# SAP ABAP ALL IMP IQ'S

BY

MR. Babu

# DATA DICTIONARY

# DDIC Objects

**1) What are the technical properties of a table and explain about them?**

**Ans:- There are 7 types of tech. properties .They are**

**a) Delivery Class: - It is used to control the Transport of the table data.**
**It is used to specify the type of the table.**

- **In Real time we mostly use the delivery class 'C'.**
- **In case of delivery class 'A' the table can maintained in any system (development, quality, production systems) using SM30.**

**b) Data Browser/table view maintenance: - This property is used to specify the whetherthe table can be maintained or not.**

- **In real time we mostly use "Display/maintenance allowed "option.**

**c) Data Class:- It is used to specify the type of the data to be maintained in the table.**

- **In Real time we mostly use APPL0 or APPL1.**
- **APPL0 – Master data – Access frequently &updated (changed) rarely. Ex: D.O.B**
- **APPL1 – Transactional data – Access frequently & updated frequently.Ex: Sal**
- **Organization & Customizing data – Updated rarely. Ex: Company Codes**

**d) Size Category: - It is used to specify the initial space required for the table in the data base.**

- **In real time we mostly use '0' category. ( 0 – 7 ) -> [ 0 – 7100]**

**e) Buffering: - It is used to control (reduce) the traffic in the data base when the table accessedby the multiple users.**

- **In Real time we mostly use the mode "Buffering not allowed " – Table Can't be buffered**
- **Buffering allowed but Switched Off – Buffering allowed but not activated**
- **Buffering Switched on – The table can be buffered. In case of this option any one of the buffering types should be selected.**
  - **Single record buffering: In case of this buffering type, record by record is buffered from the database tables.**
  - **Generic Area buffer: It is used to buffer the table data based on the specified key fields. In case of this buffering type, the no. of key fields must be specified.**
  - **Fully buffer: The entire table is buffered.**

**f) Enhancement category: - It is used to enhance the particular fields of the table.**

- **In Real time we mostly use "Can be enhanced(DEEP)".**

**g) Table Maintenance generator: - It is used to maintain multiple entries & validate the tabledata using the T-code SM30.**

**2) What is the purpose of Table Maintenance Generator (TMG)?**

**Ans: - It is used to maintain(Change, create, modify) the custom tables using the standard T-Code SM30 & the T-code SE54 is used to create TMG.**

**3) What are the events in TMG?**

**Ans: Validation can be done by Table events using SM30. Totally 38 events are there in TMG.**

**1. Before saving the data in the database.**

**2. After saving the data in the database.**

**3. Before deleting the data display.**

**4. After deleting the data display.**

**5. Creating a new entry.**

**4) What are the steps to create TMG?**

**Ans: - Following steps can be used to create the TMG.**

**a) Execute SE11**

**b) Select the Radio button DB Table**

**c) Provide the Table name & click on change**

**d) Click on utilities.**

**e) Click on TMG.**

**f)Select the auth. group as &NC& and Provide the functiongroup as your table name**

**g) Select the one-step radio button.**

**h) Enter overview screen number (could be any). [4 digit only]**

**i)Click on create icon(F6) & Click on save.**

**j) Press enter button & Click on Local Object & Click on Back.**

**5) What are the types of data base tables and differences between them?**

**Ans: - There are 3 types of data base tables.**

**1) Transparent tables: - One to One relationship – Can be Join With Other Transparent tables**

**2) Pooled tables: - Many toOne relationship.**

**3) Cluster tables: - Many to One relationship.**

- **In Real time we mostly use transparent tables.**

| Transparent Tables | Pooled Tables | Cluster tables |
|---|---|---|
| 1. They have one-one Relationship with the data base tables | Many– One | Many – One |
| 2. Secondary indexes can be created. | Not Possible | Not Possible |
| 3. Both Open & native SQL canbe used. | Only Open SQL | Only Open SQL |
| 4.The Table name ,fields name andno .of fields are similar inDDIC and Data base | Table names and filed names are different in Data base | Table names and filed names are different in Data base |
| 5.It Follows both Binary and linear Search | Binary Search | Linear Search |
| 6. It is mirror image of DB Table. | It is linked with another set of tables called table pool in the DB | It is stored with another set of tables called Table Cluster Pool |

**Open SQL: It is a set of statements which can interact with any kind of database.**

**Native SQL: It is specific to the particular database & it hits the database directly.**

**Q) How data is stored in cluster table?**

**Ans: A cluster table contains data from multiple DDIC tables.It stores data as a name value pair (varkey, vardata)**

**6)Is it possible to create custom transaction code to maintain the table? If yes, how to create it?**

**Ans: - Yes, We can create by using the custom T-Code using SE93.**

**7) What are the differences between Include structure and Append structure?**

| .INCLUDE Structure | .APPEND Structure |
|---|---|
| 1. It can be inserted at any position. | 1. It can insert only at end of the table. |
| 2. It is not possible with the standard tables | 2. It is possible to insert structure for both Standard & Custom tables |
| 3. All the steps of Include structure must be saved in either our own package or $TMP | 3. All the steps of Append structure must be saved in our own package |
| 4.The table must be in change mode | 4. It can be inserted in Display mode also |
| 5. It is possible to add the additional fields manually after Include statement. | 5. It is not possible to add the additional fields manually after Include statement |

**8) How to identify the standard tables? (Indications)**

**Ans:- 1) The tables which doesn't start with Z or Y.**

**2) The tables which can't be seen in the CHANGE mode.**

**9) How to see the relationship between the tables?**

**Ans:- 1) Using SQVI T-Code.**

**2) While maintain data in the child table (Foreign key relationship).**

**10) How many types of Indexes are there and explain about them?**

**Ans:- Indexes are used to avoid the duplicate data while accessing the table. Indexes can also be used for better performance while extracting the data the tables. Thereare 2 types of Indexes.**

1) **Primary index: It is generated by the system using the key field of the tables. Keyfields hold the unique data.**
2) **Secondary Index: These are created and maintained by the Technical Consultants/ABAPer. We can create 9 secondary indexes per table.We can create the secondary indexes for both custom& standard tables.**

**Q) What is Where Used List?**

**Ans: It is used to list out all the dependent objects & know the usage of the particular object.**

- **The T-code SQVI is used to check the relationships between the tables.**

**Q) Can we add/Enhance additional customer specific field vales to standard domains?**

**Ans: Yes, the standard domains can be enhanced to add, additional customer specific fixed values & intervals using the Menu path: Go to ->Fixed Value Append at Domain level.**

**11) What are the differences between Check table andValue table?**

| Check table | Value table |
|---|---|
| 1. It is maintained at the TABLE level. | 1. It is maintained at the DOMAIN level. |
| 2. It can be used to extract the data in the programs. | 2. It can't be used to extract data since it is linked with the domain. |
| 3. The Search Help is displayed from the Check table on the foreign key field of a foreign key table. | 3. The Search Help is displayed on each field which is linked with the same domain of the Value table. |
| 4. It forces the user to select any one of the check table field entries | 4. It does not force the user since it has empty value by default. |

**12) What are the types of Views and explain about them?**

**Ans: - Views are the virtual/imaginary tables. It does not contain the data permanently.**

**View contains the data at Run time only.**

1) **Database View: - It is used to combine the multiple tabled data by joining them.It can be used to extract the data in the programs.**
2) **Maintenance View: - It is used to maintain multiple tables data using the T-code SM30 (TMG).**
3) **Help View: - It is used in the selection method in an Elementary Search Help to provide the list of possible entries from the multiple tables.**
4) **Projection View: - It is used toHIDE the unwanted fields of the table. It can be defined using one table only.**

**13) What are the Search Help, types and explain about them?**

**Ans: - It is used to provide all the possible entries to the input variable as search help.**

**1) Elementary Search help: - It is created using one base table.**

**2) Collective Search help: - It is the collection of Elementary Search help.**

**Search Help Exit: It is a Function module to filter possible entries of a search help.**

**It is used to modify the F4 values at run time.**

**14) Is it possible to create a table/structure without creating any data element and domains?**

**Ans:- Yes , we can create using the predefined types.**

**It is used to create the table without using Data Elements/Domains.**

- **If the table or structure is created using predefined types, the symbol ' + ' is appeared instead of field labels while maintaining the tables.**
- **Reusability is not possible with the Predefined Types.**

**15) What are the Lock objects and explain about the different lock modes?**

**Ans:- They are used to control the parallel access for the same data by the multiple users. It avoids the concurrency access of multiple users on the same Database.**

- **The custom lock object name should start with 'E' followed by the table name. Ex: - E < ZB16_Student >.**
- **It generates two function modules one start with ENQUEQE and other start with DEQUEQE.**
- **ENQUEQE - Request for Lock**
- **DEQUEQE -To Release the Lock**

**Types of Locks:**

1) **Write/Exclusive: The locked data can be accessed by single user only. All other Lock requests are rejected.**
2) **Read/Shared: The multiple users can access the same data at the same time. If no user tries to edit the data, all other users cannot access this data.**
3) **Exclusive but not Cumulative: It can be requested only once by the given transaction Codes. All other lock requests are rejected.**

**SM12 : This T-code is used to list out the locked objects & to delete the locked objects.**

**16) What are the difference between Table and Structure?**

| Table | Structure |
|---|---|
| 1.Table contains the data | 1. Structuredoes notcontain the data. It is filled with single record at Run time. |
| 2.It must contains at least 1 field as Primary Fields. | 2. It does not contain primary fields. |
| 3.Database must contains Delivery class & Technical settings. | 3.It does not contains Delivery class & Technical settings. |

**17) Explain about Data element & Domain.**

**Ans:-Data Element :- It is one of the data dictionary object which can be used to maintain the field labels (descriptions) of the fields in the table or structure.**

- **Parameter ID's are maintained at Data Element level under Further Characteristics tab.**

**Domain: - It is one of the data dictionary object which can be used to maintain the technical properties of the fields in the table such are Data types (char, numc....etc), length of the data (10,20...etc)**

- It can be used to maintain Value table to provide search help on the field.
- Conversion Routines are maintained at Domain Level.

**18) How to adjust the table/View?**

**Ans :- If there was any changes made at the table level the system will throw an error as 'Structure change at field level '. The table must be adjusted to rectify this error.**

- **The T-Code SE14 or the path: Utilities --->data base objects ---> data base utility can be used to adjust the table.**
- **If the key field is changed as non-key field then the system throws an error 'Convert Table': structure changed at Table level.**

**Q) I have a Z-Table, I wants to add one more field to the table without disturbing previous data, what should I do?**

**Ans:While adjusting database table in SE14, select 'Save Data' radio button.**

**19) What are the different ways to provide search helps?**

**Ans:- We can provide in 3 ways .**

- ✓ **Using value table at Domain level.**
- ✓ **Foreign key relationship.**
- ✓ **Elementary and collective search help.**

**20) What is type group and explain?**

**Ans :- It is one of the data dictionary objects.**

- **It is used to define the Global types, Internal tables and constants which can be referred anywhere in the system.**
- **The ABAP statement TYPE-POOLS is used to call/define the type group.**

**21) What is table type and explain?**

**Ans:It is one of the data dictionary object used to define the global internal table which can be referred across the system.**

**22) What is the Maximum No. of structures that can be included in a table or Structure?**

**Ans :- 9 ( Nine )**

**23) Can you use all the Views in the ABAP Program?**

**Ans :-No . We can use only Projection view orDatabase view in the Program.**

**24) In which table are the programs, Tables, Development classes are sorted in?**

**Ans:- The Programs details are stored in the table TADIR, Data base tables in DD02L and DD02Tand the development class packages in TDEVC.**

**25)Explain about the tables TADIR and TRDIR?**

**Ans:TADIR is a table which holds the Data Dictionary objects. i.e; Data elements, Domains, Tables & TRDIR stores all the Programs details.**

**26) When youcreate new entries in the table the field values are always in Uppercase. How do you get the data with mixed case? (OR) How to make a table field to store a lowercase value?**

**Ans :-The reason for this is that the Domain for the Field in the table might have Lowercase checkbox unchecked. Check the LOWERCASE CHECKBOX to preserve the case of your data.**

- **The checkbox Lower case is checked to make character fields as a case sensitive to accept both Upper & Lower case letters. Since the system takes the 'Character' data type as Upper case letters by default.**
- **Also, we can determine whether a field stores lower case value or not by checking the "Lower Case" check box of its domain. If this is checked then it will store both lower case and upper case. If it is not checked then it will store only upper case values.**

**Steps:**

**1. Create a demo table say ZDEMO_TABLE.**

**2. Generate Table Maintenance for this.**

**3. Now create new entries with the help of transaction SM30.**

**4. Click on New entries.**

**5. Enter value and save.**

**6.Data will be saved in the capital letters or check entries using transaction SE16.**

**7. Now we want to store the entries in lowercase. Here are the steps to achieve this.**
**a. Go to the domain of the field to which you want to store values in lowercase.**
**b. Click on Change/display button.**

**c. Check the lower Case check box.**

**d. Save and activate the domain.**
**e. Now again create new entries with the help t-code SM30. Enter values and save.**
**Output:**

**Values will be stored in the given case.**

**27) What is Transaction code?**

**Ans: They are the commands/shortcuts to execute corresponding application. In SAP**

**each screen is associated with T-Code.**

**a) Standard T code: They are the system provided to work with the pre-defined applications. The**

**Standard T-codes could only be the combination of any alphabets/alpha-numeric**

**Ex: SQVI, SE11, SE38 (SE-System Engineering, SM – System Machine, SQ-System query)**

**b) Custom T code: They are created & maintained by ABAP Technical Consultant**

**SE93 is used to work with custom T-code.**

- **TSTC– It is the standard table which contains all the Transaction Codes in the system.**
- **TFDIR – Function Modules**

**28) What is Data Dictionary/DDIC?**

**Ans: Data Dictionary is central & structured source of data. The data objects can be referred anywhere in the system & the data definitions are maintained in the DDIC.**

**29) What is Client & Server?**

**Ans: Client: It is a software/hardware combination which can send the requests for services from the central system.**

**Server: It is a software/hardware combination which can provide the services to a group of clients.**

**30) Explain about 3-Tier Architecture?**

**Ans: 1. Presentation Layer (GUI): It is an environment where all the users are sent for the services.**

**2. Application Layer: It receives requests from the presentation layer to get them processed. It also sends back the services to the presentation layer. There are 4 types of components in application layer.**

a. **Dispatcher: It is a link between the presentation layer & the work process. It receives the request from the presentation layer & sends them to an appropriate work process to get them processed.**
b. **Gateway/Layer: It is a communication protocol of the cross application components such are $, BAPI etc. It provides link between two SAP systems.**
c. **Shared Memory: It is a common memory which can be shared by all the work process.**
d. **Work Process: It is a component to execute the applications.**

**3. Database Layer: It is central repository of the data. It provides data base services to group of clients.**

**31) What is Command field?**

**Ans: It is used to execute the T-codes in SAP which is located on standard tool bar.**

**/o: It is used to execute the T-code in another new session.**

**/n: It is used to execute the T-code on the same session by closing current session.**

**32) What are Workbench objects/tools?**

**Ans: It is a collection of tools which can be used to develop the applications, change the existing applications & implement the customer specific business requirements.**

**33) What is Object Navigator?**

**Ans: It is the central point of entry to work with any kind of work bench objects such as Programs, DDIC objects, Function groups etc. The T-code is: SE80.**

**34) Differences between Types & Field Strings?**

| Types | Field Strings |
|---|---|

| | |
|---|---|
| 1. They can be used as reference for Work area's & Internal tables | 1. They can be used to declare Work area's & Internal tables directly. |
| 2. The ABAP statement TYPES is used to declare Types. | 2. The ABAP statement DATA is used to declare Field Strings. |
| 3. It don't hold the data since they are not data objects. | 3. It holds thedata since they are data variables. |

**35) What is Foreign Key?**

**Ans: A key field is defined as a table field & is primary key of another table is called foreign key.**

- **The Check table field should be a primary key.**
- **The Domain name of the Check table &foreign key table should be same but field name is different.**
- **A search help is displayed to see the possible entries of a check table on the foreign key field when the foreign key table is executed.**

**36) Whatis a Client Dependent & Client Independent table?**

**Ans: The table which have the field MANDT field (Client no) is called Client Dependent table.**

**The table which don't have the MANDT field (Client no) is called Client Independent table.**

**37) What are Key field & Non-key fields?**

**Ans: The key fields don't allow duplicates. It contains unique data. The key field check boxes are checked to make the fields as primary key.**

- **The non-key fields may/mayn't allow duplicates. The key field check boxes are unchecked.**

**38) What is Initial Value Check box?**

**Ans:The Initial Value Check box is checked to take the default field values of the key fields for the 1st time.**

**39) Differences between the tables created using Data elements & Pre-defined types?**

| Data elements | Pre-defined types |
|---|---|
| 1. Data elements can be reused | 1.Reusabilty is not possible |
| 2. Foreign key relationship can be created since it contains domains. | 2. Foreign key relationship in not be created. |
| 3. Field labels(Headings) are displayed while maintain the table. | 3. '+' symbols are appeared while maintaining the table instead of field labels. |

**Q) What is a Logical Database?**

**Ans: Logical Databases are special ABAP programs that retrieve data and make it available to application programs. Use of LDB – is used to read data from database tables by linking them to executable ABAP programs.**

**40) What are the Important Tables to store the Definitions of DDIC?**

| Table | Short Text |
|---|---|
| DD01L | Domains |
| DD01T | Domain Texts |
| DD02L | SAP Tables |
| DD02T | SAP Table Texts – To find table names using Short Description |

| | |
|---|---|
| **DD03L** | **Table Fields** |
| **DD03T** | **Texts for fields** |
| **DD04L** | **Data Elements** |
| **DD04T** | **Data Element Texts** |
| **DD05S** | **Foreign Key Fields** |
| **TSTC** | **Transaction Codes** |
| **TFDIR** | **Function Modules** |

## 41) Explain about Currency or Quantity fields in DDIC?

**Ans:**

- **In SAP the currency & quantity fields are required currency key & quantity unit. Hence, the reference table & the reference field must be specified for currency & quantity fields.**
- **The data type for the currency field is 'CURR' & the data type for the currency key is 'CUKY'.**
- **The data type for the quantity field is 'QUAN' & the data type for the quantity key is 'QUAN'.**
- **The reference table could be any table which has the currency key & quantity unit fields.**
- **The same table can also be used as a reference table if it has the currency key or quantity unit fields**

**Q) What is table buffer? Which type of tables used this buffer?**

**Ans: Buffer is nothing but a memory area. Table is buffered means that table information is available on application server. When you call data from database table it will come from application server.Transparent tables and pool tables are buffered, while cluster table cannot be buffered.**

## 42) What are different types of Data Dictionary Objects?

**Ans:**

1. **Database Tables**
2. **Domains**
3. **Data types**
    a. **Data element**
    b. **Structure**
    c. **Table type**
4. **Search helps**
    a. **Elementary search help**
    b. **Collective search help**
5. **Views**
    a. **Database view**
    b. **Maintenance view**
    c. **Help view**
    d. **Projection view**
6. **Lock objects**
7. **Type groups**

- **The*Objects*Views, Match Code and Lock objects are called <u>Aggregate Objects</u> because they are formed from several related table.**

# Fundamentals

## Fundamentals

**1) Explain about ABAP Queries?**

**Ans :- It is the system provided tool to generate the reports. It doesn't require any programming Knowledge.**

**2) What are the Transaction Codes associated with ABAP Queries?**

**Ans:- The T-Codes SQ03 , SQ02, SQ01 are used to work with the ABAP Queries.**

**3) What are the purposes of SQ01, SQ02 and SQ03?**

**Ans :-**

**SQ03 :- It is used to maintain the user groups. A user group is a collection of Info sets and Queries. User groups can also be used to control the user authorization to access the corresponding queries.**

**SQ02 :- It is used to maintain the Info sets . These are used to specify the different data sources form which the data can be extracted.**

**SQ01 :- It is used to maintain the Queries . It is used to select the selection fields & list fields.**

**4) Is it possible to write an ABAP code in Queries?**

**Ans :- Yes . Using Code icon (shift+f8).**

**5) What is the difference between TYPE and LIKE?**

**Ans :-**

**TYPE: - It is used to refer the Elementary data types (I, F, P) & DDIC objects. It doesn't copy the properties of DDIC objects.**

**Syntax: - DATA : <gv-var> TYPE <I/D/F/C.....>.**

- **This Statement doesn't copy the existing properties of the objects.**

**LIKE: - It is used to refer only the data objects.**

**Syntax: - DATA : <gv-field> LIKE <dbtab_field>.**

- **This Statement copies the existing properties of the objects.**

**6) What is the default type and length of the variable?**

**Ans :- The default type is Character ( C ) and length is one (1).**

**7). Differences between Write & Move Statement?**

**Ans:Write: It is used to move the data from the source data variable to destination variable without missing conversations.i.e; Currency & Quantity fields.**

**Move: : It is used to move the data from the source data variable to destination variable with missing conversations(may be missed out).**

| In case of Write | In case of Move |
|---|---|
| <gv_currency><gv_dest> | <gv_currency><gv_dest> |
| 5,895.50 5,895.50 | 5,895.50 589550 |

**8) What are the differences between Work Area and Field Symbol?**

| Work Area | Field Symbol |
|---|---|
| 1. It is the data variable. | 1. It is a Pointer. |
| 2. The ABAP statement is DATA is used to declare or define work areas. | 2. The ABAP statement FIELD-SYMBOLS is used to declare Field symbols. |
| 3. The ABAP statement INTO is used to process record by record through work area . | 3. The ABAP statement ASSIGNING is used to assign the FIELD-SYMBOLS . |
| 4. The ABAP statement MODIFY must be used to modify the internal table . | 4.No need to use MODIFY , Since it is a pointer . |
| 5.No need to specify within the Angular brackets (<>). | 5.It Must be specify within the Angular brackets (<>) . |

**9) Differences between Work area and internal tables?**

| Work Area | Internal Tables |
|---|---|
| 1. It Holds single record at a time. | 1. Holds Multiple records. |
| 2. It doesn't have BODY. | 2. It has BODY []. |
| 3.The Record by record is processed through the work area | 3. In case of internal table with header line the record by record is processed through header line. |

**Q) What is difference between Internal Table without Header Line and Internal Table with Header Line?**

**Ans: Internal Table without Header Line creates an internal table and work area separately. When the Internal Table with header table is created then the work area is created automatically with the same name that of the internal table.**

**10). Explain about Types.**

**Ans:- TYPES :- It is used to club the multiple fields from the multiple tables as a single unit . It is used as a reference to declare the Data variables, Work areas, Internal tables etc.**

**DATA: It is used to declare the data variables such as variables, work areas, internal tablesetc in an application program.**

**Syntax: - TYPES: BEGIN OF <typ_dbtab1_dbtab2>,**

**<F1> TYPE <data element1/dbtab1-F1/dbstr_F1>,**

**.**

**<F5> TYPE <data element5/dbtab1-F5/dbstr_F5>,**

**END OF <typ_dbtab1_dbtab2>.**

**DATA:<gs-dbtab1_dbtab2> TYPE <typ-dbtab1_dbtab2>. //declare the work area**

**11) What are the different ways to declare the Internal Table and which is preferable?**

**Ans :-There are two ways to declare the Internal tables.**

1) **Syntax:-**

**DATA: <gt_dbtab> TYPE STANDARD TABLE OF <dbtab/dbstr.....>.//Preferable.**

2) **Syntax:-**

**DATA: <gt_dbtab> TYPE TABLE OF <typ_dbtab /dbtab/dbstr> WITH HEARDER LINE.**

**12) What are the differences between PARAMETERS and SELECT-OPTIONS?**

| PARAMETERS | SELECT-OPTIONS |
|---|---|
| 1. It is used to define Check boxes, Radio buttons & single I/P fields. | 1. It is used to define Single &Range input Fields. |
| 2.The ABAP statements TYPE/LIKE is used to refer the Data types | 2. The ABAP statement FOR is used. |
| 3. The Operator 'EQ' or '=' is used in the WHERE condition. | 3. The operator 'IN' is used in the WHERE condition. |
| 4. It is use as an input data variable | 4. It is system generated internal table with the fields SIGN, OPTION, HIGH & LOW |

**13)What are the different types of internal tables and explain about them?**

**Ans :-Internal Tables: -They are the intermediate tables to hold the multiple records at run time.**

- **There are 3 types of internal tables.**

1. **STANDARD Internal tables: -The standard Internal Tables are filled using the ABAP statement 'APPEND'/'INSERT'. It accepts the duplicate records. Searching of arecord is Linear Search. It is the default index table.Standard internal table can be sorted explicitly.**

   **Syntax:- DATA:<gt_dbtab> TYPE STANDARD TABLE OF <typ_dbtab/dbstr.....>.**

2. **SORTED Internal table: -The ABAP statement 'INSERT/APPEND' is used to fill Sorted Internal Table. It does not accept theDuplicate records. Searching of a record is Binary Search. Sorted internal table cannot be sorted explicitly. An additional statement with UNIQUE/NON-UNIQUE key must be used to declare Sorted Internal Tables.**

**Syntax :-DATA: <gt_dbtab> TYPE SORTED TABLE OF <dbtab/dbstr.....> WITH UNIQUE/NON-UNIQUE KEY <k-f1><k-f2>.**

3. **HASHED Internal tables: - They are Non Index tables. The Hashed Internal Tables holds huge amount of data than Standard & Sorted internal tables since they follow 'Hashed Algorithm'. The ABAP statement 'WITHUNIQUE KEY' must be used to declare Hashed Internal Tables. It holds huge amount of data (2 GB) than standard & sorted internal tables.**

**Syntax:-DATA:<gt_dbtab> TYPE HASHED TABLE OF <dbtab/dbstr.....> WITH UNIQUE KEY <k-f1><k-f2>.**

**Q) Can we sort internal table without using SORT statement. Explain?**

**Ans: We can sort internal table without using SORT statement by declaring sorted internal table .**
**Ex: DATA: IT_MARA TYPE SORTED TABLE OF MARA.**

**14) How to define SELECT-OPTIONS without specifying the table name?**

**Ans :-By declaring a Global Variable and use it as a reference while defining a select option instead of TABLES statement .**

**TABLES: This statement is used to specify the database tables/database structure names Which can be used as a reference while declaring the SELECT-OPTIONS**

- **This statement in not advisable since it creates unnecessary work areas.**

**15) Explain few of the additions to PARAMTERS and SELECT-OPTIONS statements.**

**Ans :-Obligatory: It is the keyword to make the input field as mandatory**

**Default: It is the keyword to pass the default values to the input fields.**

**As checkbox: It is the keyword to define the checkboxes in the selection screen**

**Radio button group: It is the keyword to define the radio buttons in the selection screen**

**As list box: It is used to define the dropdown list in the selection screen**

**MATCHCODE OBJECT: It is used to specify DDIC search help to the input fields**

**MODIFID: It is used to assign all the screen elements to the modification group to format the Selection fields dynamically**

**LOWER CASE: It is used to make the input fields as case sensitive to accept both upper & Lower case letters**

**Memory ID: It is the keyword to access the field value from one session to another session**

**Additions of SELECT-OPTIONS**

a) **NO-EXTENSIONS: This addition is used to hide the multiple selection button in the Selection screen.**

b) **NO-INTERVALS: This addition is used to hide the higher limit of select-options in the Selection screen.**

c) **DEFAULT…TO: This addition is used to pass the default values to lower & upper limits**

**16). Differences between Ranges & SELECT-OPTIONS?**

**Ans: It is used to define range internal tables. It is similar to SELECT-OPTIONS.**

- **SELECT-OPTIONS are displayed in selection screen where as Ranges doesn't appear in the selection screen.**

**Q) What is the Size of the internal tables?**

**Ans :- Internal Tables have 2GB(Gigabytes) of memory space and the initialize size is of 8 kb which can be extend further.**

**17) Explain Some Additions to Report statement?**

**Ans: 1.No standard page heading: This addition is used to hide the page heading in the output.**

**2. Line size: This addition is used to specify the width of a list in the output.**

**3. Line count: This addition is used to specify the length of the output list in a page.**

**4. Message ID: This addition is used to specify the message class**

**Q) Some Keywords?**

**Ans:Append:**

**It is the keyword used to append record by record from work area to at last record of the internal table. Syntax: APPEND <gs_dbtab> TO <gt_dbtab>.**

**Insert:**

**This statement is used to insert a new record at any position of an internal table.**

**Syntax : INSERT <gs_final> INDEX <n>? SY-TABIX.**

**Collect:**

**It also used like an APPEND statement. This statement is used to compare character/string fields data & summarizes the numeric field data. It is used to display sub-totals.**

**Syntax :COLLECT <gs_source> INTO <gs_collect>.**

**Lines:**

**It is the keyword which returns no of records available in the internal table.**

**Occurs:**

**It is the keyword; it allocates 8KB of memory for the internal table by default. If the data in the Internal table exceeds 8KB then it will bring one more 8KB of memory & so on up to 2GB.**

**Clear: It clears the contents of the work area & internal table with header lineITAB[ ]& variables**

**Refresh: It clears the contents of the internal table only.**

**Free: It clears the contents of internal table along with allocated memory.**

**18) Explain some of the useful additions of WRITE statement?**

**Ans:- Some of the useful Additions to WRITE statement.**

- **....WRITE: / ---->It is used to display the data in a new line.**
- **....WRITE: /n (m) ----->Displays 'm' characters form position 'n'.**
- **…UNDER OTHER OBJECT --->To display the data under the previous displayed data.**
- **....COLOR ---->It is used to display the data in different colors. color codes[0 – 7]**
- **…LEFT-JUSTIFIED/CENTERED/RIGHT-JUSTIFIED: These are used to adjust the data at left hand side, center & right hand side.**
- **....AS CHECKBOX ---->Display the checkboxes in the output.**
- **....EDIT MASK ----> It is used to display the mask on confidential data.**
- **…INPUT: It is used to make output list as editable.**

- ....NO_GAP ---->To remove the gaps in the data.
- …CURRENCY: It is used to specify the currency keys for the currency data variables.
- ....UNIT ----> It is used to specify the required units while displaying the quantity data.
- …HOTSPOT : It is used to display the hand symbols on the output data when the cursor is placed.

