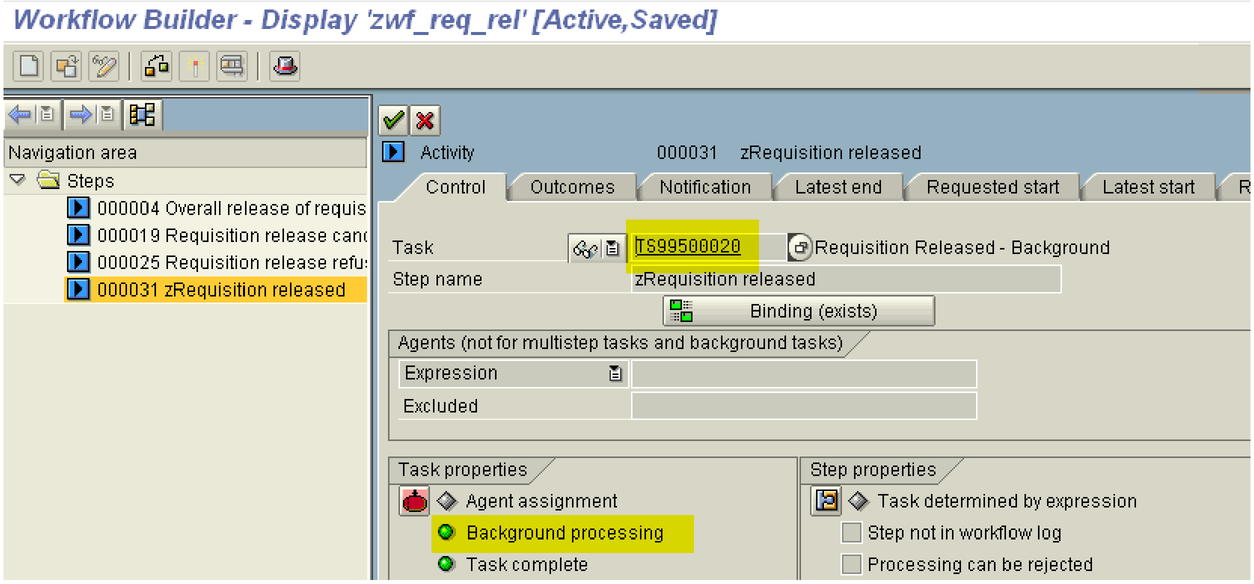
standard task), or a simple method w/ begin_method and end_method (adding text to task description and using report RSWUWFML to generate notifications for the work item). To incorporate several different notifications into a workflow process, it is sometimes simpler ***not*** *to use the standard SAP 'SendMail' task and create your own method for the process. Sometimes it's easier to dynamically create your agents for notification from a custom method rather than using the standard 'SendMail'.*


Requisition release step generated
Workflo
started
Overall release of requisition (Not in
e of purchase requisition refused
Release purchase requisition
Requisition release refused
Confirmation sent
Confirmation sent
Workflow completed

Background Task TS99500020 (Method ZBUS2105-ZNOTIFYRELEASE)

Figure 83: Workflow Builder Process Flow

To debug, use the workflow builder (transaction PFTC > enter 'Workflow template' for task type > enter task number '99500034' for Name >select display > click on the 'Workflow builder' button) to call up the workflow, and double click on the background task TS99500020 (step '00031') in the process flow diagram to launch the workflow template task detail screen.



Workflow Builder - Display 'zwf_req_rel' [Active,Saved]

Navigation area

Steps
- 000004 Overall release of requis
- 000019 Requisition release canc
- 000025 Requisition release refu:
- 000031 zRequisition released

Activity 000031 zRequisition released

Task: TS99500020 Requisition Released - Background

Step name: zRequisition released

Agents (not for multistep tasks and background tasks)

Expression:

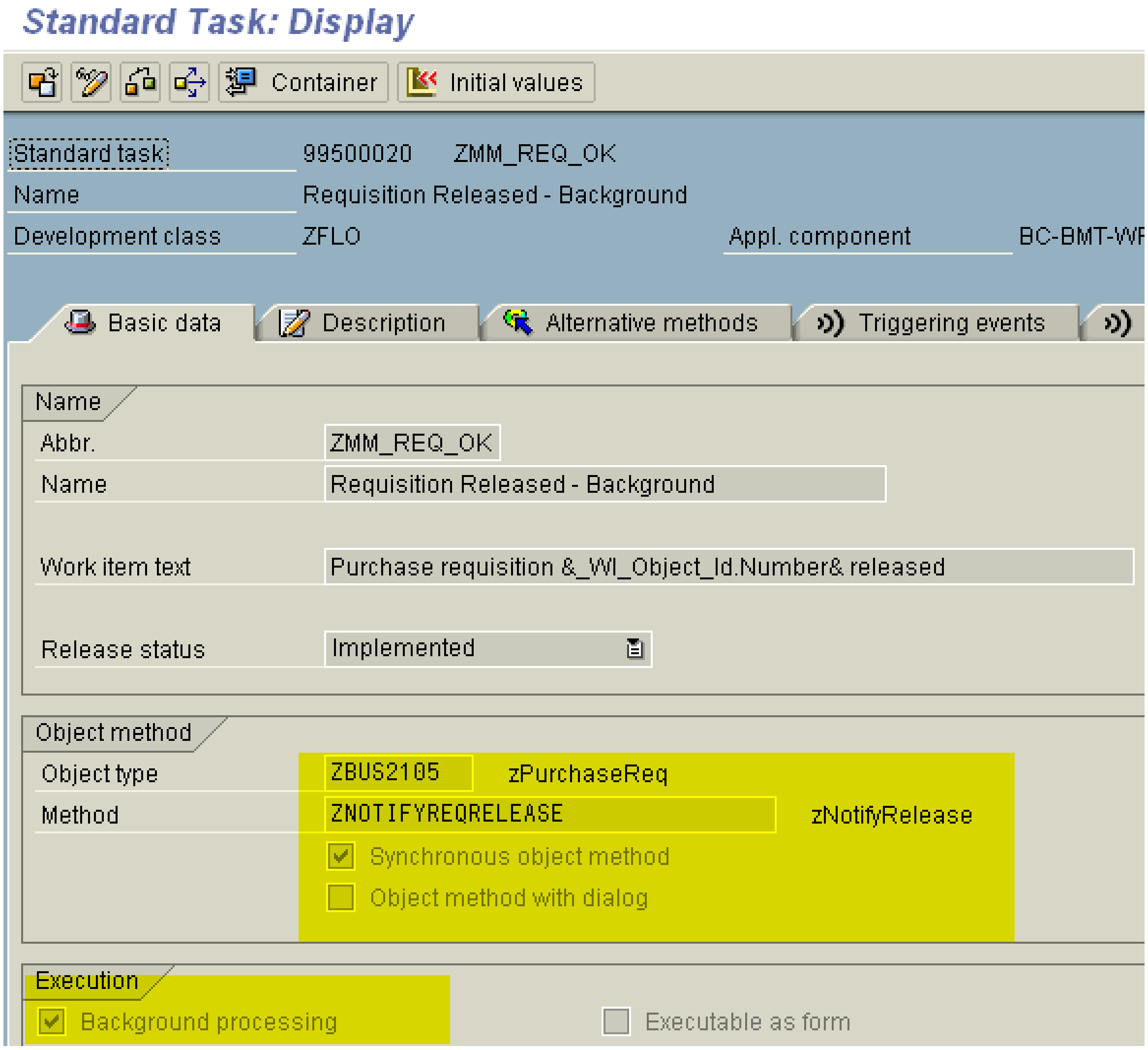
Excluded:

Task properties
- Agent assignment
- ● Background processing
- ● Task complete

Step properties
- Task determined by expression
- □ Step not in workflow log
- □ Processing can be rejected

Figure 84: Workflow Builder – Display screen

Next, double click on the task to launch the task details.



