Developing OM infotypes

by Michal Szymaczek

Although the need for creating Organizational Management (OM) infotypes is rare, who knows what challenges you may face in your own projects? For this reason, here are some details for creating OM infotypes. But first let's quickly review OM infotypes themselves.

OM infotypes — a short lesson

In some respects, OM infotypes are similar to Personnel Administration (PA) infotypes:

- Each infotype has a validity period defined by a beginning date and an ending date.
- OM infotypes are also described by time constraints — the difference between PA and OM constraints is that time constraints for OM infotypes vary for different objects.
- Infotype 1000 (Object Infotype) is a basis for all OM information. Each organizational object (e.g., organizational unit, job, position, work center) has a record of infotype 1000.

OM objects and infotypes are used when defining the organizational plan. The user can create many organizational plan versions, but only one version is "active" (i.e., visible in PA while hiring employees and reporting, for example). From the user's point of view, the organizational plan is a display of the organizational structure (see **Figure 1** on the next page) that shows the relationships between objects such as organizational units, positions, and persons.

The most frequently used organizational objects are organizational unit, position, job (general description of the position), and person (representing the employee in PA). The organizational hierarchy is created by the relationships between objects (Relationship infotype 1001).

A relationship always has two sides: passive (A) and active (B). The type of relationship is identified by a three-digit numerical code. For example:

A) The position "belongs to" (relationship A003) the organizational unit, which from the second side means that the organizational unit "incorporates" (relationship B003) the position.

B) The position X is "line supervisor" (relationship B002) to a position Y, which means that the position Y "reports to" (relationship A002) the position X.

Developing OM infotypes

There are many similarities in developing OM and PA infotypes. However, in the case of OM, you use a different transaction to support it — PPCI (Personnel Planning Infotype Copier). Let's see how it works.

Suppose for each position in the company (PD object type P), you need information about the loan limit (possible maximum loan amount the employee in this position can take). You need a new PD infotype that describes the position:

1. Use transaction SE11 (ABAP Dictionary) to create a table with the fields shown in **Figure 2** on the next page. Create a few sample records in the table (**Figure 3** on page 3).

   It is a good idea to create the elementary search help ZLOAN in transaction SE11 with the selection method ZLOAN and the export parameter ZTTYP.

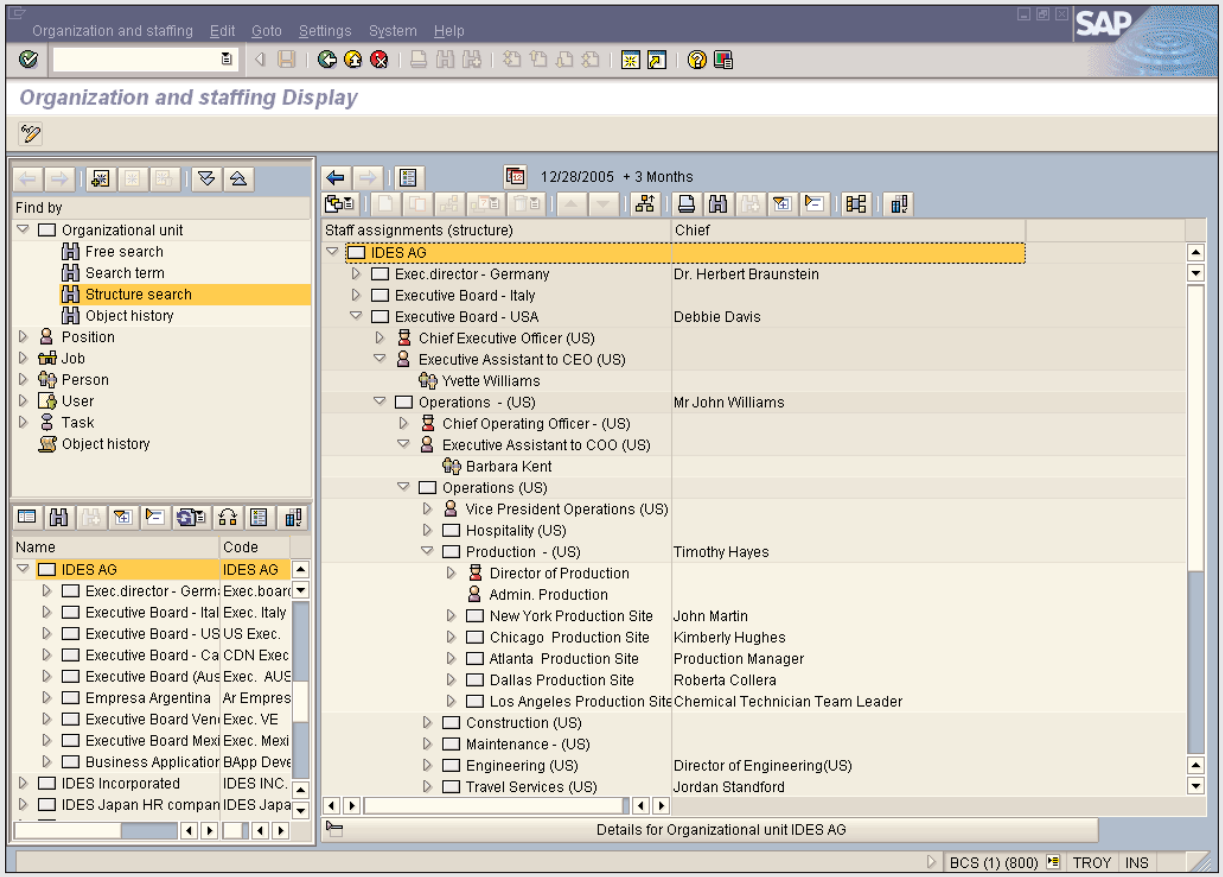

Figure 1 Sample organizational plan (accessed using transaction PPOSE)

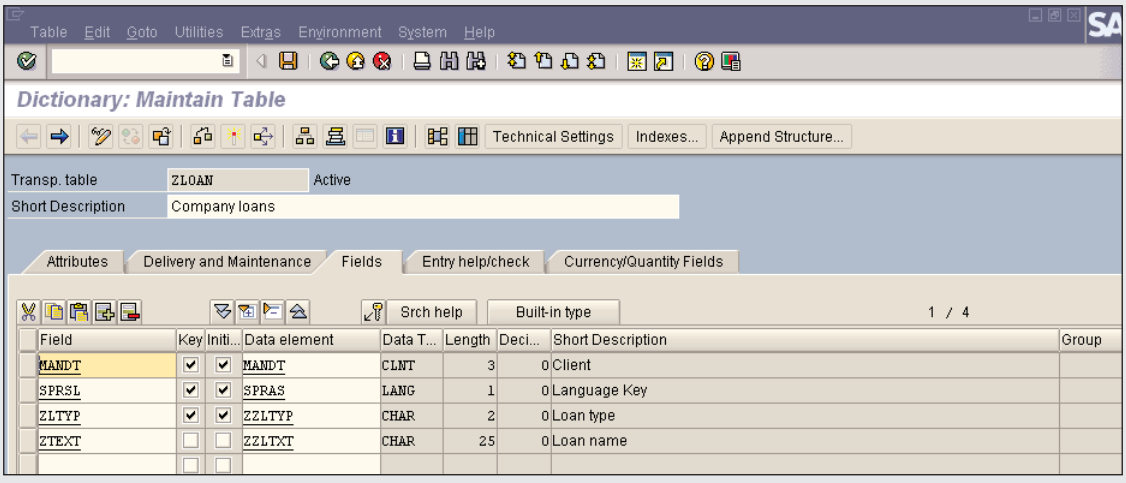

Figure 2 Loan dictionary table

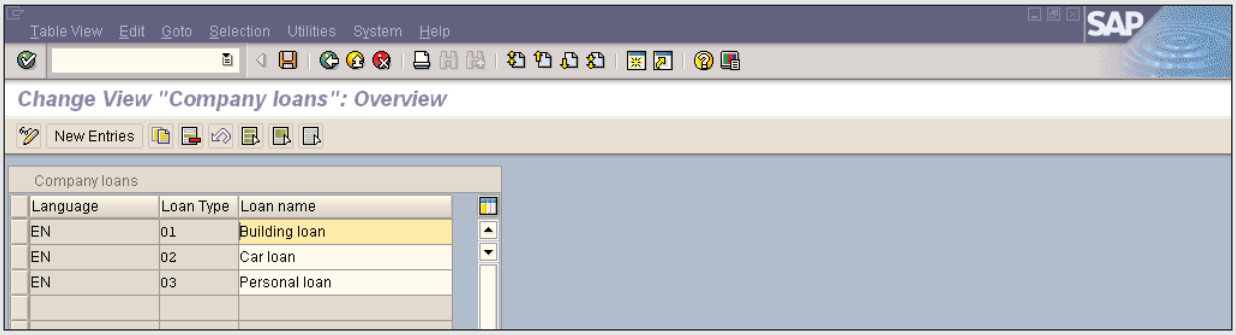

Figure 3 Sample records in ZLOAN table

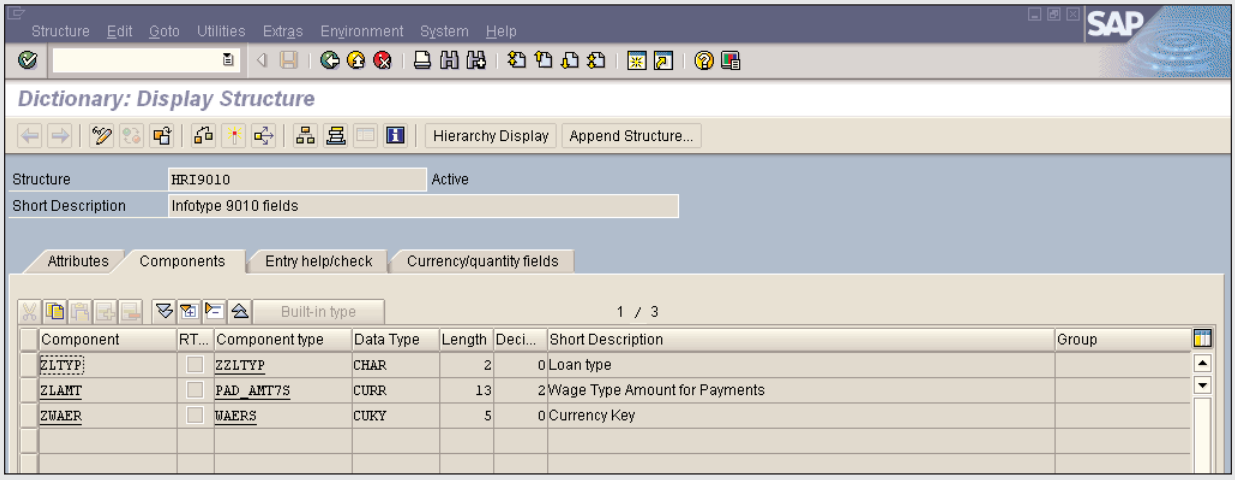

Figure 4 HRI9010 structure

Note!

- Double-clicking on component ZLTYP shows that it has been assigned search help ZLOAN.
- Component type ZLAMT is assigned reference field ZWAER on the Currency/quantity fields tab.
- Check table TCURC is assigned to component ZWAER on the Entry help/check tab.

2. Run transaction SE11 and define the HRI9010 structure with the fields shown in **Figure 4**.
3. Run transaction PPCI, enter 9010 for the new infotype (you can use any value between 9000–9999), define the infotype name (Loan limit), and click on Create. You should get the screen shown in **Figure 5** on the next page.
4. Select Field infotype and click on Create (in the Infotype block). The system generates (among others) the transparent table HRP9010, index HRP9010~1, structure P9010, module pool MP901000, screens 1000, 2000, and 3000, and some table entries.

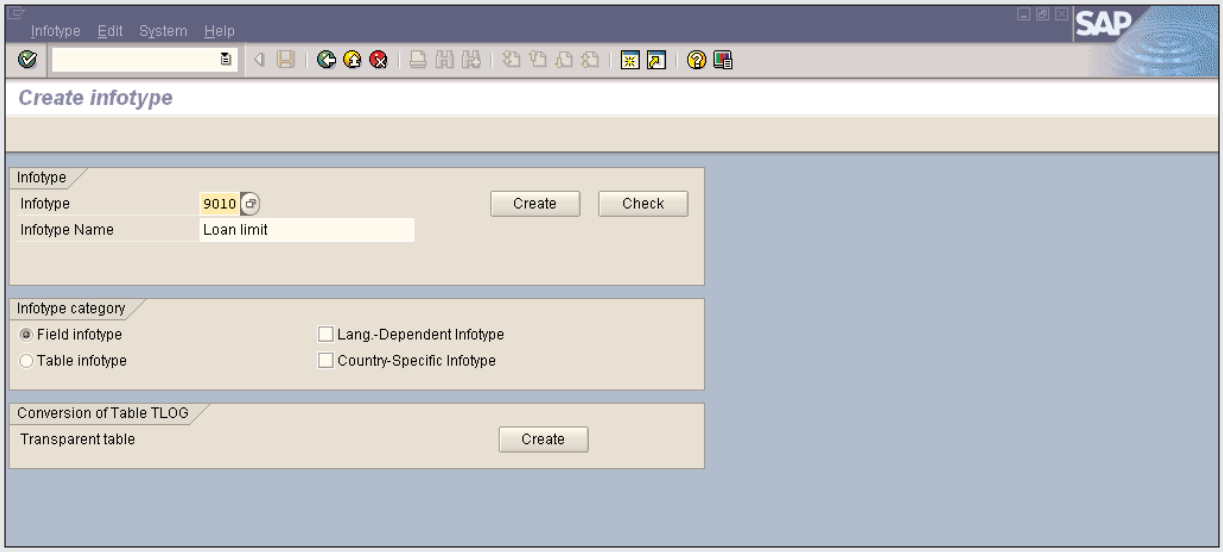

Figure 5 PPCI transaction

> ***Note!***
>
> You can choose between two infotype categories: Field infotype and Table infotype. Field infotype should be used when its fields are filled only once within the infotype itself. Table infotype lets you store data in a tabular structure in one infotype record.

5. Table T777I describes OM infotype attributes. The most important thing to define is the assignment of our new infotype to the standard object type Position. Run transaction SM30, input the table name T777I, and click on the Maintain button, which takes you to the screen shown in **Figure 6**.

6. Choose infotype 9010 in the list and double-click on Infotypes per object type in the left-hand navigation pane. Click on New Entries and define the record as shown in **Figure 7**.

7. Now, you can return to the maintenance view to define a time constraint for the new infotype. Choose infotype 9010 and double-click on Time constraint in the left-hand navigation pane. Click on New Entries and define the record as shown in **Figure 8** on page 6.

> ***Note!***
>
> The No maintenance checkbox in the Infotypes per object type view does not affect the customizing settings. This field (if it is set) merely specifies that you cannot maintain the infotype in question using the standard transactions.

> ***Note!***
>
> You can also maintain T777I entries via the Implementation Guide (IMG) by navigating to Personnel Management → Organizational Management → Basic Settings → Data Model Enhancement → Infotype Maintenance → Maintain infotypes.

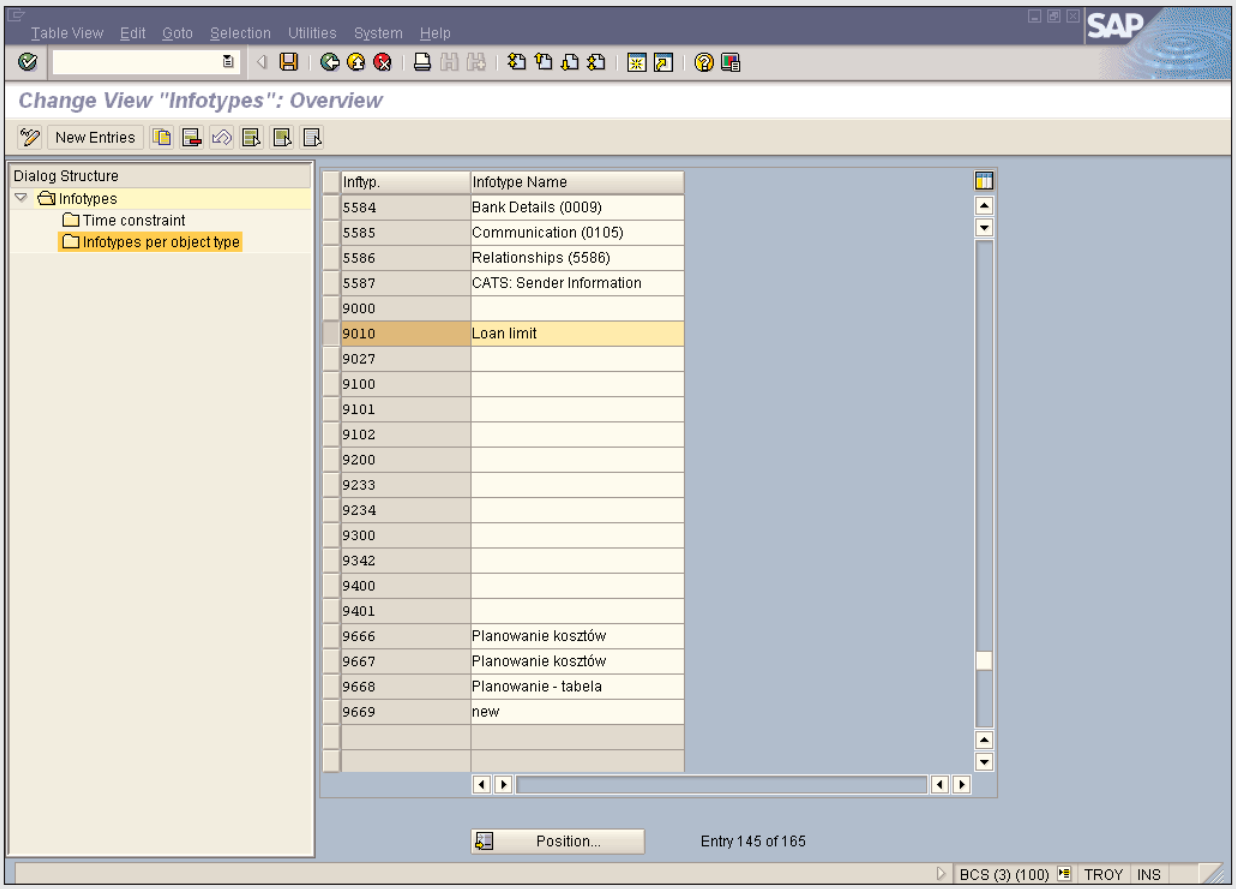

Figure 6 T777I maintenance view (transaction OOIT)

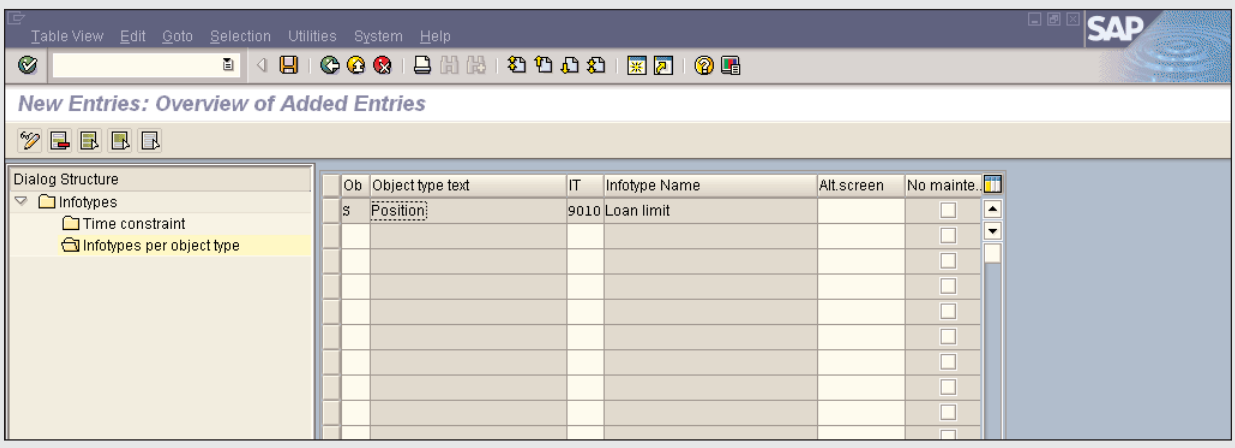

Figure 7 Infotypes per object type view

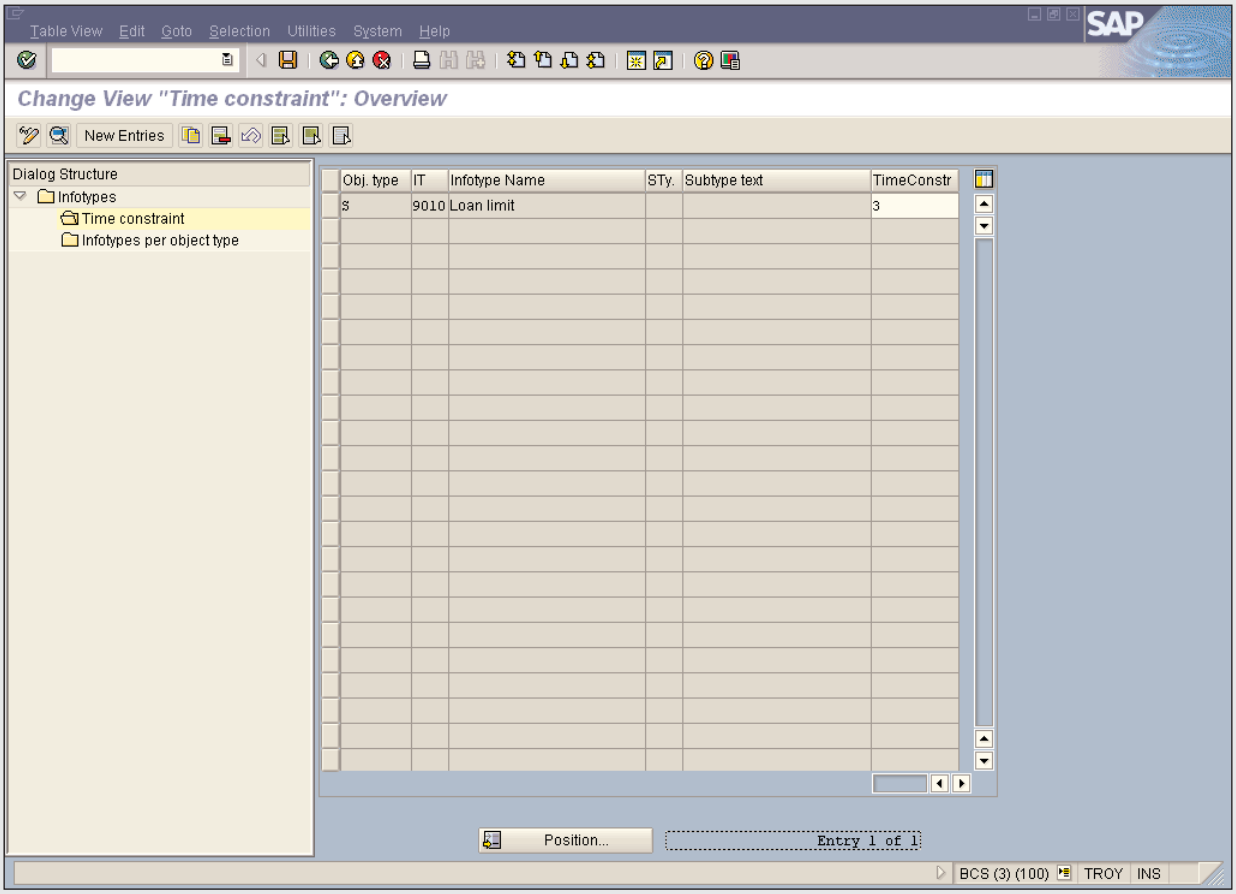

Figure 8 Time constraint maintenance for infotype 9010

8. Infotype 9010 records (entries) may now be created for positions, but it would be nice to make it more user-friendly by adding the loan type text (ZLOAN-ZTEXT) to the screen. To do this, run transaction SE80, and choose object Program and object name MP901000.
   - The table work area for ZLOAN should be defined in module pool MP901000 (**Figure 9**).
   - Now, we should change the generated PBO module INIT_9010 to ensure that the text for the loan type will be read whenever the single screen is shown (**Figure 10**).
9. We next need to place the ZLOAN-ZTEXT field in screen 2000. Go to the ABAP Workbench Screen Painter, choose screen 2000, and insert a new Input/Output field with the name ZLOAN-ZTEXT. Confirm that the field is taken from the ABAP Dictionary.
10. To check the results, run transaction PO13 (Maintain Position), enter the position you want to edit, and choose the infotype Loan limit (see **Figure 11** on page 8). Click on Create infotype (or press F5), maintain the infotype record (see **Figure 12** on page 8), and save the data.


```
* module pool infotype 9010
PROGRAM MP901000 MESSAGE-ID 5A.
INCLUDE MPH5ATOP.                       "header
TABLES: WPLOG,
        PPPAR, PPHDR, PPHDX, PPSEL, PPENQ,
        T777O, T777P, T777S, T777T,
        P1000, P1001, P9010.

TABLES: ZLOAN .  "TABLE WORK AREA FOR LOAN TYPE

INCLUDE MPHCOM00.                       "common areas
INCLUDE FHVTAB00.                       "update tables
INCLUDE FHVIEW00.                       "USER-VIEW
INCLUDE MPHFCOD0.                       "function codes
INCLUDE MPHDAT00.                       "general data
INCLUDE MPHPBO00.                       "PBO modules
INCLUDE MPHPAI00.                       "PAI modules
INCLUDE MP901020.                       "specific PAI/PBO modules
*include mpxxxxbi.                      "Batch-Input von der WPLOG
```

Figure 9 Defining a table work area for ZLOAN in module pool MP901000

```
* OUTPUT modules

*---------------------------------------------------------------------*
*       MODULE INIT_9010                                              *
*---------------------------------------------------------------------*
*       infotype specific initializations                             *
*---------------------------------------------------------------------*
MODULE INIT_9010 OUTPUT.

clear ZLOAN.

select single * from  ZLOAN
         where  sprsl       = sy-langu
         and    zltyp       = p9010-zltyp.


ENDMODULE.

* INPUT modules
```

