

# BAPI Step-by-Step

**Applies to:** SAP ABAP

## Summary:

This article demonstrates in a step-by-step process to write ABAP Reports that use BAPI function modules.

**Author(s):** Renjith Kumar. P

**Company:** Pricol Software Soutions.

**Created on:** 26 January 2007

## Author Bio

P. Renjith Kumar is SAP BI Consultant with Pricol Software solutions. He has extensive cross-functional experience and has been with end-to-end SAP implementations as ABAP Consultant.

# Table of Contents

Business Object Repository.....	3
BOR Definition.....	3
BAPI programming.....	3
Some Standard BAPI'S .....	4
GetList .....	4
GetDetail.....	4
CreateFromData.....	4
Selecting the BAPI.....	7
Frequently used BAPI: .....	9
Sales and Distribution.....	9
Material Management.....	9
Production and Planning .....	10
Finance .....	10
ABAP Report using BAPI .....	10
Testing the Report: .....	13
Disclaimer and Liability Notice.....	15

## Business Object Repository

### BOR Definition

All SAP Business Object types and SAP Interface Types and their methods are defined and described in the R/3 Business Object Repository (BOR). The Business Object Repository was introduced in R/3 Release 3.0, at the same as time as SAP Business Objects and SAP Business Workflow. Presently, the BOR is mainly used by SAP Business Workflow.

The BOR has two essential functions:

- It defines and describes SAP Business Objects and SAP Interface Types and their BAPIs.
- It creates instances of SAP Business Objects

### BAPI programming

BAPIs are defined in the Business Object Repository (BOR) as methods of SAP Business Objects or SAP Interface Types and are implemented as function modules. The separation of a BAPI definition from its actual implementation enables you to access a BAPI in two ways:

- You can call the BAPI in the BOR through object-oriented method calls
- You can make RFC calls to the function module on which the BAPI is based

## Some Standard BAPI'S

### GetList

Delivers a list of key fields objects that satisfies certain selection Criteria

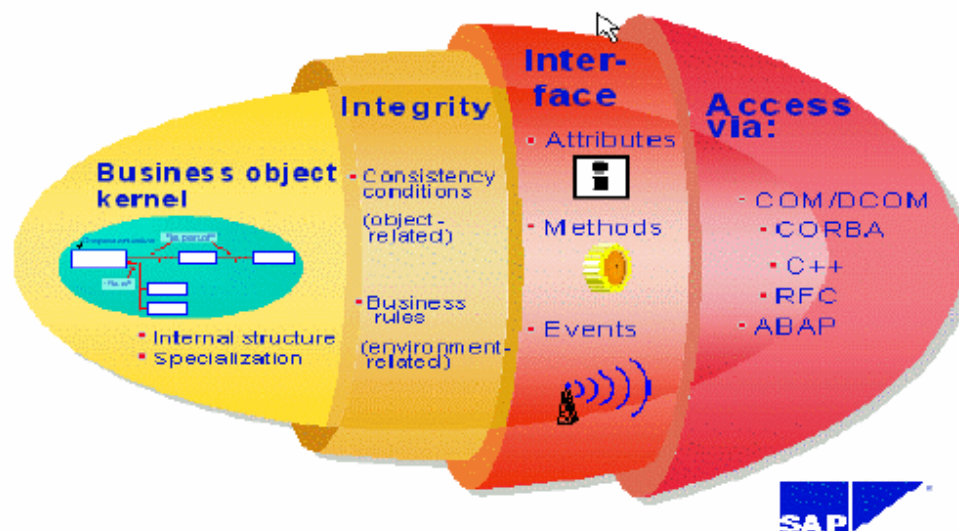
### GetDetail

Delivers detailed information of an object, whose complete key has been Specified