Standard Task: Display

Standard task 99500020 ZMM_REQ_OK
Name Requisition Released - Background
Development class ZFLO Appl. component BC-BMT-WF

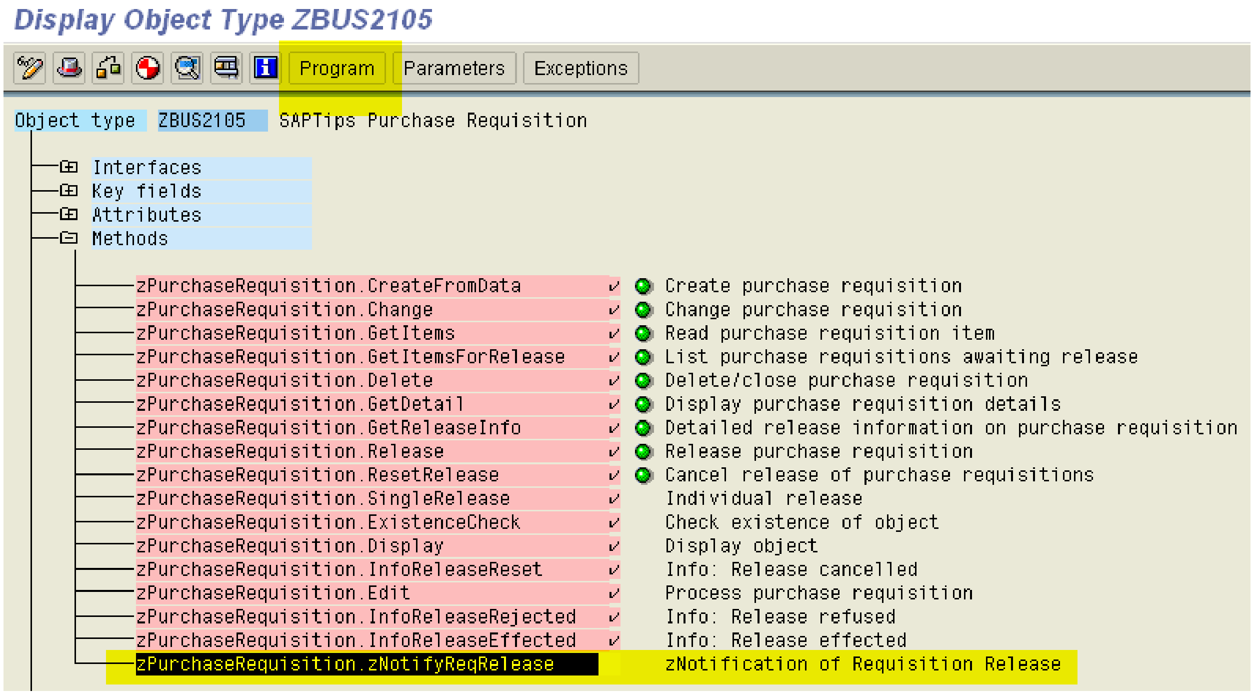
Name
Abbr. ZMM_REQ_OK
Name Requisition Released - Background
Work item text Purchase requisition &_WI_Object_Id.Number& released
Release status Implemented

Object method
Object type ZBUS2105 zPurchaseReq
Method ZNOTIFYREQRELEASE zNotifyRelease
☑ Synchronous object method
☐ Object method with dialog

Execution
☑ Background processing
☐ Executable as form

Figure 85: Standard Task: Display screen displaying Basic data tab

Double click on method 'ZNOTIFYREQRELEASE' to launch the BOR definition of the method and then place the cursor on the 'ZNOTIFYREQRELEASE' method.

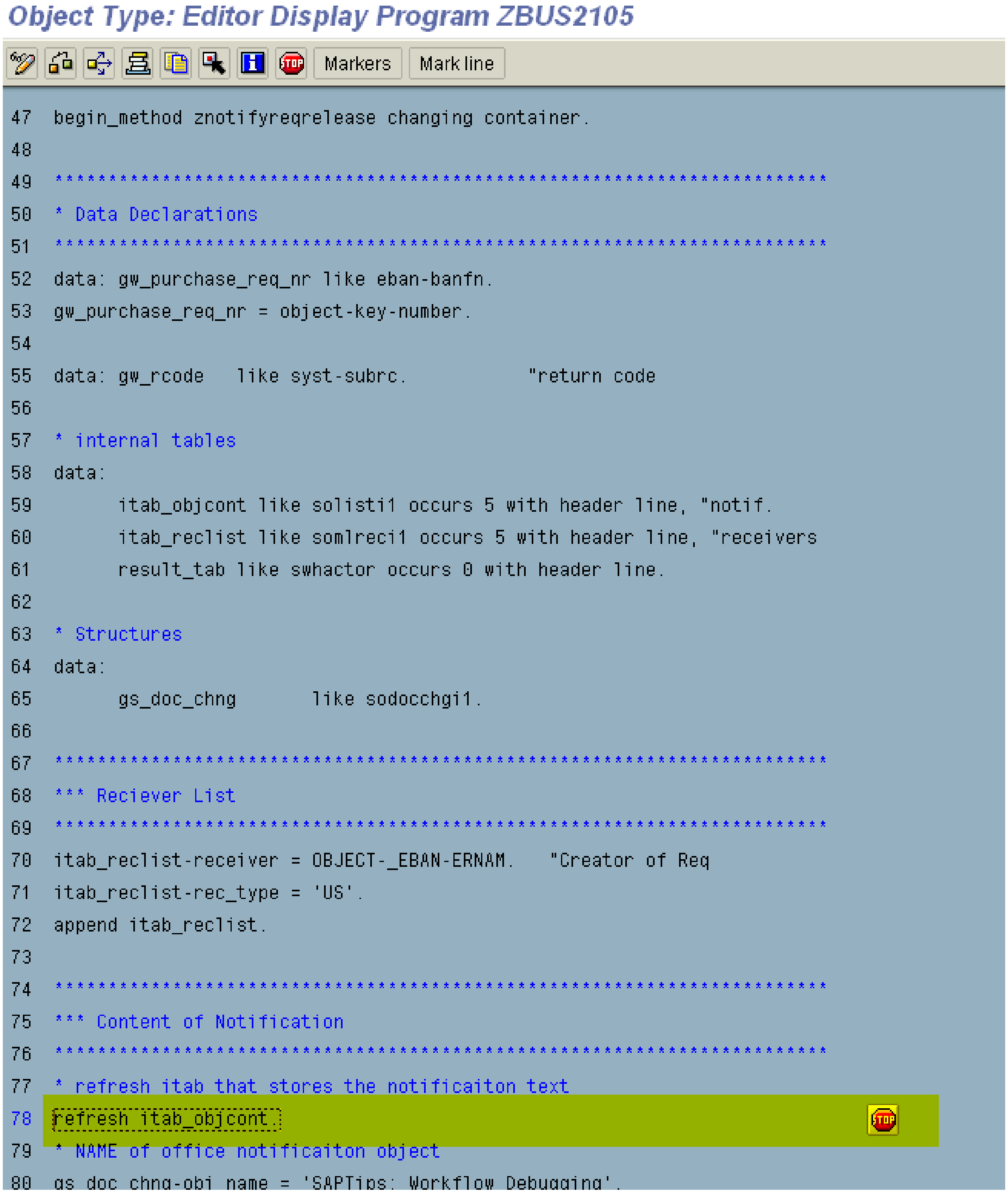


Display Object Type ZBUS2105

```
Object type  ZBUS2105   SAPTips Purchase Requisition

  Interfaces
  Key fields
  Attributes
  Methods

    zPurchaseRequisition.CreateFromData        ✓ ● Create purchase requisition
    zPurchaseRequisition.Change                ✓ ● Change purchase requisition
    zPurchaseRequisition.GetItems              ✓ ● Read purchase requisition item
    zPurchaseRequisition.GetItemsForRelease    ✓ ● List purchase requisitions awaiting release
    zPurchaseRequisition.Delete                ✓ ● Delete/close purchase requisition
    zPurchaseRequisition.GetDetail             ✓ ● Display purchase requisition details
    zPurchaseRequisition.GetReleaseInfo        ✓ ● Detailed release information on purchase requisition
    zPurchaseRequisition.Release               ✓ ● Release purchase requisition
    zPurchaseRequisition.ResetRelease          ✓ ● Cancel release of purchase requisitions
    zPurchaseRequisition.SingleRelease         ✓   Individual release
    zPurchaseRequisition.ExistenceCheck        ✓   Check existence of object
    zPurchaseRequisition.Display               ✓   Display object
    zPurchaseRequisition.InfoReleaseReset      ✓   Info: Release cancelled
    zPurchaseRequisition.Edit                  ✓   Process purchase requisition
    zPurchaseRequisition.InfoReleaseRejected   ✓   Info: Release refused
    zPurchaseRequisition.InfoReleaseEffected   ✓   Info: Release effected
    zPurchaseRequisition.zNotifyReqRelease         zNotification of Requisition Release
```

Figure 86: Display Object Type screen

Make sure the cursor is on the ‘zNotifyReqRelease’ method and click program. Insert a soft breakpoint in the code.