**Q) What is the basic Difference between internal tables and database tables? How can we differentiate by looking at the tables? And how Handling of Internal Tables?**

**Ans :- The basic Difference is database tables are Stored in DB Server and the internal tables are virtual tables these are created run time only.**

**19) Explain about the System Fields with which you have worked?**

| | |
|---|---|
| INDEX | Index of the loop |
| DATUM | Current system date |
| UZEIT | Current system time |
| SUBRC | Written code of the ABAP keywords |
| TABIX | Index of an internal table |
| DBCNT | The no. of extracted records from the database |
| BATCH | Program is running in back ground |
| REPID | Current program name |

**20) Explain about String Comparison Operations and Mathematic Functions?**

**Ans: String comparison operators: -The below string comparison operators can be used to compare the string data variables.**

| Operator |
|---|
| Contains Any (CA) |
| Contains Not Any (NA) |
| Contains Only (CO) |
| Contains Not Only (CN) |
| Contains Pattern(CP) |
| Contains No Pattern(NP) |
| Contains String (CS) |
| Contains No String(NS) |

**CA: This condition becomes true when the source string contains any one of the character in the target string.**

**NA: This condition becomes true in case of string comparison operator CA is false.**

**CO: This condition becomes true, when the source string contains only the similar string in the target string including case.**

**CN: This condition becomes true in case of string comparison operator CO is false.**

**CP: This condition becomes true, when the source string contains the specified pattern in the target string.**

**NP: This condition becomes true in case of Logical expressions with a string operator CP is false. (Opposite to CP)**

**CS:This condition becomes true, when the source string contains similar string in the target in both Upper & Lower case (not a case sensitive).**

**NS: This condition becomes true in case of string operator CS is false (Opposite to CS).**

**Mathematical Functions: - The below Mathematical operators can be used on the numbers.**

| Operator | Statement |
|---|---|
| Equal To | EQ |
| Not Equal To | NE |
| Less Than | LT |
| Greater Than | GT |
| Less Than or Equal To | LE |
| Greater Than or Equal To | GE |

## 21) Explain about Constants, Text Symbols, Selection Texts and Variants?

**Constants:-They are used to avoid the hard coded text in the programs and improve the reusability in the program. The ABAP statement 'CONSTANTS' is used to define constants.**

**Text Symbols: - The ABAP Statement TEXT followed by the symbol number (XXX) can used to define or call the text symbol.**

**Selection texts: - They are used to maintain the Field labels (descriptions) of the input fields of the selection screen. The T-code SE63 is used to translate the text elements.**

**Variants: - They are used to save the input data for further execution in the selection screen. They can also be used to avoid the data entry errors& schedule a program in the back ground.**

## 22) Explain about the useful String Operations?

| Operator | Description |
|---|---|
| STRLEN | Returns the no of chars/length of the string |
| CONCATENATE | To combine the multiple strings into single target string |
| REPLACE | To replace with one string to another target string |
| FIND | Find the string in the source String |
| SPLIT | Separate the source string into diff strings |
| CONDENSE | It compress the string without any gaps |
| MATCH OFFSET | It compress the string without any gaps |

## 23) Explain some of the useful additions of Selection-Screen?

**Ans :-**

**SELECTION-SCREEN: - This statement is used to format the selection screen. The default screen number for the selection screen is 1000.**

**Some of the useful Additions:-**

- **.....BEGIN OF BLOCK....WITH FRAME: - It is used to define the selection screen blocks with frames**

- .....WITH FRAME TITLE: - An addition TITLE with the selection screen FRAMES is used todefine the TITLES for the selection screen Frames.
- .....SKIP: - It is used to keep the blank lines in the selection screen.
- .....ULINE: - To display horizontal lines in the selection screen.
- .....BEGIN OF LINE: - It is used to define the input fields, radio buttons, push buttons, check boxes etc in the same line.
- …PUSH BUTTON: It is used to define the push buttons in the selection screen.
- …LINE: It is used to define the Input fields, Radio buttons, Check boxes, Push buttons etc in the same line.

**24) What are the differences between Normal Search (SELECT…ENDSELECT) and Pointer Search (INTO TABLE)?**

**Ans :-**

| Normal Search(SELECT…..ENDSELECT) | Pointer Search(INTO Table) |
|---|---|
| 1. It hits the data base table for each record since it is a loop statement | 1. It hits the data base table once to extract all the required data since it is a statement |
| 2.The ABAP statement 'APPEND' must be used to append record by record into an internal table | 2.No need to use 'APPEND' |
| 3.Some times it may give runtime error while debugging | 3.It doesn't give any short dumps |
| 4.Not advisable due to lack of performance& data base load | 4. It is advisable since it extracts the data faster & has good performance |

**25) What is the difference between SY-INDEX and SY-TABIX?**

**Ans:**

- SY-TABIX: SY-TABIX is a system variable which stores the index current processing record of an internal table.So when you are looping over an internal table, you use SY-TABIX.

  LOOP AT ITAB INTO WA. **SY-TABIX stores index number of internal table record ENDLOOP.

- SY-INDEX: SY-INDEX is a system variable which acts as a loop iteration counter, it stores loop iteration number.When you use DO ENDDO / WHILE for looping, there is no table involved. So you use SY-INDEX.

  Ex: DO 10 times. **SY-INDEX stores number of iteration of loop ENDDO.

**26) What are the differences between Classical Reports and ABAP Queries?**

**Ans :-**

| ABAP Queries | Classical reports |
|---|---|

| | |
|---|---|
| 1. It is the system provided tool to generate the reports. | 1. It is a Utility. |
| 2. It doesn't require any programming language. | 2. It requires the programming language. |
| 3. It is not possible to change the system generated program without having Access key. | 3. It can be change since it is in Custom name space (Z or Y). |
| 4. Column Alignment is done by the system. | 4.It is COMPLEX to Align the Columns |

**Q) What is version management in sap abap.Do we have this concept in scripts also?If at all then how do you handle in scripts and what's its usage?**

**Ans:**

- ✓ **Version is used to differentiate the current active version and previous active version. If our current active version is wrong we can retrieve the previous version.**
  **path to version management : Utilities -> versions ->version management.**
- ✓ **you can compare current version and previous active version also.**
- ✓ **For Script released versions ... go to T-code SE03 ---> select SEARCH FOR OBJECTS IN REQUESTS/TASKS**



# Modularization Techniques

## Modularization Techniques

**1) What are the Modularization techniques which you follow?**

**Ans :- They are used to organize the ABAP code in the proper way , make the program more Understandable and make use of the code re-usable to avoid the duplicate data.**

**To divide the business processing logic into reusable block of statements.**

- ✓ **Following are the different types of Modularization techniques**
  - **INCLUDES: These programs are part of another main/include programs.Include programs can't be executed independently whereas the same include program can be included in any no of executable programs. The ABAP statement INCLUDE is used to call or define the INCLUDE programs.**
  - **Subroutines: These are statements local modularization techniques. The sub-routines can be debugged at run time. Calling should be 1st& Definition should be next. The ABAP statement PERFORM is used to call the sub-routines & FORM…ENDFORM is used to define the sub-routines.**

    **There are 2 types of sub-routines.**

    I. **Internal Subroutine: These are defined & called in the same program.**
    II. **External Subroutine: These are defined in one program & called in another program.**

    **Actual Parameters: The parameters which can be called while calling the subroutines (with PERFORM statement).**

    **Formal Parameters: The parameters which can be passed while defining the subroutine (with FORM statement).**

    - ➢ **The no of actual parameters should be similar to the no of formal parameters.**

- **The sub-routines definitions should not be nested (form with in the form) but the sub-routine can be called within the sub-routine.**
- **The statements between 2 sub-routines definitions cannot be accessed**

- **Macros: These are used for the complex write statements & for long calculations. Macros can take up to 9 place holders (&1…. &9). In macros Definition should be 1st& Calling should be next. The ABAP statement, DEFINE…END-DEFINTION is used to define a macro. The macro definitions cannot be debugged at run time. The standard table TRMAC pooled table is used to maintain macros.**
- **Function Modules: These are global modularization objects. The function modules can be called any where in the same system & in another system also. The function modules are defined in function builder.**

  **The T-code SE37 is used to work with the function modules & SE80 is used to work with the function group. The function modules should be assigned to a function group & Function group is the collection of function modules.**
  - **The standard table 'TFDIR' is used to find FMs since it contains the entire Function module in thesystem.**
  - **Normal Function Module: These are defined & called in the same system to reuse the central source code.**
  - **Remote Function Module: These are defined in one system & can be called another system.**
  - **Update Function Module: This is used for SAP LUW (Logic Unit of Work). These function modules are triggered in case of implicit or explicit COMMIT work is encountered. Commit Work is used to make changes in data base permanently.**
- **Message class: This tool is used to maintain the Messages in the message pool. They are used to give messages for instructing the users.**
  **The standard T-Code SE91 is used to work with the message pool.**
  **Different types of messages are Error, Success, Warning, Information, Abort, Exit.**

- **Text symbols & Constants: The ABAP Statement TEXT followed by the symbol number (XXX) can used to define or call the text symbol. Constants are used to avoid the hard coded text and improve the reuse them in the program.**
  **The T-code SE63 is used to translate the different types of text elements.**

**2) What are the differences between Subroutines and Function Modules?**

| Subroutines | Function modules |
|---|---|
| 1. These are local. We can access the Subroutine within the server only. | 1. These are global. We can access the function module with in server as well as Outside the server. |

| | |
|---|---|
| 2. We can't execute the subroutine independently | 2. we can execute function module independently using the T-code SE37 |
| 3. Subroutines can't handle the Exceptions. | 3. Function module can handle the errors through Exceptions. |

**3) What are the differences between Macro & subroutine?**

| Macro | Subroutine |
|---|---|
| 1. Definition & calling in the same program. | 1. Definition & calling may/may not in the same Program. |
| 2. Definition should be the 1st& calling should be The next. | 2. Calling should be the 1st& Definition should Be the next. |
| 3. Macros can take up to 9 inputs. | 3. Subroutines can take any no of inputs. |
| 4. Sub-routines can be debugged at run time. | 4. Macros cannot be debugged. |

**4) What are the different parameters in Function Modules and Explain about them?**

**Ans: Following are the different types of Parameters in Function Modules**

- **Attributes:**
  **It contains technical properties of a function module such are type of function module, function group.**
- **Import parameters:**
  **They are imported from the calling program to the function module.**
- **Export parameters:**

**These are used to exported the data from the function module to the calling programs.**

- **Changing parameters:**
  **These are imported from the calling program, changed them in the function modules & exported/returned to the calling program. (Import -> Change -> Export)**
- **Tables:**
  **These can be used to import & export only for internal tables.**
- **Exceptions:**
  **They are used to raise/handle the messages in the function module based on return code (SY-SUBRC).**
- **Source code:It is used to write central ABAP source code.**

**5) Explain about different methods to pass the parameters to sub routines?**

| Method | Description | Effect |
|---|---|---|
| BY Refere | It is the default method. The pointer used the same memory allocation for both actual & formal parameters. From calling to | The Actual parameters are changed with the |

| | | |
|---|---|---|
| nce | definition & definition to calling, all parameter values are passing along with the memory. | Formal parameters. |
| By Value | The system allocates new memory. The allocated memory is freed once the subroutine ends. From calling to definition & definition to calling, the parameter values only passing. | The Actual parameters are not changed with the Formal parameters. |
| By Value and Return | It is almost similar to By value but only the difference is the Formal parameter values are returned to Actual parameters. | The Actual parameters are change with the Formal parameters. |



# Debugger

## Debugger

**1) Explain about Debugger, Break points and Watch points?**

**Ans :-Debugger :- This tool is used to check the program line by line & block by block for finding and rectifying errors or bugs.**

**Break Points:- These are Statement based. They are used to stop the program in the debugger tool for checking line by line or block by block.**

- **Break points are of 4 types. We can place up to 30 break points in the program.**
  1. **Session Break Point: An icon set/delete breakpoint icon or click on statement line at gray color vertical selection in the source code.**
  2. **Static Break Point: The ABAP statement BREAK-POINT/BREAK <user_name> is used to set the Static break point. In the real time the ABAP statement BREAK-POINT should not be used since all the users are stopped.**
  3. **Dynamic Break Point: They are used to check the program in case of not possible to find a location. The command /H is used to set dynamic break points. The program is stopped at the first statement line of the performed action.**
  4. **External Break Point: They are used to check the cross application components such as RFC, BAPI from one system to another. An icon set or delete external break-points is used to set or delete external break-points.**

**Watch Points: - These are Condition based. They are used to stop the program once the set condition is reached. The Watch points can be set inside the debugger tool. We can place up to 9 watch points in the program. We can reduce debugging time using watch points.**

## 2) Differences between Classical & New debugger?

**Ans:**

| Classical debugger | New debugger |
|---|---|
| 1.It is the old debugger | 1.It is introduced from the version ECC 5.0 |
| 2.It runs on the same session to check & resolve the issues in the program | 2.It runs on the separate session |
| 3. It has Lots of limitations such as different desktops are not there, no separate sections for local , global Variables etc | 3.It is very efficient tool to check lots of actions such as different desktops, separate actions for local & global Variables & objects at a time |

**Single Step F5**: The function key F5 is used to check Line By Line of the program.

**Execute F6**: The function key F6 is used to check Block By Block of the program.

**Return F7**: The function key F7 is used to Return from the source code of the process blocks such as sub-routines, function modules, methods etc.

**Continue F8**: The function key F8 is used to jump from one break point into the next break point then execute once the last break point is reached & then execute.

## 3) How to debugBackground Jobs?

**Ans: The Background jobs can be debug using the command 'JDBG'.**

- Execute the T-code SM37
- Check the required background job to be debug.
- Enter JDBG in the Command field & Press Enter
- Keep on pressing F7 until the required program is stopped.
- Use the Function keys F5/F6/F7 to check the program Line by line/Block by Block once it is reached.

## 4) What are the steps to debug Pop-up window?

**Ans:**

- Create a text file with the required statements & save it on Local PC/Desktop.
  **[FUNCTION]**
  **COMMAND = '/H'**
  **Title = 'Debugger'**
  **Type = 'System Command'**
- Minimize the SAP Pop-up window screen to appear both Text file & Pop-up window screen.
- Drag & drop the saved text file into Pop-up window screen.

**Note: The system gives the success message as Debugging Switched ON.**

- Perform the required action to stop the corresponding program (Yes button).

## 5) What are the types of execution modes?

**Ans: There are 2 types of executionmode to execute aprogramor the transaction codes.**

**1. Foreground: It is used to execute the programs or the transaction codes directly by pressing execute icon or F8 function key. It requires user interaction.**

**2. Background: This execution mode is used to execute the programs or the transaction codes periodically such as Hourly, Daily, Weekly, Monthly etc in background. It doesn't require any user interaction. The variants must be created before the programs are scheduled in back ground.**

- **The T-code SM36 is used to define the background jobs &**
- **The T-code SM37 is used to check the status of the background jobs.**

**The standard function modules JOB_OPEN ->To open back ground job**

**JOB_SUBMIT ->To submit job in back ground**

**JOB_CLOSE ->To close scheduled job.**

# ReportsEvents



## EVENTS

**1) What are the Selection-Screen events, explain?**

- **AT SELECTION-SCREEN -> This event is triggered after providing Input to the selection screen.**

- **AT SELECTION-SCREEN OUTPUT ->This event is triggered while performing any action in the selection screen. It is used to modify (format) the selection screen.**

- **AT SELECTION-SCREEN ON -> This event is triggered after input values providedin the selection screen based. It is used to validate/check the input dataof the selection screen.**
- **ATSELECTION-SCREEN ON VALUE REQUEST FOR ->This event is triggered when the user clicks on F4 button. The standard function module F4_INT_TABLE_VALUE_REQUEST can be used to display the possible entries as search help.**

- ✓ The function module DYNP_VALUES _READ is used to capture the selection field values in the same selection screen for Dynamic search help

- ➢ At selection -screen ON HELP-REQUEST FOR ->This event is triggered when the user clicks on F1 button .The standard function module 'HELP_OBJECT_SHOW' can be used to provide the Help document to the input variable. The T-code SE61/SO72 is used to maintain the Help documentation as per the specification.

**2) Explain about the Parallel Cursor Method?**

**Ans: - This Method is used for better Performance in case of the nested loops (Loop within the loop).**

**Steps to use this Method:-**

- ✓ **Read the 2nd internal table (Item table) within the loop of the Base/Main internal table (gt_header) with the required keys and get the index of the current record.**
- ✓ **Set the loop to the 2nd internal table (ITEM internal table) to process the data from the SET INDEX and use EXIT statement to come out of this2nd loop for the unmatched data.**

**3) How many Times the Event INITIALIZATION Triggers whiledisplaying the list of 20 pages and also TOP-OF-PAGE?**

**Ans :- INITIALIZATION Triggers Only One time and TOP-OF-PAGE triggers 20 times.**

**4) What is the pre-requisite to use "FOR ALL ENTRIES"?**

**Ans :- The Base internal table which can be used in the 'FOR ALL ENTRIES' statement must be checked whether it has data or not.**

**5) What are Joins & types of Joins?**

**Ans: Joins are used to fetch the data from more than one table.**

1. **Inner join: This statement is used to extract the data from the multiples tables by joining them. Up to 2 tables can only be joined.**
2. **For all entries:This statement is used to extract the data from the data base table based on the entries of the Base internal table.**

**6) Explain about Loop termination statements?**

**Ans :- The following statements are used to come out of the loops and continue with the next loopstatements.**

- **EXIT: - This statement is used to come out of the loop statement and continue with the next statement line outside of the loop statements in the program.**
- **STOP: - This statement is used to stop the Loop process along with the next part of the loop statement.**
- **CONTINUE: - This statement is used to stop the current loop process and continue with the next loop process.**

- **CHECK: - This statement is used to check the specified conditionsand continue with the next loop process.**
  - **CHECK = CONDITION + CONTINUE.**

**7) Explain about the events associated with Classical Reports?**

**Ans :- Following are the EVENTS associated with the Classical Reports .**

- **LOAD-OF-PROGRAM: This event triggers at the time of loading the program into a memory.**
- **INITIALIZATION: This event is triggered before the selection screen is displayed. It is used to clear & refresh the data variables and used to pass the default values to the selection fields.**
- **AT SELECTION-SCREEN: This event is triggered after providing the input in the selection screen.**
- **START-OF-SELECTION:It is the default event & mandatory to execute any executable program. It is used to fetch the data from the DB. If no event is used in the executable program, the entire program will be triggered under Start-Of-Selection.**
- **END-OF-SELECTION: This event is triggered after Start-Of-Selection event is processed. It is used to specify the data process statements.**
- **TOP-OF-PAGE: This event is triggered with the first ULINE/WRITE/SKIP/NEW PAGE statement in START_OF-SELECTION event in the program. It is used to display the column headings.**
- **END-OF-PAGE: This event is triggered at the end of each page. It is used to display the footer details of a page such as totals etc..The report addition LINE-COUNT must be specified to trigger this event.**

**8) Which is the default event?**

**Ans :-The default event is START-OF-SELECTION .If no event is used in the executable program, the entire program will be triggered Start-Of-Selection. The main program begins with this event.**

**9). Is it possible to display previous drill down list from current one?**

**Ex: If suppose am in 10th drill down list. Is it possible to display 5th drill down list from current list?**

**I have 1 basic list and 15 secondary lists in interactive report. If i am in 9th list, how can i come directly to the basic list?**

**Ans: Yes. By re-setting list index = 5. i.e; SY-LSIND = 5.**

**System variable SY-LSIND stores the list number, by using this we can go to basic list.**

**10) In interactive report if user 19 times double clicked, again he want to main screen.How can do this?**

**Ans: If Sy-ucomm ='BACK'.**

**Leave to screen 0.**

**End if.**

**11) What are the events associated with Interactive Reports?**

**Ans :- Following are the EVENTS associated with the Interactive Reports .**

- **AT LINE-SELECTION: This event is triggered at the time of user clicks on any record of any list. It is used to provide the detailed list based on the current list. The system field SY-LSIND holds the Index of the next drill down list.**
- **AT USER-COMMAND: This event is triggered at the time of user clicks on any menu item. This event is used to provide the additional functionality based on the selected push button.**
- **TOP-OF-PAGE DURING LINE-SELECTION: This event is triggered with the first WRITE/ULINE/SKIP of the drill down list. It is used to provide the column headings for the drill down list.**
- **AT PF: This event is triggered when the function key is pressed. This event is used to provide the additional functionality based on the selected menu item.**
- **SET PF-STATUS: This event is triggered at the time of attaching our own GUI to the list. It is used to call the menu status.**
- **Up to 35 Push buttons can be created on the Application tool bar.**

**12) What are differences between Classical & Interactive reports?**

**Ans:**

- **The Classical reports are used to display the entire information in a single/ basic list**
- **The Interactive Reports are used to display the summarized information in the basic list & detailed information in the next drill down list. Up to 20 secondary lists are provided& the total no of lists are 21.**

**13) Which techniques used to identify the record which was clicked by the user in the previous list?**

**Ans: HIDE & GET-CURSOR**

**HIDE: This statement is used to hide the selected field value which can be passed to the corresponding drill down list to filter the unwanted records. It is used to capture the selected field values to the next drill down list. Hide statement provides Single drill down list.**

**GET-CURSOR: This statement is used to capture the selected field values & field names based onthe cursor position. This statement provides drill multiple down lists.**

**14) What are the System Fields associated with Interactive Reports?**

**Ans :- System fields associated with Interactive reports .**

| Field name | Description |
|---|---|
| LSIND | Index of the next drill down - contents of the current list Index no |
| LISEL | Selected list line value-contents of the selected record |
| LILLI | Selected list line number – contains the exact line no of the selected record |
| UCOMM | Fun. Code of the selected push button/Menu item |
| PFKey | Function Key |

**15) What are the Control Break Statements, explain about them?**

**Ans :-**

**Control Break statements: - They are used to control the data flow of an internal table.**

- **The Control break statements start with AT and ends with ENDAT.**
- **These statements should be used within the LOOP statements only**
- **Following are the different types of Control break statements.**

- **AT FIRST: - This event is triggered at the first record of an internal table. This is used to display the Header information.**
- **AT NEW: - This event is triggered at the first record of each block. This is used to display the individual headings.**
- **AT END OF: - This event is triggered at the last record of each block. This is used to display the total/sub totals in the internal table.**
- **AT LAST: - This event is triggered at the last record of an internal table.**
- **ON CHANGE OF: - It is a special control break statement which can be used outside of the loop statement also.**
  - **It is almost similar to AT NEW statement but it doesn't consider the preceding fields.**
  - **It can be used on the multiple fields using SET operators (AND, OR, BETWEEN).**

**16) What is the difference between ATNEW and ONCHANGE OF?**

**Ans :-**

| AT NEW | ON CHANGE OF |
|---|---|
| 1. They must be used within the loop statements. | 1. It can be used within the loop statement &outside of the loop statements also |
| 2. It considers preceding fields also | 2. It doesn't consider |
| 3. It can't be used on more than one field using the SET operator 'OR' | 3. It can be used on the multiple fields using the SET operator 'OR' |
| 4. The work area is impacted by replacing with Asterisk (*******)marks | 4. It doesn't impact the Work Area |

**17) What are the differences between SELECT SINGLE and UPTO 1 ROWS?**

**Ans :-**

| SELECT SINGLE | UPTO 1 ROWS |
|---|---|
| 1. It extracts single record from the data base table | 1. It extracts all the records into a buffer and moves the first record in to an application program |
| 2. It doesn't require END SELECT statement | 2. It requires END SELECT |
| 3.Use this one, in case of all key fields can be considered | 3. Use this one, in case of not possible to use all the key fields |
| 4. It is slower | 4. It is faster |

**18) What are the differences between Normal Reports and ALV Reports?**

| Classical reports | ALV Reports |
|---|---|
| 1. Simple reports using the WRITE statements within the loops. | 1. The system generated Functions start with REUSE* are used to display the data. |
| 2. Performance is not as good as ALV Reports | 2. Performance is good since the data is |

| | |
|---|---|
| since the LOOP statement must be used to display the data. | displayed using the function modules outside of the loops. |
| 3. Not possible to edit the Fields in the Output. | 3. It can be possible to edit the Fields in the Output. |
| 4. It is difficult to display the LOGOS in the output. | 4. It is so EASY to display the LOGOS in the output. |
| 5. It is COMPLEX to Align the Columns. | 5. Column Alignment is done by the system. |

**19). what are Conversion Routines?**

- The conversion routines are the function modules used to convert the data from internal format to an external format & external format to internal format.
- The conversion routines are maintained at the Domain level of a field.

**20) How to make the field editable in ALV Grid?**

**Ans :- By passing thefield name EDIT = 'X' to the Slis_Fieldcat_Alv.**

**21) What are the events associated with ALV Reports?**

**Ans :- There are 17 events in ALV Reports .**

- ✓ Following are some of the most useful events.
  - TOP_OF_PAGE:It is an event which is triggered at the top of each page. This event is used to display the list headings & Logos.
  - PF_STATUS_SET: This event is triggered at the time of attaching our own GUI to the program. This event is used to call the menu status for adding menu items & push buttons.
  - USER_COMMAND:It is an event which is triggered at the time of user clicks on any record of any list as well as clicks on any menu item.This event is used to provide an interactive functionality based on the selected record/push button/ menu item in ALV reports.
  - END_OF_LIST: It is used to display the Footer details in ALV reports.

**22) What are the Function Modules associated with ALV Reports?**

- The standard ALV display Function module 'REUSE_ALV_LIST_DISPLAY' is used to display the populated final internal table data in ALV LIST format.
  - The Field values cannot be edited in case of ALV List format.
  - Logos/Images cannot be displayed in ALV List format.
- The standard ALV display Function module 'REUSE_ALV_GRID_DISPLAY' is used to display the populated final internal table data in ALV GRID format.
  - The Field values can be edited in case of ALV GRID format.
  - Logos/Images can be displayed in ALV GRID format.
- The Blocked ALV's are used to display the data from the multiple internal tables as separate block in the output.
  - The standard ALV display Function modules 'REUSE_ALV_BLOCK_LIST_INIT' is used to initialize the different blocks in the output.

- 'REUSE_ALV_BLOCK_LIST_APPEND' is used to append the internal tables as separate blocks.
- 'REUSE_ALV_BLOCK_LIST_DISPLAY' isused to display the appended blocks in the output.

- The standard ALV display Function module 'REUSE_ALV_HIERSEQ_LIST_DISPLAY' is used to display the Header &corresponding item details in HIERARICHAL manner.