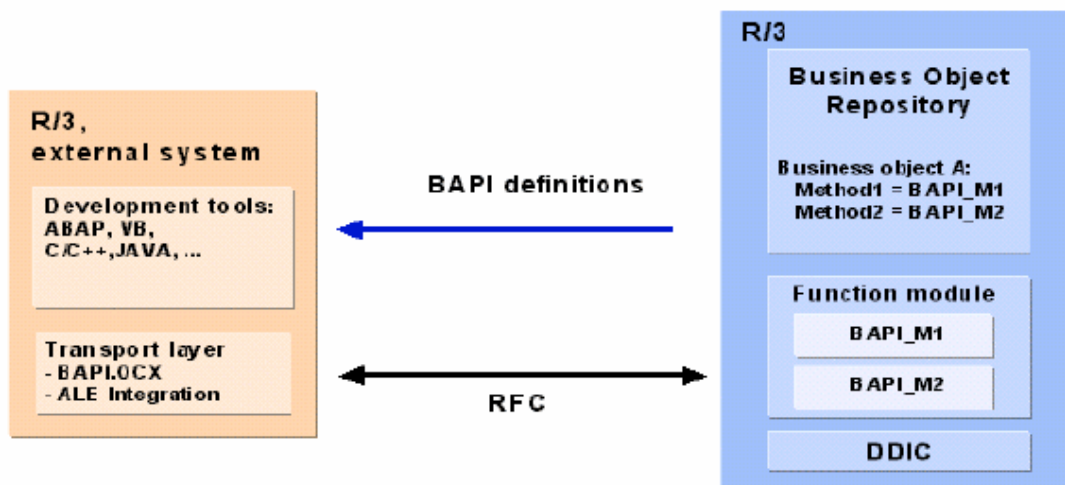
### CreateFromData

Generates new objects in R3 from key fields and returns information.

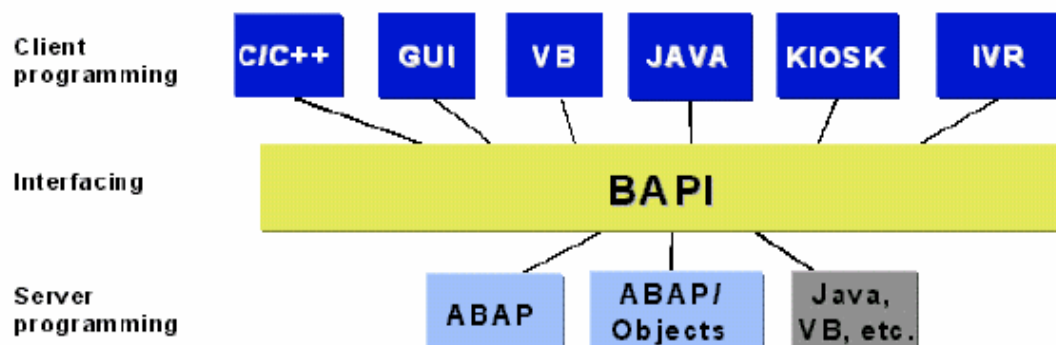
## SAP Business Object



## BAPI Communication

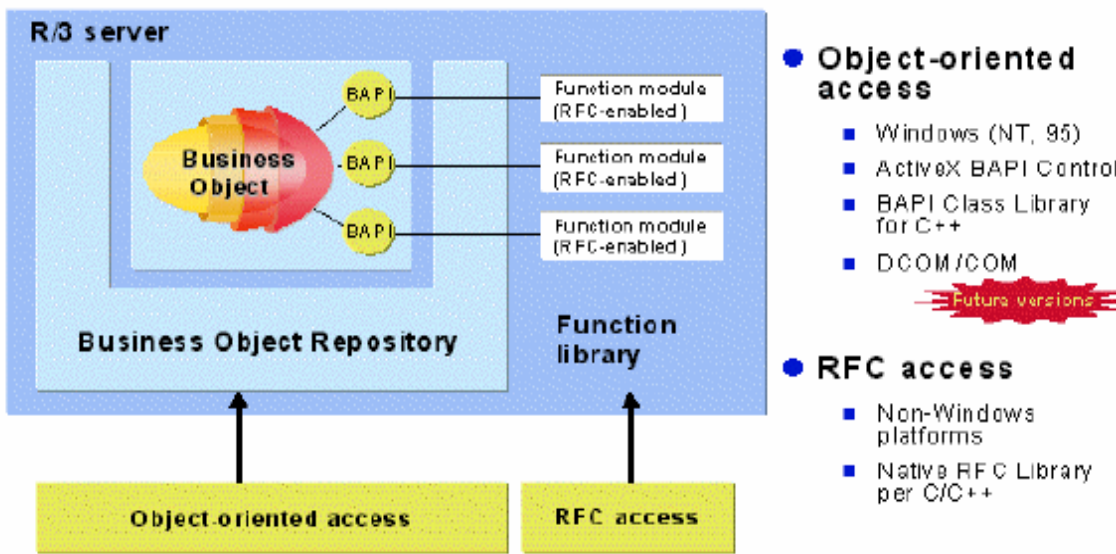


## BAPIs - separating “What” and “How”



- BAPIs are language-independent interfaces.
- Basic software technology can be changed without having to reprogram applications.

# Accessing BAPIs



The code to illustrate the steps involved when the BAPI ActiveX Control is used to access BAPIs .

- Creating a BAPI ActiveX Control object

**Set oBAPICtrl = CreateObject("SAP.BAPI.1")**

- Creating a logon control object:

**Set oLogonCtrl = CreateObject("SAP.Logoncontrol.1")**

- Creating a connection object to the R/3 System:

**Set oBAPICtrl.Connection = oLogonCtrl.NewConnection**

- Logging on to R/3 System by calling the logon method of the connection object:

**If oBAPICtrl.Connection.Logon(frmStart.hwnd,FALSE) = FALSE then**

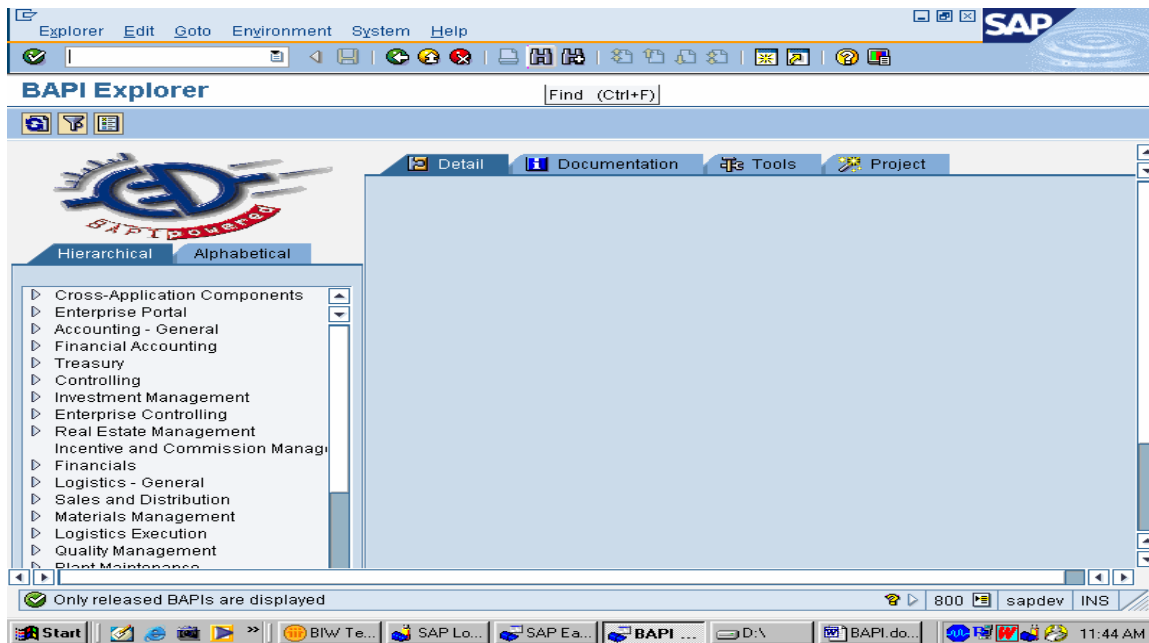
**MsgBox"R/3 Connection failed"**

**End**

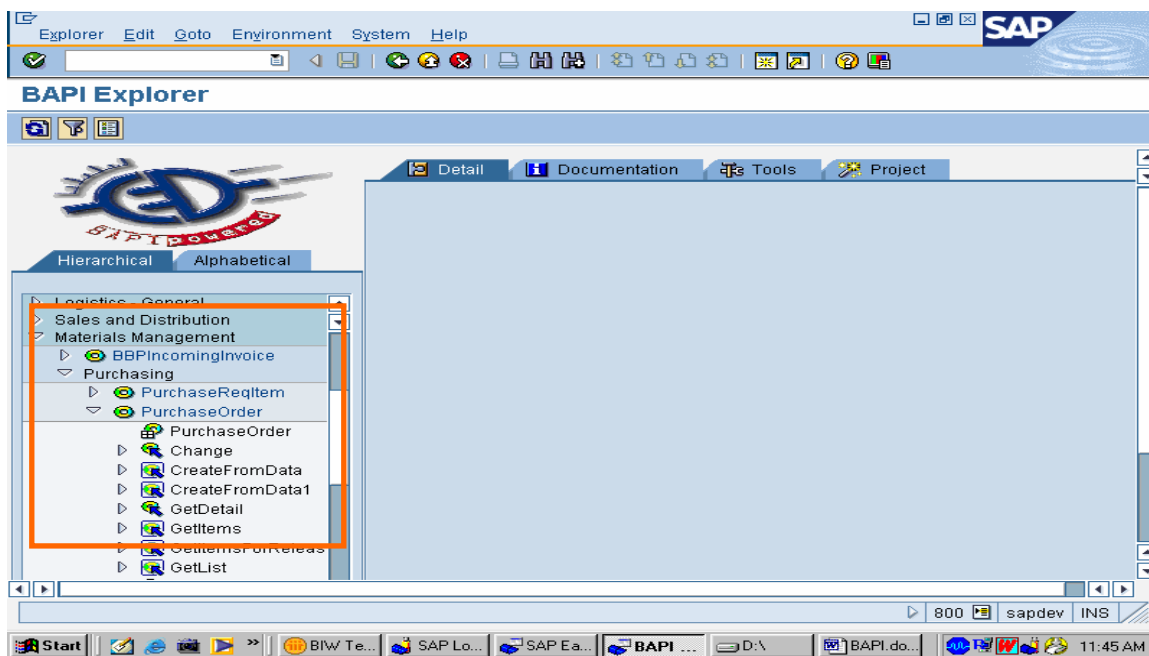
**Endif**

## Selecting the BAPI

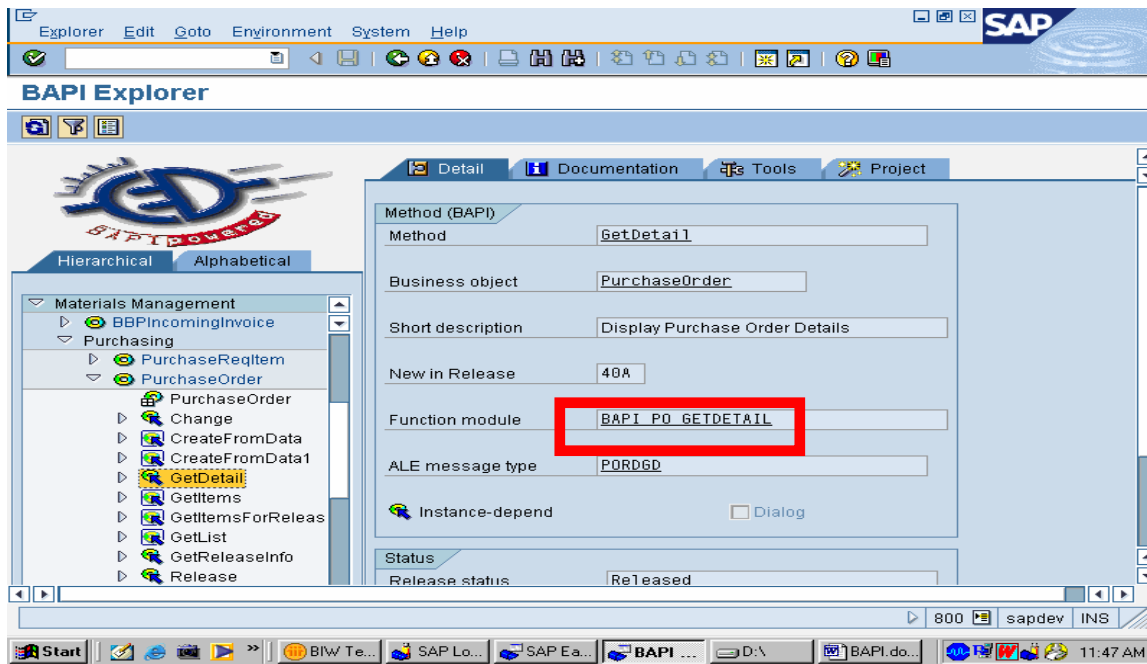
1. Give Transaction BAPI, You will get the Following Screen



Now based on your requirement you have to search for BAPI, Here we are going to search for BAPI related to Purchase Order. Purchase order is in MM so we select that from the list.