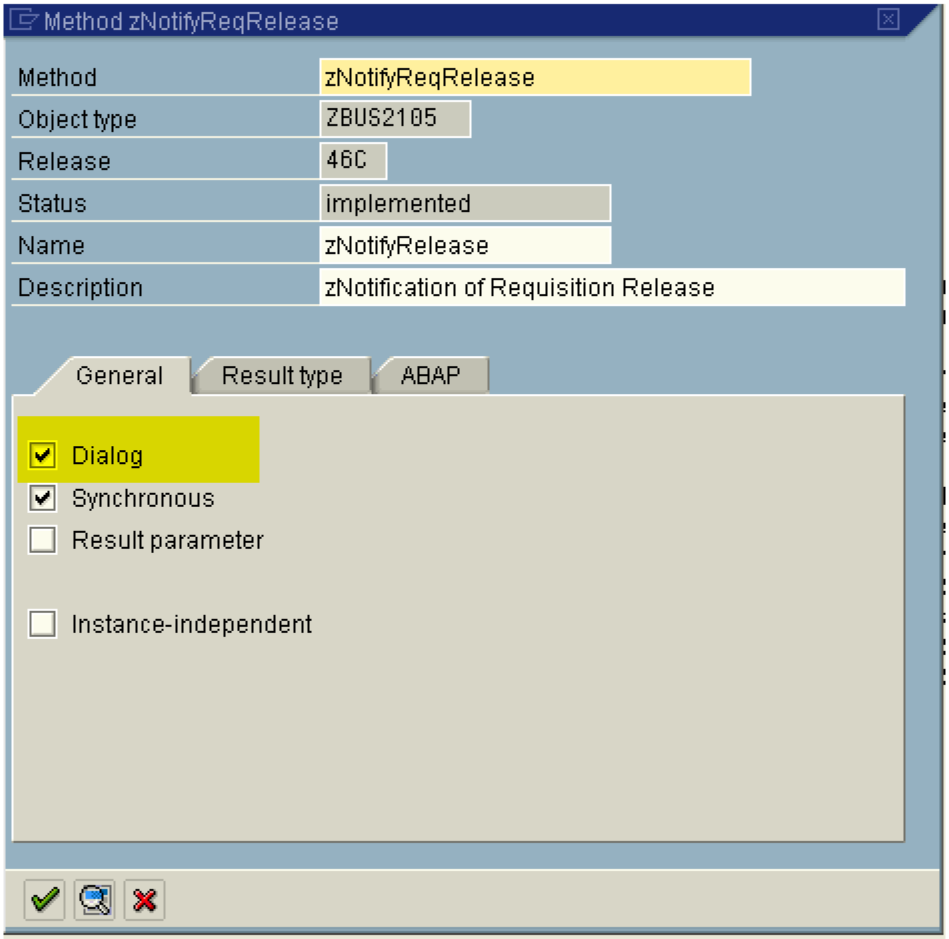


```
begin_method znotifyreqrelease changing container.

************************************************************************
* Data Declarations
************************************************************************
data: gw_purchase_req_nr like eban-banfn.
gw_purchase_req_nr = object-key-number.

data: gw_rcode   like syst-subrc.           "return code

* internal tables
data:
      itab_objcont like solisti1 occurs 5 with header line, "notif.
      itab_reclist like somlreci1 occurs 5 with header line, "receivers
      result_tab like swhactor occurs 0 with header line.

* Structures
data:
      gs_doc_chng       like sodocchgi1.

************************************************************************
*** Reciever List
************************************************************************
itab_reclist-receiver = OBJECT-_EBAN-ERNAM.   "Creator of Req
itab_reclist-rec_type = 'US'.
append itab_reclist.

************************************************************************
*** Content of Notification
************************************************************************
* refresh itab that stores the notificaiton text
refresh itab_objcont.
* NAME of office notificaiton object
gs_doc_chng-obj_name = 'SAPTips: Workflow Debugging'.
```

Figure 87: Object Type: Editor Display Program screen

Green arrow out back to the BOR definitions and (while in CHANGE mode) **double click on** ‘zPurchaseRequisition.zNotifyReqRelease’ to pull up the method attributes. **We want to set the**

dialog button so workflow does not automatically process the task. (Later we will change this back, after we have identified any issues from the debug analysis.)



Method zNotifyReqRelease

Method	zNotifyReqRelease
Object type	ZBUS2105
Release	46C
Status	implemented
Name	zNotifyRelease
Description	zNotification of Requisition Release

☑ Dialog

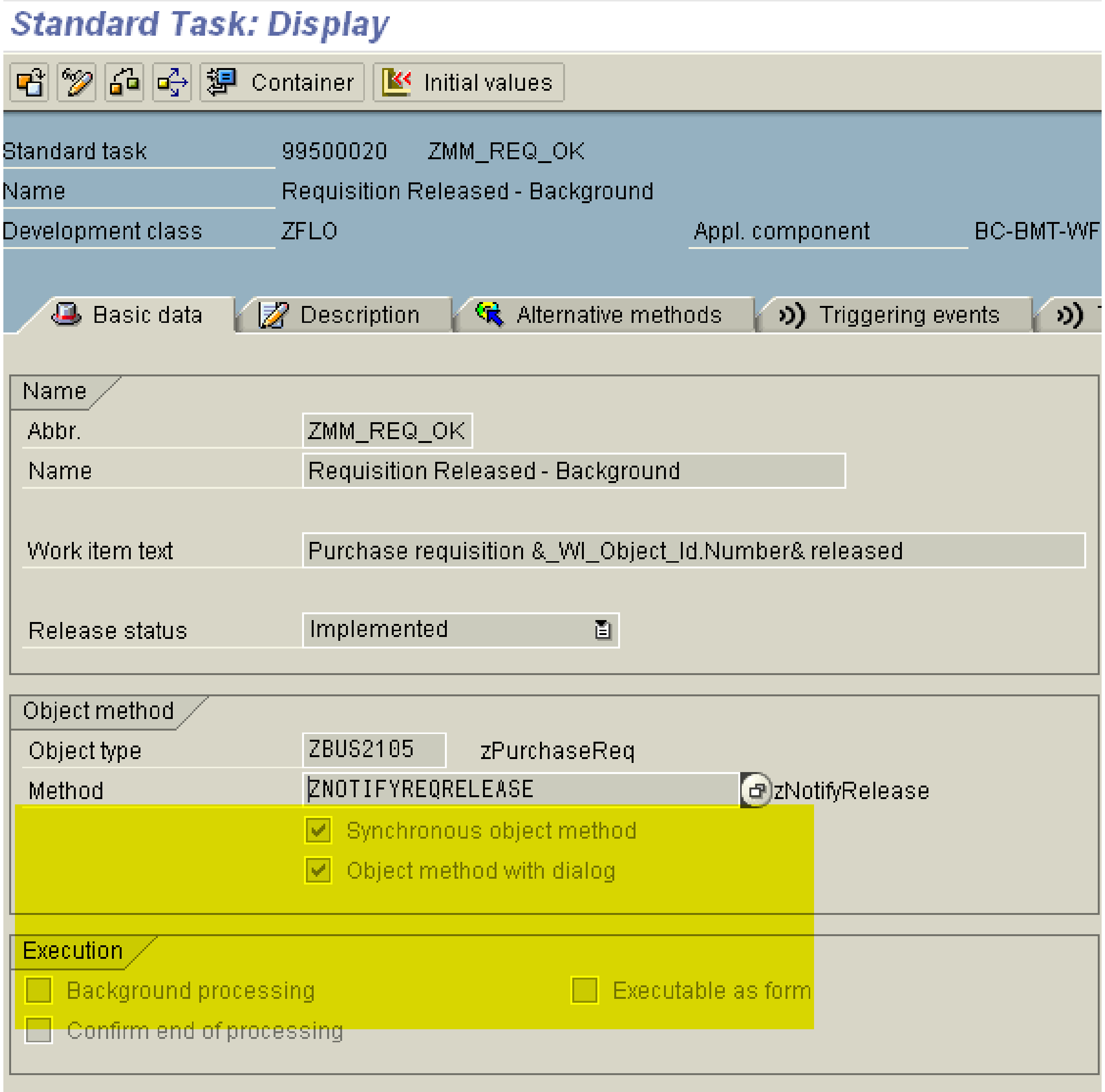
☑ Synchronous

□ Result parameter

□ Instance-independent

Figure 88: Method screen displaying General tab

Save, generate, and then green arrow back to the task description. You will notice that the attributes of the task have changed. The 'Object method with dialog' flag has been set.



Standard Task: Display

Standard task 99500020 ZMM_REQ_OK

Name Requisition Released - Background

Development class ZFLO

Appl. component BC-BMT-WF

Name

Abbr. ZMM_REQ_OK

Name Requisition Released - Background

Work item text Purchase requisition &_WI_Object_Id.Number& released

Release status Implemented

Object method

Object type ZBUS2105 zPurchaseReq

Method ZNOTIFYREQRELEASE zNotifyRelease

☑ Synchronous object method

☑ Object method with dialog

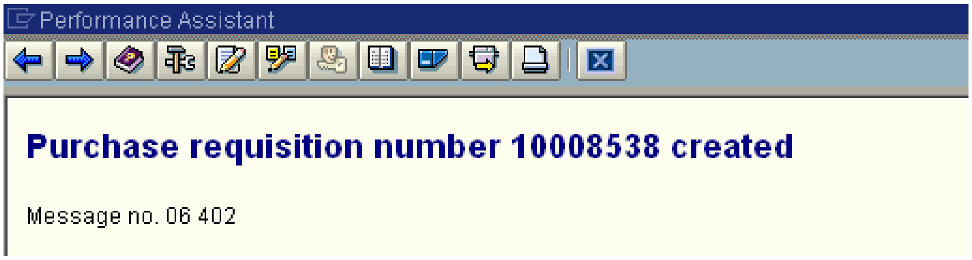
Execution

☐ Background processing ☐ Executable as form

☐ Confirm end of processing

Figure 89: Standard Task: Display screen displaying Basic data tab

Green arrow twice back to the workflow builder and regenerate your workflow. Now, we'll create a new purchase requisition (subject to the release strategy). Workflow will now stop at this task, and we will be able to execute it from the work item overview.



Performance Assistant

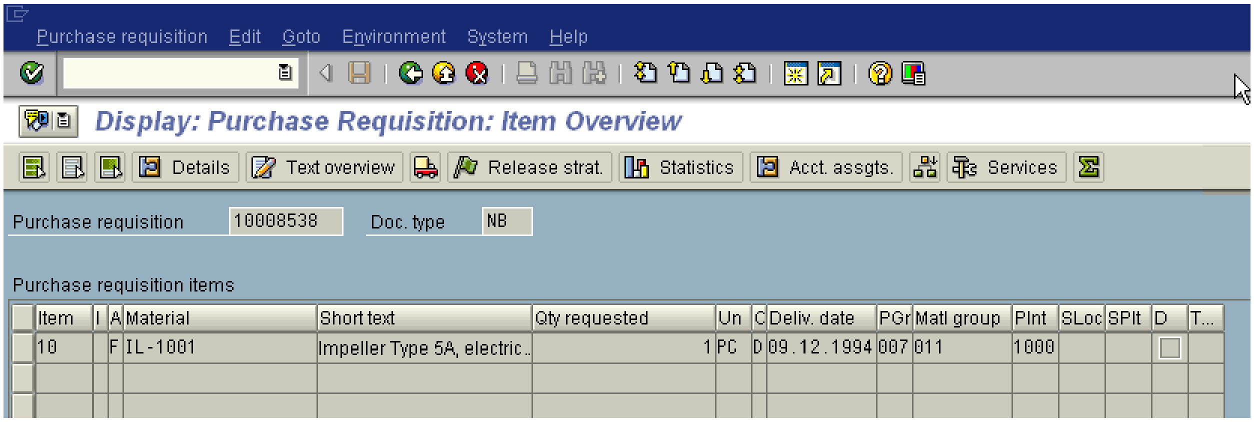
Purchase requisition number 10008538 created

Message no. 06 402

Figure 90: Performance Assistant screen

Let's use a helpful tool from SAP called Generic Object Services (GOS – available from release 4.0 onwards) to identify the workflow for the requisition we just created. GOS is activated by the application transactions and enhances the SAP functionality by allowing you to send email, add comments, review workflow history, and do many other functions. After the requisition is created, display it using transaction ME53.

> *Note: as an alternative, we can use SWI1 (work item overview to find the work item).*



Display: Purchase Requisition: Item Overview

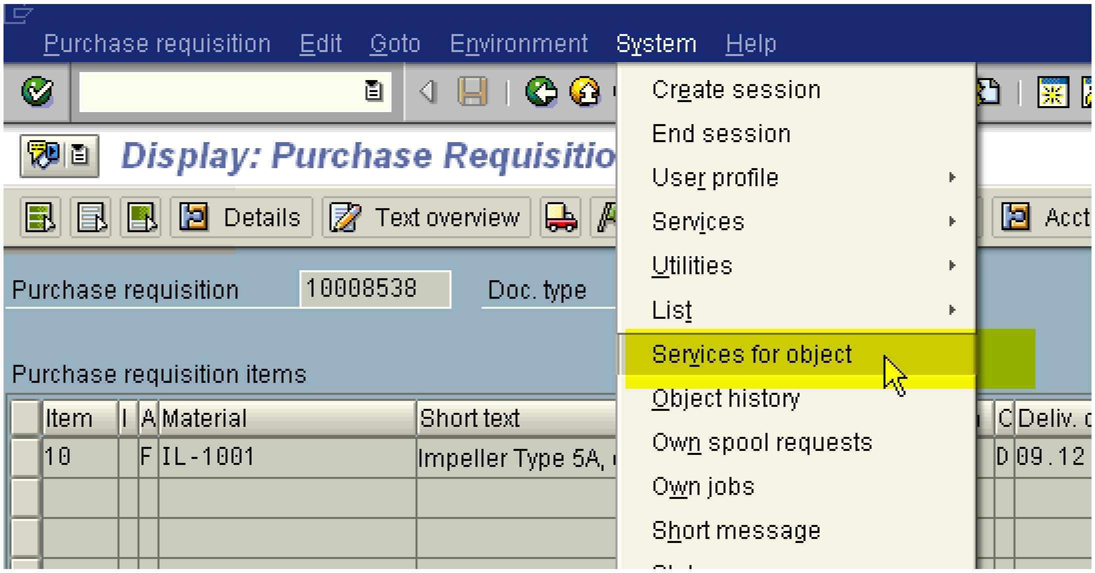
Purchase requisition 10008538 Doc. type NB

Purchase requisition items

Item	I	A	Material	Short text	Qty requested	Un	C	Deliv. date	PGr	Matl group	Plnt	SLoc	SPlt	D	T...
10		F	IL-1001	Impeller Type 5A, electric..	1	PC	D	09.12.1994	007	011	1000			□	

Figure 91: Display: Purchase Requisition: Item Overview screen

Use the menu and navigate to System > Services for Object.