**23) What are the ways to populate Field Catalog and which one do you prefer?**

**Ans :- There are two ways to populate the FILED CATALOG .**

1. **Using ALV Function Module 'REUSE_ALV_FIELDCATALOG_MERGE' // Not Advisable**
2. **Manual Population // Advisable**

**1. Steps to Populate Field Catalogusing the Function Module:**

- **Define Data dictionary structure/Internal table with the required fields.**
- **Call ALV function module 'REUSE_ALV_FIELDCATALOG_MERGE' to populate field catalog from the DDIC structure/Internal table.**
- **Modify the specified field catalog internal table 'GT_FCAT' as per the specification if required.**
- **Pass the filled field catalog internal table to an Export parameter 'IT_FIELDCAT' of the ALV Display function modules.**

**Steps to Insert Border Line for 72 Characters:**

- **Once the program is in change mode.**
- **Click on Utilities -> Settings -> Check the check box "Downwards-compatible Line Length(72).**
- **Click on Yes button.**
- **One red color vertical border line is inserted at 72 characters position.**

**2. Steps to Populate Field Catalog Manually:**

- **Populate Field Catalog Internal table with the required field values.**
- **Pass the populated Field Catalog Internal table to an export parameter of ALV display function module.**

**24) Is it possible to develop the ALV reports without populating or using any FIELD CATALOG?**

**Ans :-Yes, We can develop.**

- **Create DDIC structure with the required fields.**
- **Extract the required data from the data base tables based on the given input.**
- **Populate the Final internal tables.**
- **Use the function module either 'REUSE_ALV_GRID_DISPLAY' or 'REUSE_ALV_LIST_DISPLAY' to display final data by exporting DDIC structure name & the Final internal table.**

**25) What are the different ways to populate events in ALV report?**

**1. Manual Population:**

- **Populate an internal table of the type 'slis_t_event' with the required events & corresponding dynamic sub-routine.**
- **Define the dynamic sub-routine of an event with the required functionality.**

- **Pass the populated events internal table to Export, Import parameters of an ALV display function module 'it_event'.**

**2. By using ALV function module – REUSE_ALV_EVENT_GET:**

- **It is not advisable since it extracts all the events & loop is required.**
- **The dynamic sub-routine for each event can be populated with in the loop of event internal table.**
- **Pass the populate event internal table to an export parameter.**

**3. The call back export parameter can also be used to specify the dynamic sub-routine name of ALV events in ALV display function module.**

**26) Explain the steps to provide headings and logo in ALV reports? (Or)**

**Steps to work with an ALV event TOP_OF_PAGE?**

**Ans :-**

- **Use an ALV event 'TOP_OF_PAGE' with the corresponding dynamic subroutine name into an internal table**
- **Populate the required header data into an internal table of the type (slis_t_list_header) inbetween the definition of dynamic subroutine.**
- **Use the ALV Function module 'REUSE_ALV_COMMENTARY_WRITE' to display the Populated header and Logos.**

**27) How to provide custom Function (Push) Buttonalong with System generated Functions in ALV? (Or)How to add push buttons to ALV?**

**(Or) Steps to work with PF STATUS SET in ALV?**

**Ans :-**

- **Copy the system provided generic functions/push buttons from the standard program 'SAPLSALV'&thestatus:STANDARDintocustom program &menustatus using SE41.**
- **Add additional customer specific 'push buttons' to the copied menu status.**
- **Use an ALV event 'PF_STATUS_SET' & call the copied menu status with in the dynamic sub routine of PF_STATUS_SET.**
- **Call the copied menu status using the ABAP statement 'SET_PF_STATUS'.**

**28) How to handle Interactive Functionality? (Double Click)**

- ✓ **An interactive functionality can be provided with in the dynamic sub-routine of an ALV Event USER_COMMAND.**
- ✓ **Provide the interactive functionality such as to display another ALV drill down report/call another program when the push buttons are selected.**
- ✓ **The function code for the double click in ALV reports is '&IC1' and in Interactive classical report in 'PICK'.**

**29). How to capture selected rows in ALV output (or)**

**What are the steps to get selected rows in the next drill down list in ALV (or)**

**Interactive functionality based on the selected rows?**

**Ans:**

- **Add Single Character field to the Layout.**
- **Pass BOX single character filed name to the layout.**
- **Pass populated Layout to ALV display function modules.**
- **Copy final internal table into temporary internal table.**
- **Delete non-selected entries from temporary internal table.**

**33). What are steps to display each row in different colors in ALV report?**

**Ans:**

- **Add 4 Character field to the final internal table form which the data should be displayed.**
- **Populate color code dynamically.**
- **Pass color code field name to the field 'info_fieldname' of the layout.**

**34). How to display Traffic lights in ALV Report?**

**Ans:**

- **Add 4 character field to the final internal table.**
- **Populate traffic lights based on the status.**
- **Populate Field Catalog with traffic lights field name.**

**30) Explain about some of the useful types with which you have worked in SLIS Type Group.**

**Ans :-**

| Field Name | Description |
|---|---|
| Tab name | Final Internal Name |
| Tab index | Index no. of the selected record in the final internal table . |
| Value | Selected field value |
| Field name | Field name of a selected row |

**31). Explain some useful types in the type group 'SLIS'?**

**Ans:**

I. **SLIS_T_FIELDCAT_ALV: It is used to format the columns in ALV reports such as Populate column headings, Edit the columns, Display hand symbols.**
II. **SLIS_LAYOUT_ALV: It is used to format the entire layout in ALV reports such as display data in different colors. Instead of format statement we use layout.**
III. **SLIS_T__EVENT: It is used to populate an ALV event.**
IV. **SLIS_T__LIST_HEADER: It is used to display the List Headings & Logos in ALV.**
V. **SLIS_T__EXTAB: It is used to populate the function codes of the push buttons to be hidden in ALV report/output.**
VI. **SLIS_T__SELFIELD: It is used to capture the selected row/line value in ALV.**

VII. **SLIS_KEYINFO_ALV: It is used to populate the key field's information to display Header & the corresponding Item data in Hierarchical ALV.**

VIII. **SLIS_T_SORT_INFO: It is used to provide sub totals & grand totals in ALV report.**

## 32)What are the types of Memories, explain about them?

**Ans :- The Memory concept is used to access the data across the internal or External Sessions.**

- ✓ **There are 2 types of memories.**

1. **SAP Memory: - It is used to access the data from one session to another session through Parameter ID or Memory ID.**
   - **The ABAP statements SET PARAMETER ID and GET PARAMETER ID are used to work with the SAP memory.**
   - **SET PARAMETER ID is used to pass the field value from an application program into the specified SAP Memory ID or Parameter ID.**
   - **GET PARAMETER ID is used to get the data from the specified memory ID or Parameter ID into filed of an application program.**
   - **The Parameter ID's are maintained at Data Element level under Further Characteristics.**

**2. ABAP memory: - It is used to access the data across internal sessions. The ABAP statements IMPORT & EXPORT are used to work with the ABAP Memory.**

- **The IMPORT statement is used to import the data from the specified memory ID into data variable.**
- **The EXPORT statement is used to export the data into the specified memory ID from the program.**

## 35) Can we place More than one selection screen Element in one line, If Yes, How?

**Ans :- Yes.**

```
SELECTION-SCREEN BEGIN OF LINE.

* List of PARAMETERS

SELECTION-SCREEN END OF LINE.
```

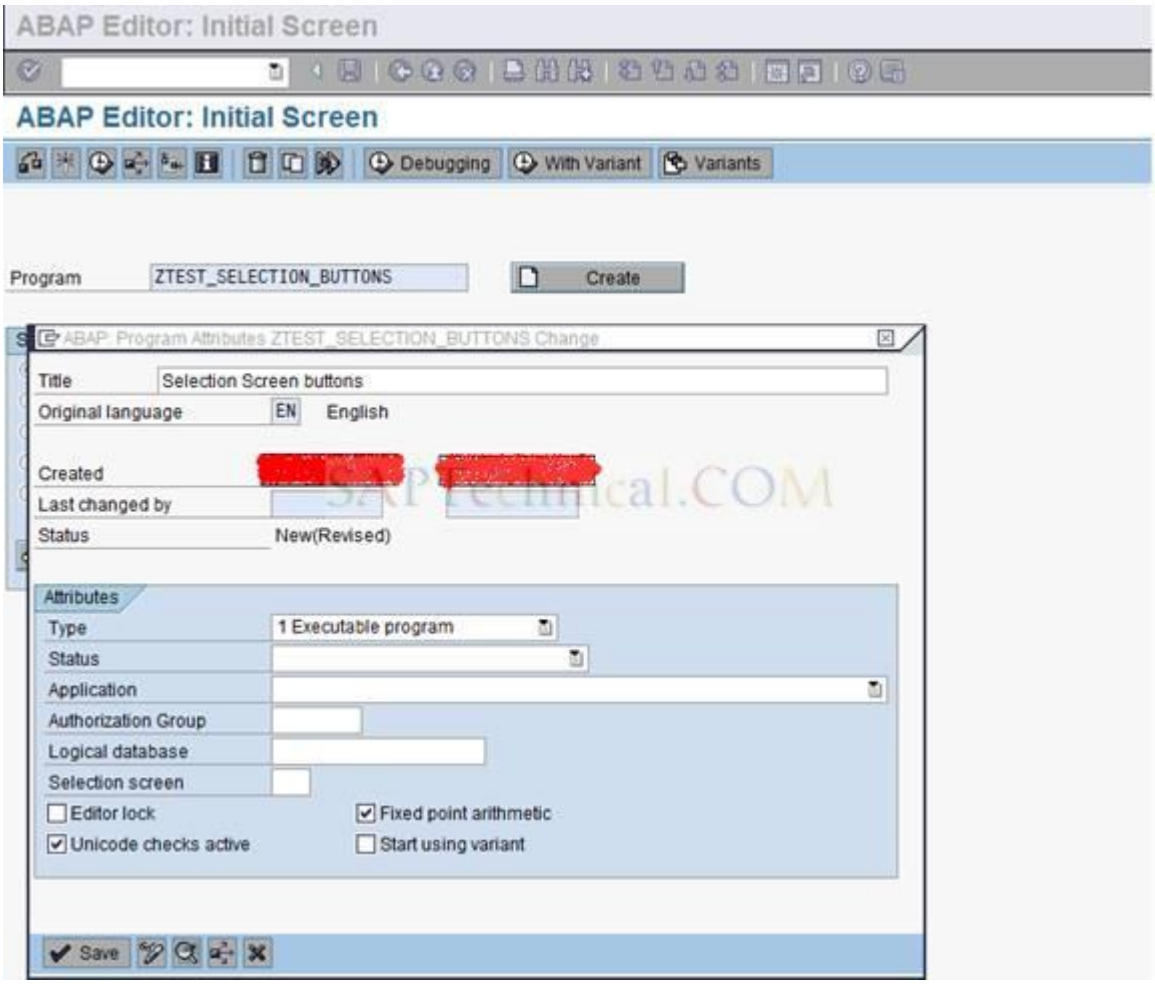
## 36) How to place the PUSH buttons (Function keys) on Selection Screens?

**Ans:-Requirement:**

Three buttons display, change, create to be created along with their respective icons on the selection screen.

**Steps to be followed:**

**Step1: Create program in SE38 with some meaningful description.**




**Step2: Create a selection screen with two or three fields.**

```
12
13    SELECT-OPTIONS : s_plant FOR vbap-werks,    " Plant
14                     s_class FOR klah-clint.    " Class
15
```


**Write the below code in the program after the above selection screen statement.**

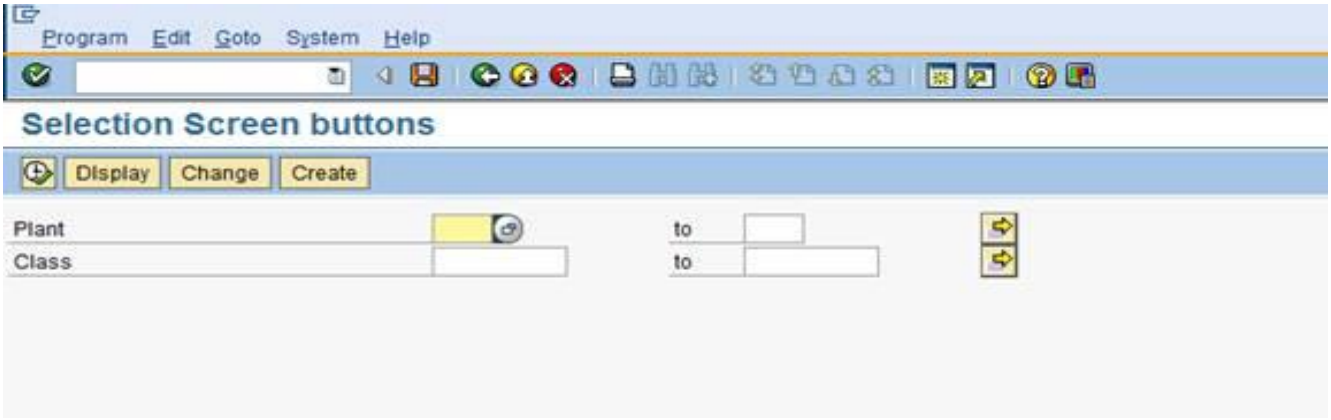
```
17
18    * In our cases we are activating three buttons
19    SELECTION-SCREEN: FUNCTION KEY 1,
20                      FUNCTION KEY 2,
21                      FUNCTION KEY 3.
22
```

**Write the below code in the program in the Initialization event.**

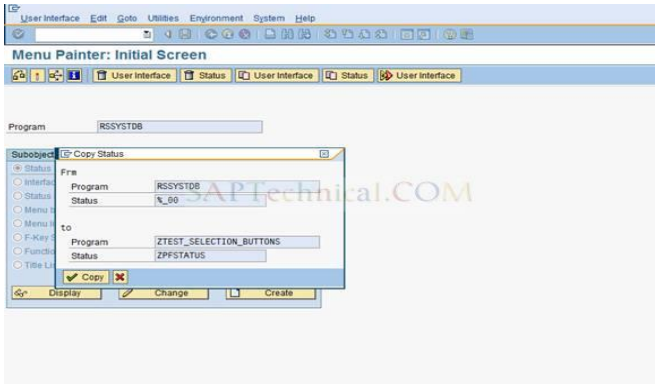
```
21
22
23  INITIALIZATION.
24
25    DATA : functxt TYPE smp_dyntxt.
26
27  * Button for display on selection screen
28    CLEAR functxt.
29    functxt-text = 'Display'.
30    sscrfields-functxt_01 = functxt.
31
32  * Button for change on selection screen
33    CLEAR functxt.
34    functxt-text = 'Change'.
35    sscrfields-functxt_02 = functxt.
36
37  * Button for Create on selection screen
38    CLEAR functxt.
39    functxt-text      = 'Create'.
40    sscrfields-functxt_03 = functxt.
```



**Execute the report and you will see three push buttons on the selection screen as below.**

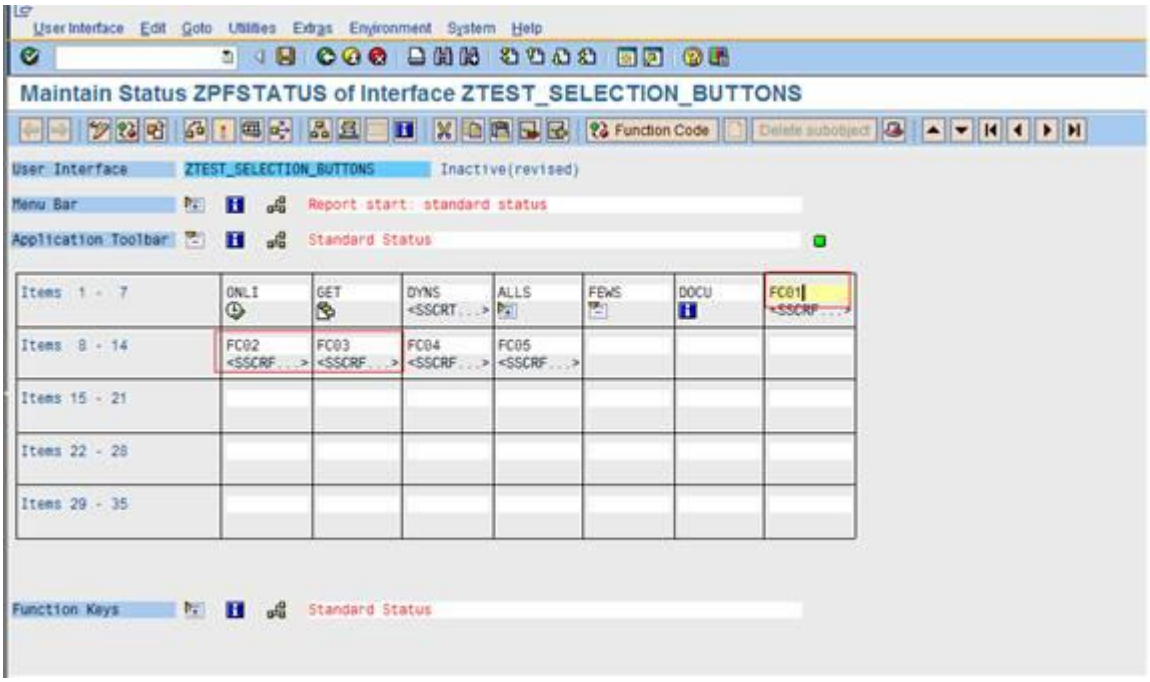



**Now to handle the buttons in the program, first we need to create a custom GUI status copied from the standard GUI as below**






**RSSYSTDB& %_00 refer to the standard GUI status.**

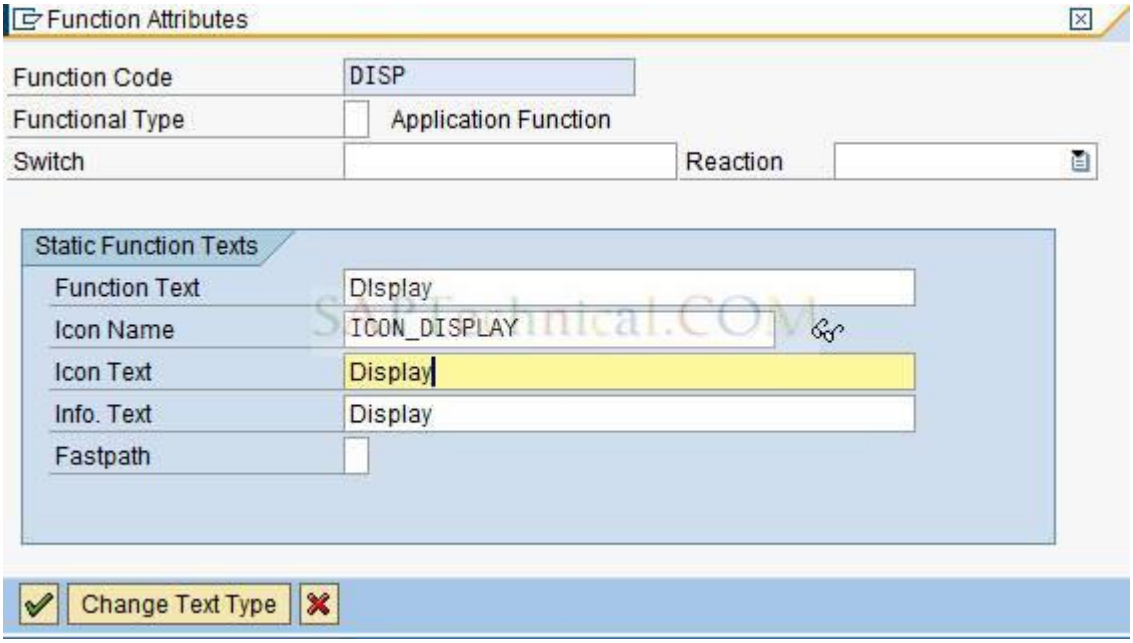



**Now we need to add the function codes for the above three buttons created on the selection screen.**

**FC01, FC02 and FC03 refer to the three buttons which we have activated above.**

**Here we will add the function codes for the three buttons. These function codes will be captured in the system variable SY-UCOMM field and this can be used in the program logic.**

**Function code 'DISP' is used here for the first push button (FC01), similarly 'CHAN' and 'CREA' for second and third push buttons respectively**.




**After entering the function codes for all the push buttons active the pf-status.**

**Now come back to the program and write the below code in the**

**AT SELECTION SCREEN OUTPUT event.**

```
44
45   AT SELECTION-SCREEN OUTPUT.
46
47     SET PF-STATUS 'ZPFSTATUS'.
48
```

**Now execute the program and you can find the below selection screen.**


**<u>Summary</u>: In this way we create buttons along with the icons on the selection screen.**

# Module Pool Programming/Screen Painter

## Module Pool Programming/Screen Painter

- This concept is used to maintain different types of screens & corresponding flow logic.
- This concept contains mainly 2 parts.
  - Module pool program
  - Screen painter

**1) What is the Screen Painter (Layout) and explain different components in screen painter?**

**Ans: Screen painter: -**

- It is a tool used to maintain different types of screens.
- The T-code SE51 is used to work with screen painter.

a) **Normal screen:**
   - The normal/main screens which can be called through transaction codes/using ABAP statement 'CALL SCREEN'.

b) **Sub screen:**
   - The sub screens are part of main screens (normal)
   - The sub screen can't be executed independently
   - The flow logic ABAP statement 'CALL SUBSCREEN' is uued to call the sub screen areas in the normal screens
   - The push buttons/menu status is not possible on the sub screen
   - It is not possible to call sub screen with in the sub screen

c) **Modal dialog screen:**
   - It is used to populate/display the data on pop dialog box with some generic functions ( To display icons at bottom)

d) **Selection screen:**
   - The selection screen is reserved for Reports/Executable programs.
   - The screen number 1000 is reserved for selection screen.

✓ **Different Components/Elements :-**

1. **Attributes: It contains the technical details of the screen such are type of the screen, next screen number, package etc.**

2. **Element List: It contains the screen elements, attributes & properties of the screen elements in the normal screen & sub screen & attributes such are general attributes, texts or input output templates, special attributes, display attributes, modification groups.**
   - Make an input list as drop down list by selecting, option under the tab input/output.
   - This component is used to make the screen elements visible, invisible, change the dimensions etc.
   - The modification groups which can be used to format the screen dynamically are specified by using modification groups.

3. **Flow Logic:**
   - This component is used to provide the link between the module pool program & screens.
   - It can also be used to control the data flow between the screens

- The screen related ABAP statements such are modules, MODULE, CALL SUBSCREEN, CHAIN etc. statements can be used in flow logic.

3. Layout: It is used to design the screens with different screen elements
   a. Text fields: To define the text fields for screen field labels on the screen.
   b. Input/output field: It is used to define input/output fields on the screen.
   c. Checkbox: To define the check box on the screen.
   d. Push button: It is used to define the push button on the screen.
   e. Tab strip control: It is used to define the tab strip control to work with the different subareas on the screen.
   f. Tab strip (with wizard): it is used to define the tab strip control using the wizard.
   g. Sub screen area: It is used to define the sub screen areas on the main scrren/on the tab strip control.
   h. Table control: It is used to define the table controls to maintain multiple line items or records on the screen.
   i. Table control with wizard: It is used to define the table controls using wizards.
   j. Custom control: It is used to define the custom controls in the screens.
   k. Status icon: It is used to define the status icons on the screen.
   l. Box: it is used to define the boxes around screen elements in the screens.

**2) What are the Screen painter attributes?**

Ans: They are used to maintain different attributes/settings of a screen elements such are function code, function type, drop down list etc.

a) Drop down: There are 2 types of drop down options
   I. List Box: To list out only the values.
   II. List box with key: To list out the values with corresponding keys when the drop down icon is pressed
- There must be foreign key relationship between value & text fields of a text table.

b) Groups: They are used to specify the radio button group name or modification group name.
c) FCT code: Function code of a push buttons or menu items.
d) Dictionary Attributes:
- It is used to maintain the attributes of the data dictionary objects such are table or structure fields.
- Some of the data dictionary objects are: date format, search help, conversion exit etc.

e) Program Attributes:
- It is used to maintain the different attributes of input or output fields.
  I. Required: To make the screen input field as mandatory
  II. Possible: It is default icon to accept the input values at run time.
  III. Not possible: In case of this option, input is not possible but the field can be used as output field.
  IV. Recommended: To make the input field as optional. The symbol question mark is appeared in the input or output field.
- Check the check box output only to restrict the input data.
  V. Display Attributes:To maintain different attributes such are invisible, output styles, 2D display etc.

**3) What is the difference between Flow Logic and Module Pool program?**

- **Flow Logic is used to define the Flow Logic of the screen elements. The modules are used to communicate with the Module Pool Program.**
- **Module Pool Program is used to define the modules of the Flow Logic in screens. The T-code SE38/SE80 is used to develop & create the M.P.P.**
- **It can have 4 Include programs such are *TOP, *F01, *I01, *O01.**

**4) What are the systems generated Include programs in Module Pool program?**

**Ans :- There are 4 Include programs in module pool programs.**

a) **TOP Include: It is used to declare global data declarations which can be accessed across the screens.**

b) **F01 Include: It is used to define the FORMS (Subroutines) which can be used in module definitions.**

c) **O01 Include: It is used to define the PBO (Process Before Output) modules.**

d) **I01 Include: It is used to define the PAI (Process After Input) modules.**

- **The module pool name 7 scrren program name must be similar.**

**5) What are the events associated with Screen Painter?**

**Ans :- The below events are used in the MPP.**

**1. PROCESS BEFORE OUTPUT (PBO):**

- **This event is triggered before the screen is displayed.**
- **It can also be triggered while performing any action on the screen.**
- **This is used to provide the default values to the screen fields.**
- **It can also be used to format the screen elements dynamically.**

**2. PROCESS AFTER INPUT (PAI):**

- **This event is triggered after the Input values are provided to the screen.**
- **This event is used to validate the input data on the screens.**
- **It can also be used to process the screen data based on the performed action (pressed push button menu item etc)**

**3. PROCESS ON HELP-REQUEST (POH):**

- **This event is triggered when the function key F1 or help icon (?) is pressed.**
- **This event is used to provide the help documentation on the screen fields.**
- **The T-code: SE61/SO72 is used to maintain help documentation.**
- **The standard function module 'HELP_OBJECT_SHOW' is used to display the maintain help documentation.**

**4. PROCESS ON VALUE-REQUEST (POV):**

- This event is triggered in case of in case of function key F4 or search help icon is pressed.
- This event is used to populate & display the custom search help on screen input fields.
- The standard function module "F4IF_INT_TABLE_VALUE_REQUEST" is used to display the populated internal table data as search help.

**6) What are the steps to provide Search Help?**

I. Use an Event PROCESS ON VALUE-REQUEST.
II. Specify or define the Module on the required Screen Field.
III. Use the Standard Fn. Module F4IF_INT_TABLE_VALUE_REQUEST to display the Populated data in the search help Window.

**7) What are the steps to provide Help Documentation?**

a) Use the Flow logic event 'PROCESS ON HELP-REQUEST'.
b) Specify the module name on the particular screen field.
c) Use the standard function module HELP_OBJECT_SHOW (with technical description) or DSYS_SHOW_FOR_F1HELP (without technical description) to display the maintained Help Documentation.

**8) What are the different ways to execute the Screens?**

Ans :- There are 2 ways to execute the screens.

1. Using the T code: The screens can be executed through the T codes to which the Module pool program & the screen no is assigned.
2. Using an Executable program: The screens can be called & defined using the Statement 'CALL SCREEN'. CALL SCREEN is used to call the specified screen number.

**9) How to validate/check the fields (screen) in Module Pool program?**

Ans: - a) Use an event Process After Input to validate the Input//screen field values.
b) Once the Flow logic is in Change mode, define the module with the field statement between CHAIN…ENDCHAIN screen statement.
C) Fetch the data from the database & give the proper message if data doesn't exist.

- The statement CHAIN….ENDCHAIN is sued to make all the screen input fields enables to enter input data.
- The screen validations without CHAIN….ENDCHAIN statements make the screen input fields grayed out (display mode).

**10) How to make screen field as mandatory in Screen painter?**

Ans :-In screen painter doubleclick the screen field
Go to ATTRIBUTES->PROGRAM
There select input as REQUIRED
Required: To make the screen input fields as mandatory.

**11) How to make the Screen fields invisible based on certain conditions dynamically and which**

**Event should be used?**

**Ans :- The event AT SELECTION SCREEN OUTPUT is used to format the selection screen dynamically by setting a loop to screen table (field Active can be used to Hide/Appear the screens).**

**12) Module screen flow statements?**

**Ans: This statement is used in flow logic to call the define process module pool programs.**

**Syntax: [FIELD<screen-field>]MODULE<module-name>.**

**Additions:**

I. **….AT EXIT-COMMAND: This command is used to trigger the dialogue module in case of push button or menu item with the function type 'E' exit is pressed.**
II. **…AT CURSOR-SELECTION: This addition is used to trigger the corresponding module when the cursor is placed on the input field of a screen. It can also be used to trigger corresponding module in case of a push button or menu item with f-type 'S' (system) or CS (cursor selection) is pressed.**
III. **CHAIN…ENDCHAIN:This statement is used in flow logic to define the process chains. It can be used to make all the screen input fields enables to enter input data.**

**Useful additions to CHAIN…ENDCHAIN statement:**

I. **MODULE ON CHAIN-REQUEST: The module with ON CHAIN-REQUEST is used to trigger the corresponding module on the specified screen fields (in the sequence).**
II. **MODULE ON CHAIN INPUT: The module ON CHAIN INPUT is used to check the individual conditions of the screen fields in the sequence.**

**13) What are the steps to work with Tab Strip Control?**

**Ans :-**

- **It is a set of pages & used to place different sub screens of application on single screen.**
- **Each tab page contains title & its page area.**
- **Create tab strip control with title on Main screen.**
- **Assign sub screen area to the corresponding tab.**
- **Make the necessary code changes in flow logic.**
- **Write processing logic in Module Pool Program.**

**14) What are the steps to work with Table Control?**

**Ans :-**

- **It is used to maintain multiple entries in tabular format. Multiple rows are processed using Loop with control.**
- **Define or declare tab strip control in Module Pool program.**
- **Define or create Table control on screen by filling all necessary fields.**
- **Make necessary code changes in flow logic & write process logic in MPP.**

# SAP SCRIPTS

# SAP SCRIPTS

## 1) What are the Components of SAP Script (Form Painter/Layout set)?

**Ans: SAP Script formcontains 2 parts.**

**1. Form painter (Layout Set)  2. Print Program**

1. **Header: It is used to maintain the Administrative & basic settings data.**
   a. **Administrative data:**
   - **It contains the technical details such are Package name, Changed by, Created by etc..**

   b. **Basic data:**
   - **It is used to maintain the default settings which can be applied across the forms.**
   - **TheBasic settings are page format, orientation, font size etc.**
   - **The first page & the default paragraph names must be specified once they are created.**

2. **Pages:**
   - **They are the rectangular format of the documents.**
   - **This element is used to maintain the next page, print mode, page counter mode etc.**

3. **Windows:**
   - **They are the small components on the pages.**
   - **We can place the same window in any no of pages.**
   - **The windows are not positioned on the pages unless they are assigned to the pages.**

4. **Page window:**
   - **They are used to position the window on the pages.**
   - **Different position attributes Left margin, upper margin, Window width, Height are maintained. It is used to place the window on the page with co-ordinates.**

5. **Paragraph format:**
   - **They are used to print/format the entire paragraph with a required format & style.**
   a. **Standard Attributes: It is used to maintain the standard attributes such are alignment, line spacing, left margin, right margin etc.**
   b. **Font Attributes: They are used to maintain the font setting such are font family, font size, bold, italic etc.**

6. **Character format:**
   - **They are used to print/format the particular text/string with the required font & style.**
   - **They can also be used to select the Bar codes.**
   a. **Standard Attributes:It isused to select barcode, super script, sub scripts etc..**
   b. **Font Attributes:They are used to maintain the font type,font size, bold, italic etc.**

2. **Print Program or Driver Program:**

- The Driver program is used to extract the data from the database tables or views, process the extracted data & pass the processed data to SAP Script form to display/print on the document.

