Class: ZCL_IM__EVENTOS_WF_CA_SP

Status: Active
Implemented interfaces
IF_EX_WORKBREAKDOWN_UPDATE (BAdI-Interface IF_EX_WORKBREAKDOWN_UPDATE)

Attributes

Description: Classe de implementação para implementação add in Z_EVENTOS_
Instantiation: Public
Final

Not released
Fixed pt.arithmetic
Category: General Object Type
Package: ZBRPS
Original lang.: PT

Created by: ABAP
Created on: 20.01.2013
Last changed by: ABAP
Last changed on: 21.02.2014

Forward declaration type grps.
ABAP

Attribute

Private attribute

Attrib.	Cat	Description	Ref. Type	Init. value
AT_CALLSTACK	Inst	Pilha de chamadas ABAP	TYPE ABAP_CALLSTACK	
AO_TVARV	Stat.	TVARV Retriever	TYPE REF TO ZCL_TVARV_RETRIEVER	

Methods

Public methods

IF_EX_WORKBREAKDOWN_UPDATE~AT_SAVE

```
METHOD if_ex_workbreakdown_update~at_save.
ENDMETHOD.
```

IF_EX_WORKBREAKDOWN_UPDATE~BEFORE_UPDATE

```
METHOD if_ex_workbreakdown_update~before_update.
  DATA: st_wbs_element TYPE cjbai_wbselement,
        lt_tj02t       TYPE TABLE OF tj02t.
  DATA: l_obtyp   TYPE jsto-obtyp,
        l_stsma   TYPE jsto-stsma,
        l_stonr   TYPE tj30-stonr,
        lt_status TYPE TABLE OF jstat,
        st_status TYPE jstat.
  DATA: l_object_type TYPE swr_struct-object_typ,
        l_object_key  TYPE swr_struct-object_key,
        l_event       TYPE swr_struct-event,
        lt_worklist   TYPE TABLE OF swr_wihdr.
```

```
* >>> chamado #958_2 - r.quadros - Justificativa de alteração de status
  DATA: lo_api         TYPE REF TO lcl_api,
        lo_retriever   TYPE REF TO zclps_nt_email_addr_retriever,
        dref           TYPE REF TO data,
        lv_stsma       TYPE j_stsma,
        lt_text        TYPE catsxt_longtext_itab,
        lt_container   TYPE swrtcont,
        lw_container   TYPE swr_cont.
  FIELD-SYMBOLS: <lv_text>   TYPE txline,
                 <lw_status> TYPE jstat.
* <<< chamado #958_4 - r.quadros
  IF sy-uname = 'WF-BATCH'.
    RETURN.
  ENDIF.
* >>> chamado #958_4 - r.quadros
  lo_api ?= me->get_api( ).
  ao_tvarv->get_single( EXPORTING i_vname = 'STSMA_J'
                        IMPORTING ex_value = lv_stsma ).
* <<< chamado #958_2 - r.quadros - Justificativa de alteração de status
* Lê os elementos PEP
  LOOP AT it_wbs_element INTO st_wbs_element
    WHERE stufe = 2.
    CLEAR: l_obtyp,
           l_stsma,
           l_stonr.
    REFRESH lt_status.
*   Busca os status que estão sendo salvos
    CALL FUNCTION 'STATUS_READ'
      EXPORTING
        objnr            = st_wbs_element-objnr
      IMPORTING
        obtyp            = l_obtyp
        stsma            = l_stsma
        stonr            = l_stonr
      TABLES
        status           = lt_status
      EXCEPTIONS
        object_not_found = 1
        OTHERS           = 2.
    CHECK sy-subrc EQ 0.
* <<< chamado #958_4 - r.quadros
    IF l_stsma <> lv_stsma.
      CONTINUE.
    ENDIF.
* >>> chamado #958_4 - r.quadros
*   Mantém apenas os status desativados
    DELETE lt_status WHERE inact IS INITIAL.
*   Mantém apenas os status "Contrato assinado"(E0001) e "Sinal pago" (E0002)
    DELETE lt_status WHERE stat NE 'E0001'
                       AND stat NE 'E0002'.
*                       AND STAT NE 'I0005'.
    CHECK lt_status[] IS NOT INITIAL.
    LOOP AT lt_status INTO st_status.