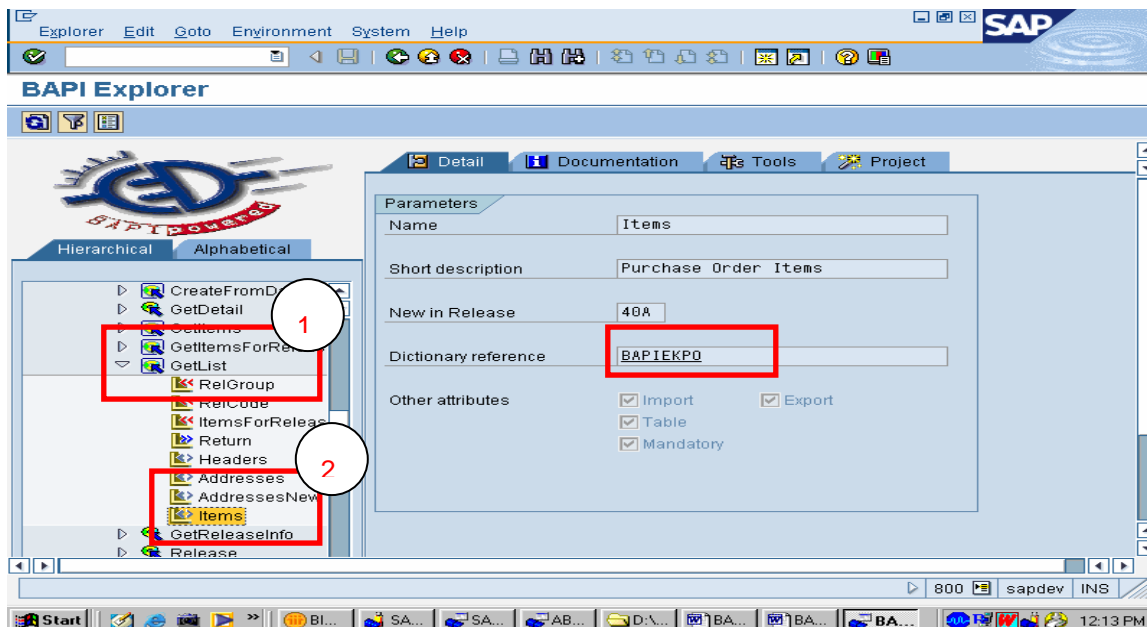


We have to get purchase order detail, So Double click on the '**GetDetail**'. The right frame will show the BAPI name.