**2) What are the difference types of Windows?**

**Ans: Windows: - They are used to define the different types of windows.**

- **Following are the different types of windows.**
  - **Main Window: It is the default widow. It is used to display the continuous text on the pages. Without main window we can't design SAP scripts. The system generates the pages dynamically for main window. Up to 99 main windows can be created in a form. The window type is 'MAIN'.**
  - **Constant Window: This window contains the static content which can be formatted only once to print/display the same contents on all the pages.**
  - **Variable Window: It is used to print the data based on the window size. It can be formatted for each page to print/display the different contents. Without a variable window also we can design the SAP script.**
  - **Graphic Window: It is used to print/display the graphics/logos on the page.**

**3) What are the different types of Symbols in Scripts?**

**Ans: - The Symbols are the constants which can be inserted n the page windows to display/print the corresponding contents/data. Symbols are of 3 Types.**

**1. System Symbols: They are used to display the system data such as date, time etc …**

**Ex: &DATE&, &DAY&, &MONTH&**

| Symbol | Description |
|---|---|
| &DATE& | System Date |
| &DAY& | Day of the System |
| &MONTH& | Month of the System |
| &YEAR& | Year of the system |
| &TIME& | System Time |
| &HOURS& | System Hours (HH) |
| &ULINE& | Horizontal Line |

- **The system provided standard structure SAPSCRIPT contains some of the SAP Script system fields.**

**Note: Number of pages like "1 of 10" can be displayed using the system symbols PAGE & FORM PAGE as below.**

**Ex:&PAGE&of&SAPSCRIPT-FORMPAGES&**

**Current Page number, Total no of pages**

2. Standard Symbols: They are the predefined symbols. The pooled table TTDTG is used to maintain the standard symbols.

3. Application Symbols:These are used to display/print the processed data from the print program or database. Ex: &WA_T001-BUKRS&

**4) What are the different TAG columns in SAP Script form?**

| Ans: Tag | Meaning |
|---|---|
| * | Default paragraph format |
| = | Extended Line |
| /: | Command Line |
| /* | Comment Line |
| /E | Text Element -> Display multiple entries of an internal table |
| P | Paragraph format |
| /= | Line feed & Extended line |

**5)What are the Function Modules associated with Scripts?**

Ans:-The below Function Modules are used in the Scripts.

1. OPEN_FORM: (*)
   - This function module is used to provide the link between the print program & SAP script form.
   - The global data variables of the print program can be accessed in SAP Script form after this function module is called to open a form/used.

2. START_FORM:
   - This function module is used to specify the starting page from where the document to be printed or displayed.

3. WRITE_FORM:(*)
   - This function module is used to pass the multiple entries of an internal table into the SAP Script form to print/display the multiple entries or lineitems through the Text Elements.

4. END_FORM:
   - It is an optional function module which is used to End the started Form.

5.CLOSE_FORM: (*)
   - This function module is used to close the Opened Form & send the Spool Request/Print to the device.

- It can also be used to get the SAP Script form data into OTF(Other Text) format.

**Note**: The standard function module 'CONVERT_OTF' or 'CONVERT_OTF_2_PDF' can be used to convert the SAP Script OTF Data into PDF format.

- The standard function module SO_NEW_DOCUMENT_ATT_SEND_API1 is used to send an Email with the required attachments in different formats.

**6. CONTROL_FORM**:

- This function module is used to control the form by creating the pages or windows dynamically.

**Note**: An icon Transport (Lorry icon) or in the menu path, Graphic -> Transport is used to generate transport request to release LOGO's form one system to another system.

**6) What are the different ways to debug the Script?**

Ans: - There are two ways to debug the Script Form.

- The path SE71 --->Utilities --->Activate debugger.
- By executing Standard Program/ Report 'RSTXDBUG'.

**7) Differentiate between Page1 & Page2 format in sapscript?**

Ans: Page1 format: In this all pages have the same format.

Page2 format: In this there is variation in page format i.e. first page has different format than second page.

**8) Some of the useful Control commands/Statements in SAP scripts?**

Ans:

1. **/: Address…Endaddress**: It is used to display the address details of the given address number in the standard format.

Syntax: /: ADDRESS

/: ADDRESS number &GS_KNA1-ADRNR&.

/: ENDADDRESS

2. **/: Set Date mask**: This command is used to print/format the system Date in the required format.

Syntax: SET DATE MASK = '<DD.MMMM.YYYY>'.

* Date: &DATE&

3. **/: Set Time Mask**: This command is used to print/format the system Time in the required format.

Syntax: /: SET TIME MASK = ' SS:MM:HH '.

Note: Pass blank spaces to the formatted date & time to format them into default format.

- There is no loop statements in SAP Script forms but the text elements can be used to communication with the function module WRITE_FORM used within the loop of the print program to display the multiple entries.

**Text Element:**

- Text Element is the name give to the block of statements in the page window. If we pass the text element name to the WRITE_FORM function module then the WRITE_FORM function module transfers the data form driver program to all the statements which are available under Text Element.

- **Whenever we are working with main window, then we must provide Text Element name, otherwise the 1st information printed twice.**

4. **Protect…End protect: It is used to print the continuous text without any page breaks in SAP Script forms.1st the system checks each & every page which page is having the enough place to print the continuous text. If no page is having enough space it simply break the text & printing in different pages.**

   **Syntax: /: PROTECT**

   **/E ELE**

   **/: ENDPROTECT**

5. **Define: This command is used to define/declare the symbols in SAP Script.**

   **Syntax: /: Define &GV_SYMBOL&**

6. **Box: This command is used to draw the Boxes/Frames on the page windows.**

   **Syntax: /: BOX XPOS <**> cm/mm YPOS <yy> cm/mm**

   **/: WIDTH <ww> cm/mm HEIGHT <yy> cm/mm**

   **/: FRAME <FF> TW INTENSITY <yy> TW**

   **Note: An addition INTENSITY is used to gray out the boxes.**

   - **The command position & size can also be used to define the box position & sizes.**

8. **Top…End Top: This command is used to display/print the contents at the Top portion/Header of the specified window.**

   **Syntax: /: TOP**

   **/* Display/Print at Top of page window.**

   **/: ENDTOP**

9. **Bottom…End bottom: This command is used to display/print the contents at the Bottom/Footer portion of specified window.**

   **Syntax: /: BOTTOM**

   **/* Display/Print at Bottom of page window.**

   **/: ENDBOTTOM**

10. **New-Page: This command is used to generate the next pages.**

    **Syntax: /: NEW-PAGE.**

11. **New-Window: This command is used to generate the new windows.**

    **Syntax: /: NEW-WINDOW.**

**10) How to add, additional functionality to script without modifying the print (driver) program?**

**Ans: - The Form Routines (PERFORM .....END PERFORM) are used to add the additional functionality to the SAP Script form without making any changes in the print program.**

**11) What are client dependant objects in abap/sap?**

**Ans: SAP Script layout, text element, and some DDIC objects.**

**12) Can we call another Form from same print program in sap script?**

**Ans: Yes, we can call other forms in the same print program/driver program.**

**13) How to copy the Standard Script and make the changes?**

Ans :- The path SE71--->Utilities---> Copy From Client can be used and make the necessary changes

**14) How to convert the language from DE to our required language?**

Ans: In SE71, Utilities ->Convert Original Language, Provide the To Language & press Enter

**15) How to transport text elements in SAP ABAP?**

Ans: If it is first time, they will automatically transport along with the program, if you change them next time, it will ask for TR, you can move that TR

**16) What are the Standard Texts?**

Ans :- The Standard texts are used to maintain the Long text using the T-Code 'SO10'.

- The command INCLUDE….OBJECT is used to insert the standard text on page window.
  Syntax: /: INCLUDE <ZSTP_TEXT_NAME> OBJECT <obj-name>
  ID <id_name> LANGAUGE <EN/..>
- The standard function module 'READ_TEXT' can also be used in the programs to read standard text & application text.
- By Default the System doesn't generate any transport request for standard texts.
- The standard Report 'RSTXTRAN' is used to generate the Transport Request number for Standard texts.

**17) Tell some of the Standard Scripts and Print programs?**

Ans :-Some of the useful Standard Script Forms and Print programs :-

| Form name | Print program | Form routine | Application |
|---|---|---|---|
| MEDRUCK | SAP M & SP/SAPFM06P | Entry_new | Purchase order only |
| RVOROADDRS | RVADOR01 | ENTRY | Sales order |
| RVDELNOTE | RVADDN01 | ENTRY | Shipping |
| RVADIN01 | RVINVOICE01 | ENTRY | Billing |

**18) How to execute the program in back ground?**

Ans: Execute SE38, provide the program name & Execute, provide the Input
In the Menu bar, click on program -> Execute in Back ground
Provide the output device LP01 & click on enter
Click on Immediate & save.

**19) How to copy the SAP script from 800 client to 810 client?**

Ans: In the Target client 810, Execute SE38 & Provide the program name: RSTXFCPY & Execute
Provide the Form name, In source client 800, Provide the Target Form name & Execute.

- All the standard script forms are available in the client 000.

**20) Explain about some of the Symbol Formatting Options in Scripts?**

| SYMBOL FORMAT | DESCRIPTION |
|---|---|
| &SYMBOL& | Displays/Print the contents of symbols without any format |

| | |
|---|---|
| &SYMBOL(C)& | To compress the content without spaces |
| &SYMBOL(Z)& | To remove the unwanted preceding zeros |
| &SYMBOL(R)& | Displays the contents by justifying the Right hand side |
| &SYMBOL(K)& | To avoid Conversion Routine |
| &SYMBOL(<)& | To display the symbols sign at Left hand side |
| &SYMBOL(S)& | Display/Print the sign of the numbers |
| &SYMBOL(N)& | Display/Print 'N' no of characters |
| &SYMBOL+N(M)& | Display/Print 'M' no of characters form the Nth position |
| &SYMBOL(.D)& | Display/Print 'D' no of Decimals |

**21) How to format the date, time and country in Scripts?**

**Ans :- The commands SET DATE MASK, SET TIME MASK , SET COUNTRY can be used to format the date ,time and country.**

**22) How do we display the Application Texts?**

**Ans :- Using the Function Module 'READ_TEXT' .**

**23) How to split the line into the next line in Scripts?**

**Ans :-Tthe Function Module ' RKD_WORD_WRAP' is used to split the text into multiple lines..**

**24) What are the fields in ITCSY Structure?**

**Ans :-**

| Field | Description |
|---|---|
| NAME | Name of the programSymbol (without &) |
| VALUE | Text Symbol Value |

**The standard structure is use as a reference Import & Export the symbol with the form routines in SAP Script form.**

**25) How do we create/maintain different styles in Scripts?**

**Ans :- The T-Code SE72 is used to maintain the SAP Script Form Styles such are Paragraph Formats, Character formats. Barcodes are maintained in the Character Format.**

**26) What are the ways to upload the Logos and display?**

**Ans: - The T-Code SE78 (or) the standard report 'RSTXLDMC' is used to upload the LOGOS/IMAGES**

- **Whenever we are working with .BMP image then we must convert .BMP to graphics by using SE78 T-code.**
- **Whenever we are working with .TIFF image then we must convert TIFF to text image by using RSTXLDMC standard program.**

**27) How can you pass the data to the Script from the Print program without using WRITE_FORM?**

**Ans: - After Accessing the Fun. Module 'OPEN_FORM' we can declare the Global data.**

**28) How to develop the SAP Scripts in Diff. languages?**

**Ans: - In SE63, Click Translation --->Long Texts --->Sap scripts---> Forms.**

**29) How to convert sap script spools request to PDF?**

**Ans: - RSTXPDFT4.**

**30) How to convert/migrate sap scripts to Smart Forms?**

**Ans: - Execute SMARTFORMS & Enter the New Smart Form name**

**The path SMARTFORMS --->Utilities --->Migration ---> Import SAP script Form**

**Enter the Existing SAP script Form name & click on Yes button.**

**31) How to set the output device LP01 by default in our own pc?**

**Ans:Execute SU01 & provide the user name as: SAPUSER, click on change mode**

**Click on Default tab & provide the output device LP01**

**Select the required decimal notation (1,234,576.89) & date format & click on save.**

**32) How to identify the driver program based on the form name?**

**Ans: Execute SE11, open the Table TTXFP & Click on Display & Click on Contents**

**Provide the Form name & Execute, Identify the Driver program.**

**33) What are the Transaction Codes associated with Scripts?**

| T-Code | Description |
|---|---|
| NACE | To maintain Output types |
| SE71 | Form painter |
| SE72 | Form Style |
| SE73 | To Maintain different types of forms & Barcodes |
| SE78 | Graphics or LOGO |
| SE63 | Translations to translate the text symbols & text elements in different languages |
| SO10 | Standard texts |

- **Bar codes can be selected under 'Standard Attributes tab.**
- **The T-code SLXT is used to generate Transport Request for Translations**

**34) Standard reports associated with SAP scripts?**

**Ans: RSTXDBUG -> Activate/Deactivate form debugger**

**RSTXLDMC -> To upload Graphics/Logo**

**RSTXSCRP -> Download/Upload the Forms, Styles & Standard Text**

**RSTXPDFT4 -> To convert the spool job of the SAP script into PDF format**

**RSTXFCPY -> Copy the SAP script from one client to other**

**RSTXTRAN -> To create Transport Request for standard text**

**Q) What are the tables associated with Scripts?**

**Ans :- TNAPR**

**Q) How to generate Transport Request for Translations?**

**Ans:**

- **Execute the T-code SLXT**
- **Select the required target language (DE)**

- **Enter the required text element/text symbol name (%TEXT1) in description**
- **Select a radio button workbench request**
- **Click on execute**



# SMARTFORMS

## SMARTFORMS

**1) What are the major differences between Scripts and Smart Forms?**

**Ans :-**

| SAP Script | SMART FORMS |
|---|---|
| 1. They are Client dependent. | 1. They are Client Independent. |
| 2. The back Ground Images are not possible. | 2. Back Ground Images are possible |
| 3. It is not possible to create SAP Script without Main Window. | 3. It Can be created without Main Window also. |
| 4. Multiple page Formats are not possible. | 4. Multiple page Formats are possible. |
| 5. It cannot be executed without Print/Driver Program. | 5. It can beexecuted without Driver Programby executing the system generated Function Modules |
| 6. Color fonts cannot be displayed. | 6. Color fonts can be maintained. |
| 7. The data extract logic (ABAP CODE) can't be written. | 7. The data extract logic (ABAP CODE) can be written under INITIALIZATION and program lines node. |
| 8. The Tables & Templates are not possible but the Command box is used to draw the boxes. | 8. The Tables & Templates can be drawn to display static or dynamic data. |
| 9. By using RSTXDBUG standard program we can debug the SAP script. | 9. By using BREAK-POINT keyword we can debug the smart form. |

**2) Why the Scripts are Client dependent and Smart Forms are Client independent?**

**Ans :-**

- **SAP scripts are client dependent that means, if we design the SAP script in one client that is not reflected to other clients in the same server.**

**The Scripts are client dependent because the contents or data displayed using the text elements. The text elements are client dependent hence the scripts are Client dependent.**

- **SMART FORMS are client dependent that means if we design the SAP script in one client that is automatically reflected to all other clients in the same server.**

**The Smart forms are client independent because one UNIQUE FUNCTION MODULE is generated by the system while activating the smart forms. The generated function module is client independent hence the smart forms are client independent.**

**3) What are the Transaction Codes associated with Smart Forms?**

**Ans :- 1. SMARTFORMS - To maintain SMARTFORMS.**

**2. SMARTSTYLES - To maintain the styles for smart forms.**

**3. SMARTFORM_TRACE - To debug the SMARTFORMS and activate the databasetracer.**

- **Tex t module is used to maintain the long texts.**

**4) How to display Address window dynamically?**

**Ans :- The Path %PAGE ---> CREATE ---> ADDRESS ---> Type DETERMINE DYNAMICALLY**

**5) How to print Logos in Smart Forms?**

**Ans: The Path %PAGE ---> CREATE ---> GRAPHIC --->LOGO**

**6) what are the components of Smart Form ?**

**Ans: 1.Smart Form Layout**

**2. Function Module**

**3. Print Program**

**7) What are the main components in Smart Form?**

**Ans: The main components are, 1. Global Settings 2. Pages & Windows**

**1. Global Settings: It is used to maintain the global settings, definitions which can be applied across the mart forms. It is collection of Form Attributes, Form Interface & Global Definition**

**I. Form Attributes: These are used to maintain the smart form attributes & output such are page format, style name etc. The default style for smart form is 'SYSTEM'.**

**II. Form Interface: These are used to declare the variables, work areas & internal tables which are needed to transfer the data from print program to layout. It is the system generated function module with all the parameters except changing parameters.**

**III. Global Definitions: It is used to maintain the global definitions which can be used across the different Global definitions.**

**a) Global Data: It is used to maintain the global data definitions such are variables, work areas, Internal Tables etc.**

**b) Types: It is used to define the types which can be referred to declare internal tables & work areas.**

**c) Field Symbols: It is used to define the field symbols.**

**d) Initialization: It is processed before the smart form pages & windows are processed. It is used to extract all the required data from the database table & process the extracted data. Import & Export parameters must be specified before they are used.**

e) **Form routines**: It is used to define the subroutines which can be called across the smart form. It is used to define the currency/quantity fields as like the DDIC tables/structures.

f) **Currency/Quantity Field**: It is used to define the Currency/Quantity Fields as like the DDIC tables/structures.

2. **Pages & Windows**: It is used to design the different page formats & windows. Following are the different nodes in smart forms.

a) **Windows**: They are positioned on the pages. Following are different type of windows.

- **Main Window**: It is used to display the continuous text on the pages. This window is the default widow. Without main window also we can design Smart forms.
- **Secondary window**: It is used to print the different contents on each page (similar to variable window in SAP Scripts).
- **Copies window**: It is used to the print the same document with multiple copies with different headings.
- **Final window**: This is used to print the total amount after all the lines or line items are printed.

b) **Graphics**: It is used to create a graphic node (window) to display the graphics/logos.

c) **Address**: It is used to define the address node/window to display/print the organization address/personal address/work place add/ress.

- ***The address can be determined dynamically also using an option Determine Dynamically.***

d) **Text**: It is used to display/print the contents of the different symbols/fields.

e) **Table**: A table is used to print the multiple entries/dynamic data of an internal table. The Table node can be used as template to display static record by un-checking the checkbox internal table. Line types can be created under Table node.

f) **Template**: A Template is used to display the static data (single record). This node does not have the separate header, main area & footer whereas, the table node contains the different sessions such are header, main area & footer. The template node can be used as table by defining it under loop node to display the dynamic data as like the table node.

g) **Flow logic**: The flow logic nodes are used to control the nodes & define the program lines in smart forms. Following are the different flow logic nodes.

I. **Loop**: It is used to process the corresponding nodes based on the number of entries of an internal table.

II. **Program lines**: It is used to write ABAP source code. Input & Output parameters must be filled with the data variables before they are used.

III. **Alternative**: It is used to control the corresponding nodes based on the specified condition. It is used to process the alternativeeither True/False process blocksbased on the SET conditions.

IV. **Command**: This node is used to go to a new page based on the specified conditions. This is used to break the page based on the condition.

## 8) What are the ways to debug the Smart Form?

**Ans :-**

**1St Way: - By setting the Session Break point in the Generated Function module.**

- **The SMARTFORMS can be debug by setting the Session Break points at the required statement line in the system Generated Function module.**
- **The system generated internal table %TEXT contains the following fields which are filled with the text node contents at Run time to debug text elements.**

| Field name | Description |
|---|---|
| TDFORMAT | Tag column of the text elements |
| TDLINE | Contents of the text element/text node |

**2nd Way :- By setting the Static Break points in the SMARTFORMS.**

- **The SMARTFORMS can be debug by setting the Static Break points in the required program lines /initialization node under global definitions of the SMARTFORM. The statement BREAK followed by user name can be used**

## 9) What are the Smart Form events?

**Ans :- 1. Only On First Page: To Print/Display the data on the 1st page only.**

**2. Not on First Page: The corresponding contents are not to Print/Display on the 1st page.**

**3. Only after end of Main Window: To Print/Display the corresponding contents after the MAIN window is processed.**

**4. Only Before end of Main Window: To Print/Display the corresponding contents of the node or text node before theMAIN window is ended.**

**5. Only On page: To print/Display the data on the specified page.**

## 10) What are the differences between Template and Table?

**Ans :-**

- **A Table is used to print the multiple entries/dynamic data of an internal table. The Table node can be used as template to display static record by un-checking the checkbox internal table. Line types can be created under Table node.**
- **A Template is used to display the static data (single record). This node does not have the separate header, main area & footer whereas, the table node contains the different sessions such are header, main area & footer. The template node can be used as table by defining it under loop node to display the dynamic data as like the table node.**

## 11) How to find the system generated function module?

I. **Execute the T-code 'SMART FORMS'.**
II. **Enter the required smart form & click on Display button.**

III. Click on Environment menu item.
IV. Click on function module name.

- The system generated function module name for smart forms could be different from system to system [R3D -> R3Q -> R3P].
- Hence capture the system generated function module name dynamically through an Export parameter 'FM_NAME' = lv_name to the function module 'SSF_FUNCTION_MODULE_NAME'.

**12) How to find the system generated function module of the smart form?**

Ans:Execute T-code …> SMARTFORMS, Enter the Form name & click in Display button

Click on Environment …> Function module name.

**13) Can you move a Smartform from one SAP system to another without using transports?**

Ans: Yes, this can be achieved using the Upload/Download feature for Smartforms.

One can download the Smartform from one system and save it as an XML file.

Once that is done, the XML file can be used to upload the Smartform in another system.

**14) How to upload the smart form?**

I. Execute the T-code SMART FORMS.
II. Click on utilities menu item & click on upload form.
III. Enter new form name to be created by uploading the existing downloaded smart form.
IV. Click on yes button.
V. Select the required downloaded smart form file (xml format)
VI. Click on open button
VII. Enter package name & click on save icon.
VIII. Create the 'Transport request' number by pressing create request icon.
IX. Click on yes button & click on change button.
X. Make the necessary changes if required.
XI. Click on activate icon 7 click on back icon.

**15) What are the System Fields associated with Smart Forms?**

Ans :- SFSY is the system provided structure which contains a smart form system fields.

| Field | Description |
|---|---|
| PAGE | Current page Number |
| FORMPAGES | Total No. of pages |
| JOBPAGES | Total No. of pages of print job |
| DATE | Current date of Application Server |
| TIME | Current Time of Application Server |
| SUBRC | Return value |
| USERNAME | user name |

| PAGENAME | page name + variant |
|---|---|

**Note:**

- ✓ Smart Styles are used to create the paragraph & character formats. The T-code is SMARTSTYLES.
- ✓ SSF_FUNCTION_MODULE_NAME is the function module which is used to generate the smart form function module no based on the smart form name.
- ✓ SPELL_AMOUNT is the function module which is used to convert the amount in words.
- ✓ The button SWAP is used to change the Source language to Target Language.
- ✓ The T-code SOST is used to check the send mails & push them.

**16) What is the role of TNAPR table in SAP?**

**Ans: It Stores all the forms and driver programs and as well as all NACE settings.**

**17) What is the role of NAST table in SAP?**

**Ans: After the final selection of the output is done and the application document is saved, entries are created in the NAST table with application ID, Application document number, output type, output medium, output timing and Status code.**

**18) How to find print program and form if an 'output type' is given?**

**Ans :-The T-code NACE/ NACO or the standard table TNAPR is used to find the Form details or Output type details in the system.**

**19) Explain about NACE transaction code?**

**Ans: - NACE: - It is used to maintain the Output types and find the form name and print program names. If you create the output type using NACE then it will be automatically visible in table NAST and TNAPR.**

**The standard program TNAPR can also be used to find the form & print program name.**

- ➢ Execute theT-code NACE.
- ➢ Select an appropriate required application.
- ➢ Clickon output button on tool bar
- ➢ Click on position button.
- ➢ Enter the required Output type: NEU& Press Enter button.
- ➢ Select the font Output type (NEU).
- ➢ Double click onprocessing routines folder.
- ➢ Click on display/change icon.
- ➢ Change processing routine details such are print program, form routine, form name etc.
- ➢ Click on save & click on create request icon.
- ➢ Enter short description as per the project standards.
- ➢ Click on save icon & click on Yes button.

**20)What are the steps to send a FORM/REPORT as PDF/XLS/.TXT attachment through an E-Mail?**

**Ans:**

- ➢ Pass flag 'X' to field 'GETOTF' of export parameter control_parameters.

- **Get the form data in OTF format using the system generated function module of the SMART FORM.**

  **Note: Use the standard function module 'CLOSE_FORM' to get SAP Script form data in OTF format.**
- **Convert the OTF data of the forms into a PDF format using the function module CONVERT_OTF.**
- **Populate the mail subject, mail body, receivers list etc.**
- **Use the standard function module 'SO_NEW_DOCUMENT_ATT_SEND_API1' to send an email with an attachment such as PDF or Excel etc.**

**Q) How do you achieve Page Protection in Smart form?**

**Ans: While one can use the PROTECT..... ENDPROTECT command for SAP-Scripts, for Smartforms the Page-Protection checkbox can be used to ensure page protection**

- **You have Page-protection property only for Text Elements in the Main Window.**

# File Handling Techniques



## File Handling Techniques

**1) What are the Function Modules associated with Local PC/Front End system?**

**Ans :- GUI_UPLOAD - To upload the flat file from the specified file path of Local PC/Front End into internal table.**

**GUI_DOWNLOAD - To download the data from aninternaltable into the specified file path of Local PC/Front End.**

**2) How to handle the files in Application Server?**

**Ans: -**

- **It is used to maintain files securely.**
- **The T-Code 'AL11' is used to work with Application server.**
  **In the application server each file is called one Dataset. We can't create the file directly in the application server. Through program only we can create the file in the application server (download the data into application server).**

**3) What are the steps involved in handling Application server?**

**Ans :- The T-Code AL11 is used to work with the Application server.**

- **The below ABAP statements are used to Download/Transfer the data from an internal table into the specified file path of an application server.**

1. **OPEN DATASET FOR OUTPUT: This statement is used to open the required files in writemode in application server.**
2. **TRANSFER: This statement is used to transfer the concatenated data into the opened file.**
3. **CLOSE DATASET:It is used to close the opened file in the server.**

- **The below ABAP statements are used to Upload/Read the data from file in application server into an internal table of the program.**

1. **OPEN DATASET FOR INPUT: This statement is used to open files in read mode.**
2. **READ DATA SET: This statement is used to read record by record from the opened file in Application server.**
3. **CLOSE DATASET: This statement is used to close the opened file.**

**4) How to upload the file from presentation server (Front end) to Application server manually?**

- ✓ **The Code 'CG3Z' is used to upload the files from the Front end system into the specified file path of an application server.**
- ✓ **Once the file is open in 'AL11'.**
- ✓ **Execute the T-Code 'CG3Z'.**
- ✓ **Enter the source file name on the Front end system.**
- ✓ **Enter the Target file name on an application server.**
- ✓ **Select the required file type and file name on the front end system by pressing the search help icon(BIN/ASC).**
- ✓ **Click on Upload icon.**

**5) What are the Function Modules to provide Search Help on Presentation server/local PC and Application Server as well?**

**Ans :-**

- **The Function module 'F4_FILENAME' is used to provide the search help to select the required file path of the presentation server/Local PC.**
- **The Function modules 'F4_DXFILENAME_TOPRECURSION ' OR '/SAPDMC/LSM_F4_SERVER_FILE' is used to provide the search help to choose the files from the Application server.**

**NOTE: - The function module 'F4_DXFILENAME_TOPRECURSION ' can be used for both Presentation and Application server.**

**6) How to convert logical file in to corresponding physical file path?**

**Ans :-UsingFile transaction code.**

- **The function module 'FILE_GET_NAME' can be used to get the logical file path of the physical file path.**

**Note: ALSM_EXCEL_TO_INTERNAL_TABLE is the function module which is used to upload the excel sheet information into internal table.**

- **The T-code CG3Y is used to download the files from the Application server/Back end into the specified file path of the Presentation server/Front end.**
- **The T-code CG3Z is used to upload the files from the Presentation server/Front end into the specified file path of the Application server/Back end into.**

# DATA MIGRATION/CONVERSION



## DATA MIGRATION/CONVERSION

**1) What are the common steps involved in Data Migration/BDC?**

**Ans :-**

**Common steps involved in Data Migration:-**

- **Record the Screen Flow of the required functional T-code using the T-Code 'SHDB'.**
- **Convert the recorded screen Flow into a temporary program..**
- **Upload the Flat file data from the Legacy System (Local PC /Al11) into an Internal Table.**
- **Convert the uploaded flat file data within the loop of flat file internal table(gt_flat_file).**
- **Copy the system generated subroutines which starts with BDC_DYNPRO& BDC-FIELD from the temporary program & paste them within the loop of Flat file internal table.**
- **Comment screen default values & pass the flat file data.**
- **Use any one of either session/call transaction to update/migrate the data.**

**Note: An icon export is used to download the recording into local pc.**

**An icon read form file is used to generate a program using the given recording file.**

**2) What is LSMW?**

**Ans :- Legacy System Migration Workbench .**

- **It is the system provided tool to migrate or convert the legacy system data.**
- **The T-Code 'LSMW' is used to work with this tool.**
- **An icon export is used to download the recording into Local PC.**
- **An option read from file is used to generate a program using the given recording file.**

**3) What are the steps for LSMW?**

**Ans: There are totally 14 steps in LSMW.**

1. **Maintain object attributes**
2. **Maintain source structures**
3. **Maintain source fields**
4. **Maintain structure relations**
5. **Maintain field mapping conversion rules**
6. **Maintain fixed values, translations & user-defined routines**
7. **Specify files**
8. **Assign files**
9. **Read data**
10. **Display read data**
11. **Convert data**
12. **Display converted data**
13. **Create batch input session**
14. **Run batch input session**

**4) What are the differences between LSMW and BDC?**

**Ans :-**

| LSMW | BDC |
|---|---|
| 1. It is the system provided tool. | 1. It is the Utility to develop a program. |
| 2. Doesn't require any programmingknowledge. | 2. It requiresprogrammingknowledge.. |
| 3. The data can be imported in different import methods such are Batch InputRecording, BAPI and IDOC. | 3. The data is processed or migrated in Batch Input Method only |
| 4. The Field mapping is done automatically by the System. | 4. The Field mapping should be done manually by passing the Flat data. |
| 5. The T-Code LSMW is used. | 5. The T-Code SM35 is used to develop BDC program |
| 6. Mostly used for both Huge & small amount (Master) of data. | 6. BDC can be used for Small amount of data only. |
| 7. LSMW is possible for custom screens/standard with custom fields of applications. | 7. BDC can be used custom screens/custom fields of application also. |

**SHDB: It is the T-code used to record the screen flow as the T-codes.**

**5) What are the differences between Session Method and Call Transaction Method?**

**Ans :-**

| Session | Call Transaction |
|---|---|
| 1. The standard Function modules 'BDC_OPEN_GROUP', 'BDC_INSERT' and 'BDC_CLOSE_GROUP' are used to work with the session method. | 1. The ABAP statement CALL TRANSACTION ….is used. |
| 2. The data is updated in Synchronously mode (Record by Record). | 2. The data is updated in both synchronous and Asynchronous modes. |
| 3. An Error LOG File is generated by the system to handle the errors. | 3. The messages/errors are manually handled explicitly using the structure BDCMSGCOLL & the function modules 'FORMAT_MESSAGE' or 'WRITE_MESSAGE' or the table T100. |
| 4. Session method can process any no of transactions at a time. | 4. Call Transaction can process only one transaction at a time. |
| 5. After processing the session through SM35 only, the database is updated. | 5. Immediate database updation. |
| 6. Session method is slower. | 6. Call transaction method is faster |
| 7. We can schedule the session method in background. | 7. We can't schedule the call transaction in background. |

**Synchronous: In this mode, the data is not continuously updated in case of error records.**

**Asynchronous: It is used to update the data continuously by skipping the error records.**

**6) Is it possible to write an ABAP code in LSMW?**

**Ans :- Yes, by clicking on the change(pencil) icon that appears beside the recording structure field.**

**7) What are the Function Modules associated with BDC Session Method?**

**Ans :- The System provided Functional Modules are**

- **'BDC_OPEN_GROUP ': It is used to create the session using the T-code SM35.**
- **'BDC_INSERT': It is used to insert the flat file data of the structure 'BDCDATA' with transaction into session.**

**Note: An export parameter "CTUPARMS" is used to handle the screen resolution in case of table control to process the multiple line items in session method.**

- **'BDC_CLOSE_GROUP': It is used to close the opened batch input session.**

**8) How to handle messages in Call Transaction Method?**

- **Get the message details such are message type, message number into an internal table (gt_bdcmsgcoll) of the type BDCMSGCOLL using an addition "….MESSAGES INTO" with CALL TRANSACTION statement.**
- **Set a loop into an internal table (gt_bdcmsgcoll) of the type "bdcmsgcoll".**

- Use any of the function modules “FORAMT_MESSAGE”/”WRITE_MESSAGE” to get the corresponding message text, since the structure “BDCMSGCOLL” does not have any message text field.
- Display the populated messages with message text either list or ALV.