Figure 10 Modifying the generated PBO module INIT_9010

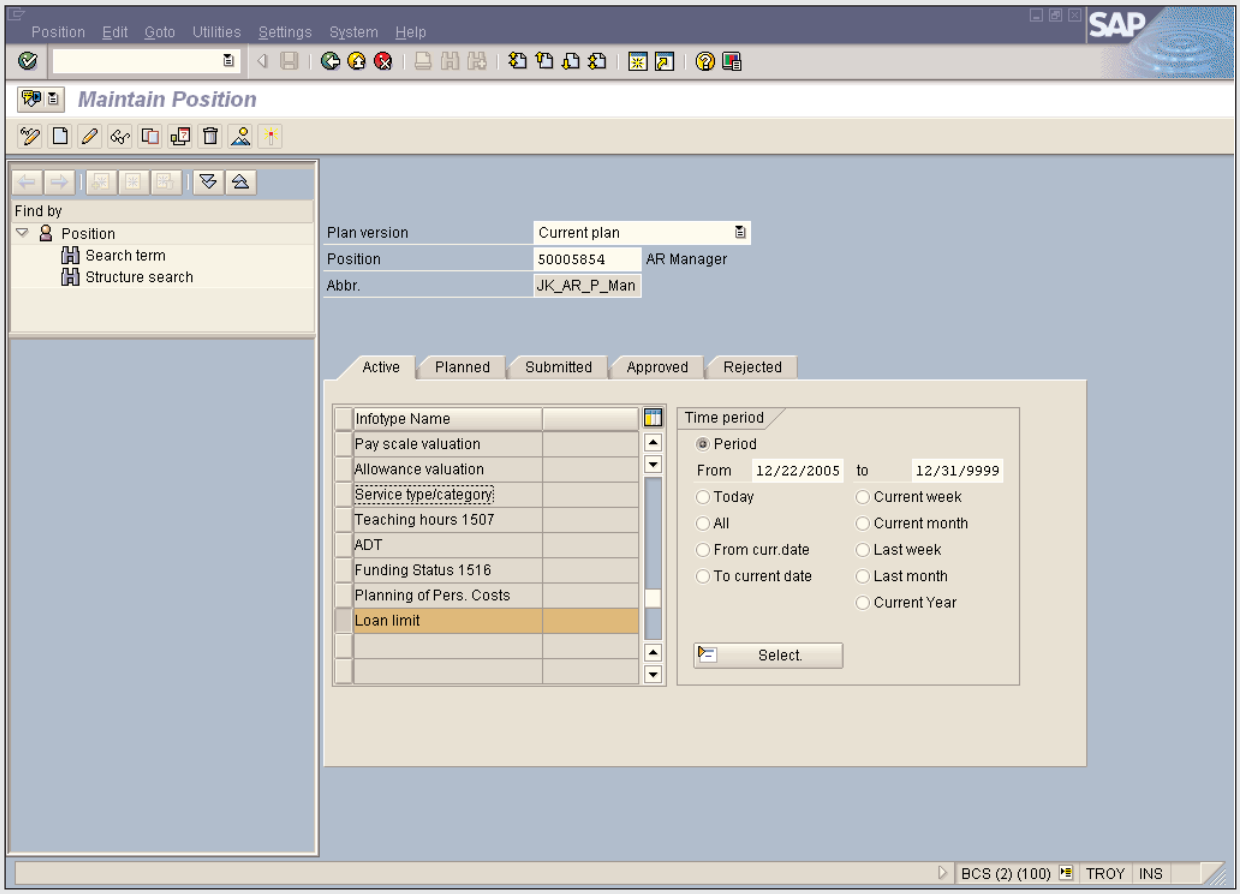

Figure 11 Transaction PO13 for position maintenance

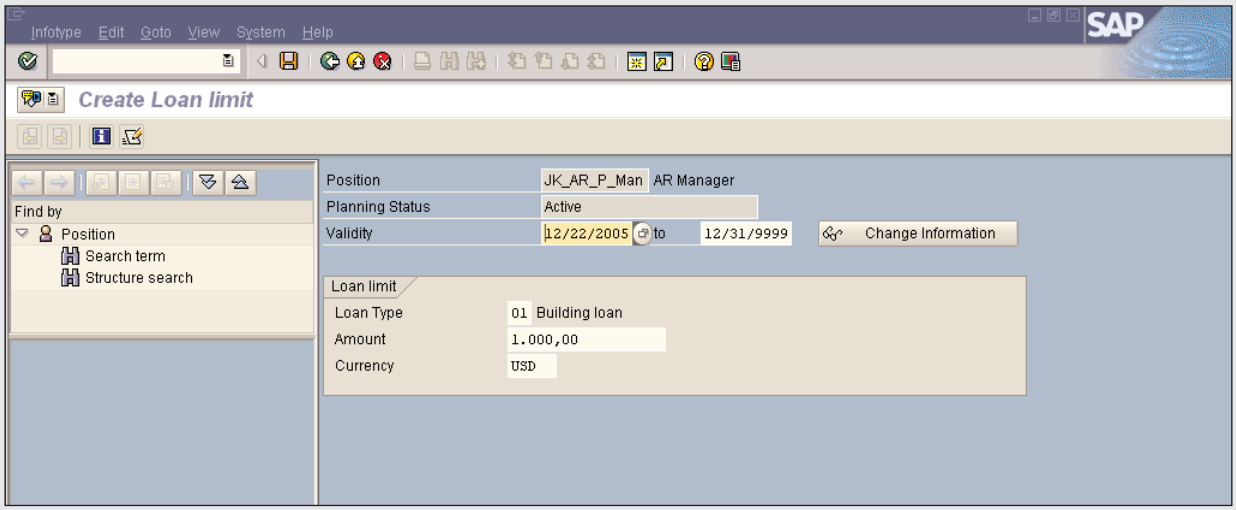

Figure 12 Loan limit maintenance screen