      SELECT COUNT( * ) FROM jcds
```

```
        UP TO 1 ROWS
        WHERE objnr EQ st_wbs_element-objnr
          AND stat  EQ st_status-stat
          AND inact EQ st_status-inact
          AND chgnr GT 1.
      CHECK sy-subrc NE 0.
*     Verifica se existe algum usuário parametrizado
*      SELECT COUNT( * ) FROM ztbps_wf01 * >>> chamado #958_2
*        UP TO 1 ROWS
*        WHERE posid EQ st_wbs_element-posid
*          AND estat EQ st_status-stat
*          AND ativo EQ 'X'.
*      CHECK sy-subrc EQ 0. << chamado #958_2
* >>> chamado #958_2 - r.quadros - Justificativa de alteração de status
      FREE lo_retriever.
      CREATE OBJECT lo_retriever
        EXPORTING
          p_posid = st_wbs_element-posid
          p_estat = st_status-stat.
      CHECK lo_retriever->are_there_recipients( ) = abap_true.
* <<< chamado #958_2 - r.quadros - Justificativa de alteração de status
      CLEAR: l_object_type,
             l_object_key.
      l_object_type = 'Z_CA_SP'.
      CONCATENATE st_wbs_element-pspnr st_wbs_element-poski
        INTO l_object_key.
      CASE st_status-stat.
        WHEN 'E0001'.     " Contrato assinado
          l_event       = 'Assinado'.
        WHEN 'E0002'. " Sinal Pago
          l_event       = 'Pago'.
        WHEN OTHERS.
          CONTINUE.
      ENDCASE.
* >>> chamado #958_2 - r.quadros - Justificativa de alteração de status
      CREATE DATA dref TYPE txline.
      ASSIGN dref->* TO <lv_text>.
      CASE abap_true.
        WHEN sy-batch.
          <lv_text> = 'Processamento em background'(002).
          APPEND <lv_text> TO lt_text.
        WHEN me->was_called_by( 'SAPLCNIF_STATUS_2054' ).
          <lv_text> = 'Processamento por BAPI'(003).
          APPEND <lv_text> TO lt_text.
        WHEN OTHERS.
          lt_text = me->get_justification( ).
      ENDCASE.
* <<< chamado #958_2 - r.quadros - Justificativa de alteração de status
*      CALL FUNCTION 'SAP_WAPI_CREATE_EVENT' >>> chamado #958_2
*        EXPORTING
*          object_type = l_object_type
*          object_key  = l_object_key
*          event       = l_event
*          commit_work = space. <<< chamado #958_2
* >>> chamado #958_2 - r.quadros - Justificativa de alteração de status
      lw_container-element = 'Justification'.
      LOOP AT lt_text ASSIGNING <lv_text>.
        lw_container-value = <lv_text>.
        APPEND lw_container TO lt_container.
      ENDLOOP.
      TRY .
          lo_api->raise_event( p_objty      = l_object_type
                               p_objky      = l_object_key
                               p_event      = l_event
                               pt_container = lt_container ).
        CATCH cx_bapi_error.
          CONTINUE.
      ENDTRY.
* <<< chamado #958_2 - r.quadros - Justificativa de alteração de status
    ENDLOOP.
  ENDLOOP.
* >>> chamado #958_2 - r.quadros - Justificativa de alteração de status
  FREE: lo_api, lo_retriever.
* <<< chamado #958_2 - r.quadros - Justificativa de alteração de status
ENDMETHOD.
```

CLASS_CONSTRUCTOR

Description: CLASS_CONSTRUCTOR
Static method

```
method CLASS_CONSTRUCTOR.
*{   INSERT         ED2K932611                                        1
  CREATE OBJECT ao_tvarv
    EXPORTING
      p_name = 'Z958'.
*
*}   INSERT
endmethod.
```

Private methods

GET_API

Description: Provides an API calls mediator.
Instance mthd

Returning parameter

```
 VALUE(RO_API) TYPE REF TO OBJECT
```

```
method GET_API.
*{   INSERT         ED2K931383                                        1
  DATA lo_api TYPE REF TO lcl_api.
  CREATE OBJECT lo_api.
  RO_API ?= lo_api.
  FREE lo_api.
*
*}   INSERT
endmethod.
```

SET_CALLSTACK

Description: Updates the call stack to its current status.
Instance mthd

```
method SET_CALLSTACK.
*{   INSERT         ED2K931873                                        1
  DATA lo_api TYPE REF TO lcl_api.
  lo_api ?= get_api( ).
  me->at_callstack = lo_api->SYSTEM_CALLSTACK( ).
  free lo_api.
*
*}   INSERT
endmethod.
```

WAS_CALLED_BY

Description: Returns TRUE if the argument is in the call stack.
Instance mthd

Importing parameter
`P_REPID TYPE SYREPID` (Programa ABAP, programa básico atual)

Returning parameter
`VALUE(R_BOOLEAN) TYPE ABAP_BOOL` (Boolean variable.)

```
method WAS_CALLED_BY.
*{   INSERT         ED2K931873                                        1
  clear me->at_callstack.
  me->SET_CALLSTACK( ).
  READ TABLE me->at_callstack WITH KEY mainprogram = p_repid
   TRANSPORTING NO FIELDS.
  IF sy-subrc = 0.
    r_boolean = abap_true.
  ENDIF.
*
*}   INSERT
endmethod.
```

GET_JUSTIFICATION

Description: Returns the Justification text.
Static method

Returning parameter
`VALUE(RT_TEXT) TYPE CATSXT_LONGTEXT_ITAB` (Tabela de textos descritivos para CATSXT)

```
method GET_JUSTIFICATION.
*{   INSERT         ED2K931920                                        1
  DATA: lv_title  TYPE sychar70,
        ls_text1  TYPE string,
        ls_text2  TYPE string,
        ls_title  TYPE string,
        lv_answer TYPE char01,
        lo_api    TYPE REF TO lcl_api.
  lv_title = 'Por favor, justifique a alteração de estado.'(001).
  ls_title = 'Justificativa Obrigatória!'(004).
  ls_text1 =
        'É necessário que você justifique a alteração de status.'(005).
  ls_text2 = 'Deseja cancelar e perder as alterações?'(006).
  create OBJECT lo_api.
  CLEAR rt_text.
  WHILE rt_text IS INITIAL .
    rt_text = lo_api->get_user_text( lv_title ).
    IF rt_text is INITIAL.
      CLEAR lv_answer.
      TRY .
        lv_answer = lo_api->cancelling_decision( p_text1 = ls_text1
                                                 p_text2 = ls_text2
                                                 p_title = ls_title ).
      CATCH cx_bapi_error.
        CONTINUE.
      ENDTRY.
      IF lv_answer = '1'.
        free lo_api.
        LEAVE to CURRENT TRANSACTION AND SKIP FIRST SCREEN.
      ENDIF.
    ENDIF.
  ENDWHILE.
  free lo_api.
*
*}   INSERT
endmethod.
```

Redefined Methods

Local Types

```
*"* local classes for public class ZCL_IM__EVENTOS_WF_CA_SP
*"* use this source file for any type declarations you need
*"* in your method implementation
```