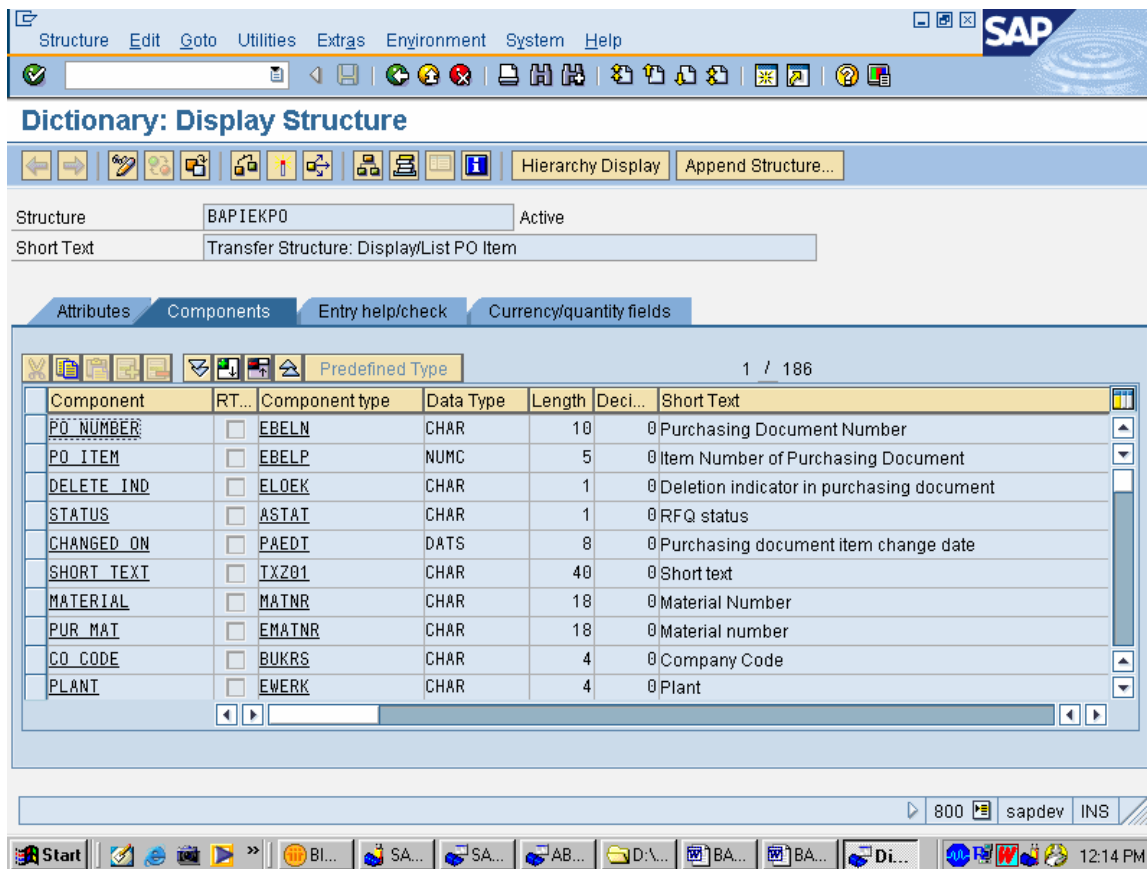


Now Expand The GetList by pressing > , you will get the screen as below

We need Purchase Order Header detail, so double Click Header; you will get Right Frame with the Dictionary Reference. The Dictionary Reference is BAPIEKKO; Note that you have to use that in Report. Double click that BAPIEKKO to see the fields that can be displayed in Report.







Note down the Fields name from the Components Column that you need in the Report output.

### Frequently used BAPI:

#### Sales and Distribution

Customer Material Info	BAPI_CUSTMATINFO_GETDETAILM
Sales order	BAPI_SALESORDER_GETLIST
Sales order	BAPI_SALESORDER_GETSTATUS

#### Material Management

Purchase Req Item	BAPI_REQUIREMENT_GET_LIST
Purchase order	BAPI_PO_GETDETAIL
Purchase order	BAPI_PO_GETITEMS

Purchase order	BAPI_PO_GETITEMSREL
Purchase order	BAPI_PO_GET_LIST
Purchasing info	BAPI_INFORECORD_GETLIST

### Production and Planning

Planned order	BAPI_PLANNEDORDER_GET_DETAIL
Planned order	BAPI_PLANNEDORDER_GET_DET_LIST
Planned Indep Reqmt	BAPI_REQUIREMENTS_GETDETAIL

### Finance

AP Account	BAPI_AP_ACC_GETOPENITEMS
AP Account	BAPI_AP_ACC_GETOPENITEMS
Debtor Credit Account	BAPI_CR_ACC_GETDETAIL
AR Account	BAPI_AR_ACC_GETOPENITEMS
AR Account	BAPI_AR_ACC_GETPERIODBALANCES
AR Account	BAPI_AR_ACC_GETSTATEMENT

### ABAP Report using BAPI

Now in The ABAP editor (SE38) create a new report and write the Code

```
*&-----*
*& Report  ZBAPI_1
*&
*&-----*
*&
*&      BAPI TO GET PO ITEM DETAILS
*&-----*
```

```
REPORT  ZBAPI_1.
```

```
DATA : BEGIN OF I_POITEM OCCURS 0.
```

```
    INCLUDE STRUCTURE BAPIEKPO.  "Include the Structure of dictionary Ref.
```

DATA : END OF I\_POITEM.

PARAMETERS P\_EBELN LIKE EKK0-EBELN default '4500012164'.. “ Input .

CALL FUNCTION 'BAPI\_PO\_GETDETAIL'

EXPORTING

PURCHASEORDER	=	P_EBELN
* ITEMS	=	'X'
* ACCOUNT_ASSIGNMENT	=	' '
* SCHEDULES	=	' '
* HISTORY	=	' '
* ITEM_TEXTS	=	' '
* HEADER_TEXTS	=	' '
* SERVICES	=	' '
* CONFIRMATIONS	=	' '
* SERVICE_TEXTS	=	' '
* EXTENSIONS	=	' '
* IMPORTING		
* PO_HEADER	=	
* PO_ADDRESS	=	

TABLES

* PO_HEADER_TEXTS	=	
PO_ITEMS	=	I_POITEM. “ Assign the Internal Table
* PO_ITEM_ACCOUNT_ASSIGNMENT	=	
* PO_ITEM_SCHEDULES	=	
* PO_ITEM_CONFIRMATIONS	=	
* PO_ITEM_TEXTS	=	.
* PO_ITEM_HISTORY	=	
* PO_ITEM_HISTORY_TOTALS	=	
* PO_ITEM_LIMITS	=	
* PO_ITEM_CONTRACT_LIMITS	=	
* PO_ITEM_SERVICES	=	
* PO_ITEM_SRV_ACCASS_VALUES	=	
* RETURN	=	
* PO_SERVICES_TEXTS	=	
* EXTENSIONOUT	=	

LOOP AT I\_POITEM.