Display: Purchase Requisitio

Purchase requisition 10008538 Doc. type

Purchase requisition items

Item	I	A	Material	Short text	C	Deliv. d
10		F	IL-1001	Impeller Type 5A,	D	09.12

Figure 92: Display: Purchase Requisition screen showing System pull down menu

A new window appears with services > click on the workflow overview to display all workflows assigned to the purchase requisition.

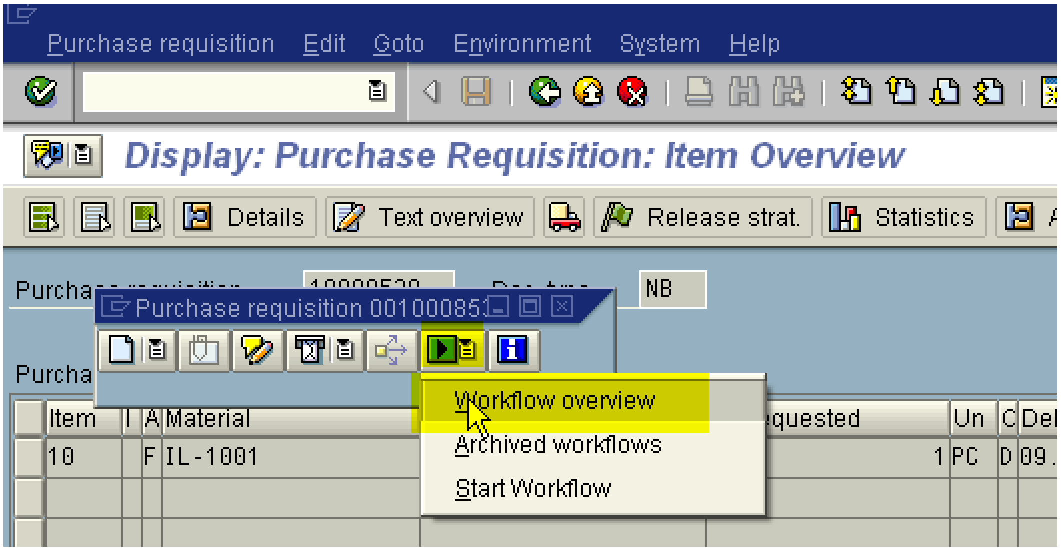
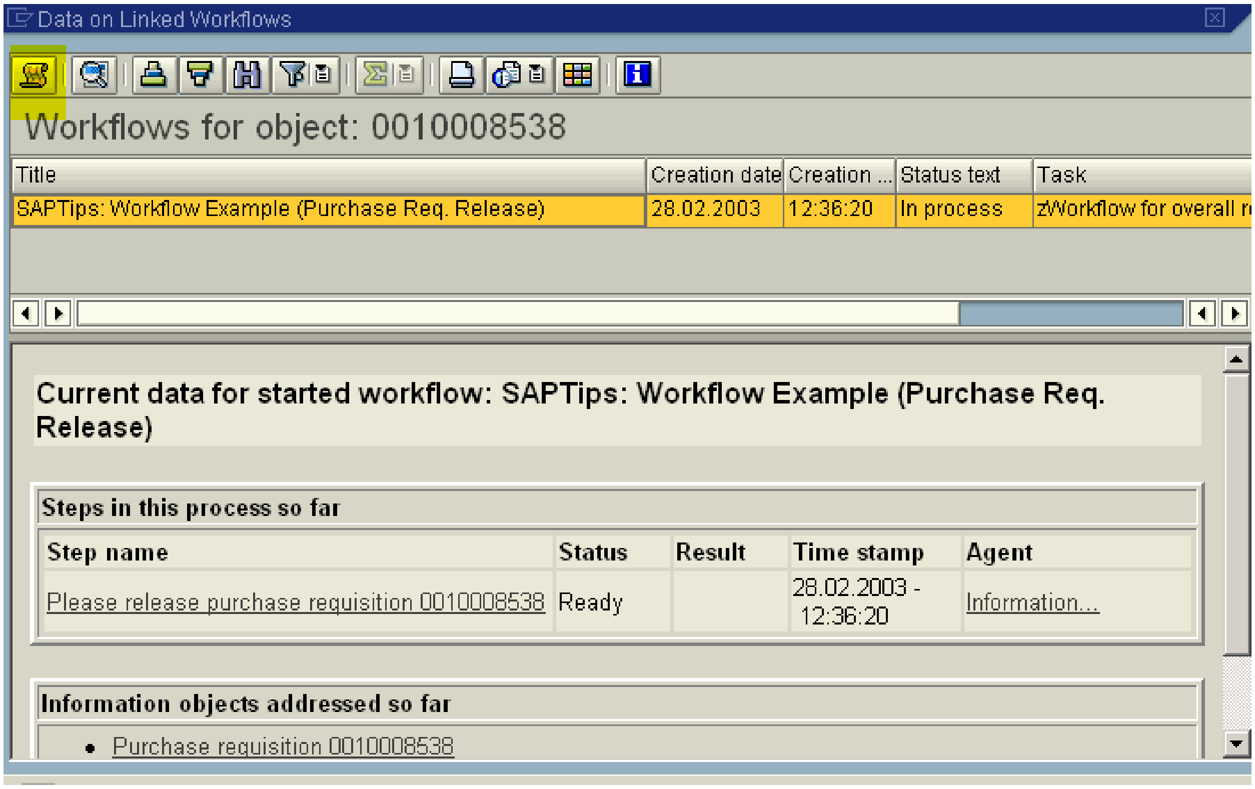


Figure 93: Display: Purchase Requisition: Item Overview screen showing Purchase requisition menu bar

Data on linked workflows appears > click on the workflow log (scroll icon).


Data on Linked Workflows

Workflows for object: 0010008538

Title	Creation date	Creation ...	Status text	Task
SAPTips: Workflow Example (Purchase Req. Release)	28.02.2003	12:36:20	In process	zWorkflow for overall r

Current data for started workflow: SAPTips: Workflow Example (Purchase Req. Release)

Steps in this process so far

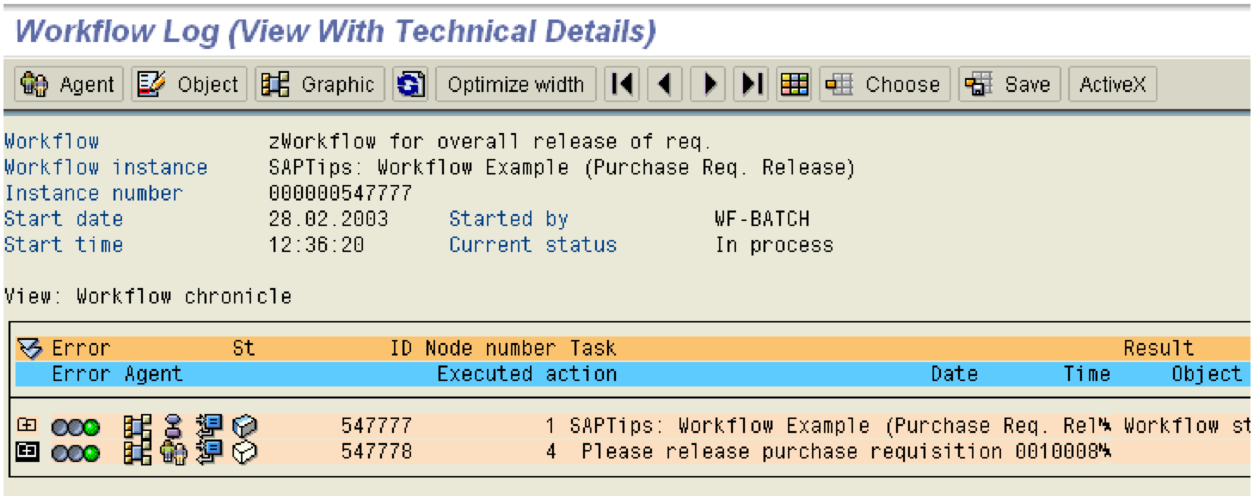
Step name	Status	Result	Time stamp	Agent
Please release purchase requisition 0010008538	Ready		28.02.2003 - 12:36:20	Information...