Local class definitions

```
*{   INSERT         ED2K931383                                        1
*---------------------------------------------------------------------*
* Class LCL_API
*  OVERVIEW: Objects of this class mediates APIs calls. It aims to
*            isolate this development from external dependencies, as
*            might be desirable in tests.
*            The instances of this object is provided by the private
*            method GET_API of the main class
*            (ZCL_IM_WF_US_ST_RL_JUSTIF). By redefining the method
*            GET_API one can provide appropriated stubs, which redefine
*            the desired methods of LCL_API, instead of calling the
*            real API.
*---------------------------------------------------------------------*
CLASS lcl_api DEFINITION.
  PUBLIC SECTION.
  METHODS: status_read IMPORTING p_objnr TYPE j_objnr
    RETURNING VALUE(rt_status) TYPE tt_jstat
    RAISING cx_bapi_error,
*---------------------------------------------------------------------*
```

```
* Method STATUS_READ
*----------------------------------------------------------------------*
* REQUIRES: The object number must be provided.
* EFFECTS.: If the API processing fails, returns exception type
*           CX_BAPI_ERROR, otherwise returns a table with status data.
*----------------------------------------------------------------------*
* ->   P_OBJNR      Object number.
* <-   RT_STATUS    Table type of JSTAT.
*----------------------------------------------------------------------*
           get_user_text IMPORTING p_title    TYPE sychar70
                     RETURNING VALUE(rt_text) TYPE catsxt_longtext_itab,
*----------------------------------------------------------------------*
* Method GET_USER_TEXT
*----------------------------------------------------------------------*
* REQUIRES: The title of the pop up window must be provided.
* EFFECTS.: Returns a table with the lines of the text provided by user
*           in a pop up for text edition.
*----------------------------------------------------------------------*
* ->   P_TITLE      The text to be shown as pop_up.
* <-   RT_TEXT      The text typed by the user.
*----------------------------------------------------------------------*
           raise_event IMPORTING p_objty      TYPE SWO_OBJTYP
                                 p_objky      TYPE SWO_TYPEID
                                 p_event      TYPE SWO_EVENT
                                 pt_container TYPE SWRTCONT OPTIONAL
                       RAISING cx_bapi_error,
*----------------------------------------------------------------------*
* Method RAISE_EVENT
*----------------------------------------------------------------------*
* REQUIRES: BOR object type, key and event name related to the event to
*           be raised must be provided.
* EFFECTS.: If the API processing fails, returns exception type
*           CX_BAPI_ERROR, otherwise raises the related event.
* MODIFIES: Publishes the BOR event specified in the environment.
*----------------------------------------------------------------------*
* ->   P_OBJTY      BOR Object Type.
* ->   P_OBJKY      Object key.
* ->   P_EVENT      Event name.
* ->   PT_CONTAINER Information to be passed to the event.
*----------------------------------------------------------------------*
           SYSTEM_CALLSTACK
                      RETURNING VALUE(rt_callstack) TYPE ABAP_CALLSTACK,
*----------------------------------------------------------------------*
* Method SYSTEM_CALLSTACK
*----------------------------------------------------------------------*
* EFFECTS.: Returns the call stack.
*----------------------------------------------------------------------*
* <-   RT_CALLSTACK   ABAP Call Stack.
*----------------------------------------------------------------------*
           cancelling_decision IMPORTING p_title TYPE string OPTIONAL
                                         p_text1 TYPE string OPTIONAL
                                         p_text2 type string OPTIONAL
                               RETURNING VALUE(r_answer) TYPE char01
                               RAISING cx_bapi_error.
```