WRITE :/ 'PO NUMBER        = ' , I\_POITEM-PO\_NUMBER COLOR COL\_HEADING,

```

/ 'ITEM          = ' , I_POITEM-PO_ITEM,
/ 'MATERIAL NAME = ' , I_POITEM-MATERIAL,
/ 'MATERIAL      = ' , I_POITEM-PUR_MAT,
/ 'CHANGED ON   = ' , I_POITEM-CHANGED_ON,
/ 'SHORT TEXT    = ' , I_POITEM-SHORT_TEXT,
/ 'COMPANY CODE  = ' , I_POITEM-CO_CODE,
/ 'PLANT         = ' , I_POITEM-PLANT,
/ 'MATERIAL GROUP = ' , I_POITEM-MAT_GRP,
/ 'QUANTITY      = ' , I_POITEM-QUANTITY LEFT-JUSTIFIED,
/ 'UNIT          = ' , I_POITEM-UNIT,
/ 'NET PRICE     = ' , I_POITEM-NET_PRICE LEFT-JUSTIFIED.

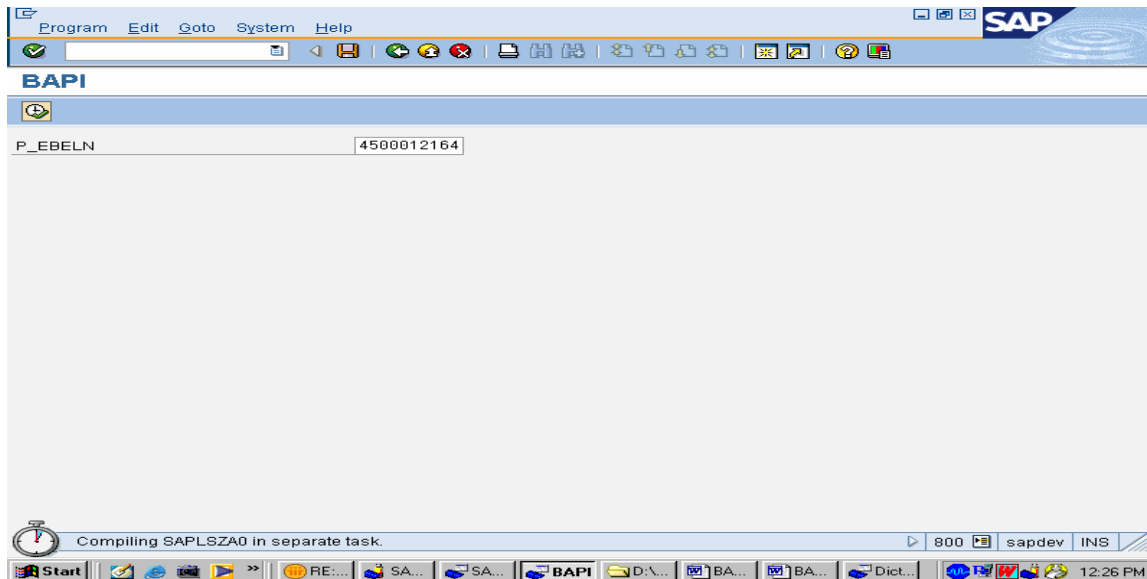
```