Note:The standard table ‘T100’ can also be used to populate the message text.

**9) What is the syntax to make use of Call Transaction?**

Ans :- CALL TRANSACTION '<T-CODE>' USING <gt_bdcdata> MODE <A/N/E> UPDATE<S/A>
MODE <A/N/E> UPDATE <S/A>
MESSAGES INTO <GT_BDCMSGCOLL>.

- Here <T-CODE> is the required transaction code through which the flat file data to be migrated.
- <gt_bdcdata> is an internal table of type “bdcdata” structure
- <A/N/E> are different data process modes
  A = All screens, N = No screens, E = Error screens
- <S/A> are the different update modes

**10)Which method do you prefer to migrate the data if the method is not mentioned?**

Ans :- CALL TRANSACTION.

**11) What are the fields in BDCMSGCOLL structure?**

Ans:BDCMSGCOLL is one structure in DDIC which contains the below fields

| Field | Description |
|---|---|
| TCODE | BDC Transaction code |
| DYNAME | Screen program name |
| **DYNUMB | Screen number |
| **MSGTYP | Message Type (E/W/I/S….) |
| ** MSGSPRA | Message language. |
| MSGID | Message ID |
| MSGNR | Message Number |

**12) What are the fields in BDC DATA structure?**

Ans :- It is a structure in DDIC which contains 5 fields.

| Field | Description |
|---|---|
| PROGRAM | BDC Module pool |
| DYNPRO | BDC Screen number |
| DYNBEGIN | BDC Screen Start |
| FNAM | field Name of the screen |
| FVAL | BDC Field Value(flat file) |

**13) Can you set up background processing using CALL TRANSACTION?**

Ans: Yes, Using No Screen Mode in 'CALL TRANSACTION'

**14) Can we use call transaction and session method in the same program? Explain?**

**Ans: Yes, we can call ,call transaction and session method in same program, generally we use call transaction for updating and session method for logging errors in the same program .**

**15) How to run the Session in Background?**

**Ans :-The standard report program RSBDCSUB is used to schedule the Batch Input Sessions in Background. (OR)**

**We can also SUBMIT the program RSBDCSUB within the BDC session program itself.**

**16) What Is BDCRECCX1?**

**Ans:- In a standard program, Generated from recording which contains all the re-usable declarations and also the re-usable Subroutine Definitions for all the BDC Programs .**

**17) How can you PRINT a session log file?**

**Ans :- Execute RSBDCLOG .**

**18) How to Export the Session?**

**Ans :- From SM35 ---> Utilities ---> Export Session .**

**19 ) While Uploading a flat file through BDC Call Transaction , The system(not SAP Server i.e. the currently working Machine) suddenly get CRASHED . How do iknow how many records have been updated ?**

**Ans:- Even though it is BULK Data Processing ,but updating the data base is always record by record only. So After Calling the Transaction, Maintain the status of the calling transaction into one custom table including the record details. So that we can know the custom table for the latest details.**

**20) After running a BDC program in background, next day morning when you see the results, few records are not updated(error records). What will you do then?**

**Ans:We will look into incorrect session, analyze the error screen, and reprocess the session after correcting the data.**

**21) You are given functional specs for a BDC program and you need to decide whether to write a method call transaction or a session. How u will decide?**

**Ans: Based on the amount of data(number of records to update), I will decide what method to use.**

**Ex: If data is less the 5000, I will use call transaction.**

**If data is more than 5000 records I prefer session method.**

**22). How to handle screen Resolution?**

**Ans:**

- **Declare a structure data variable of the type 'CTU_PARAMS'.**
- **Pass the required data such as default size, display mode, update mode etc to the defined ctu_params structure variable.**

- **Use the populated structure variable of type 'ctu_params' with CALL TRANSACTION using an addition 'OPTIONS'.**

**23) How to handle Table Control in BDC?**

**Ans :- Table control is used to migrate/update the multiple records(line items) in BDC.**

**Note:**

- ✓ **Press Page Down in the table control to record the function code for page down.**
- ✓ **The next item no is the 1st item no when the Page Down is pressed.**
  - **Record the Screen flow with line items using the T-Code "SHDB".**
  - **Convert the recording into a program using the button program.**
  - **Make the changes in the original main program like,**
    - **Upload the flat file data which contains both Header & Item data.**
    - **Split the uploaded Flat file data into the Header and Item tables, using the Control Break Statements.**
- ✓ **Set a loop to Header internal table then process the Header related data by copy & paste the system generated header related sub-routines from the temporary recording program**
- ✓ **Set a loop to Item table data to process the multiple line items.**
- ✓ **Make the Item number as dynamic by concatenating the screen field name with the screen number.'**
- ✓ **Use the function code "p+" to handle page down in table control.**
- ✓ **Use any one of methods with BDC_INSERT(Session) or CALL TRANSACTION to migrate the data from internal table GT_BDCDATA.**

**Note: TheHeader & Item data can be processed separately using the control break statements without splitting the flat file data.**

**The control break statements can also be used to process Header & Item data separately by avoiding nested loops.**

**24) What are the problems in processing batch input sessions? How is batch input process different from processing on line?**

**Ans: 1) If the user forgets to opt for keep session then the session will be automatically removed from the session queue (log remains). However, if session is processed we may delete it manually.**

**2) If session processing fails, data will not be transferred to SAP database table.**

# ENHANCEMENTS



## Enhancements

**1) What is Enhancements and what are the different techniques and explain them?**

**Ans :-Enhancements are used to add some additional customer specific functionality to the standard applications/programs.**

1. **Customer Exits: These are the Function Modules available in most of the functional areas such as SD, MM, FICO etc.They are used to add the additional functionalities in the standard applications/programs without using access key to break them. The access key is the unique key provided by the BASIS consultant to break the standard applications. The application will become custom development once they are broken.**

- **Field Exits: These are used to change the field labels of the standard data elements.**
- **Function Exits: These are used to add the additional functionality to the standard programs without breaking them. Each Function Exit contain a custom include program which starts with Z* to implement the customer specific additional business logic. The Function Exit is the 3 digit no which is called using the ABAP statement CALL CUSTOMER_FUNCTION.**
- **Menu Exits: These are used to add the custom specific menu items & corresponding additional functionality based on the selected menu item.**
- **Screen Exits: These are used to add the additional screen elements & the corresponding additional functionality to the standard screens/transaction.**
- **T-Code SMOD: It is used to find the customer exits of the standard package.**
- **T-code CMOD: It is used to assign an enhancement/exit to project.**

2. **User Exits: These are the Sub-routines which start with USEREXIT_*& these are defined in the standard Include programs. They are mostly used in SD functional area. The Access key is required to break the standard Include programs of the User Exits for the first time. The T-code SPRO is used to find the User Exits.**
3. **BTE (Business Transaction Events): These are the Source code plug-ins to add an additional functionality in the standard program.**
   - **T-code BERE: It is used to find the BTE's.**
   - **T-code FIBF: It is used to work with BTE's.**
4. **Enhancement Framework: The enhancement frame work is advanced to user exits for adding an additional functionality without breaking the standard programs. It is used to attach/hook the source code plug-ins such as enhancement section or enhancement points in the standard programs.**

✓ **In User-exits we write code only in the form routines(ie., in Form...End Form). and When we go for upgrade the versions, we again need to change the code in the user-exit whichwe written in previous versions but with Enhancement Frame work there is not much code change is required when we go for upgrades.**

- **BADI: These are new enhancement technique to add the additional customer specific business logic to the standard programs.**

**2). what are the ways to find the Exits?**

I. **Using the T-code Package name of the applications**
II. **Using the T-code SE84**
   - **Find the package name of the required T-code.**
   - **Execute the T-code SE84.**
   - **Expand Enhancement folder**
   - **Double click on enhancements**
   - **Enter the required package name/Exit name**

- Click on execute icon

III. Using the Find function in standard programs.

IV. Using the standard Tables MODSAP & MODACT

**3) What are the types of Enhancement Frame work?**

**Ans:**

- It is an advanced enhancements concept introduced from the version ECC6.
- It is used to attach/hook the source code plug-ins such as enhancements section or enhancement points in the standard programs.
- There are 2 types of enhancement frame works.

1. Implicit Enhancement Section/Point: They are provided implicitly at the particular locations by SAP. They are mostly presented at the beginning & end of the sub routines (FORM-ENDFORM), beginning& end of the function modules, beginning & end of the programs.

2. Explicit Enhancement Section/Point: They are pre coded by SAP & provided at the specific location in the standard programs. There are 2 types of Explicit Enhancement.

- Enhancement-Section: It is used to copy, change the existing standard source code & add additional business logic to the standard program. The ABAP statement ENHANCEMENT-SECTION is used to specify the Enhancement Sections.
- Enhancement-Point: It is used to add the additional customer specific business logic to the standard program.

**4) What are the differences between Customer Exits and User Exits?**

| Customer Exits | User Exit |
|---|---|
| 1. These are the Function modules | 1.These are the subroutines |
| 2. The Customer Exit have different type such as function exit, menu exit, screen exit & field exit | 2.User Exit do not have different types |
| 3. These are available under most of the functional areas such as SD,MM,FI etc | 3.These are mostly available in SD functional area |
| 4.Access key is not required to add the additional functionality | 4.Access key is required to break the standard program |
| 5.The Customer Exits are called using CALLCUSTOMER_Function | 5. The User Exit starts with USEREXITS(subroutines in standard include) |

**5) What are the Transaction Codes associated with BADIs?**

**Ans :-**

**T-code SE18 & SE19: are used to work with the BADI's**

**T-code SE18: used to define the BADI's**

**T-code SE19: used to implement the BADI'**

**6) What are the types & sub types of BADIs and explain about them?**

**Ans :- It is used to add the additional customer specific business logic to the standard programs.**

1. **Classic/Old BADI: These are mostly used in the functional areas like SD, MM, FI/CO. The standard method "GET_INSTANCE" of the standard class "CL_EXIT_HANDLER" is used to work with the classic BADI's.**
2. **Kernel/New BADI: The ABAP statements "GET BADI & CALL BADI" are used to work with New BADI's. It is faster than Classical BADI's.**

   **GET BADI: This statement is used to generate a New BADI object & sets the BADI reference to the objects in the BADI reference variables.**

   **CALL BADI: This statement is used to call the Methods of the New BADI's.**

**Subtypes:**

a. **Single use BADI: The single use BADI can't beimplemented for multiple times. This BADI can have only one implementation.**
b. **Multiple use BADI: The multiple use BADI's can't be implemented for the multiple times.**
c. **Filter Dependent BADI: It is used to control the BADI implementation based on the specified filter value. The system generates an Import parameter 'FLT_VAL' to pass the filter values for controlling the business logic of the BADI implementation.**
   - ***Interface: It is a separate structure which can be used to extend the scope of a class.***

## 7)What is FALL BACK class in BADI?

**Ans :- If a BADI doesn't have any active implementations, the program goes to run time error. The Fall back classes are used to call the default methods in case of a BADI doesn't have any active Implementations.**

- **The exceptions can also be raised to avoid such kind of errors.**

## 8) What are the differences between Exits and BADIs?

**Ans :-**

| Exits | BADIs |
|---|---|
| 1. They are procedural ABAP based | 1. They are Object Oriented ABAP based |
| 2. It is not possible to assign an Exit to more than oneproject | 2. The multiple use BADI can be implemented For multiple times |
| 3. The Filter values can't be set to the Exits | 3. The Filter values can be set for Filter Dependent BADI's |
| 4. The T-codes SMOD & CMOD are used | 4. The T-codes SE18 & SE19 are used |

## 9) What are the ways to find BADIs?

**Ans :-**

- ✓ **Most of the Standard BADI's are classic BADI's.**
- ✓ **They are used to add the additional customer specific functionality/business logic to the standard programs/T-codes.**
- ✓ **There are 4 ways to find the BADI's**
  - **I. Using the standard class CL_EXITHANDLER**
    - **Execute the T-code: SE24 to maintain the global class.**

- Enter the standard class name CL_EXITHANDLER
- Click on display button.
- Double click on the method GET_INSTANCE
- Set a session break point on the method GET_CLASS_NAME_BY_INTERFACE
- Execute the required functional T-code Ex: MM02
- Perform the required action Ex: save material
- Double click on a passing parameter (EXT_NAME) to list out the corresponding BADI. Ex: BADI_MATERAIL_CHECK

II. Using the T-code SE84

III. Using the T-code ST05 (SQL Tracer)

IV. Using the T-code SPRO

## 10) What are the differences between Classic (Old) BADI and New(Kernel) BADI?

**Ans :-**

| Classic or Old BADI | New or Kernel BADI |
|---|---|
| 1. The standard method GET_INSTANCE of the standard class CL_EXIT_HANDLER is used to call | 1. The ABAP statements GET BADI & CALL BADI are used to work with New BADI's. |
| 2. An interface of a BADI referred to create the Reference object | 2. The definition of a BADI is referred to create the Reference object |
| 3.It is not faster as like the Kernel BADI | 3. It is faster than the Classic BADI |
| 4. It can be a multiple use BADI | 4. It must be a single use BADI |
| 5. The Fall back class can't be defined | 5. The Fall back class can be defined |
| 6. It can be a Filter Dependent BADI | 6. It is not possible to maintain the Filter values |

## 11) What are the statements not be used in Exits and BADIs?

**Ans :- The ABAP statements COMMIT work & ROLLBACK work should not be used in EXITS.**
**The ABAP statements STOP & EXIT are also not advisable.**

## 12) What are the Enhancement Spots?

**Ans :-**

- The enhancement spots are used to manage explicit enhancement options that means you can add your code in standard ABAP code without need of access key,which implies that the standard code is not disturbed.
- Enhancement spots specify the places where we can add our code in standard SAP.

Each enhancement spot element definition must be assigned to at least one enhancement spot.

**13) What are the steps to insert the Enhancement Spot and work?**

**Ans:**

- **Implicit enhancementSpots are available in every program on some predefined spots as a begin of FM, end of FM, begin of some event and so on. It means that you can easily modify any program, FM, subroutine with some limitations.**

- **If you want to see all implicit spots in program,**
  **go toSE80 -> Edit -> Enhancement Operations -> Show Implicit Enhancement Options**

# Cross Applications/ Interfaces (RFC, BAPI, IDOC)



## Cross Applications (RFC,BAPI,IDOC)

**1. What is the Cross Application/Interfaces concept?**

**Ans: -This concept is used to distribute the data between SAP to SAP systems (SAP R/3 <-> SAP CRM) &SAP to Non SAP systems (SAP <-> .NET, JAVA)**

**2) What is Remote Function Call (RFC)& types of RFC's?**

**Ans:**

- **It is a SAP specific protocol to provide communication between different systems.**
- **It is the process ofcalling Function modules from another system.**
- **SAP Uses CPIC (Common Programming Interface for Communication) Protocol to transfer data between Systems.**

**Types:**

1. **Synchronous RFC: In case of this RFC both the Sender/Source and Receiver/Target systems should be available while distributing the data. The next part of calling program isn't continued until call function is completed.**

2. **Asynchronous RFC: In case of this RFC both the Sender/Source and Receiver/Target systems should not be available while distributing the data. The next part of calling program is continued without completing call function.**

3. **Transactional RFC (TRFC):It is almost similar to Asynchronous RFC. The Transactional RFC is executed only once in an RFC server & save the corresponding dataunder unique transaction ID in the Database.**

4. **Queued RFC: In case of this RFC the multiple transactional RFC's are serialized in the sequence using the Function module 'TRFC_SET_QUEUE_NAME'.**

- **An additional statement 'Destination' is used while calling Remote Enabled Functions.**

**3) How to Debug RFC Function module?**

**Ans:**

- **SE38 -> Utilities -> Settings -> ABAP Editor -> Debugging activate the external debugging and choose the New Debugger option in ABAP debugger.**
- **Go to the particular place in the code and put break point, pop will appear then choose the HTTP break point.**
- **If you are triggering the RFC from SAP portal make sure that both the user ID should be same If the users are different then provide the XI/Portal User ID in the users field.**

**4) How do you connect to the remote server if you are working from the office for the client in remote place?**

**Ans: WAS ->Web Application Server or ITS (Internet Transaction Server) are generally used for this purpose. If you are sitting at your office with a server which is in the system and the other server is at the clients place you can generate IDOC, Intermediate Documents which carry the data you want to transfer or the documents you want to transfer, these IDOC are interpreted by the system at the receiving end with the message class with which it is bound with. If you want to logon a system which is very distant, then remote login can be used this depends on the internet speed.**

**5) What are the steps to work with Remote Enabled Function/maintain RFC connection?**

**Ans: An additional statement *'Destination'* is used while calling remote enabled functions.**

1. **Define RFC Destination:** 2. **Creation of Remote Enabled Function: SE80**

- **Execute the T-Code : SM59**
- **Select the required RFC connection such as ABAP connections/Type 3 connections.**
- **Click on create icon**

- Enter RFC destination name & Description : SAP R/3 to SAP CRM
- Enter the target application server name/address in the target host field : CRM system
- Enter the target system number Ex: 33
- Click on “Logon & Security” tab
- Enter the logon credentials of the Receiver/Target system

**Note: The check box Current user can be checked to make use of the same User name & password for the target system**

- Click on save icon & click on yes button
- Click on ‘Connection Test’ button to test RFC destination
- Click on ‘Remote Logon’ button to test the remote connection

**6) How to make Destination (system-id) as dynamic?**

**Ans:** IF Sy-sysid EQ ‘CMD.

<gv_dest> = ‘R3D’.

ELSEIF Sy-sysid EQ ‘CMQ.

<gv_dest> = ‘R3Q’.

ELSE Sy-sysid EQ ‘CMP.

<gv_dest> = ‘R3P’.

*Call RFC

CALL FUNCTION ‘ZRFC/RFC’

DESTINATION <gv_dest> “R3D

**7) What are the differences between RFC and BAPI?**

| RFC | BAPI |
|---|---|
| 1. RFC’s are used to connecting from SAP to SAP only | 1. BAPI is used to connecting from SAP to SAP as well as SAP to NON SAP |
| 2. TheT-code SE37 can be used to find RFC’s | 2. TheTcode BAPI &SE37 can be used to find the BAPI |
| 3. RFC doesn’t generate any methods | 3. BAPI generates the unique method in BOR to use in the external system directly |
| 4. It doesn’t have standard RFC’s to migrate the data | 4. The standard BAPI which starts with BAPI can be used to migrate the data |

**8) What are the steps involved in creating Custom BAPI?**

**Ans :-**

1. Create the BAPI structure (SE11) starts with ZBAPI*.
2. Create the BAPI function module (SE37) starts with ZBAPI* which is
   a) Remote Enabled

b) All parameters are pass by value

c) Must contain Return parameter

3. Create method & object for the BAPI remote enabled function in BOR using the T-code 'SW01'.
4. Generate, implement & release the created BAPI to the outside using the T-code - 'SQO1'.

## 9) What are the ways to find Standard BAPIs?

Ans :- There are 2 ways to find BAPI.

- Using the T-code : BAPI
- Using the T-code : SE37

## 10) How to find/check the Standard BAPIs?

Ans :-1. Execute the T-code BAPI

2. Expand the required Hierarchy. Ex: Sales & Distribution etc
3. Expand the required sub-functional area. Ex: Sales/Billing etc
4. Double click on the required BAPI
   - Corresponding BAPI details such as method name, business object name, BAPI function module &etc are displayed under details tab
   - The tab Documentation is used to know more details about the BAPI's (standard)
   - The tab Alphabetic can also be used to find the BAPI's in the alphabetical order
5. Click on back

## 11) What are the steps to work with standard BAPI?

Ans:

- Upload the flat file data into an internal table
- Convert the flat file data into BAPI compatible Work area's/internal tables with in the loop of the flat file internal table.
- Call the BAPI function module & pass the BAPI internal tables.

## 12) How to handle exceptions in RFCs and BAPIs?

Ans :-You can handle the exceptions by using a structure BAPIE1RET2 which returns the messages captured in BAPI. In Tables parameter of your BAPI, you can create a structure as of typeBAPIE1RET2 so that you can capture the 'RETURN'parameters from BAPI.

## 13) How to handle error messages in standard BAPI's?

Ans:

1. Get all the message details into a written table of the type BAPIRET such as Message ID, Message type, Message variable etc.

2. **Use the standard function modules ,**
    - **MESSAGE_INITIALIZE ( Initialize messages),**
    - **MESSAGE_STORE (To store messages) &**
    - **MESSAGES_SHOW (To display messages in the Dialog box/Model dialog)**

## 14) What is Extended BAPI?

**Ans:**

- **Standard BAPI's can be extended to add the customer specific additional field.**
- **Standard BAPI contains table's parameter EXTENSIONIN, EXTENSIONOUT in case they are extended BAPI's.**
- **BAPI extensionin is used to update custom fields of a table (ex: MARA).**
- **BAPI extensionout is used to retrieve custom fields of a table.**

## 15) What are the differences between BAPI and normal BDC?

| BAPI | BDC |
|---|---|
| BAPI is faster than BDC. | BDC is relatively slower than BAPI. |
| BAPI's are used to upload the data from the flat file to SAP system directly. | BDC's are used to upload the data from file to SAP system via screens hence it is slower. |
| No such processing options are available in BAPI. | Background and Foreground processing options are available for BDC. |
| BAPI would generally used for small data uploads. | BDCs would be preferred for large volumes of data upload since background processing option is available. |
| BAPI never cause to terminate the program. Whenever an error occurred in the BAPI, it returns those errors through Return parameter. This parameter returns exception messages or success messages to the calling program. | Errors can be processed in SM35 for session method and in the batch input program for Call Transaction method. |
| In BAPI recording is not required | In BDC, recording is required |

## 16) What are Authorization Objects?

**Ans:**

- **These are used to protect/prevent user access to the data & T-codes/Tables.**
- **The authorization objects are maintained by BASIS consultants.**
- **The T-code: SU24 is used to find authorization objects in the system.**
- **The ABAP statement 'AUTHORITY_CHECK' or the standard Function Modules 'AUTHORITY_CHECK*…' are used to check the authorizations.**

## 17) What is ALE& EDI?

**Ans :-**

- **ALE stands for Application Link Enabling. It is used to distribute the data between two different SAP systems/Servers.**
- **EDI stands for Electronic Data Interchange. It is used to distribute the databetween SAP & non-SAP systems/servers.**

**18) What is an IDOC?**

**Ans :-**

- **IDOC stands for Intermediate Document.**
- **IDOC is the collection of segments & each segment is the collection of fields.**
- **IDOC is the carrier to carry the data betweenthe distributed systems in electronic form without user interaction.**

**19) What is an IDOC Structure?**

**Ans:-Each IDOC has 3 records**

I. **Control Record:**
   - **It contains IDOC type & Message type information, Partner information such as partner system number, Port number of the sender & receiver system.**
   - **It is also contain technical & address information of sender & recipients.**

II. **Data Record:**
   - **These are collection of segments. The data record of an IDOC contains segment.**
   - **Segment is collection of fields which can be distributed the data between the systems.**

III. **Status record:**
   - **It contains the status of the processed IDOC to know where an IDOC has been successful/failed.**
   - **The standard table EDIDC contains an IDOC information such as control records, different statuses etc.**

**Note:**

- **The Sender/Source system is also called as 'Outbound system'.**
- **The Receiver/Target system is also called as 'Inbound system'.**
- **The T-code 'WE30' is used to maintain different types of IDOC's such as Basic IDOC, Extended IDOC.**
- **The T-codes WE02/WE05 are used to list out the generated IDOC's.**
- **The T-code 'WE19' is used to test/check the IDOC.**

**Custom Segments:**

- **The custom segments are starts with Z*/Y* & they are maintained by technical & functional consultants**

**Standard Segment:**

- **These are system provided segments**
- **The T-code : WE31 is used to maintain the segments**

**Message Types:**

- These are identifiers of the application data to be distributed between the systems
- The T-code WE81 is used to create & WE82 is to assign the message types to the IDOC (Data carrier)
- The standard table EDIDC contains all available message type

**System provided standard message types?**

| Message Type | Description |
|---|---|
| MATMAS | Material Master |
| DEBMAS | Customer Master |
| CREMAS | Vendor Master |

**Partner Profiles:**

- These are used to maintain the partner system details with which data to be distributed such as partner number, Partner type & Message type of outbound parameters.
- The T-code WE20 is used to maintain partner profiles

**Ports:**

- These are used to specify the way in which data to be distributed through an IDOC
- The system generates one unique port number
- The T-code WE21 is used to maintain ports for IDOC process

**Customer Distribution Model:**

- It is used to define data distribution model to exchange the messages between two logical systems (SAP – SAP/SAP – non SAP)
- The T-code BD64 is used to define/maintain customer distribution model

**Process codes;**

- It is used to identify type of the data to be distributed through an IDOC interface.
- The T-code WE41/WE42 is used to work with the process code

**15) What are the Transaction Codes/Reports to send/generate and receive the data to an IDOCs?**

**Ans :-**

| T-code | Description |
|---|---|
| BD10 | Send Material |
| BD11 | Get Material |
| BD12 | Send Customer |
| BD13 | Get Customer |
| BD14 | Send Vendor |
| BD15 | Get Vendor |
| BD16 | Send Cost center |
| BD17 | Get Cost center |
| BD87 | Reprocess IDOC |

**11) How to check the IDOCs?**

**Ans :-WE19 is the T-code to check the IDOC's.**

1. **Execute WE02/WE05**
2. **Provide the Message type: CREMAS (Logical Sys), Partner no: LS810 & Click on Execute**

**3. After we get the status code 03, if we want to know the IDOC is reached to destination or notthen we execute RBDMOIND standard program.**

**4. The status of an IDOC (3) use in Green color (turn 10 to12) means it is delivered Successfully.Yellow trafficlight indicates/the status code remains same, IDOC is in Progress or in the Transaction RFC & Red traffic light/the status code turns 10 to 11 indicates the Error/the IDOC is damaged.**

**5. Expand the Data Records folder & click on each segment to see the data to be distributed.**

## 12) What are the Transaction Codes associated with ALE, IDOCs?

**Ans :-**

| T-code | Description |
|---|---|
| SALE | Basic configuration of ALE/IDOC |
| WE30 | Maintain(create & display) IDOC's |
| WE31 | Maintain(create & display) Segments |
| WE20 | Maintain partner profiles |
| WE21 | Maintain ports in IDCO processing |
| WE41 | Maintain process codes (Outbound system) |
| WE42 | Maintain process codes (Inbound system) |
| WE81 | Maintain Logical Message Types |
| WE82 | Assign message type to IDOC type |
| NACE | Maintain output type |

- **Most of the T-codes which are related to ALE/IDOC starts with WE* & BD*.**
- **The T-code WE30 is used to create & display the different types of IDOC's such as Basic IDOC & Extended IDOC.**
- **The T-Code WE05/WE02 is used to list out the IDOC.**
- **The T-code WE19 is used to check the IDOC.**

## 13) What is the Transaction Code to create/maintain the segments?

**Ans :- The T-code WE31 is used to create & display the segments.**

## 14) What is the Transaction Code to create a Custom IDOC?

**Ans :-Create Segment ( WE31)**
**Create Idoc Type ( WE30 )**
**Create Message Type ( WE81 )**
**Assign Idoc Type to Message Type ( WE82 )**

## 16) What are the Partner profiles and its Transaction Codes?

**Ans :- Partner profiles are maintained/created to specify the partner details (system) to which the data to be transmitted. The T-code WE20 is used to create the partner profile.**

**17) What are the Message Types and the Transaction Code?**

**Ans :-**

- **The message type is the data identifier of an application.**
- **The control record of an IDOC contains the Message Type.**
- **The T-code WE81 is used to Create & WE82 is to assign the Message Type to the IDOCS.**

**18) How to add Segments to an Existing IDOC?**

**Ans :-**

- **You can do that using WE30 and WE31.WE30 you can create extension to the existing Idoc type and WE31 you can add the fields to the sub segment. So first develop the segment using WE31.you need to create the extension first &then addit to the segment which ever you want using WE30 with extension option.**
- **Here you have to create enhanced idoctype. In we31 create the segment type as per your requirement and go to we30 and create enhanced Idoc type by coping the standard Idoc.After that you will select the segment under which you want to add the new segmnt and press create button on the application tool bar and give the details of your new segment and save it.**

**19) What are the advantages of ALE, IDOCs and EDI?**

**Ans :-**

- **The data is distributed between the systems in electronic form without user interaction.**
- **It reduces the data errors, man power & paper work.**
- **It is used to distribute the data between SAP & NON SAP systems.**

**Q) How to reprocess error IDOC's?**

**Ans: We can reprocess error IDOC's(status 51) using program RBDAGAIN.**

**Q) How to reprocess edited IDOC's?**

**Ans: We can reprocess edited IDOC's using program RBDAGAIE (Se38 program).**

**Q) How do you get output from IDOC?**

**Ans: Data in IDOC is stored in segments; the output from IDOC is obtained by reading the data stored in its respective segments.**

**Q) How do we generate IDOC in SAP?**

**Ans: The function module MASTER_IDOC_DISTRIBUTE is used to generate the IDOC in the custom report**

**Q) Requirement: All the students data will be created in 800 client and same will be send to 810 client using IDOCS.**

***Steps to be done in 800 client***

**Login to 800 client and do following.**

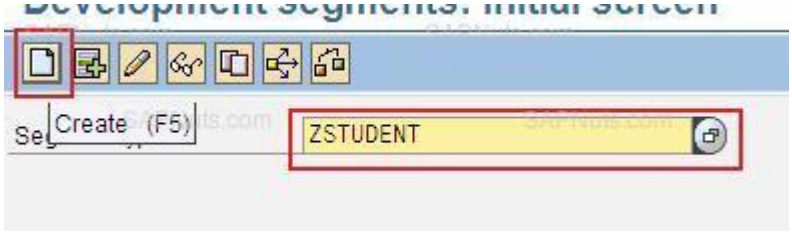
1. **Create a table (client dependent) : SE11**
2. **Create segments : WE31**
3. **Create IDOC type : WE30**
4. **Create message type : WE81**
5. **Assign message type to IDOC type : WE82**
6. **Define logical system : SALE**
7. **Assign logical system to client.**
8. **Create RFC destination: SM59**
9. **Create port : WE21**
10. **Check partner profiles : WE20**
11. **Create outbound program.**

***Steps to be done in 810(Target System) client.***

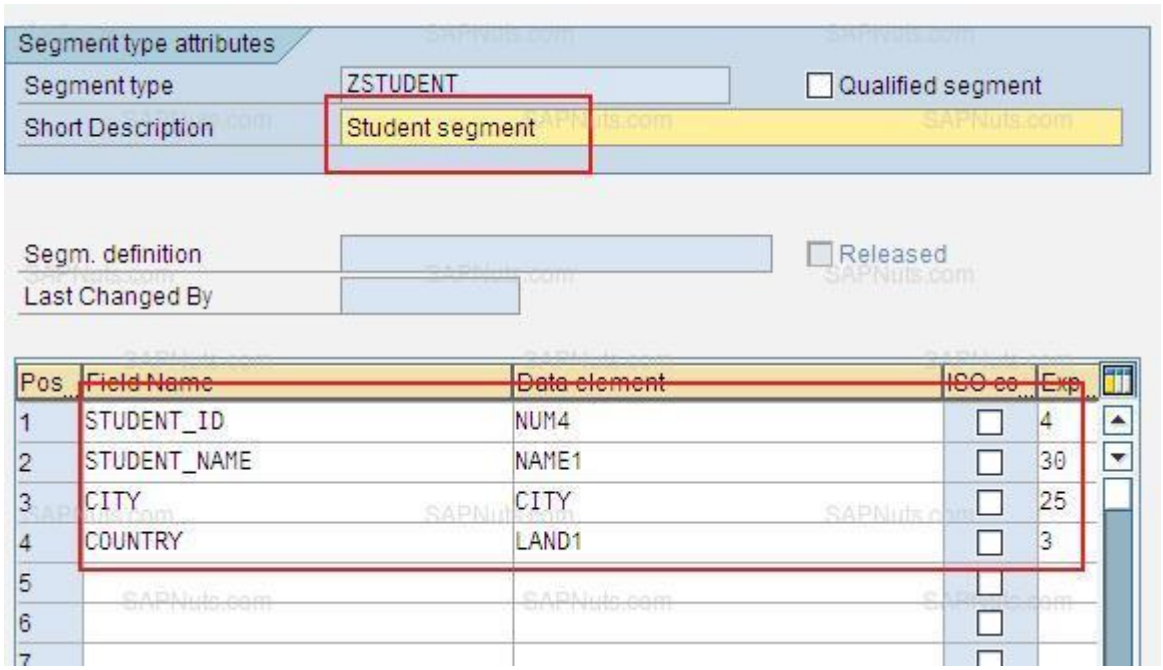
1. **Create a table (client dependent).**
2. **Create segments.**
3. **Create IDOC type.**
4. **Create message type.**
5. **Assign message type to IDOC type.**
6. **Define logical system.**
7. **Assign logical system to client.**
8. **Create inbound function module SE37.**
9. **Register function module characteristics.**
10. **Assign function module to message and IDOC type.**
11. **Create process code WE42.**
12. **Check partner profiles for inbound.**

1. **<u>Create a table (client dependent).</u> – SE11 : ZSTUDENTand click on create**
2. **<u>Create segments.</u> - WE31 : provide segment as ZSTUDENT and click on create**

**Go to t-code WE31, provide as segment ZSTUDENT and click on create.**

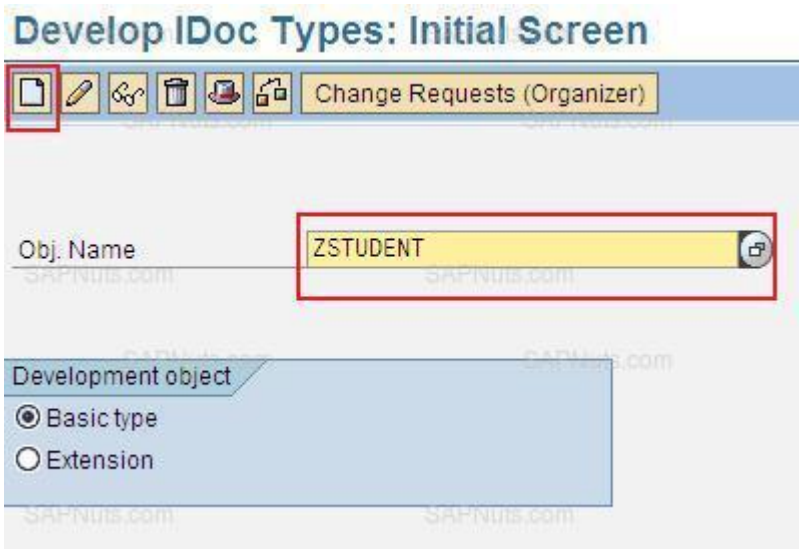



**Just click enter if you find any information message and add fields as below.**

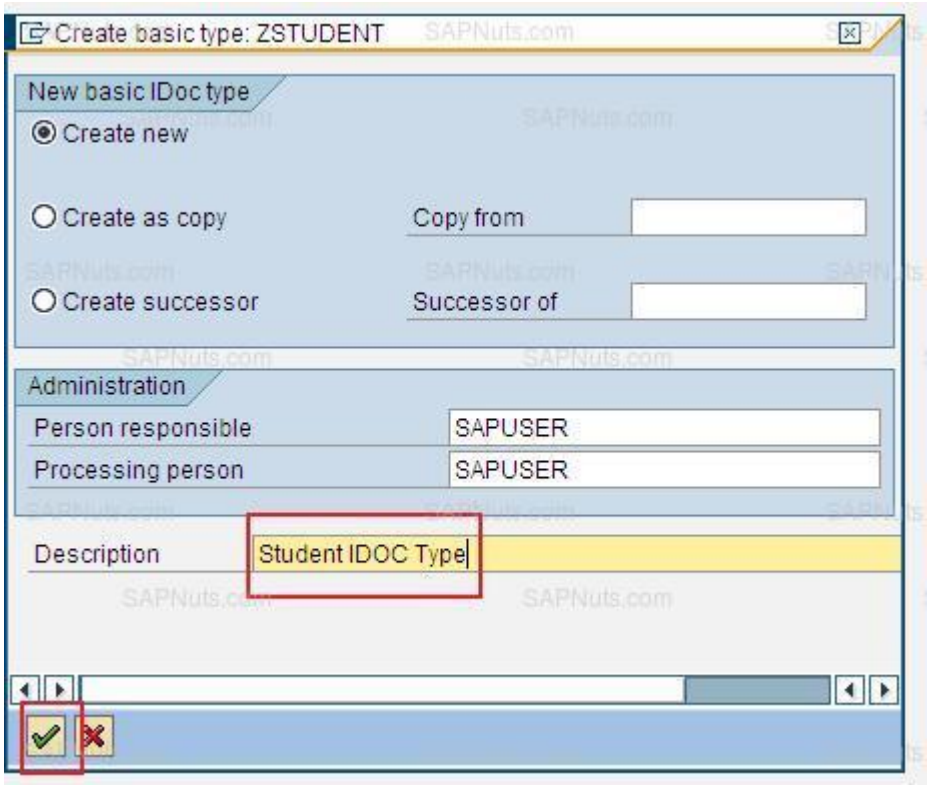



**Save it in a local object.**

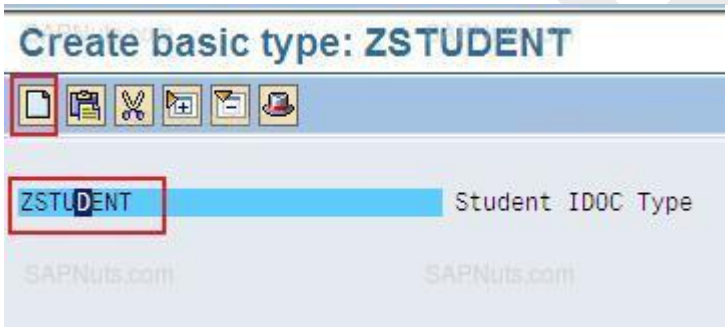
3. **<u>Create IDOC type.</u>Go to WE30, provide name as ZSTUDENT and click on create**



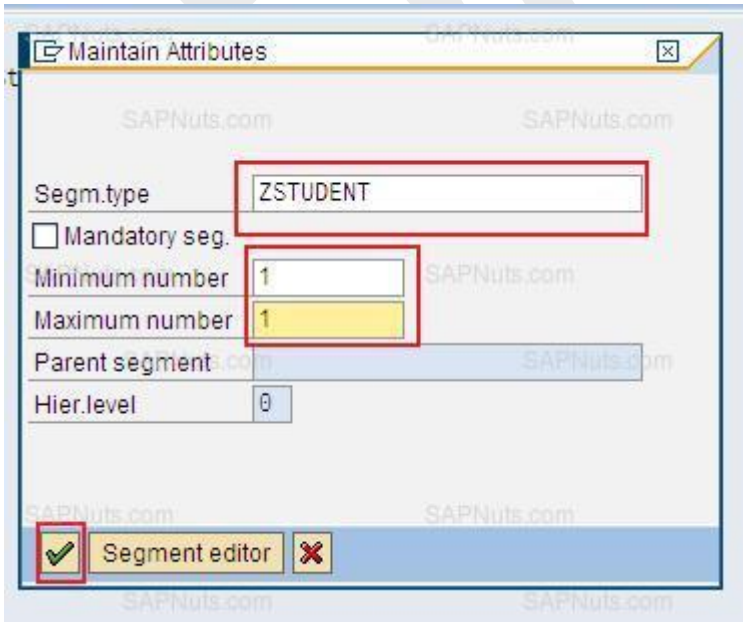
**Provide description and press enter.**




**Select ZSTUDENT and click on create.**

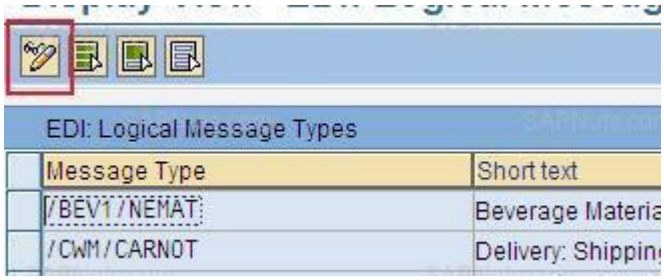



**Provide a segment name (which we have created in step2), maximum and minimum numbers and enter.**

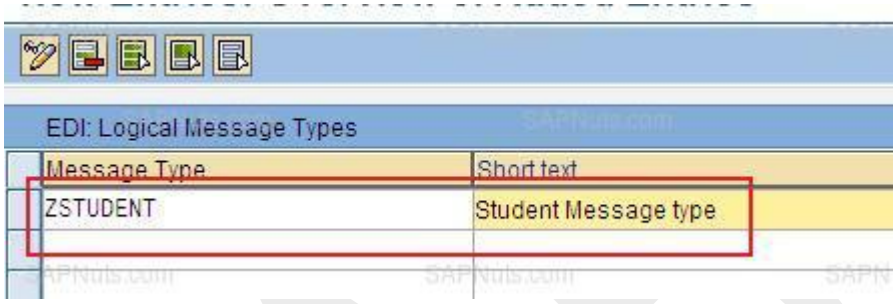


**Save it in a local object.**

4. **Create message type.Go to t-code WE81 and click on change icon.**

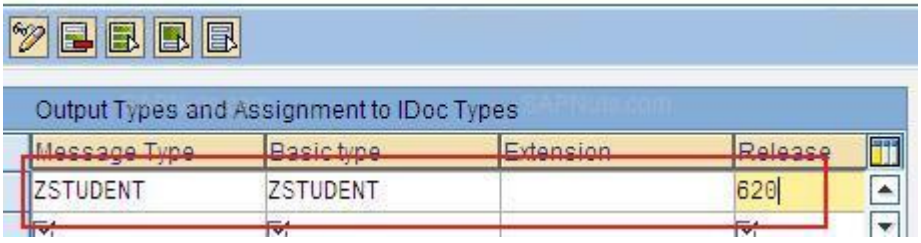



**Provide message type as ZSTUDENT and description.**




5. **<u>Assign message type to IDOC type.</u> Go to t-code WE82, click on change and click on new entries.**

**Provide message type, IDOC type and release as below.**




**Save it in a transport request.**

6. **<u>Define logical system.</u>**

**This is the most confusing step in ALE configurations, try to understand carefully.**

**What is logical system?**

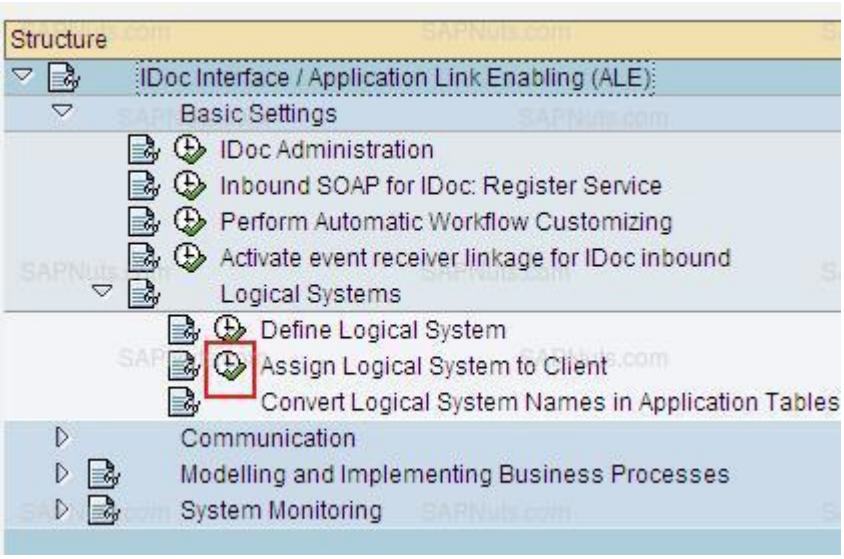
**'Logical system' is used to identify an individual client in a system, for ALE communication between SAP systems. That's why you see a field for 'logical system' in the client master data in SCC4 (table T000). You use logical systems in ALE configuration - this should be documented further in the IMG guide, or SALE transaction.**

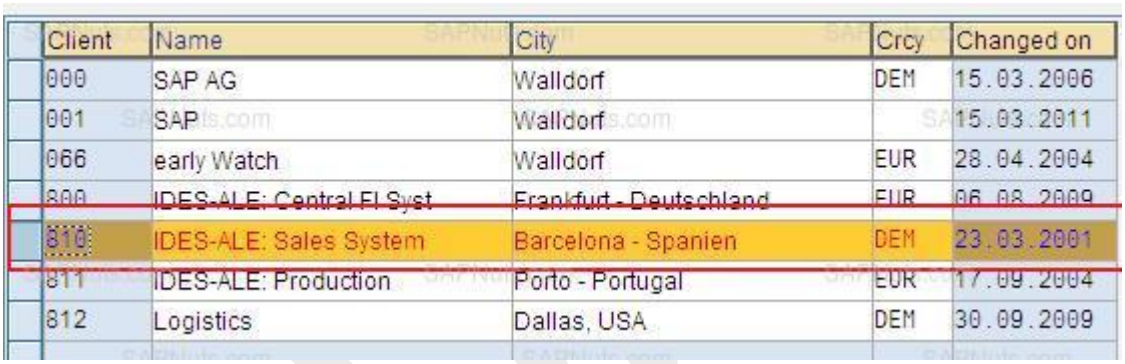
**Before creating a logical system please follows below steps.**

**In a blind way logical system assigned in target system client(in our example 810 client)**

**Should be in logical system definition in source client.**

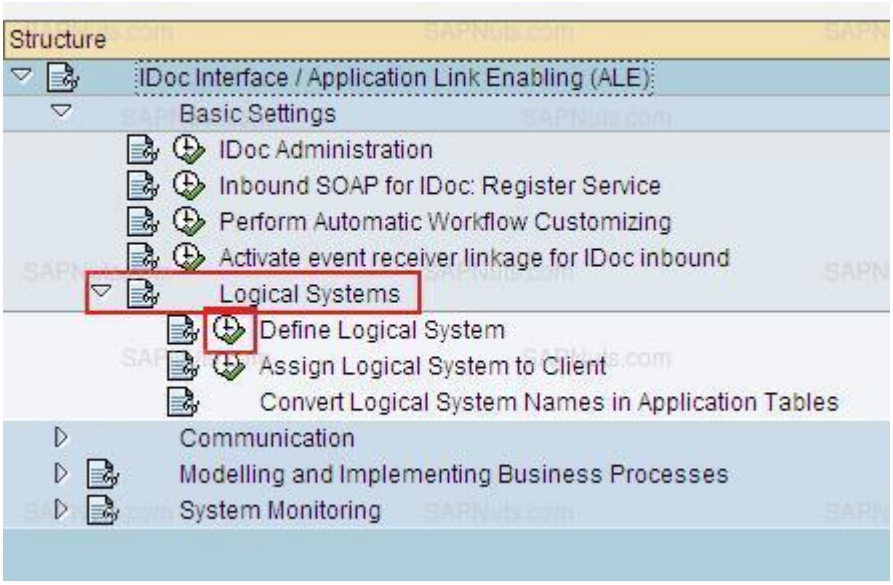
**Step1: Go to target system( In our example 810 client).**
**(Get logical system from t-code SALE ->Basic Settings ->Logical Systems -> assign logical system to client -> 810 (double click and copy logical system)).**



| Client | Name | City | Crcy | Changed on |
|---|---|---|---|---|
| 000 | SAP AG | Walldorf | DEM | 15.03.2006 |
| 001 | SAP | Walldorf | | 15.03.2011 |
| 066 | early Watch | Walldorf | EUR | 28.04.2004 |
| 800 | IDES-ALE: Central FI Syst | Frankfurt - Deutschland | EUR | 06.08.2009 |
| 810 | IDES-ALE: Sales System | Barcelona - Spanien | DEM | 23.03.2001 |
| 811 | IDES-ALE: Production | Porto - Portugal | EUR | 17.09.2004 |
| 812 | Logistics | Dallas, USA | DEM | 30.09.2009 |

**Step2: Go to SALE.**
**Expand basic settings -> define logical system.**

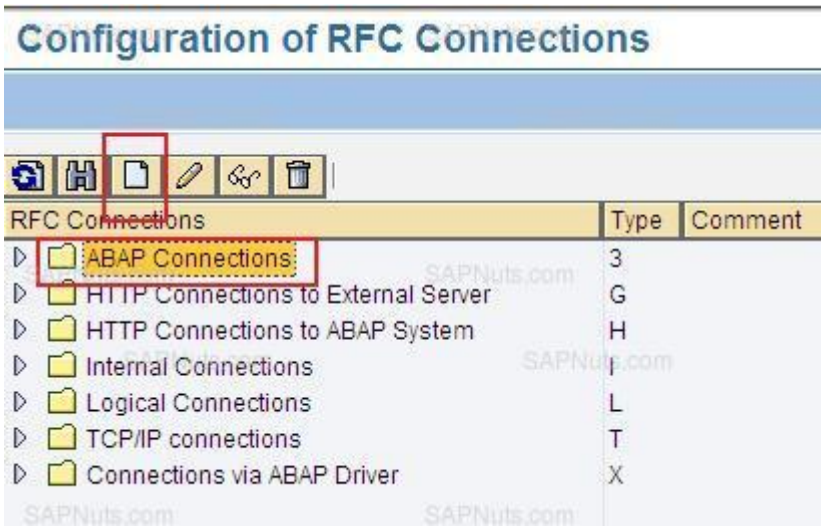



**Check for logical system existence(which we get from 810 client) if it is there, no need to add, if it is not there add it.**

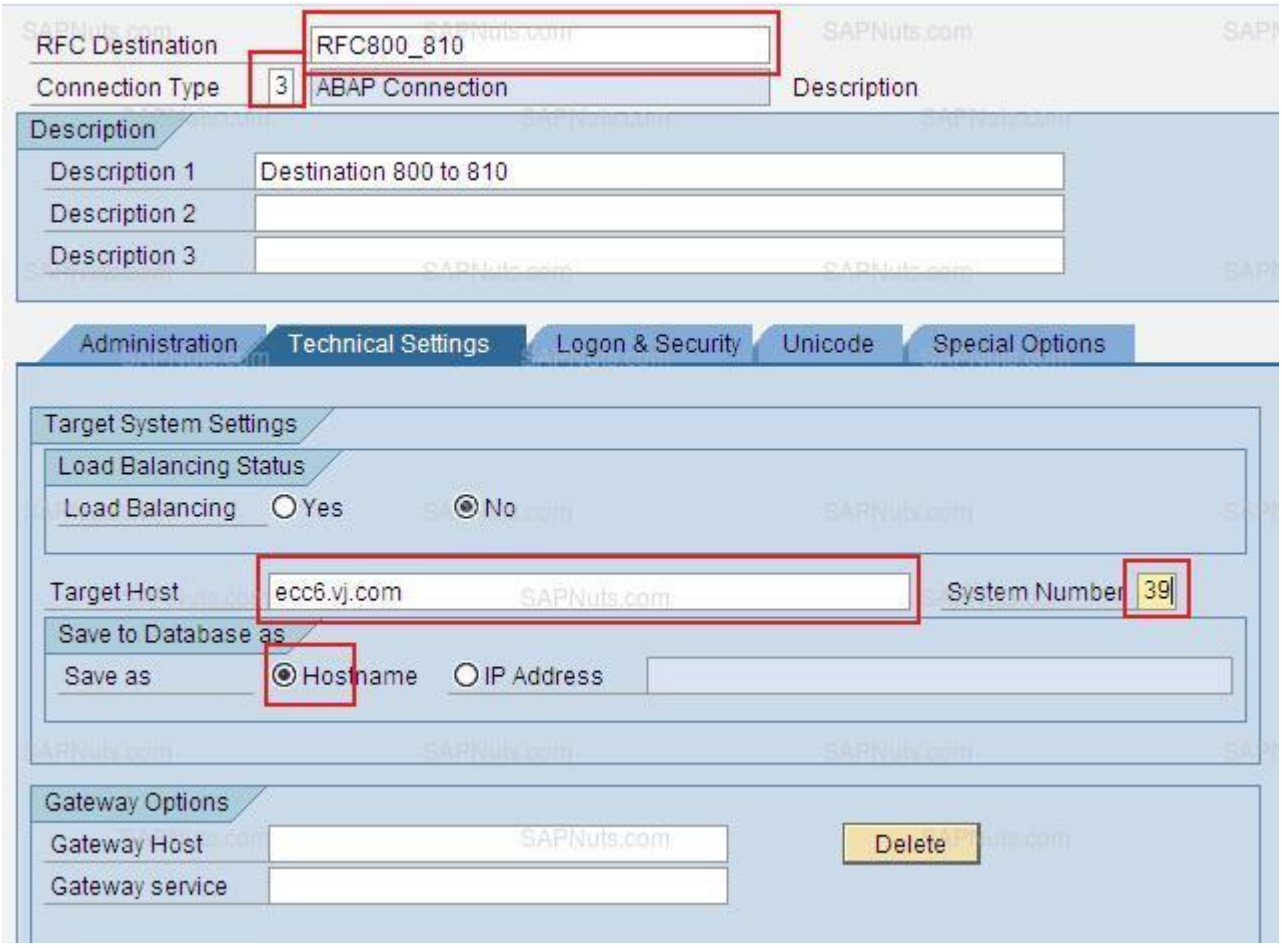
7. **<u>Assign logical system to client.</u>For now skip this.**

8. **<u>Create RFC destination.</u>**

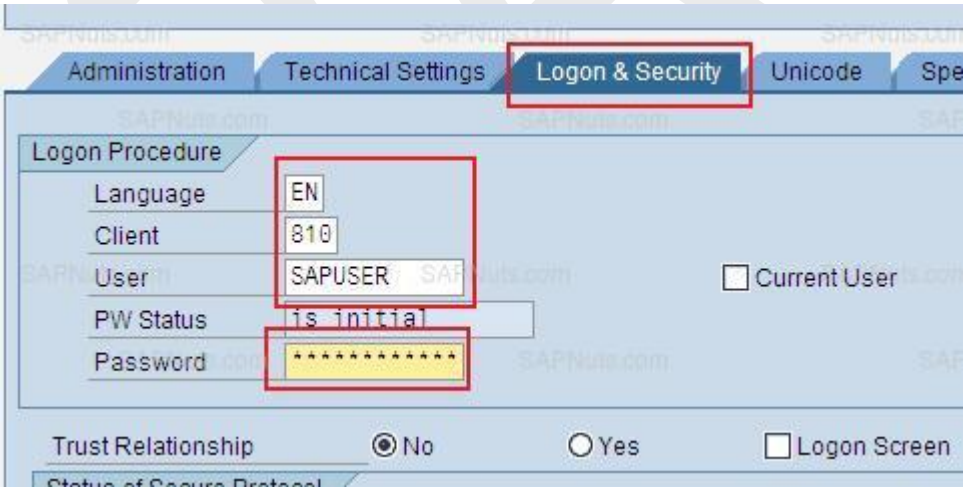
   **Go to T-code SM59, select ABAP connections and click on create.**



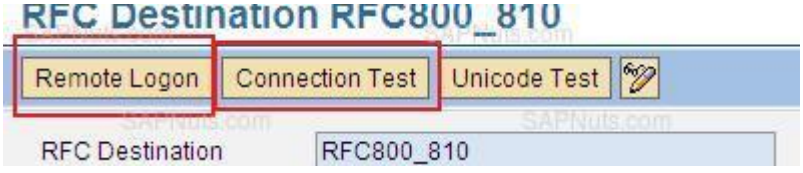

   **Provide RFC destination, description, hostname and system number.**

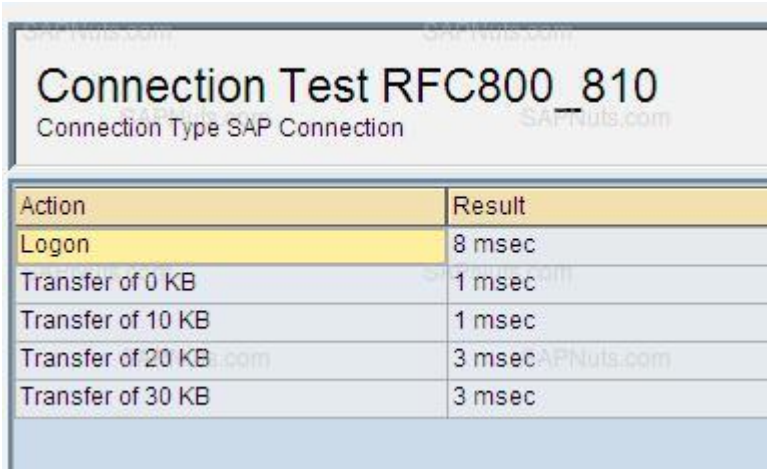



**Click on logon and security tab, provide client, user name, password and save.**

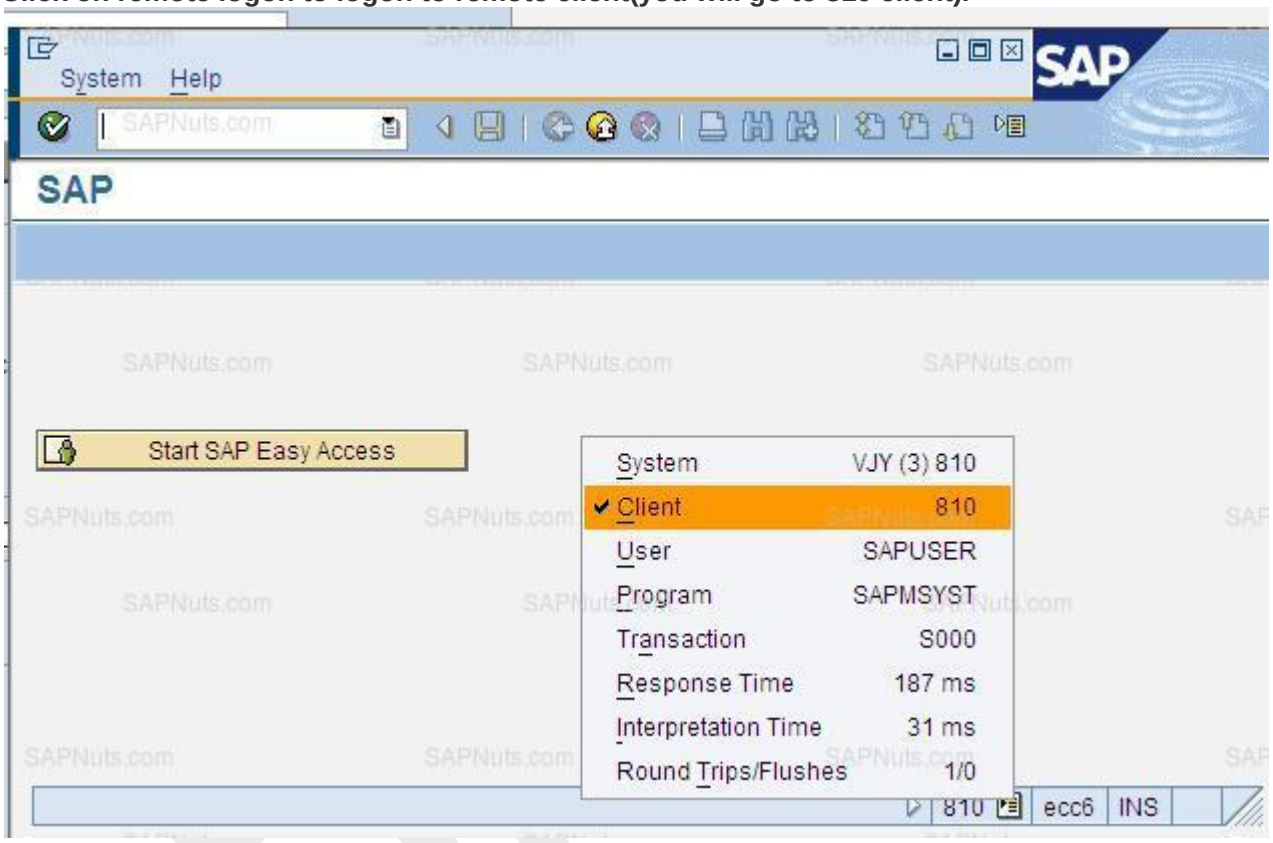



**Click on connection test to test connection.**



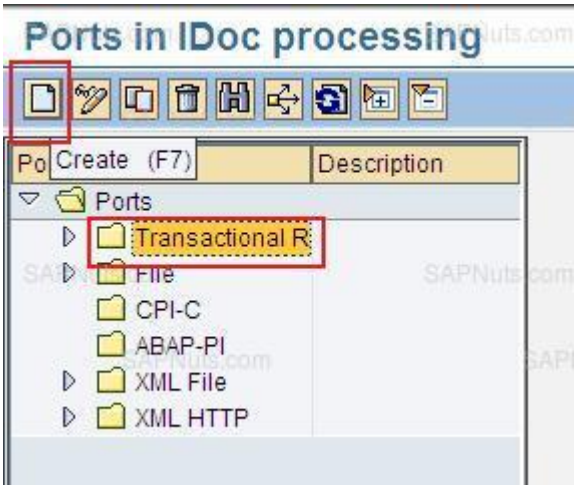



**Click on remote logon to logon to remote client(you will go to 810 client).**

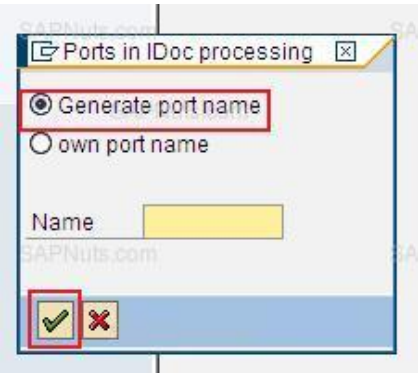



**Step9: Create PORT in WE21.**

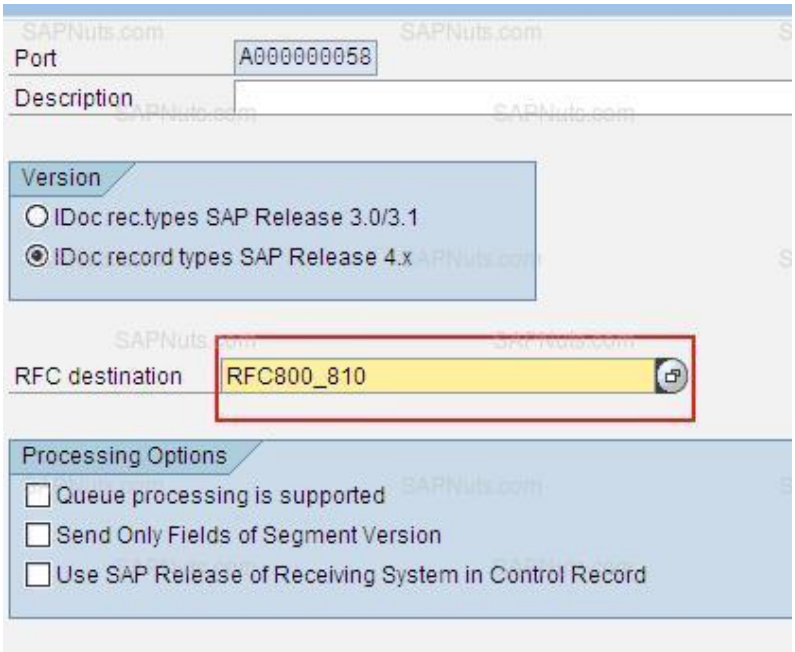
**Go to t-code WE21, select transactional RFC and click on create.**



**Select generate port name and enter.**



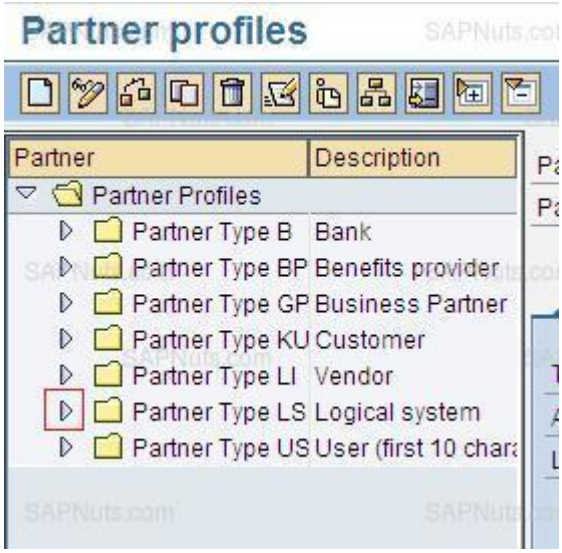

**Provide RFC destination, which we created in step8.**



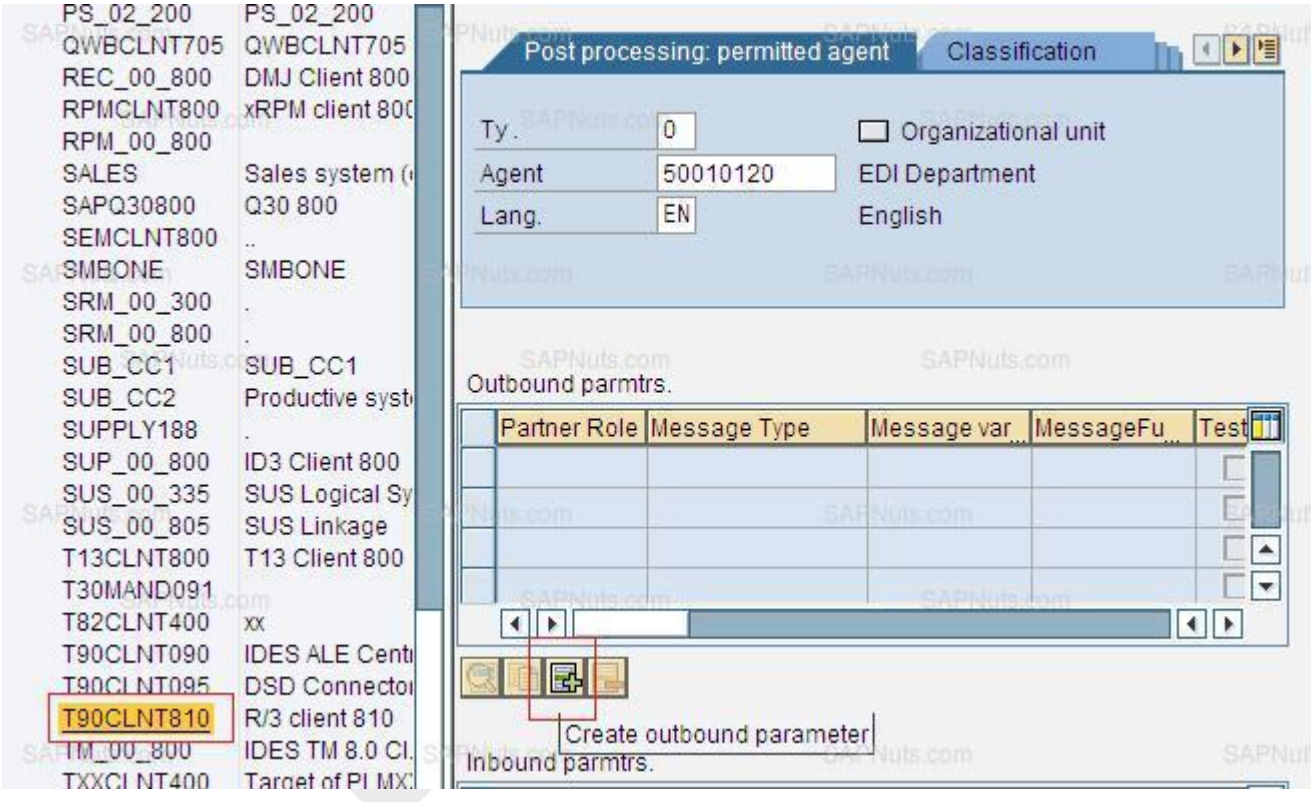

**Save.**

**10.<u>Check partner profiles in WE20.</u>**

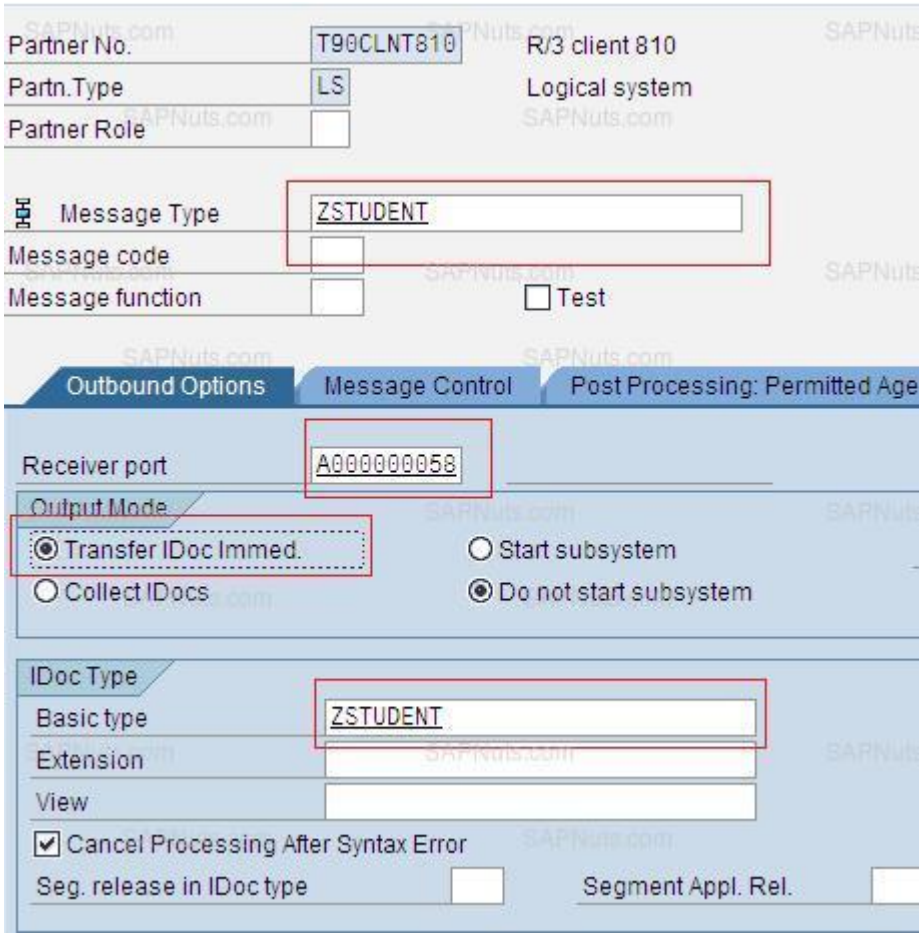
**Go to WE20, expand partner type logical system and search for logical system (Get logical system from t-code SALE ->Basic Settings ->Logical Systems -> assign logical system to client -> 810 (double click and copy logical system)).**




**If logical system is already available in partner profiles in WE20, we don`t need to create partner profile again instead we will add message type at out bound parameters level. In my case, logical system is already available in partner profiles so I will add ZSTUDENT message type to it. To add message type, select logical system, click on add icon (see below image).**




**Provide message type, receiver port (which we have created in step9), select transfer IDOC immediately, basic type and save.**




**If logical system is not available in partner profile in WE20, select partner type logical system and click on create.**

**Provide partner no as T90CLNT810 (Get logical system from t-code SALE ->Basic Settings ->Logical Systems -> assign logical system to client -> 810 (double click and copy logical system). Save and add message type at out bound partner level.**

***Steps to be done in 810 (Target System) clients.***

**Continue 1 to 7 steps as same as 800 client.**

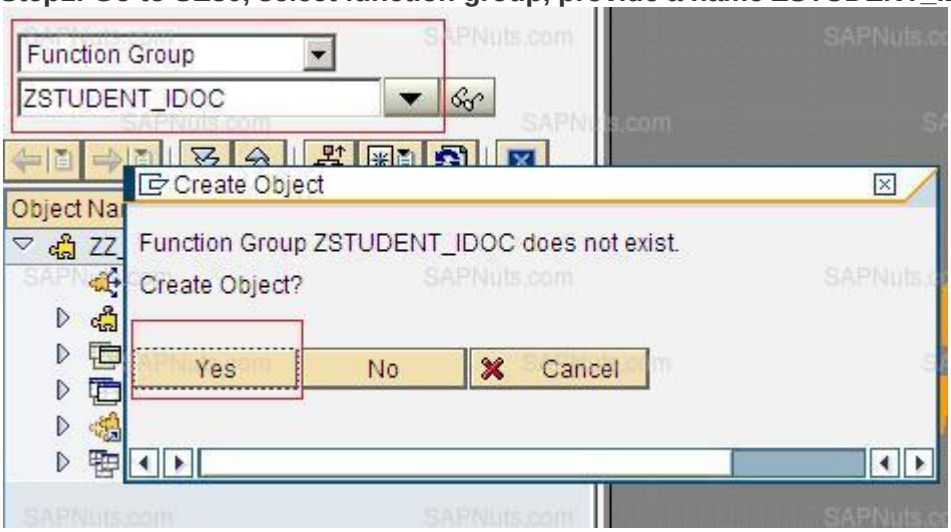
1. **Create a table (client dependent).**
2. **Create segments.**
3. **Create IDOC type.**
4. **Create message type.**
5. **Assign message type to IDOC type.**
6. **Define logical system.**
7. **Assign logical system to client.**
8. **Create inbound function module SE37.**

**Every inbound IDOC will have process code, every process code is associated with a inbound process ex: Function Module or Work Flow Task.**

**Follow below steps to create inbound function module.**

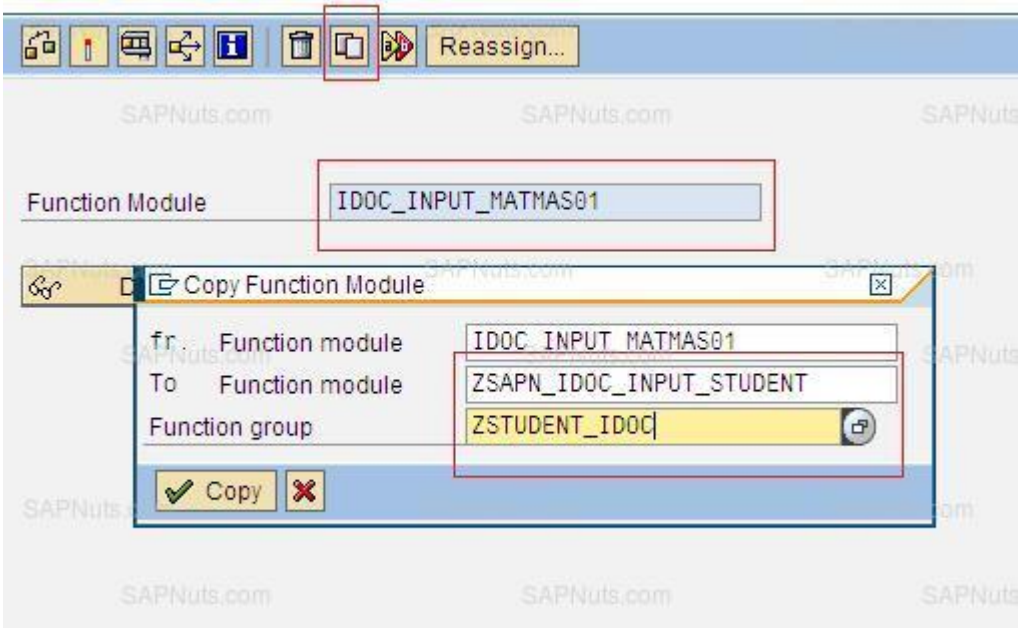
1. **Create a function group in SE80.**
2. **Copy Function module IDOC_INPUT_MATMAS01.**
3. **Add your own code.**

**Step1: Go to SE80, select function group, provide a name ZSTUDENT_IDOC, enter.**

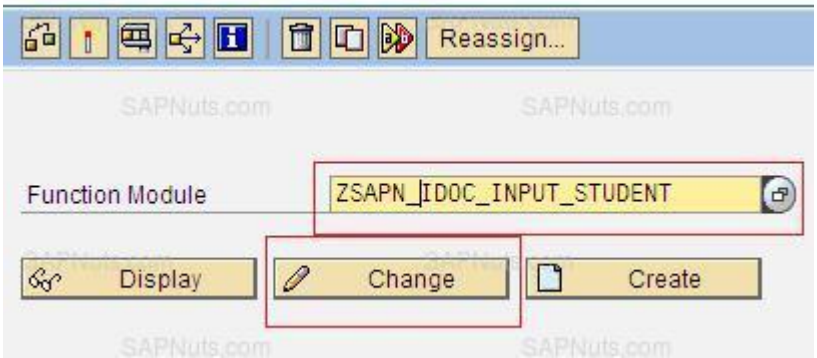



**Save and activate function group.**

**Step2: Go to SE37, provide function module name as IDOC_INPUT_MATMAS01, click on copy, provide function module name as function group and enter.**




**Step3:Add your own code in function module. Go to Se37, provide name as ZSAPN_IDOC_INPUT_STUDENT, and click on change. Remove all code and add below code.**




```
FUNCTION ZSAPN_IDOC_INPUT_STUDENT.
*"----------------------------------------------------------------------
*"*"Local Interface:
```

```
*"  IMPORTING
*"     VALUE(INPUT_METHOD) LIKE  BDWFAP_PAR-INPUTMETHD
*"     VALUE(MASS_PROCESSING) LIKE  BDWFAP_PAR-MASS_PROC
*"     VALUE(NO_APPLICATION_LOG) LIKE  SY-DATAR OPTIONAL
*"     VALUE(MASSSAVEINFOS) LIKE  MASSSAVINF STRUCTURE  MASSSAVINF
*"       OPTIONAL
*"     VALUE(KZ_TEST) LIKE  MDAT1-KZ_TEST DEFAULT SPACE
*"     VALUE(ONLY_MAPPING) LIKE  MDAT1-KZ_TEST DEFAULT SPACE
*"  EXPORTING
*"     VALUE(WORKFLOW_RESULT) LIKE  BDWF_PARAM-RESULT
*"     VALUE(APPLICATION_VARIABLE) LIKE  BDWF_PARAM-APPL_VAR
*"     VALUE(IN_UPDATE_TASK) LIKE  BDWFAP_PAR-UPDATETASK
*"     VALUE(CALL_TRANSACTION_DONE) LIKE  BDWFAP_PAR-CALLTRANS
*"  TABLES
*"      IDOC_CONTRL STRUCTURE  EDIDC
*"      IDOC_DATA STRUCTURE  EDIDD
*"      IDOC_STATUS STRUCTURE  BDIDOCSTAT
*"      RETURN_VARIABLES STRUCTURE  BDWFRETVAR
*"      SERIALIZATION_INFO STRUCTURE  BDI_SER
*"  EXCEPTIONS
*"      WRONG_FUNCTION_CALLED
*"----------------------------------------------------------------------
  SORT IDOC_DATA BY DOCNUM.
  SORT IDOC_CONTRL BY DOCNUM.
DATA : T_EDIDD            TYPE STANDARD TABLE OF EDIDD,             "Data Record
         T_EDIDC            TYPE STANDARD TABLE OF EDIDC,             "Control Record
         T_EDIDS            TYPE STANDARD TABLE OF BDIDOCSTAT.        "Status Record
*----------*Global Work Area Declarations *-----------------------*
DATA : W_EDIDC            LIKE EDIDC,
         W_EDIDD            TYPE EDIDD,
         W_EDIDS            TYPE BDIDOCSTAT.
  DATA: W_IDOC_DATA TYPE EDIDD.
DATA : L_INDX TYPE SYTABIX.
  DATA: IT_STUDENT TYPE TABLE OF ZSTUDENT.
DATA : WA_STUDENT TYPE ZSTUDENT.
DATA : ERROR_MESSAGE TYPE STRING.
DATA : T_STUDENTS TYPE TABLE OF ZSTUDENTS.
DATA : W_STUDENTS TYPE ZSTUDENTS.
DATA : ERROR_FLG TYPE CHAR1.
```

```
* Go through all IDocs
  LOOP AT IDOC_CONTRL.
    CLEAR W_EDIDC.
    W_EDIDC = IDOC_CONTRL.
    READ TABLE IDOC_DATA INTO W_IDOC_DATA WITH KEY DOCNUM = IDOC_CONTRL-DOCNUM BINARY SEARCH.
    IF SY-SUBRC = 0.
      L_INDX = SY-TABIX.
      LOOP AT IDOC_DATA FROM L_INDX.
IF IDOC_DATA-DOCNUM = IDOC_CONTRL-DOCNUM.
APPEND IDOC_DATA TO T_EDIDD.
          CLEAR IDOC_DATA.
        ELSE.
          RETURN.
        ENDIF.
      ENDLOOP.
    ELSE.
      CONTINUE.
    ENDIF.
    SORT T_EDIDD.

    CLEAR W_EDIDD.
    LOOP AT T_EDIDD INTO W_EDIDD.
      IF W_EDIDD-SEGNAM = 'ZSTUDENT'.
        CLEAR WA_STUDENT.
        WA_STUDENT = W_EDIDD-SDATA.
        IF WA_STUDENT IS NOT INITIAL.
          APPEND WA_STUDENT TO IT_STUDENT.
        ENDIF.
        CONTINUE.
      ENDIF.
    ENDLOOP.
    IF IT_STUDENT IS NOT INITIAL. "if data is there in segement
      LOOP AT  IT_STUDENT INTO WA_STUDENT.
        MOVE-CORRESPONDING WA_STUDENT TO W_STUDENTS.
        MODIFY ZSTUDENTS FROM W_STUDENTS. "update data base table
        IF SY-SUBRC <> 0.
          ERROR_FLG = 'X'.
          ERROR_MESSAGE = 'Error occured in updating ZSTUDENT table'.
        ENDIF.
```

```
      ENDLOOP.
    ENDIF.
    IF ERROR_FLG = 'X'. "add error status to IDOC
      IDOC_STATUS-STATUS = '51'.
IDOC_STATUS-DOCNUM = W_EDIDC-DOCNUM.
      IDOC_STATUS-MSGTY  = 'E'.
      IDOC_STATUS-MSGID  = '00'.
      IDOC_STATUS-MSGNO  = '398'.
      IDOC_STATUS-MSGV1  = ERROR_MESSAGE+0(25). "t_return_ecm-message
      IDOC_STATUS-MSGV2  = ERROR_MESSAGE+25(25).
IDOC_STATUS-MSGV3  = ERROR_MESSAGE+50(25).
      IDOC_STATUS-MSGV4  = ERROR_MESSAGE+75(25).
CLEAR ERROR_MESSAGE.
      APPEND IDOC_STATUS TO  IDOC_STATUS .
      CLEAR IDOC_STATUS  .
else. "add success status to IDOC
      IDOC_STATUS-STATUS = '53'.
      IDOC_STATUS-DOCNUM = W_EDIDC-DOCNUM.
      APPEND IDOC_STATUS TO  IDOC_STATUS .
      CLEAR IDOC_STATUS  .
    ENDIF.
  ENDLOOP.
ENDFUNCTION.
```

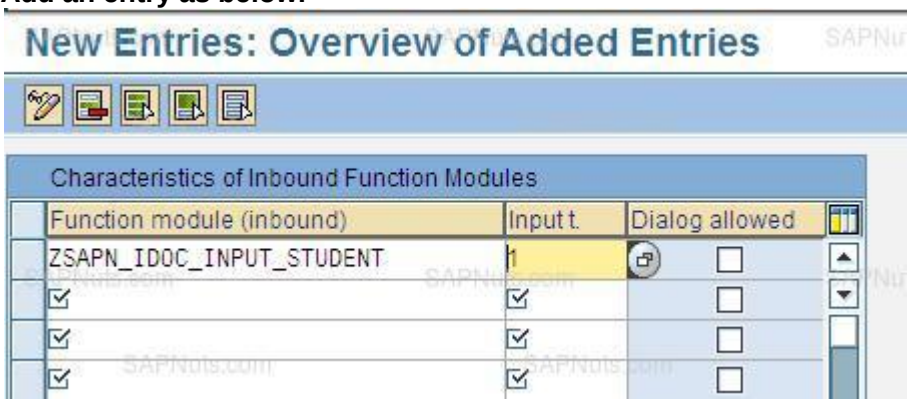
**Save and activate.**

9. **Register function module characteristics.**

**Go to BD51, click on change and click on new entries.**

**IMP Note: If you don`t find change icon or if you find a message like Ã¢â‚¬Ëœ You are not allowed to cross client customizationÃ¢â‚¬â„¢ log on to 800 client and do this step.**

**Add an entry as below.**

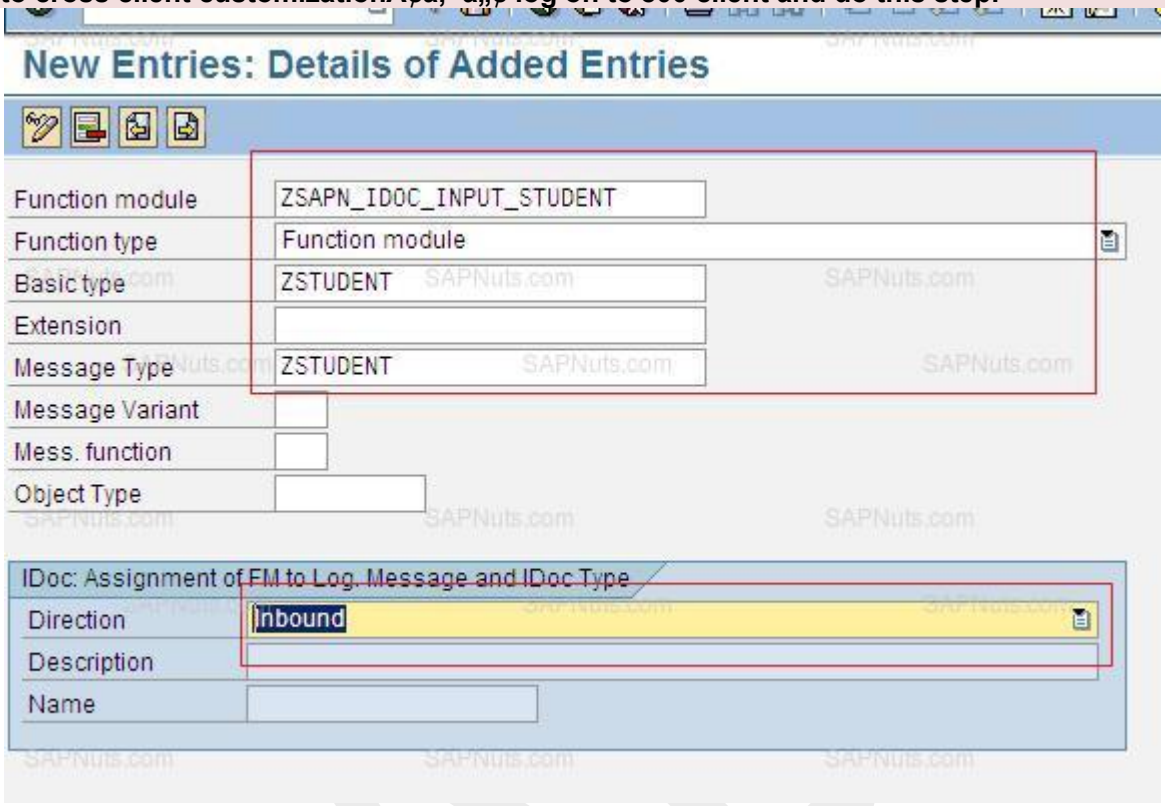


**Save it in a transport request.**

**10. <u>Assign function module to message and IDOC type.</u>**

**Go to WE57, click on change and new entries.**

**IMP Note: If you don`t find change icon or if you find a message like Ã¢â‚¬Ëœ You are not allowed to cross client customizationÃ¢â‚¬â„¢ log on to 800 client and do this step.**