```
*--------------------------------------------------------------------*
* Method CANCELLING_DECISION
*--------------------------------------------------------------------*
* EFFECTS.: If the API processing fails, returns exception type
*           CX_BAPI_ERROR, otherwise shows up a pop up that allows the
*           user to decide whether she wants to abort an operation and
*           returs her decision.
*--------------------------------------------------------------------*
* ->   R_TITEL    Title line of dialog box.
* ->   R_TEXT1    First line of dialog box.
* ->   R_TEXT2    Second line of dialog box.
* <-   R_ANSWER   User decision.
*--------------------------------------------------------------------*
ENDCLASS.                    "lcl_api DEFINITION
*----------------------------------------------------------------------*
*       CLASS lcl_api IMPLEMENTATION
*----------------------------------------------------------------------*
*
*----------------------------------------------------------------------*
CLASS lcl_api IMPLEMENTATION.
  METHOD status_read.
    CALL FUNCTION 'STATUS_READ'
    EXPORTING
      objnr            = p_objnr
    TABLES
      status           = rt_status
    EXCEPTIONS
      object_not_found = 1
      OTHERS           = 2.
    IF sy-subrc <> 0.
      RAISE EXCEPTION TYPE cx_bapi_error
        EXPORTING
          t100_msgid = sy-msgid
          t100_msgno = sy-msgno
          t100_msgv1 = sy-msgv1
          t100_msgv2 = sy-msgv2
          t100_msgv3 = sy-msgv3
          t100_msgv4 = sy-msgv4.
    ENDIF.
  ENDMETHOD.                    "STATUS_READ
  METHOD get_user_text.
    CALL FUNCTION 'CATSXT_SIMPLE_TEXT_EDITOR'
    EXPORTING
      im_title              = p_title
    CHANGING
      ch_text               = rt_text.
  ENDMETHOD.
  METHOD raise_event.
    DATA: lt_msg_data TYPE SWR_MSGTAB,
          lt_return   TYPE bapiret2_t,
          lw_return   TYPE bapiret2.
    FIELD-SYMBOLS <lw_msg_data> TYPE SWR_MSTRUC.
    CALL FUNCTION 'SAP_WAPI_CREATE_EVENT'
    EXPORTING
    object_type             = p_objty
    object_key              = p_objky
    EVENT                   = p_event
    COMMIT_WORK             = space
  TABLES
    INPUT_CONTAINER         = pt_container
    MESSAGE_STRUCT          = lt_msg_data.
  READ TABLE lt_msg_data WITH KEY MSGID = 'E' TRANSPORTING NO FIELDS.
  IF sy-subrc = 0.
    LOOP AT lt_msg_data ASSIGNING <lw_msg_data>.
      lw_return-ID         = <lw_msg_data>-msgid.
      lw_return-TYPE       = <lw_msg_data>-msgty.
      lw_return-NUMBER     = <lw_msg_data>-msgno.
      lw_return-message_v1 = <lw_msg_data>-msgv1.
      lw_return-message_v2 = <lw_msg_data>-msgv2.
      lw_return-message_v3 = <lw_msg_data>-msgv3.
      lw_return-message_v4 = <lw_msg_data>-msgv4.
      MESSAGE ID     <lw_msg_data>-msgid
      TYPE   <lw_msg_data>-msgty
      NUMBER <lw_msg_data>-msgno
      WITH   <lw_msg_data>-msgv1 <lw_msg_data>-msgv2
      <lw_msg_data>-msgv3 <lw_msg_data>-msgv4
      INTO lw_return-MESSAGE.
      APPEND lw_return TO lt_return.
    ENDLOOP.
    RAISE EXCEPTION TYPE cx_bapi_error EXPORTING
      status = lt_return.
  ENDIF.
ENDMETHOD.
METHOD SYSTEM_CALLSTACK.
  CALL FUNCTION 'SYSTEM_CALLSTACK'
   IMPORTING
     CALLSTACK          = rt_callstack.
ENDMETHOD.
METHOD cancelling_decision.
  data  ls_question TYPE string.
  CONCATENATE p_text1 p_text2 INTO ls_question SEPARATED BY space.
  CALL FUNCTION 'POPUP_TO_CONFIRM'
    EXPORTING
      TITLEBAR                    = p_title
      text_question               = ls_question
    IMPORTING
      ANSWER                      = r_answer
    EXCEPTIONS
      TEXT_NOT_FOUND              = 1
      OTHERS                      = 2.
  IF sy-subrc <> 0.
    RAISE EXCEPTION TYPE cx_bapi_error
    EXPORTING
      t100_msgid = sy-msgid
      t100_msgno = sy-msgno
      t100_msgv1 = sy-msgv1
      t100_msgv2 = sy-msgv2
      t100_msgv3 = sy-msgv3
```

```
        t100_msgv4 = sy-msgv4.
    ENDIF.
  ENDMETHOD.
ENDCLASS.                    "lcl_api IMPLEMENTATION
*
*}   INSERT
```

Macros

```
*"* use this source file for any macro definitions you need
*"* in the implementation part of the class
```

Overview