How to add customer specific data fields to ECTR objects

Request

There are additional data fields (like the full name of the user who changed the object the last time) which shall be displayed at ECTR objects e.g. document or material master.

Examples

Document

Show a special classification fields in ECTR level 1 data.

Material

Show plant related or storage related data from several views in ECTR.

Solution

Backend

In the Backend a BAdI has to be implemented to determine that special data and return it to the frontend. This can be done with the following BAdI.

BAdI: /DSCSAG/ENH_DOC
Method: FOL_OPENDETAILS_AFTER

The value which should be added to an object has to be appended to the table out_customer_data.

If you decide to use customer fields for a type of object in ECTR, you have to fill these fields for all instances of this object. E.g. it will have strange side effects if you fill a customer field key only for documents of dtype "PINT", but not for dtype "PEXT".

Example code for document info records:
DATA: ls_document TYPE /dscsag/doc_infolder,
ls_doc_data TYPE bapi_doc_draw2,
ls_return TYPE bapiret2,
ls_customer_data TYPE /dscsag/name_value_dokob,
lv_classname TYPE classname, "<-- ARBITRARY CLASS NAME
lv_charname TYPE atnam, "<-- ARBITRARY CHARACTER NAME
lv_classtype TYPE klassenart VALUE '017', "<-- ARBITRARY CLASS TYPE
lt_char_val TYPE TABLE OF bapi_characteristic_values,
l_t_cl_alloc TYPE TABLE OF bapi_class_allocation.

FIELD-SYMBOLS: <ls_cl_char> TYPE bapi_characteristic_values.

LOOP AT out_document INTO ls_document.
  CLEAR: ls_doc_data, ls_return, lt_cl_alloc[], lt_char_val[].
  CALL FUNCTION 'BAPI_DOCUMENT_GETDETAIL2'
    EXPORTING
      documenttype = ls_document-dokar
      documentnumber = ls_document-doknr
      documentpart = ls_document-doktl
      documentversion = ls_document-dokvr
      getclassification = 'X'
      GETACTIVEFILES = SPACE
      GETDOCDESCRIPTIONS = SPACE
      GETDOCFILES = SPACE
      INHERITED = 'X'
    IMPORTING
      documentdata = ls_doc_data
      return = ls_return
    TABLES
      characteristicvalues = lt_char_val
      classallocations = lt_cl_alloc.
    IF ls_return-type CA 'EA'.
      APPEND ls_return TO attr_returns[].
      CONTINUE.
    ENDIF.
  *  Always fill the name for the attribute
  CLEAR: ls_customer_data.
  ls_customer_data-dokob = 'DRAW'.
  MOVE ls_document-dokar TO ls_customer_data-objky+0(3).
  MOVE ls_document-doknr TO ls_customer_data-objky+3(25).
  MOVE ls_document-dokvl TO ls_customer_data-objky+28(2).
  MOVE ls_document-dokvl1 TO ls_customer_data-objky+30(3).
  ls_customer_data-name = 'ARBITRARY_NAME'. "<-- CAN BE CHARACTER NAME HERE
  IF lt_cl_alloc[] IS NOT INITIAL AND
    lt_char_val[] IS NOT INITIAL.
    READ TABLE lt_cl_alloc TRANSPORTING NO FIELDS WITH KEY classtype = lv_classtype classname = lv_classname.
    IF sy-subrc EQ 0.
READ TABLE lt_char_val ASSIGNING <ls_cl_char> WITH KEY
charname  = lv_charname.
IF sy-subrc EQ 0.
    ls_customer_data-value = <ls_cl_char>-charvalue.
ENDIF.
ENDIF. *lt_class_alloc
ENDIF.
APPEND ls_customer_data TO out_customer_data. "<- ALSO ATTRIBUTE WITH EMPTY VALUE CAN BE APPENDED
ENDLOOP.

DISCLAIMER: This is an example to show how to use the BAdI Interface. The usage of the "fat" BAPI_DOCUMENT_GETDETAIL2 to read the classification in a loop will slow down performance of ECTR. If you know what you have to do, better make a direct select from AUSP table to read the classification values with "FOR ALL ENTRIES IN".

Example code for materials:

```
DATA: ls_material TYPE /DSCSAG/MATERIAL_TAB,
  ls_return TYPE bapiret2,
  ls_customer_data TYPE /dscsag/name_value_dokob.
LOOP AT out_matnums INTO ls_material.
  CLEAR: ls_customer_data.
  ls_customer_data-dokob = 'MARA'.
  MOVE ls_material-material TO ls_customer_data-objky.
  ls_customer_data-name = 'ZMSTAE'. "<- Must match with config in dictionary
  ls_customer_data-value = 'your new great value'.
  APPEND ls_customer_data TO out_customer_data. "<- ALSO ATTRIBUTE WITH EMPTY VALUE CAN BE APPENDED
ENDLOOP.
```

## Frontend

The additional data are read in the backend and returned to the frontend with the upper BAdI.

Now the frontend has to be configured how to show this additional data in ECTR. In the configuration a mapping for customer_data has to be defined.

In the customer dictionary file must be defined how the fields of the returned structure should be named in ECTR:
Dictionary mapping for customer data fields

# Dictionary mapping for customer data fields
plm.customdata.field.<customer_data-dokob>.<customer_data-name>=...

# Customer fields for documents
plm.customdata.field.DRAW.ZOWNER = Owner

# Customer field for change number (AENR)
# plm.customdata.field.AENR.TRUCKLIST = Trucks

# Customer fields for materials
plm.customdata.field.MARA.ZMSTAE = Z - Status

There are other data which is send from backend without any need to implement a BAdl. E.g. the append structure to the table AENR or the append structure to table STPO (via CI_STPO).

This data can be addressed in this way for materials and change numbers.

default.txt:

# Shop CI_STPO - fields in assemblies window
plm.docstructure.showZfields = ZZBOMREL;

# Show CI_STPO - fields in BOM window
plm.bom.customdata.fields = ZZPFAD

Dictionary:

Mapping of Z fields in customer specific CI - dictionary

#Mapping of Z fields in customer specific AENR - append
plm.ecm.customerfield.ZZPSPNR = WBS Element

#Mapping of Z fields in customer specific CI_STPO for document structure window
plm.docstructure.mapping.ZZBOMREL = ZZ Bom Relevant

#Mapping of Z fields in customer specific CI_STPO for BOM window
plm.customdata.field.STPO.ZZPFAD = ZZ BOM Path
Adding a new property to materials (screenshots)

1. Implement the example code for material as described above.
   Now ECTR has a new field:
   ![Image of material property implementation](image1.png)
   If this is the display, you have forgotten to add a specific name into the dictionary.

2. Fill the `customer.txt` to name the field
   ```
   DSCSAG ECTR-51 customize dictionary en
   fnc.api.generic2(ZM18_LAUNCH_DEMO_WF_ON_ECN) = plm.customdata.field.MARA.ZMSTAE = Z - Status
   ```

3. Open the Preferences in the Object Browser:
   ![Image of object browser](image2.png)
   The name has changed as intended
4. Select your value, add it to displayed columns and press ‘Apply’:

5. Now ECTR shows your value in the Object Browser: