## Preface

This assignment suffers from a bad timing, at least they do not fit my schedule. My first exams (not the oral exam for ERP systems !) begin just next week so I better prepare for them and skip questions 3 and 4. Dr. Schuldt underlined that these questions do not play an important role in the ERP exam. Thanks.

As always, I am registered under the nickname `developer30` in the SAP R/3 system.

## Question 1

*After the implementation of reports in the last exercise, we will design and implement dialog transactions that support the creation of new entries in our banking scenario.*

*This dialog transaction is supposed to consist of a single Dynpro (screen) in which entry data can be specified. In addition to the graphical user interface and the data fields contained in this screen, also control flow has to be defined (PBO and PAI modules) in a module pool (which is a special ABAP program).*

*Some nice colleague of yours has already started implementing such a dialog transaction. Unfortunately, he did not manage to complete this task. The name of the module pool he has prepared is ZHPI00SingleDynpro. Copy this program to your name space (i.e., to ZHPI30SingleDynpro) and save it in your development class. For copying, you can go to the overview on program objects in the Object Navigator (Development → ABAP Workbench → Overview → Object Navigator). There, you can choose Program and ZHPI00SingleDynpro; display this program (using the icon with glasses). With right mouse → copy ... on the name ZHPI00SingleDynpro in the list of object names, you can copy the module pool and all associated objects. Check all objects that are listed (and not only the module pool itself), i.e., also screens and user interfaces.*

*Display your copy ZHPI30SingleDynpro in the Object Navigator. Your dialog transaction consists of a single screen (Dynpro) with number 0100. In the list of program objects, you can navigate to this screen. Essentially, this screen consists of the definition of control flow (Flow Logic) and a specification of the data elements (Element List) used in a dialog screen. Go to the flow logic specification first. Here, only module calls are possible (but no ABAP commands !). The screen painter (the tool where the graphical layout of the screen is defined) can be reached by using Layout. Fields that have the same name as variables in the PBO and PAI module, respectively, are automatically copied to/from this dialog mask (when PBO or PAI is activated). Important hint: In here, xx has of course to be replaced by your login number (Remark by me: it's 30).*

*You find the ABAP sources of the PBO module and the PAI module in module pool ZHPI30SingleDynpro. Complete all these program skeletons. The dialog transaction is supposed to collect all required information on an entry (customer number, account number, and amount). The entry ID shall be determined automatically (by incrementing the maximum entry ID of this account). This information shall be written to the database in the PAI module (create a new account entry record), but only if the account balance remains positive. In addition, the account balance shall be consistently updated.*

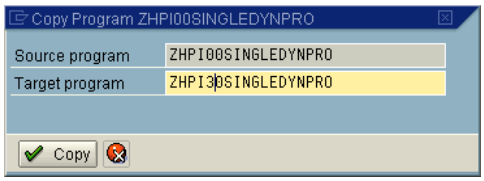
*Hint: Do not forget to explicitly activate all Dynpros. Please bear also in mind that variables are only copied to/from the GUI when they have the same name than variables in the associated modules.*

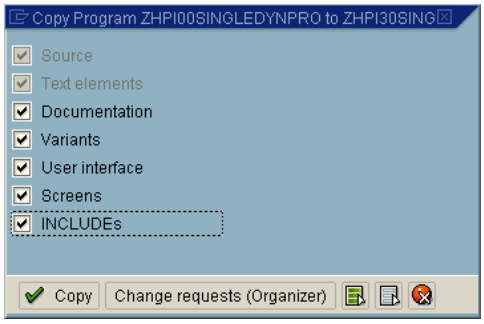
*After having implemented the Dynpro, you can define a dialog transaction. Use transaction ID ZSD30 for this purpose (via Screen → Other object ...). In the dialog box, check transaction and specify the transaction ID. You then have to specify a program name and the number of the first screen to be displayed in this transaction. After having defined a transaction, you can invoke your Dynpro via the transaction code /nZSD30 in the command field (for the ease of usage, you can add this transaction ID to your personal favourites). Without the definition of a transaction ID, you can only simulate this transaction, but you cannot execute it.*

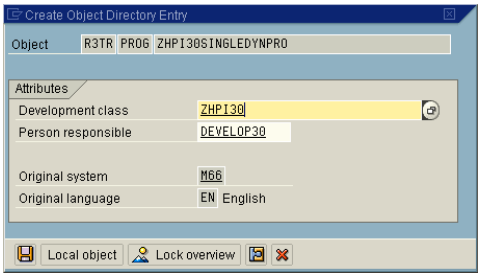
*Further Hint: If you cannot find the command field, it can be displayed by clicking on the small triangle in the upper left corner of the SAPGui window which will then make the command field appear.*

Phew, quite a lot to do. Here we go:

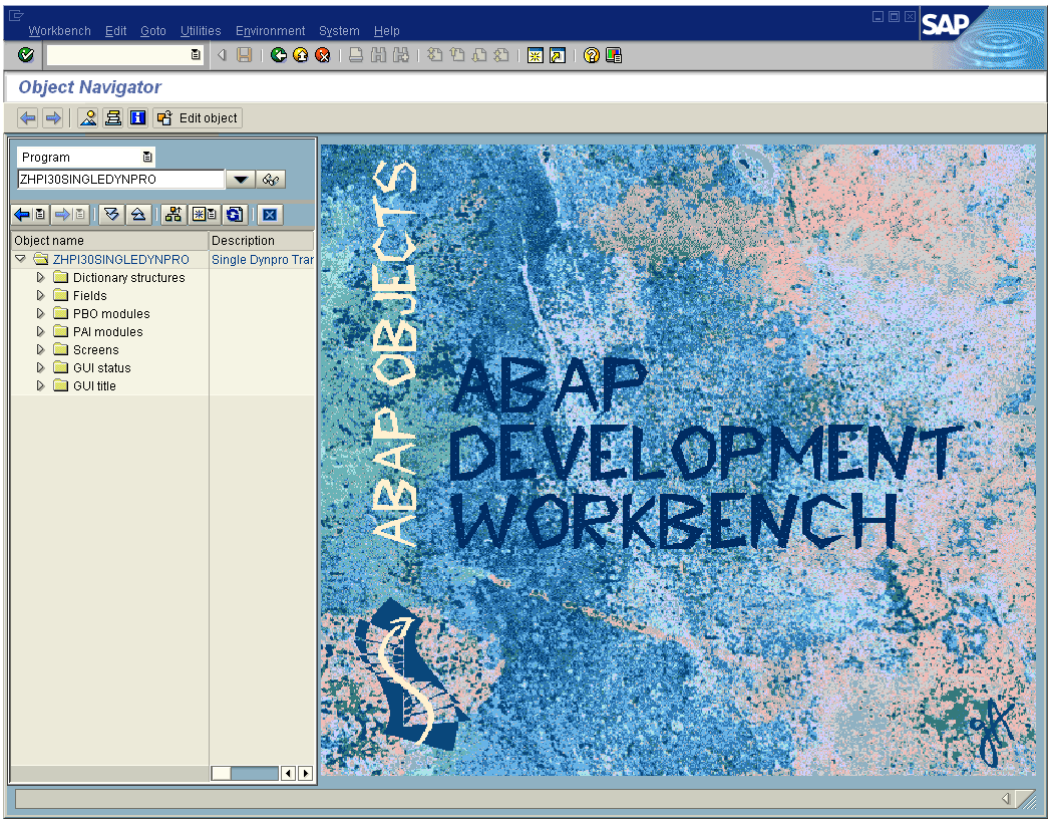
Copying the program to my personal development class does not require the use of any intellectual masterpieces. The description given above is very detailled – writing these lines takes more time than clicking around in R/3.



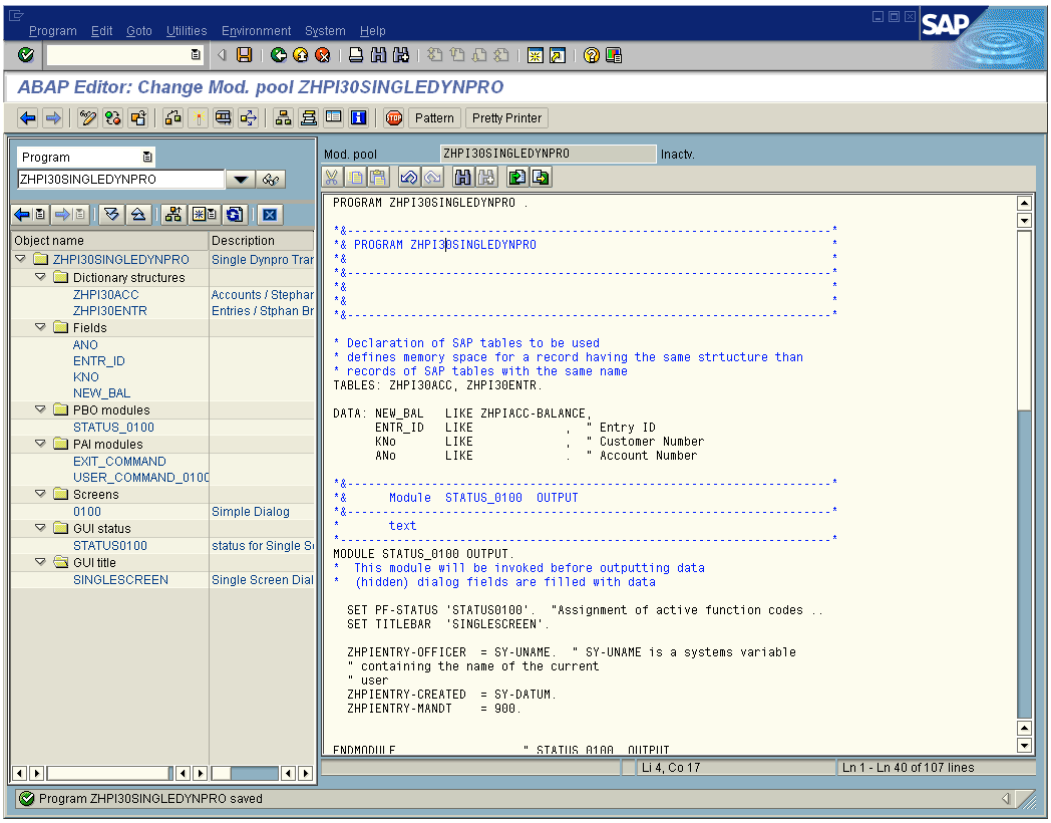




The ABAP workbench welcomes me with a warm and hearty screen. Well done, Walldorf.



Fairly incomplete, the source code was not easy to correct. Unfortunately, the comments did not give enough hints. On the left side of the screenshot one may notice all the modules, screens, etc. attached to the program.




I present the original – still incomplete – source code first, later I give my corrected version.

```
PROGRAM ZHPI30SINGLEDYNPRO .

*&---------------------------------------------------------------------*
*& PROGRAM ZHPI30SINGLEDYNPRO                                          *
*&                                                                     *
*&---------------------------------------------------------------------*
*&                                                                     *
*&                                                                     *
*&---------------------------------------------------------------------*

* Declaration of SAP tables to be used
* defines memory space for a record having the same structure than
* records of SAP tables with the same name
TABLES: ZHPI30ACC, ZHPI30ENTR.

DATA: NEW_BAL   LIKE ZHPI30ACC-BALANCE,
      ENTR_ID   LIKE,  " Entry ID
      KNo       LIKE,  " Customer Number
      ANo       LIKE.  " Account Number

*&---------------------------------------------------------------------*
*&      Module  STATUS_0100  OUTPUT
*&---------------------------------------------------------------------*
*       text
*---------------------------------------------------------------------*
MODULE STATUS_0100 OUTPUT.
*  This module will be invoked before outputting data
*  (hidden) dialog fields are filled with data

  SET PF-STATUS 'STATUS0100'.  "Assignment of active function codes ..
  SET TITLEBAR  'SINGLESCREEN'.
```

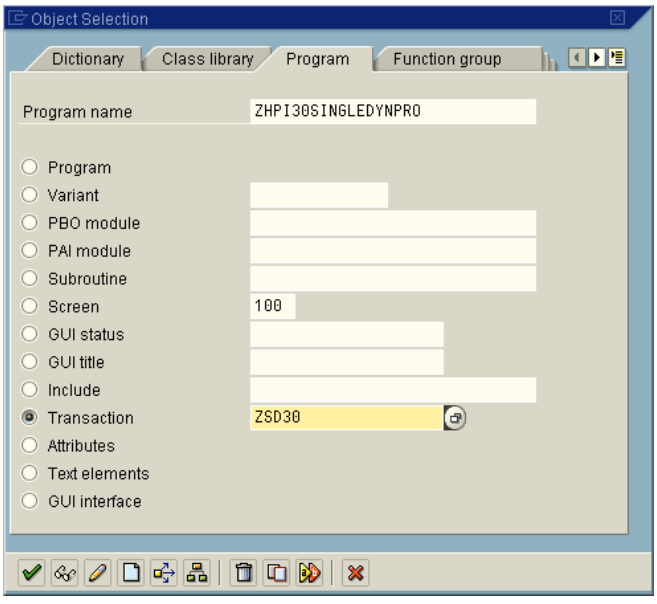
```
  ZHPI30ENTR-OFFICER  = SY-UNAME.  " SY-UNAME is a systems variable
  " containing the name of the current
  " user
  ZHPI30ENTR-CREATED  = SY-DATUM.
  ZHPI30ENTR-MANDT    = 900.


ENDMODULE.                " STATUS_0100  OUTPUT

*&---------------------------------------------------------------------*
*&      Module  USER_COMMAND_0100  INPUT
*&---------------------------------------------------------------------*
MODULE USER_COMMAND_0100 INPUT.
* This module will be executed when a function code is used or
* when a button is pressed

* Perform changes to the database
* fill records zhpi30entr and zhpi30acc with data
* and INSERT data to the database table having the same name

  " Determine Account Data

  SELECT * FROM
    WHERE ACCTNO  =
    AND KUNNR =.
  ENDSELECT.

  NEW_BAL =    .

  IF NEW_BAL >= 0.
    " Update account balance and modify account record

             .
    MODIFY   .

    KNO =  -KUNNR.
    ANO =  .

    " Determine Entry ID
    " first select the maximum entry id for this account/customer
    " then create a new ID by incrementing the maximum

    SELECT          INTO
    FROM
    WHERE  KUNNR = KNO AND ACCTNO = ANO.

    ZHPIENTRY-ENTRYID =  .

    " insert new entry record
              .

    COMMIT WORK.

  ELSE.

    " Failure handling (balance negative)
    ROLLBACK WORK.

  ENDIF.


  SET SCREEN 0.   "Terminate Dialog
  LEAVE SCREEN.

ENDMODULE.                " USER_COMMAND_0100  INPUT
*&---------------------------------------------------------------------*
*&      Module  EXIT_COMMAND  INPUT
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
MODULE EXIT_COMMAND INPUT.
  ROLLBACK WORK.
  SET SCREEN 0.
  LEAVE SCREEN.
ENDMODULE.                " EXIT_COMMAND  INPUT
```

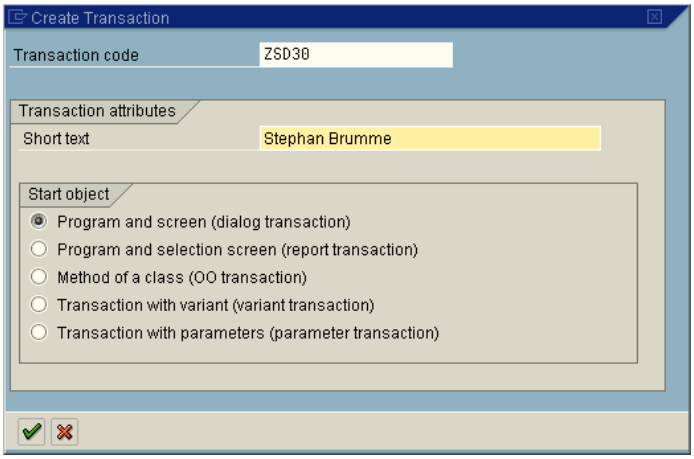
All modifications, additions and added comments are highlighted.

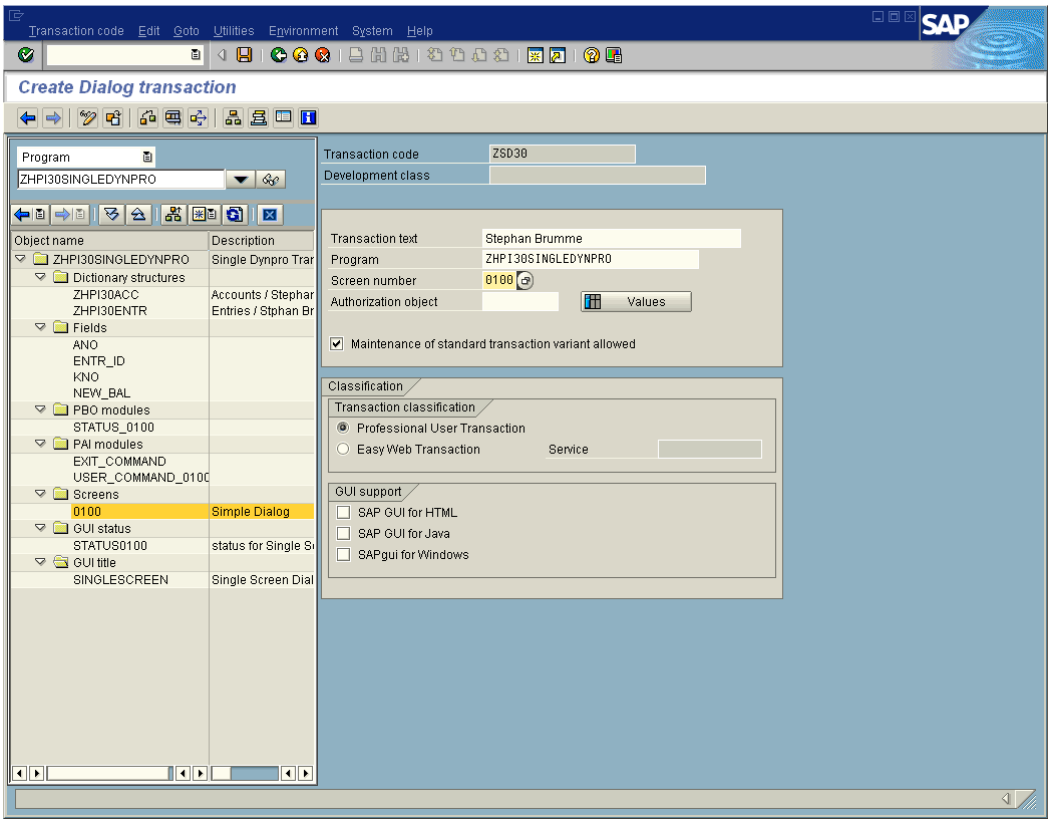
```
PROGRAM ZHPI30SINGLEDYNPRO .

*&---------------------------------------------------------------------*
*& PROGRAM ZHPI30SINGLEDYNPRO                                          *
*&                                                                     *
*&---------------------------------------------------------------------*
*&                                                                     *
*&                                                                     *
*&---------------------------------------------------------------------*

* Declaration of SAP tables to be used
* defines memory space for a record having the same strtucture than
* records of SAP tables with the same name
TABLES: ZHPI30ACC, ZHPI30ENTR.

DATA: NEW_BAL   LIKE ZHPI30ACC-BALANCE,
      ENTR_ID   LIKE ZHPI30ENTR-ENTRYID           ,  " Entry ID
      KNo       LIKE ZHPI30ACC-KUNNR            ,  " Customer Number
      ANo       LIKE ZHPI30ACC-ACCTNO            .  " Account Number

*&---------------------------------------------------------------------*
*&      Module  STATUS_0100  OUTPUT
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
MODULE STATUS_0100 OUTPUT.
*  This module will be invoked before outputting data
*  (hidden) dialog fields are filled with data

  SET PF-STATUS 'STATUS0100'.  "Assignment of active function codes ..
  SET TITLEBAR  'SINGLESCREEN'.

  ZHPI30ENTR-OFFICER  = SY-UNAME.  " SY-UNAME is a systems variable
  " containing the name of the current
  " user
  ZHPI30ENTR-CREATED  = SY-DATUM.
  ZHPI30ENTR-MANDT    = 900.


ENDMODULE.                 " STATUS_0100  OUTPUT

*&---------------------------------------------------------------------*
*&      Module  USER_COMMAND_0100  INPUT
*&---------------------------------------------------------------------*
MODULE USER_COMMAND_0100 INPUT.
* This module will be executed when a function code is used or
* when a button is pressed

* Perform changes to the database
* fill records zhpi30entr and zhpi30acc with data
* and INSERT data to the database table having the same name

  " Determine Account Data

  SELECT * FROM ZHPI30ACC
    WHERE ACCTNO = ZHPI30ENTR-ACCTNO
    AND KUNNR = ZHPI30ENTR-KUNNR.
  ENDSELECT.

  NEW_BAL = ZHPI30ENTR-AMOUNT + ZHPI30ACC-BALANCE  .

  IF NEW_BAL >= 0.
    " Update account balance and modify account record
    ZHPI30ACC-BALANCE = NEW_BAL .

    " Modify balance
    MODIFY ZHPI30ACC  .

    KNO =  ZHPI30ENTR-KUNNR.
    ANO =  ZHPI30ENTR-ACCTNO.

    " Determine Entry ID
    " first select the maximum entry id for this account/customer
```

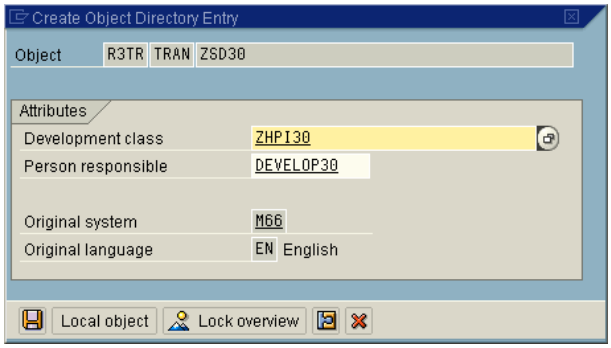
```
    " then create a new ID by incrementing the maximum

    SELECT MAX( ENTRYID ) INTO ENTR_ID
    FROM ZHPI30ENTR
    WHERE  KUNNR = KNO AND ACCTNO = ANO .

    ZHPI30ENTR-ENTRYID = ENTR_ID + 1 .
    ZHPI30ENTR-KUNNR   = KNO .
    ZHPI30ENTR-MANDT   = 900 .
    ZHPI30ENTR-ACCTNO  = ANO .
    ZHPI30ENTR-AMOUNT  = ZHPI30ENTR-AMOUNT .
    ZHPI30ENTR-OFFICER = SY-UNAME .
    ZHPI30ENTR-CREATED = SY-DATUM .

    " insert new entry record
    INSERT ZHPI30ENTR .

    COMMIT WORK.

  ELSE.

    " Failure handling (balance negative)
    ROLLBACK WORK.

  ENDIF.


  SET SCREEN 0.   "Terminate Dialog
  LEAVE SCREEN.

ENDMODULE.                 " USER_COMMAND_0100  INPUT
*&---------------------------------------------------------------------*
*&      Module  EXIT_COMMAND  INPUT
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
MODULE EXIT_COMMAND INPUT.
  ROLLBACK WORK.
  SET SCREEN 0.
  LEAVE SCREEN.
ENDMODULE.                 " EXIT_COMMAND  INPUT
```

The new transaction ZSD30 does not cause any headache since the question describes all necessary steps in an understandable manner.



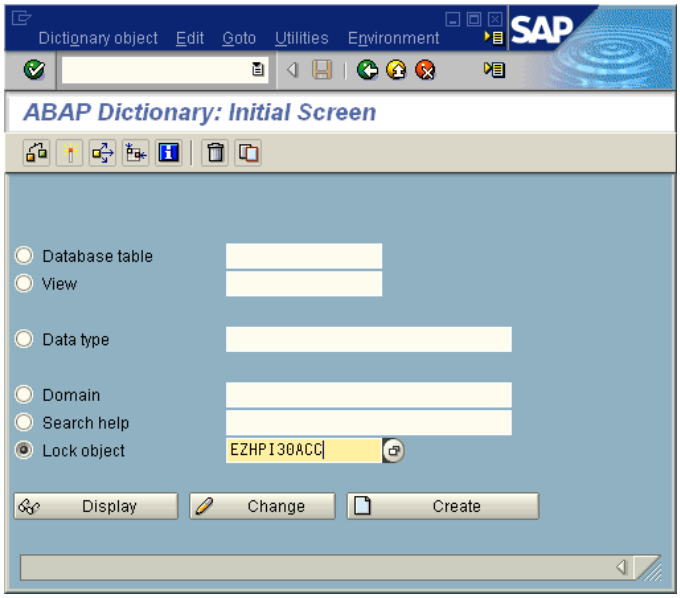

Create Transaction
Transaction code
ZSD30
Transaction attributes
Short text
Stephan Brumme
Start object
Program and screen (dialog transaction)
Program and selection screen (report transaction)
Method of a class (OO transaction)
Transaction with variant (variant transaction)
Transaction with parameters (parameter transaction)


Transaction code Edit Goto Utilities Environment System Help
SAP
Create Dialog transaction
Program
ZHPI30SINGLEDYNPRO
Object name
Description
ZHPI30SINGLEDYNPRO
Single Dynpro Trar
Dictionary structures
ZHPI30ACC
Accounts / Stephar
ZHPI30ENTR
Entries / Stphan Br
Fields
ANO
ENTR_ID
KNO
NEW_BAL
PBO modules
STATUS_0100
PAI modules
EXIT_COMMAND
USER_COMMAND_0100
Screens
0100
Simple Dialog
GUI status
STATUS0100
status for Single Sc
GUI title
SINGLESCREEN
Single Screen Dial
Transaction code
ZSD30
Development class
Transaction text
Stephan Brumme
Program
ZHPI30SINGLEDYNPRO
Screen number
0100
Authorization object
Values
Maintenance of standard transaction variant allowed
Classification
Transaction classification
Professional User Transaction
Easy Web Transaction
Service
GUI support
SAP GUI for HTML
SAP GUI for Java
SAPgui for Windows

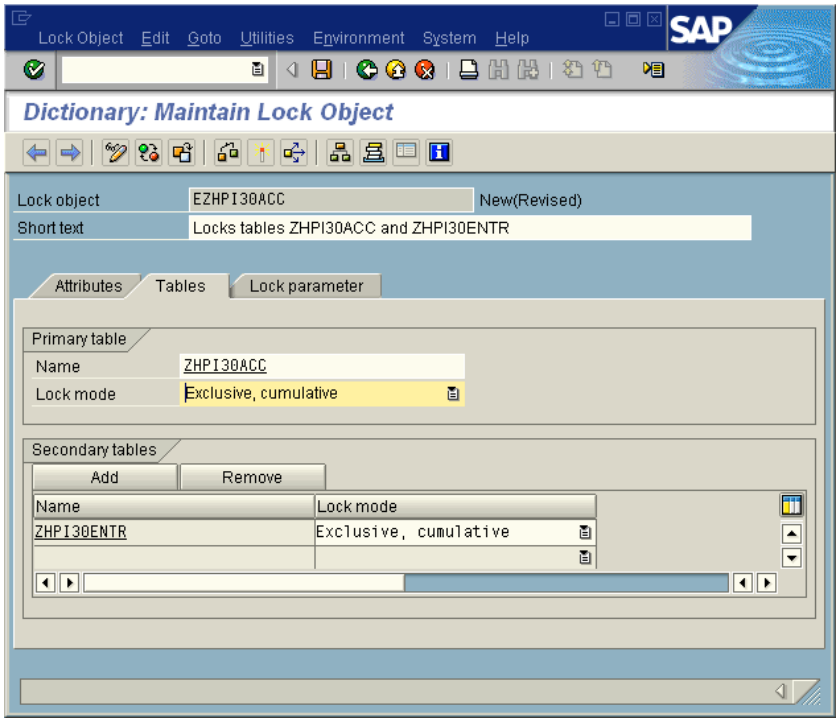

Create Object Directory Entry
Object
R3TR
TRAN
ZSD30
Attributes
Development class
ZHPI30
Person responsible
DEVELOP30
Original system
M66
Original language
EN
English
Local object
Lock overview

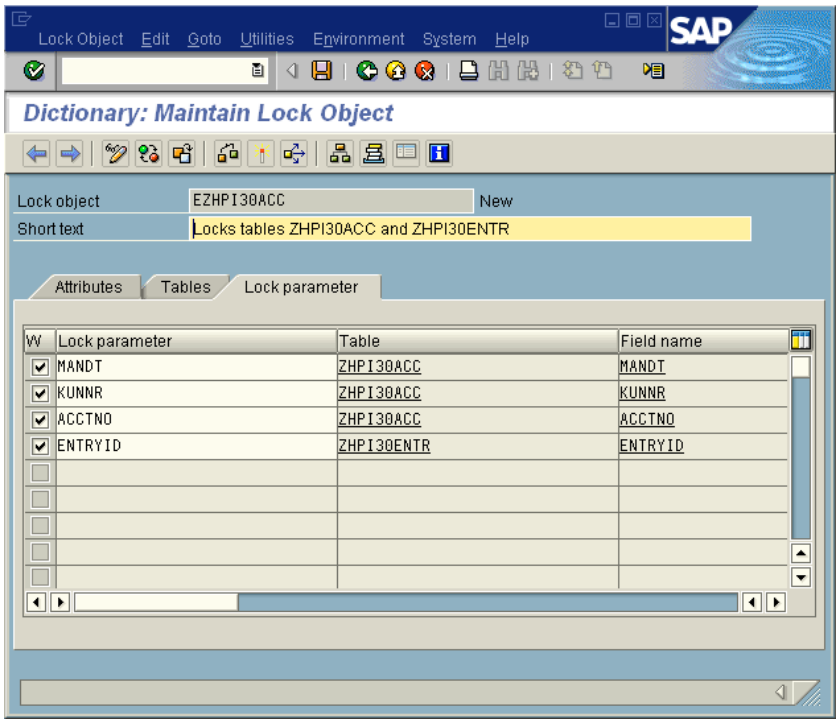
## Question 2

*The first Dynpro was implemented with only one Dynpro. When implementing more complex sequences of dialog masks with several Dynpros, we have to apply the SAP lock management to guarantee the consistency of data even across several Dynpros. To this end, you have to define a lock object first. The lock objects shall combine semantically related records of the account and of the entries table and shall restrict the access to these records. Lock objects are dictionary objects and are therefore managed by the ABAP dictionary. Create a lock object and name it EZHPIxxACC, where again xx has to be replaced by your login number (Remark: xx=30). Note the exception from the naming scheme here: user-defined lock objects have to start with the prefix EZ. The lock object shall be defined over the primary table ZHPI30ACC and should have ZHPI30ENTR as secondary table. Choose a lock mode that allows to consistently modify records and/or to insert new entries. Which lock parameters are automatically generated for the lock object ?*



Cumulative locks are generated automatically as one can easily infer from the screenshot.



## Question 3 & 4

Skipped as explained in the preface. Sorry for that.