Information objects addressed so far

- Purchase requisition 0010008538

Figure 94: Workflows for Object screen

The workflow log appears. Since the background task we have created occurs after the first dialog (overall release) task, we must first execute the overall release. Click on work item ID 547778 (overall release task).

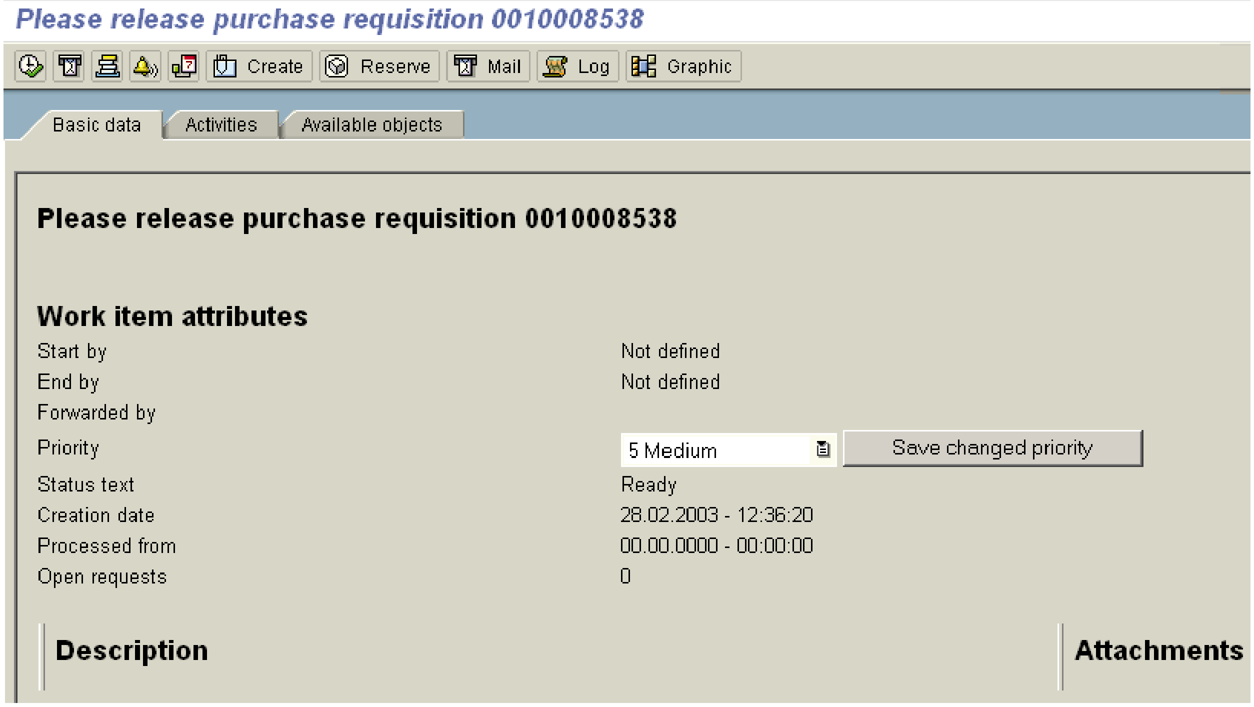


```
Workflow              zWorkflow for overall release of req.
Workflow instance     SAPTips: Workflow Example (Purchase Req. Release)
Instance number       000000547777
Start date            28.02.2003     Started by           WF-BATCH
Start time            12:36:20       Current status       In process

View: Workflow chronicle

 Error           St            ID Node number Task                                  Result
 Error Agent                      Executed action                      Date    Time    Object

                           547777         1 SAPTips: Workflow Example (Purchase Req. Rel Workflow st
                           547778         4  Please release purchase requisition 0010008
```

Figure 95: Workflow Log (View With Technical Details)

By clicking overall release work item 547778, the work item execution window appears. Execute the work item.

> *Note: if you cannot execute the work item from SWI1, then you are most likely not assigned as an agent to the task. The easiest way to resolve this is to change the task attribute to 'General' as described previously in this White Paper.*



Please release purchase requisition 0010008538

Please release purchase requisition 0010008538

Work item attributes

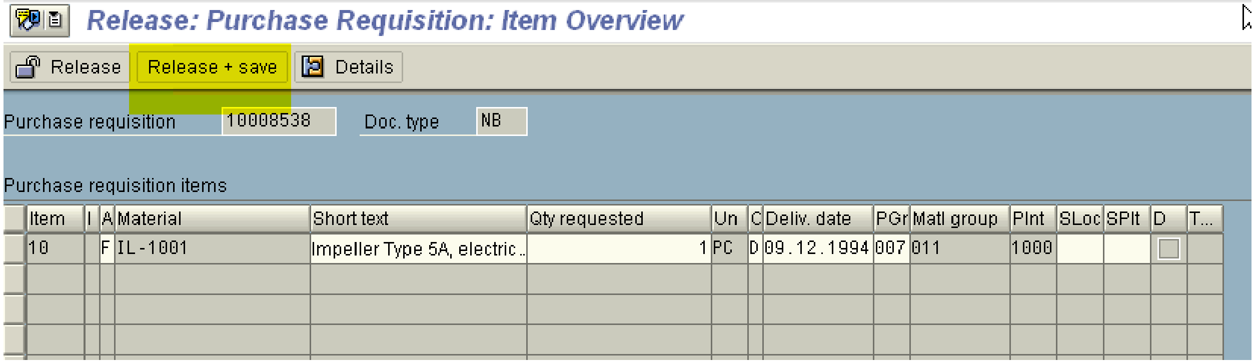
Start by	Not defined
End by	Not defined
Forwarded by	
Priority	5 Medium
Status text	Ready
Creation date	28.02.2003 - 12:36:20
Processed from	00.00.0000 - 00:00:00
Open requests	0

Description

Attachments

Figure 96: Please release purchase requisition screen displaying Basic data tab

By executing the work item, the task (method) calls transaction ME54 to release the req. Click ‘Release + save’ and release the requisition.



Release: Purchase Requisition: Item Overview

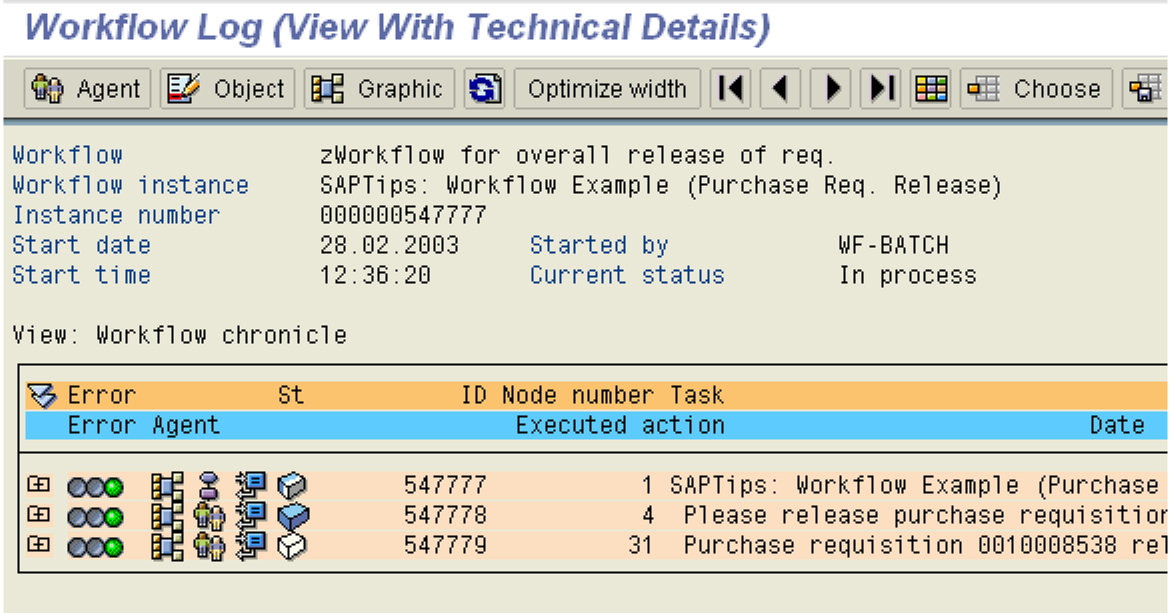
Purchase requisition 10008538 Doc. type NB

Purchase requisition items

Item	I	A	Material	Short text	Qty requested	Un	C	Deliv. date	PGr	Matl group	Plnt	SLoc	SPlt	D	T...
10		F	IL-1001	Impeller Type 5A, electric..	1	PC	D	09.12.1994	007	011	1000			□	

Figure 97: Release: Purchase Requisition: Item Overview screen

After the requisition is released, you return to the work item > green arrow back to the workflow builder. You will notice that workflow now stopped on our background task (here it is work item 547779) **because we set it to a dialog task**.



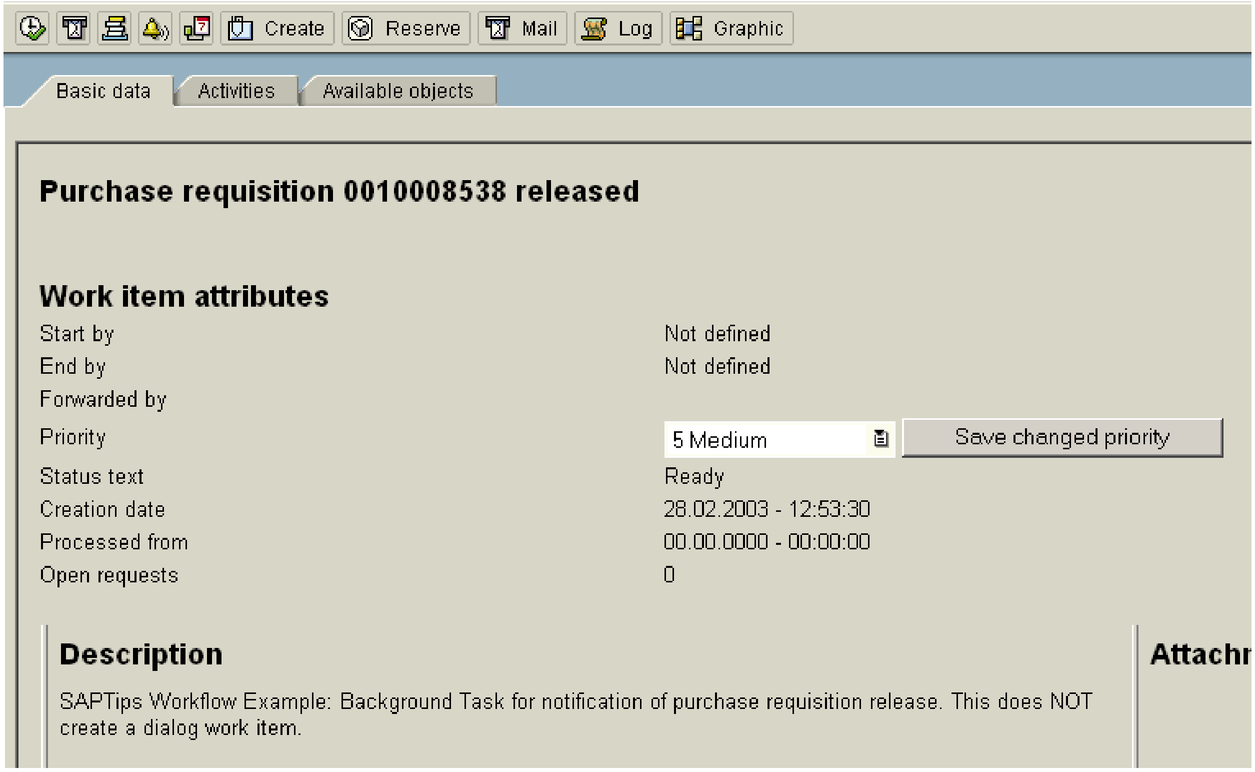
Workflow zWorkflow for overall release of req.
Workflow instance SAPTips: Workflow Example (Purchase Req. Release)
Instance number 000000547777
Start date 28.02.2003 Started by WF-BATCH
Start time 12:36:20 Current status In process

View: Workflow chronicle

Error	St	ID	Node number	Task
Error Agent		Executed action		Date
		547777	1	SAPTips: Workflow Example (Purchase
		547778	4	Please release purchase requisitior
		547779	31	Purchase requisition 0010008538 rel

Figure 98: Workflow Log (View With Technical Details) screen

We can click work item id 547779 to launch the work item execution screen.



Purchase requisition 0010008538 released

Work item attributes

Start by	Not defined
End by	Not defined
Forwarded by	
Priority	5 Medium
Status text	Ready
Creation date	28.02.2003 - 12:53:30
Processed from	00.00.0000 - 00:00:00
Open requests	0

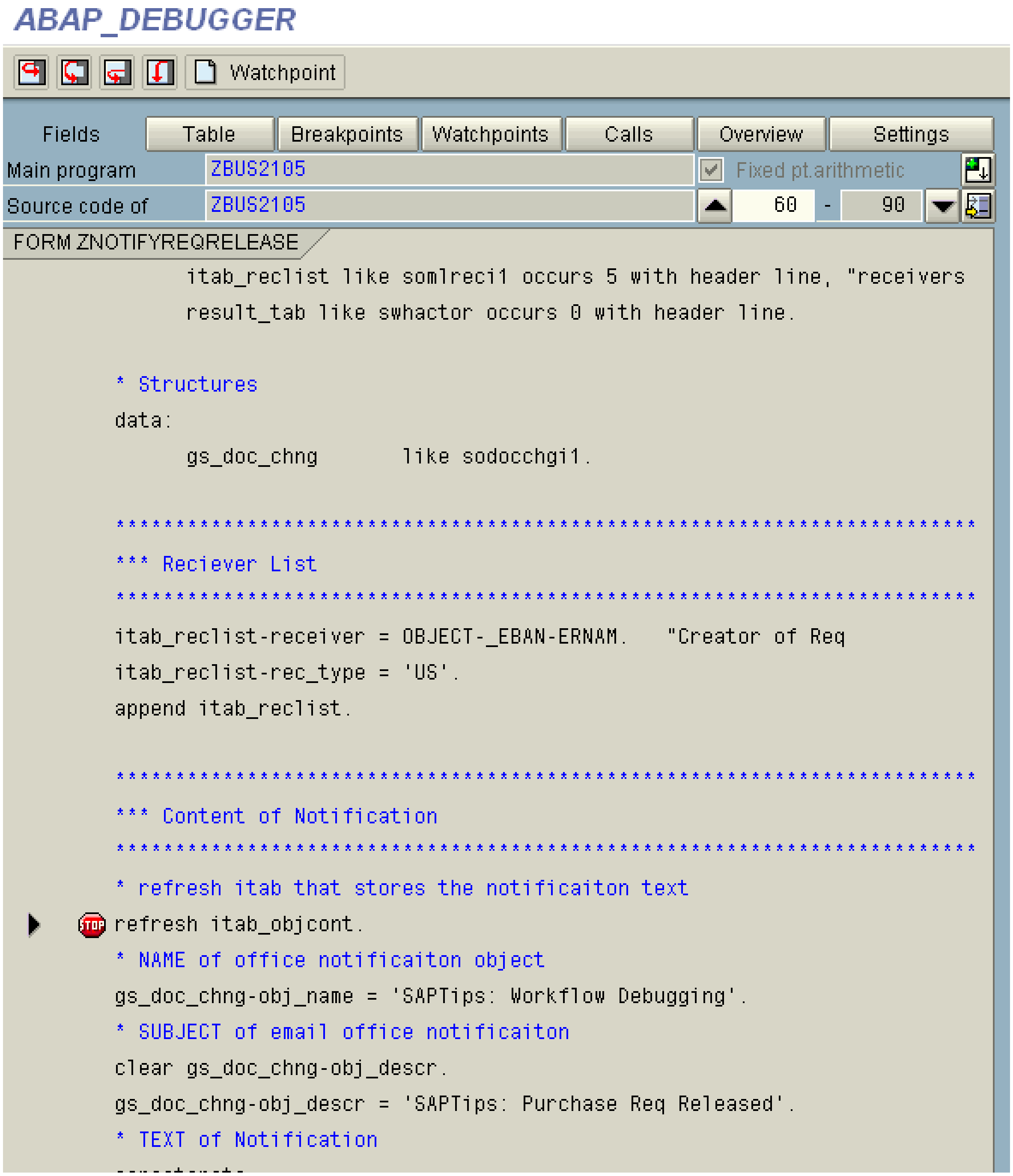
Description

SAPTips Workflow Example: Background Task for notification of purchase requisition release. This does NOT create a dialog work item.

Attachr

Figure 99: Work item execution screen displaying Basic data tab

Execute the work item. You will now be able to debug the background method associated with the task. Technically, we have made the method dialog to debug it. After you have finished debugging the method, be sure to change the attributes back to background (switch off the dialog indicator in SWO1) and then regenerate the workflow via transaction PFTC.



Main program ZBUS2105 ☑ Fixed pt.arithmetic

Source code of ZBUS2105 60 - 90

FORM ZNOTIFYREQRELEASE

```
      itab_reclist like somlreci1 occurs 5 with header line, "receivers
      result_tab like swhactor occurs 0 with header line.


* Structures
data:
      gs_doc_chng       like sodocchgi1.

**********************************************************************
*** Reciever List
**********************************************************************
itab_reclist-receiver = OBJECT-_EBAN-ERNAM.   "Creator of Req
itab_reclist-rec_type = 'US'.
append itab_reclist.

**********************************************************************
*** Content of Notification
**********************************************************************
* refresh itab that stores the notificaiton text
refresh itab_objcont.
* NAME of office notificaiton object
gs_doc_chng-obj_name = 'SAPTips: Workflow Debugging'.
* SUBJECT of email office notificaiton
clear gs_doc_chng-obj_descr.
gs_doc_chng-obj_descr = 'SAPTips: Purchase Req Released'.
* TEXT of Notification
```

Figure 100: ABAP_DEBUGGER screen

Summary of Trouble Shooting Transactions

Once you are able to initiate the debugging sessions, all the standard ABAP debugging principles apply. Here's a review of some important transactions to assist in troubleshooting your workflows.

#	Transaction	Description	Relevant Notes
1.	SWI1	Work item overview	This is the most used transaction to display workflows created in the SAP systems. Use the technical workflow log to analyze single-step tasks in a multi-step workflow. Review the container element parameter values passed in to each task, change container element parameters to test different variations of the workflow, and determine to whom the work items are assigned (agent resolution).
2.	PFTC	This is the general task maintenance transaction used to develop multi-step and single-step tasks. This is the entrance point to the workflow builder process diagram.	Later versions of SAP use the following transactions: PFTC_CHG Change Tasks PFTC_COP Copy Tasks PFTC_DEL Delete Tasks PFTC_DIS Display Tasks PFTC_INS Create Tasks PFTC_STR Task Maintenance -> Be sure to request authorization to PFTC since it is less cumbersome to use than the individual transactions. In single-step tasks, use the menu and go to 'Additional Data' > 'Agent Assignment' to check the task attributes. Remember 'General' tasks can be executed by anybody.
3.	SWU3	The area menu IMG for setting up the workflow engine.	Several steps are required to set up the engine. Errors usually occur w/ the RFC destination (see transaction SWUB) below. Use the 'Start verification workflow' to validate that the workflow engine is set up correctly.
4.	SWUB	Workflow RFC destination	Synchronize the password and use the test button to make sure the RFC destination is working correctly. Often errors exist because someone

#	Transaction	Description	Relevant Notes
			changed the WF-BATCH user id password.
5.	SM21	System Log	If you cannot figure out why a particular workflow is **not** working, always check the system log.
6.	ST22	Short Dump Log	If you cannot figure out why a particular workflow is **not** working, always check for short dumps.
7.	SWE2	Event Linkage Table	If you are expecting workflows and not receiving any, make sure the event is linked to a workflow receiver.
8.	SWELS	Turn on/off the event trace log	This can help resolving event issues. Validate, at minimum, that an event is being created and then check SWE2 for the event linkage. If no event is being created, then there's most likely a problem with the activation of events in the application configuration or other areas. **IMPORTANT**: **in production, you should turn off the event log for daily use and only turn it on when troubleshooting**. If the log is on, it has a tendency to fill up the log tables and cause errors if the system is not properly maintained.
9.	SWEL	Display Event Trace	Display any events you expected to be triggered.
10.	SWU_OBUF	Synchronize run time buffer	As of 4,6, SAP includes a task and org assignment buffer for increased performance. Any time org assignments to tasks have been made, **be sure to sync the buffers**. Otherwise, new agent assignment rules may not be taken into consideration, i.e., you may have fixed an issue but the buffer is using the old data!

#	Transaction	Description	Relevant Notes
11.	SWUS	The workflow test tool.	This is very helpful to start workflows for transactions where no workflows were created but were supposed to have been. It's also a GREAT tool to start debugging tasks after you inserted a breakpoint. Review other work items for similar tasks via transaction SWI1 (display the container elements) if you need help identifying values for the 'Input Data'.
12.	SWUE	Workflow event test tool.	Similar to the SWUS, but this creates the event. Again, use SWI1 to identify container elements (event > ws).
13.	SM58	tRFC buffer errors	If any workflow engine issues exist, the workflow will log errors in the tRFC event tables. The use of SM21, ST22, and SM58 are essential in identifying workflow issues.
14.	SCOT	SAP Connect Configuration	Any notifications (not work items) will be logged here. If there are issues with end users receiving notifications review SCOT. Most often, the issue is that program RSCONN01 is not running to send notification out to the SENDMAIL or exchange server.
15.	SWLD	Old area menu for workflow development	Exit out of the easy menu (/n), then enter /nSWLD to get to the old area menu.
16.	SWPC	Re-start workflows after system crash	
17.	SWIA	Execute work item w/o agent check	This is a great transaction to bypass agent resolution issues and focus on other problems.
18.	SWPR	Re-start workflow after error.	

SAP Workflow: A Comprehensive Guide to Troubleshooting and Debugging

Thomas Nittmann, Dataxstream, LLC. Thomas is a principal at Dataxstream, LLC, and has over nine years of SAP experience. After implementing SAP Plant Maintenance and Materials Management in Germany, he joined Ernst and Young's Management Consulting practice in 1995, and later co-founded Dataxstream. He focuses on SAP Cross Application integration with Logistics modules, and specializes in workflow automation for both R/3 and mySAP® products. Thomas can be reached at Thomas@dataxstream.com or visit www.dataxstream.com

NO WARRANTY: This documentation is delivered as is, and Klee Associates, Inc. makes no warranty as to its accuracy or use. Any use of this documentation is at the risk of the user. Although we make every good faith effort to ensure accuracy, this document may include technical or other inaccuracies or typographical errors. Klee Associates, Inc. reserves the right to make changes without prior notice.

NO AFFILIATION: Klee Associates, Inc. and this publication are not affiliated with or endorsed by SAP AG. SAP AG software referenced on this site is furnished under license agreements between SAP AG and its customers and can be used only within the terms of such agreements. SAP AG and mySAP are registered trademarks of SAP AG.

All other company and product names used herein may be trademarks or registered trademarks of their respective owners.