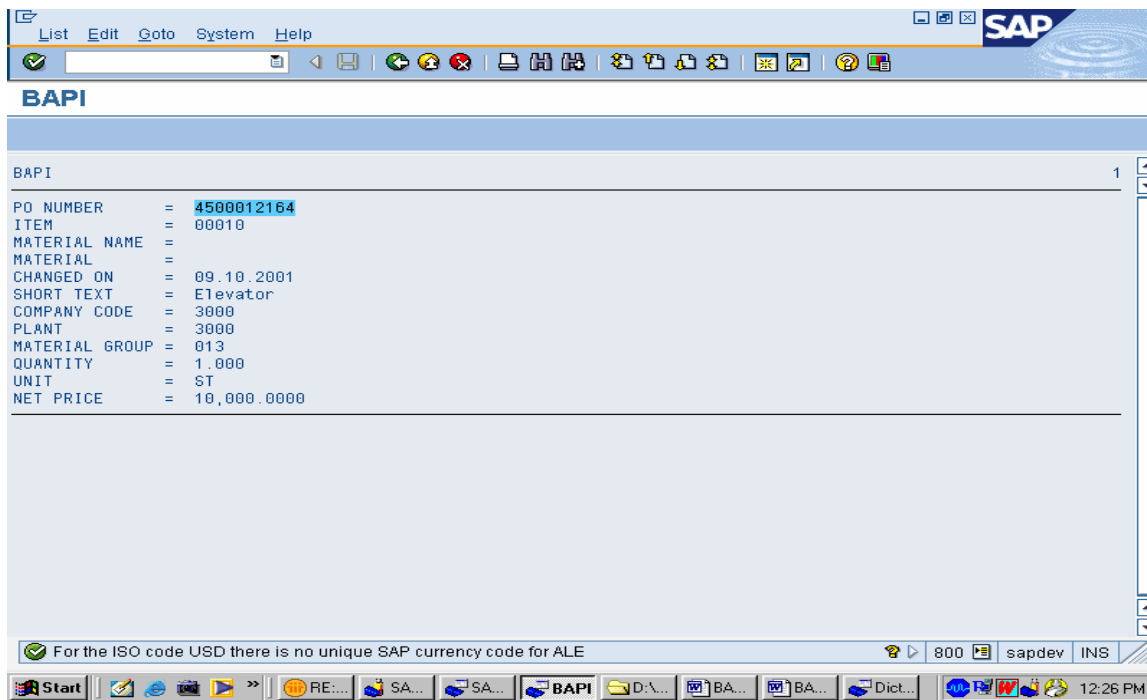
ULINE.

ENDLOOP.

Note: You can write any no fields from the Dictionary BAPIEKPO in the output, just  
 Note the field and give that in the Write statement within LOOP...ENDLOOP

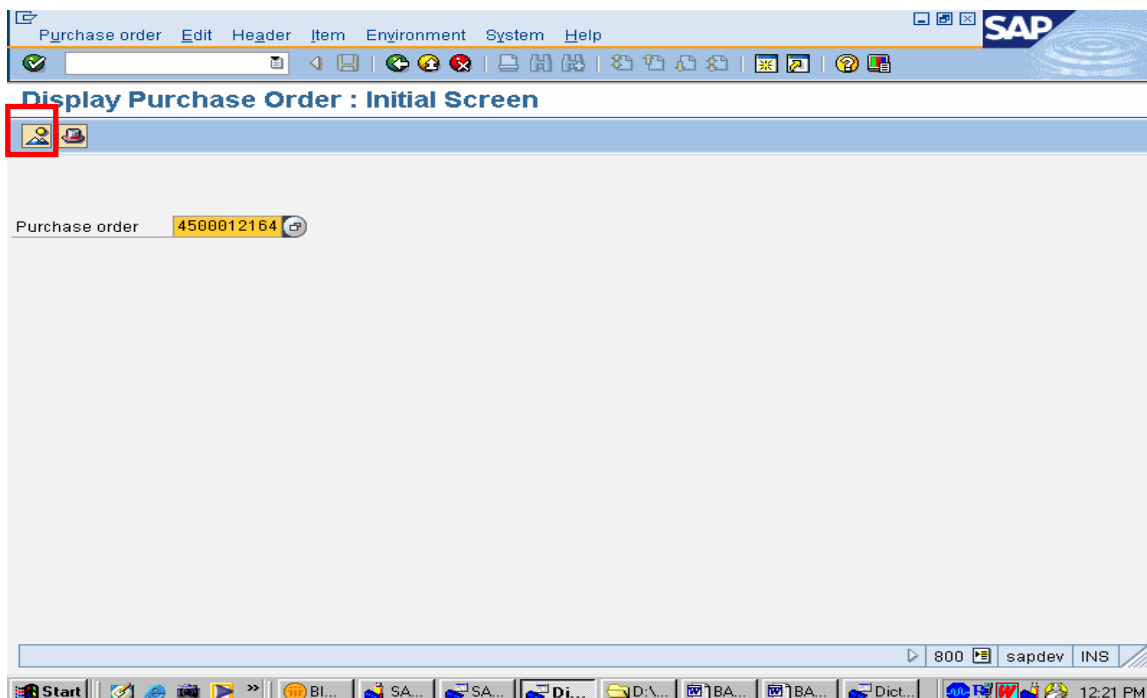
### Output of the Report





### Testing the Report:

Give transaction ME23 and give the Purchase order no, you can see the details of the report there.



[illegible]

## Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.