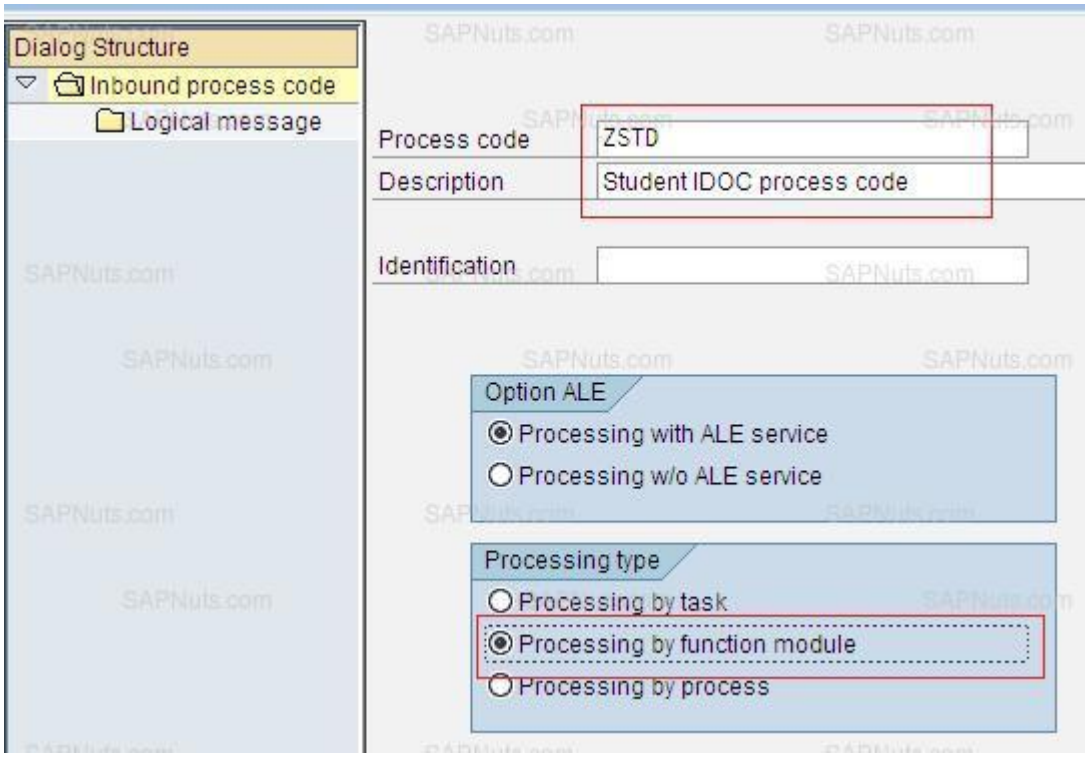



**Save it in a transport request.**

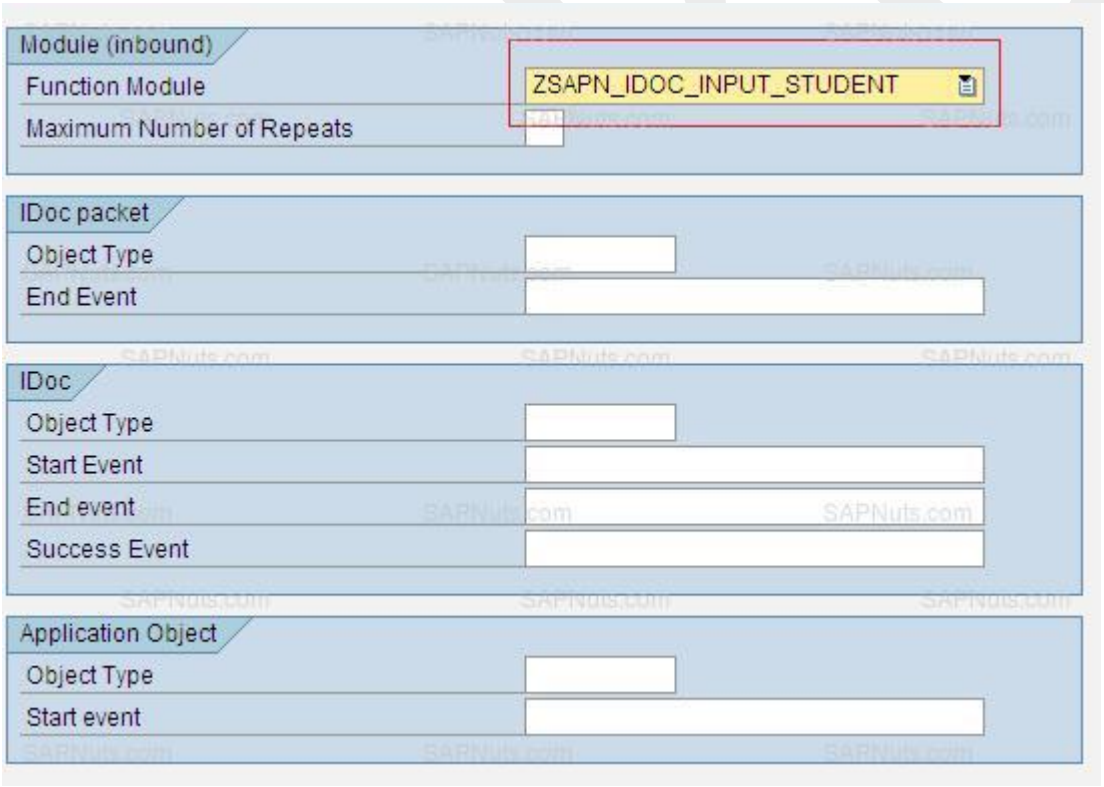
**11. <u>Create process code WE42.</u>**

**Go to WE42, click on change and click on new entries.**

**Provide a process code, descriptions.**



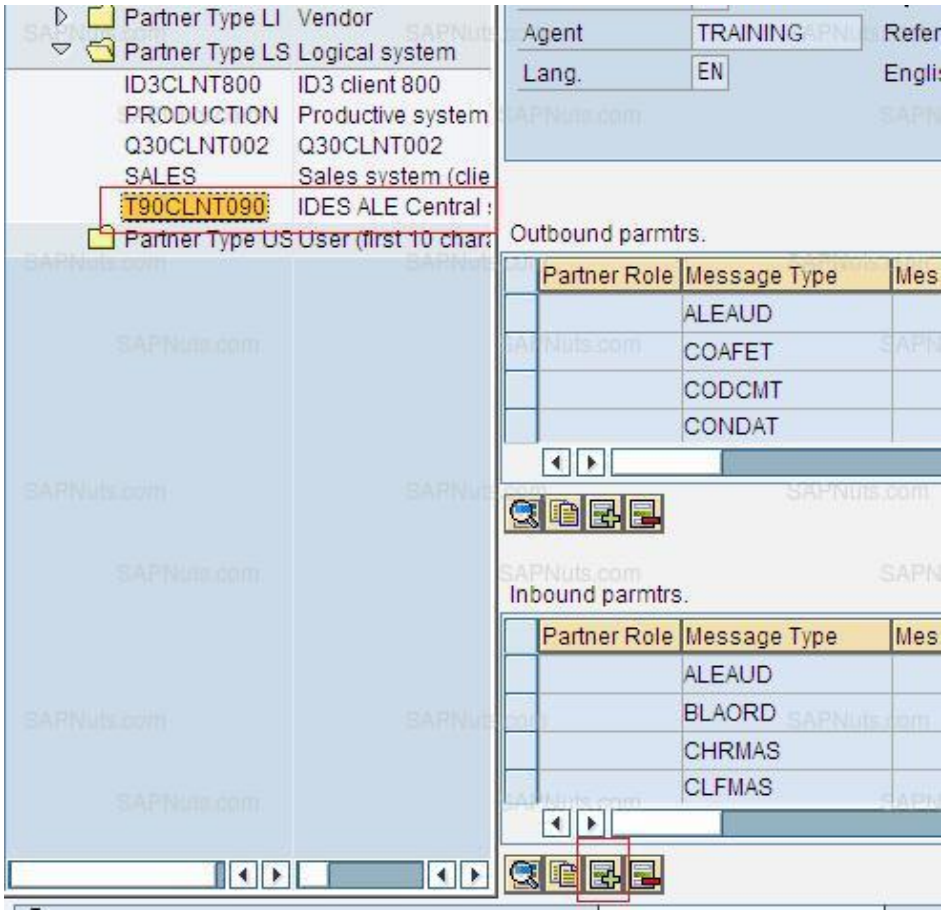

**Click Save, create a transport request. Select function module from drop down and save it in transport request.**



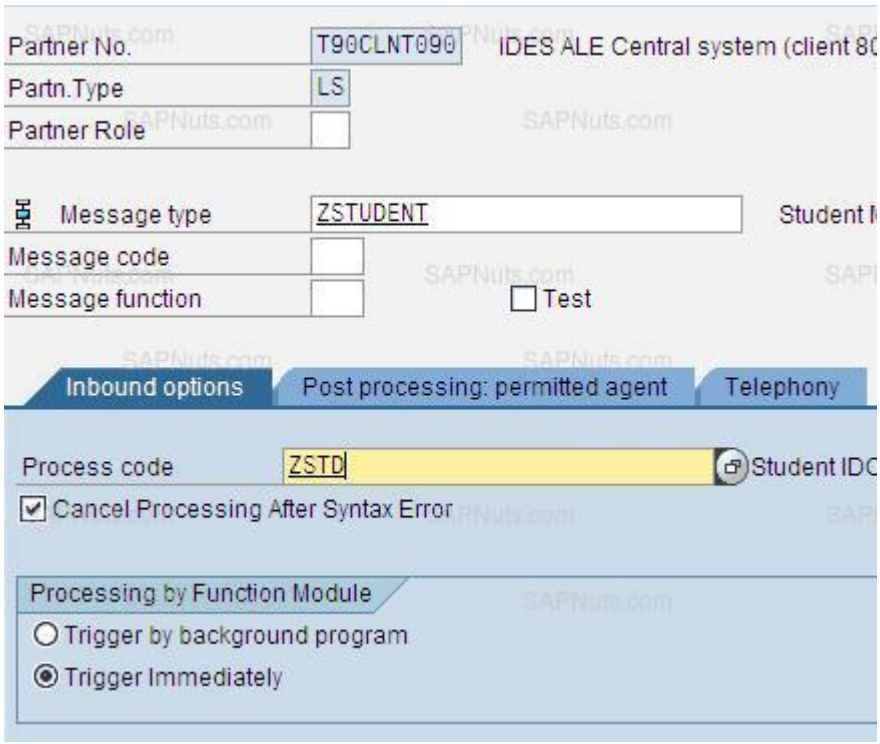

**12. <u>Check partner profiles for inbound.</u>**

**Go to WE20, expand logical system and select logical system(Get logical system from t-code SALE ->Basic Settings ->Logical Systems -> assign logical system to client -> 800 (double click and copy logical system)).**

**Click on add icon under inbound parameters.**




**Add entries as below.**




**Save.**

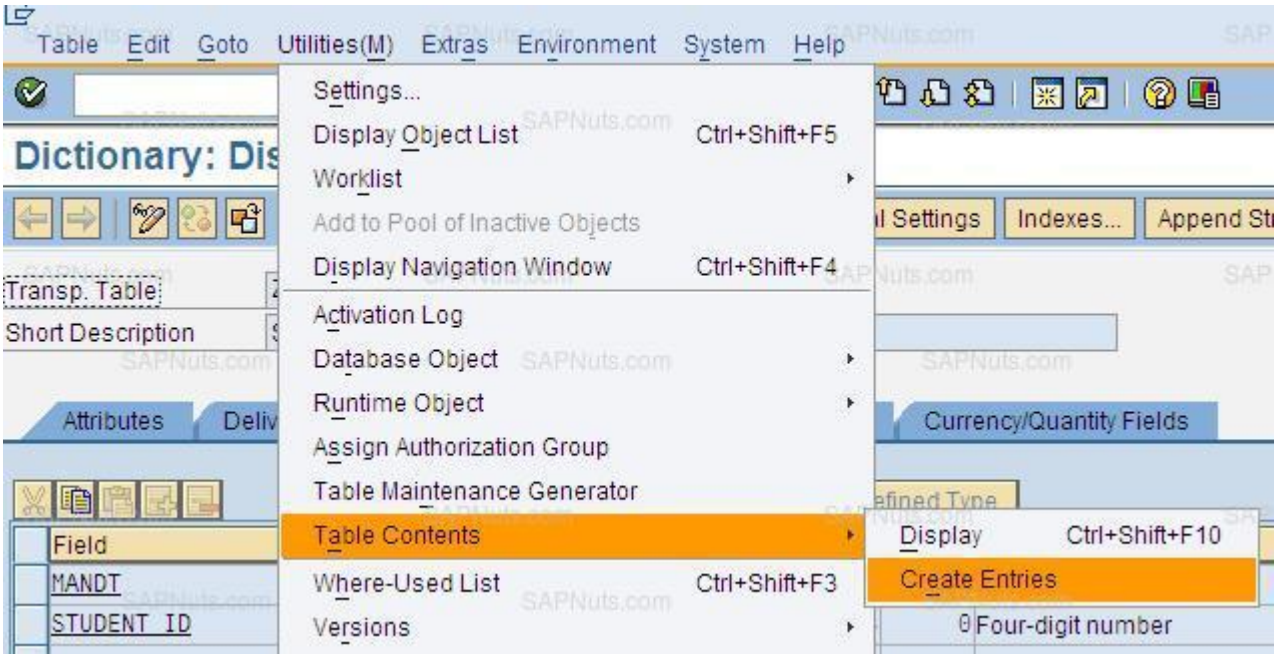
# Testing the development

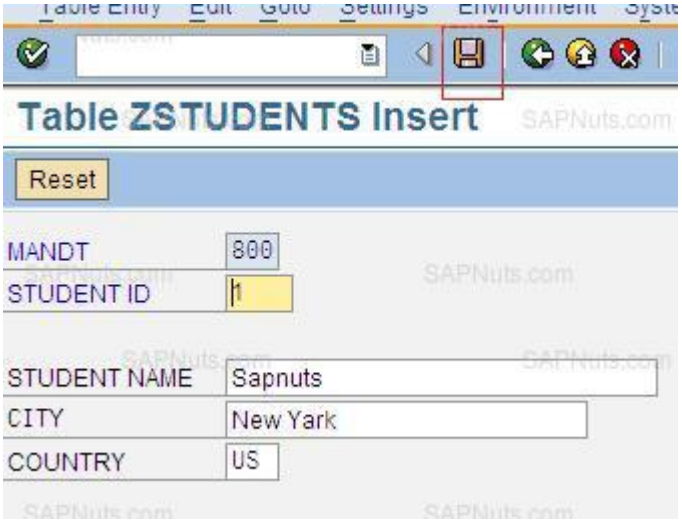
**Follow the below steps to test the above development.**

1. **Create entries in ZSTUDENTS table in 800 client.**
2. **Create outbound program.**
3. **Execute the program and send students from 800 to 810 clients.**

**Create entries in 800 client.**
**Go to SE11, provide name as ZSTUDENTS and click on display.**

**Go to utilities -> table entries -> create entries.**




**Similarly add some more entries.**

**Go to SE38, create a program to send students to 810 client.**

**Create a program with name ZSAPN_SEND_STUDENTS and add below code.**

```
REPORT  ZSEND_STUDENTS.


DATA: ZSTUDENTS TYPE TABLE OF ZSTUDENTS.
```

```
DATA : WA_STUDENTS TYPE ZSTUDENTS.

TYPES : BEGIN OF TY_ZSTUDENT. "STRUCTURE FOR SEGMENT DATA
        INCLUDE STRUCTURE ZSTUDENT.
TYPES : END OF TY_ZSTUDENT.
DATA : ZSTUDENT TYPE TY_ZSTUDENT.
DATA: BEGIN OF T_EDIDD OCCURS 0.
        INCLUDE STRUCTURE EDIDD.
DATA: END OF T_EDIDD.

DATA: BEGIN OF F_EDIDC.
        INCLUDE STRUCTURE EDIDC.
DATA: END OF F_EDIDC.

DATA: BEGIN OF T_EDIDC OCCURS 0.
        INCLUDE STRUCTURE EDIDC.
DATA: END OF T_EDIDC.
SELECT-OPTIONS: S_STD FOR WA_STUDENTS-STUDENT_ID.

START-OF-SELECTION.
  SELECT * FROM ZSTUDENTS INTO TABLE ZSTUDENTS WHERE STUDENT_ID IN S_STD.
  LOOP AT ZSTUDENTS INTO WA_STUDENTS. "send students on eby one
    MOVE-CORRESPONDING WA_STUDENTS TO ZSTUDENT.
    CLEAR T_EDIDD.
    T_EDIDD-SEGNAM = 'ZSTUDENT'. "segment name
    T_EDIDD-SDATA  = ZSTUDENT. "IDOC data record
    APPEND T_EDIDD.
* Fill control record
    CLEAR F_EDIDC.
    F_EDIDC-MESTYP = 'ZSTUDENT'.           "Message type
    F_EDIDC-DOCTYP = 'ZSTUDENT'.         "IDOC type
    F_EDIDC-RCVPRT = 'LS'.               "Partner type
    F_EDIDC-RCVPRN = 'T90CLNT810'.         "Receiver partner



    CALL FUNCTION 'MASTER_IDOC_DISTRIBUTE'
      EXPORTING
        MASTER_IDOC_CONTROL            = F_EDIDC  ">>>>>
      TABLES
```

```
        COMMUNICATION_IDOC_CONTROL     = T_EDIDC  "<<<<<
MASTER_IDOC_DATA               = T_EDIDD  ">>>>>
      EXCEPTIONS
        ERROR_IN_IDOC_CONTROL          = 1
        ERROR_WRITING_IDOC_STATUS      = 2
        ERROR_IN_IDOC_DATA             = 3
        SENDING_LOGICAL_SYSTEM_UNKNOWN = 4
        OTHERS                         = 5.


    COMMIT WORK.
CLEAR : WA_STUDENTS, ZSTUDENT.
REFRESH : T_EDIDD.
  ENDLOOP.
```

**Execute the above program and provide students ids to send.**

**Now go to WE09 and check status in 800 client.**

**Go to 810 client and check table ZSTUDENTS for entries(entries should be created).**

# PROJECT

# PROJECT

## 1) What is System Landscape?

**Ans: Landscape is an arrangement of servers.**

a) **Development system: It is an environment where all the developments, configurations & Changes are done. This system contains at least 2 clients & an optional sand box client.**
   - **Development client: It is used to develop the objects, change the existing objects & maintain configuration changes.**
   - **Unit test client: It is used to perform unit test for preparing Unit Test Document.**

**Sandbox system: It is an environment to develop the objects, make the changes & maintain configurations atinitial stages of the projects. All the consultants are authorized for any transaction code in this system. This system is an optional system.**

b) **Quality system: It is an environment where all the developments, configurations & changes are tested without any errors by the functional, technical consultants & end users.**

c) **Production system: It is an environment where all the end uses work. It is the live system Where all the original data is maintained. This system contains a single client.**

## 2) What is Package & Transportable Objects?

- **Package: It is a collection of transportable objects. The package is used to save all the transportable objects.**
- **Transportable Objects: They must be saved under the required package. The system generates the unique Transport Request number while saving the objects under the package. The transportable objects can be released from one system to another system.**

## 3) Explain about Correction and Transportation System (CTS) or Transport Organizer?

**Ans :- This tool is used to record the changes made in the system.**

- **It is also used to release the Transport Request (TR) from one system to another system (Dev --->Quality ---->Production).**
- **When the change request is created either manually or automatically,the system assigns anumber to it automatically and this number is known as change request number.The change request records all modifications made to development object.When the changes have been made and the change tasks have been released, the changerequest can be released.**
- **The T-Codes SE01/SE09/SE10 is used to work with this tool (CTS).**

**Note: The T-code SE03 is used to find TR of objects**

**The T-code STMS is used to Import TR's**

- **This documentation provides you with an overview of how to manage changes with the CTS and essential information on setting up your system and client landscape and deciding on a transport strategy. Read and follow this documentation when planning your development project**

**4) What are the differences between SE01, SE09 and SE10?**

**Ans : SE01 - Correction & Transport Organizer**

**SE09 - Workbench Organizer**

**SE10 - Customizing Organizer**

**5) How to import the Client dependent/Script changes/Table entries into the Unit Test system?**

**Ans :- Using the SCC1 in the test client (220)**

- **Enter the TR no**
- **The source client no (200) is appeared by default**
- **Check the check box Including Request Sub Tasks**
- **Click on Start Immediately button & press Enter button.**

**6) What are types of Transport Requests (TR) and Explain about them?**

**Ans: Transport request: - It is a unique request number generated by the system while saving the object under the package name. The TR's are generated in development client only & release them from development client. The TR no is starts with development system id followed by K-series. Ex: <R3D>k9<****>**

**The Transport requests of 4 types.**

- **Customizing Requests: These are used to record & release the configuration changes in the system.**
- **Work bench requests: They are used to record the changes/workbench objects such as DDIC objects (Tables, Structures, Transaction codes), Programs, Function modules etc& release them from one system to other.**
- **Transport of Copies: They are created to transport the configurations, developments & changes from one system into another system.**
- **Relocations: They are used to change the objects from one package to another package.**

**7) How to generate Transport Request (TR)?**

**Ans:**

- **Enter the package name at the time of saving with objects Ex: ZDEV/ZSD/ZMM…**
- **Click on save icon**
- **Click on create request icon**
- **Enter short description as per the project standards Ex: <CR547895 – Sales Report>**
- **Click on save icon & click on yes button**

**8) How to change short description of TR?**

**Ans: Once the transport requests are displayed in transport organizer,**

- **Double click on the required TR -> click on properties tab**
- **Click on display/change**
- **Change short description as per the project standards -> click on save & back**

**9) How to release Transport Request?**

**Ans:**

- **Execute the T-code: SE09/SE10/SE01.**
- **Check all the check boxes**
  - **Customizing requests**
  - **Workbench requests**
  - **Modifiable**
  - **Released**
- **Click on display button.**
- **Expand the required TR no under modifiable status.**
- **Place your cursor on the task number.**
- **Click on release directly or F9.**

**Note:**

- **The objects under the TR must be activated before the transport requests are released.**
- **The tick marks are appeared beside the task numbers once they are released.**
- **Place a cursor on the TR number after all the corresponding tasks are releases.**

**10) How to delete Transport Request/Object from TR?**

**Ans:**

- **Execute the T-code SE09/SE10/SE01**
- **Click on display button.**
- **Expand the required TR under modifiable status**
- **Expand the corresponding task number -> right click on object name -> click on delete -> click on yes button -> click on yes button.**

**Note: Follow the similar steps to remove any number of objects from the task/TR.**

- **Place the cursor on the task number after all the corresponding objects are deleted.**
- **Click on delete & click on yes button**
- **Place the cursor on TR after all the corresponding tasks are deleted**
- **Click on delete icon -> click on yes button.**

**11) How to change package name of an object?**

- **Execute the required T-code : SE38, SE11 Etc**
- **Click on 'Go to' from menu**
- **Click on 'object directory entry'**
- **Click on display/change icon**
- **Change the package number Ex: $TMP -> ZPACK**
- **Create the TR by pressing create request**
- **Click on tick (yes)**

**12) What is Technical Specification (TS)?**

**Ans: It is a document prepared by technical consultant (ABAPer).This contains all the technical details such as the Technical solution for the requirement. The detailed TS contains all the details**

**such as starting from designing the Selection screen, Declarations, all the Function Modules used & the processing logic to meet the customer requirements, unit test cases etc.**

**13) Roles & responsibilities for support project?**

**Ans:Implementation Process**

- **Getting the business Requirement document from functional consultant / functional analyst.**
- **Analyze the business Requirement.**
- **Analyze the estimated time for development.**
- **Development of Object (ABAP application).**
- **Unit Testing by Developer.**
- **Releasing the object to testing environment.**
- **Prepare technical document of the development.**

**Support Process**

- **Receiving tickets/issues from business.**
- **Analyzing the ticket.**
- **Resolving the issue in time.**

- **Developer testing and transporting.**

**14) What are the performance tips do you follow while developing an object? (Or)**

**What are the performance tips do you follow to improve the performance in the given program?**

**Ans: 1) Don't use Select - End selectloop (Normal search)since it hits the database for each record & Use INTO table statement (Pointer search) to extract all the required data into an internal tab in single step.**

**2) Don't use select * while extracting data since it extracts all fields & Use select followed by the required fields in the similar order of the database tables.**

**3) The internal Table must be must sort internal tables before using control break statements.**

**4) The internal Table must be sorted internal tables before using binary search addition with READ table statement for faster access.**

**5) Copy and delete internal tables/ db tables outside of the loop.**

**6) The Base internal table must be checked whether it has data or not before using the FOR ALL ENTRIES Statement otherwise Select statement extracts all the records form the database without considering FOR ALL ENTRIES.**

**7) The internal Table must be SORTED before DELETE ADJUSCENT DUPLICATES statement.**

**8) The TABLES statement is not advisable since it creates lot of implicit Work Areas.**

**9) Prefer Parallel Cursor method in case of Nested Loops.**

**10) Secondary Indexes are not advisable since the data is filtered at data base level for all the users.**

11) Up to 2 tables can be joined then use For All Entries while extracting the data from the next required tables.

12) For All Entries with Inner Joins are advisable since For All Entries are another extra filter criteria as like Select-Options.

13) Don't use Select statements within the Loops. Use For All Entries outside of the loop statements then use Read Table statement with Loop statements.

**15) What is a Functional Specification?**

**Ans: Functional Specification is the official document for a developer which describes the features and design of expected functionality. It contains the technical expectations and required data for the design and development. The FS is also called as GAP. This can be prepared only after discussing with the end users & understand their requirements & document the End users/ clients requirement.**

**16) How do you get functional specs when you are assigned some object?**

**Ans: Generally Functional Specifications will be sending through emails(In support process, they may come in a specific tool like: Remedy, Radix).**

**17) How do you receive Functional Specifications?**

**Ans: In general a functional specification will be sent to a consultant through organizational Email like : Outlook, IBM Lotus etc.**

**Functions of Functional Specifications:**

- **Requirement.**
- **Design Summary.**
- **Desired Functionality.**
- **Finished product.**
- **Test Cases.**

**Functional Specification format/contents may differ from company to company, every company follow their own formatting style.**

**18) Who prepares Functional Specifications?**

**Ans: As per standard Organizational structure, functional consultants will prepare functional specifications, in some cases Techno-Functional consultants will prepare them. (In some companies Senior-Technical Consultants may prepare).**

**19) What is the next step after Functional Specifications?**

**Ans: Once Functional Specification is prepared, the document will be sent for approval to functional head or project manager, once approved, it will send to technical lead. The technical lead will distribute the Functional Specification to any developer.**

**20) Client concept: If there are two clients in development server, Ex: client 200 for development and client300 for configurations, what does it actually mean?**

**Ans: Even if there are two clients 200 and 300, 200 for development and 300 is for configuration, you cannot access 200 client data in 300 client.**

**In real-time, functional people need accurate data to test business configuration(SD/MM/FI etc), when it comes to technical people, they may create junk data to test their application(Creating material, junk master data, junk transaction data etc), so they maintain one separate client(200) for development activities and separate client for configurations.**

**Note: Transport path will be same for two clients (200 - QA) and (300 - QA) i.e.; same QA client.**

## 21) Explain SD (Sales & Distribution) Flow?

| Flow | T-Codes | Tables |
|---|---|---|
| Inquiry | VA11, VA12, VA13 | VBAK, VBAP |
| Quotation | VA21, VA22, VA23 | VBAK, VBAP |
| Purchase Order | ME21N, ME22N, ME23N | EKKO, EKPO |
| Sales Order | VA01, VA02, VA03 | VBAK, VBAP |
| Delivery | VL01, VL02, VL03 | LIKP, LIPS |
| Shipping | VT01 | VTTK, VTTP |
| Billing | VF01, VF02, VF03 | VBRK, VBRP |
| Invoice | VF21 | VBRK, VBRP |

- Customer enquires about the Products services that were sold by a company.
- Company Gives a Quotation for the products and Services to a Customer
- Based on purchase order sales order will be created.
- Once sales order is created, delivery will be done
- Next the goods should be delivered through shipping
- Every bill has invoice...next provide invoice.
- Invoice end of sales flow.

## 22) MM(Material Management) Flow?

**Ans:**

| Flow | T-Codes | Tables |
|---|---|---|
| Purchase Requisition | ME51, ME52, ME53 | EKPO, EINE |
| Request For Quotation | ME41, ME42, ME43 | EKKO, EKPO |
| Quotation | ME47 | EINE, EINA |
| Price Comparison | ME49 | EKKO, EKPO |
| Purchase Order (PO) | ME21, ME22, ME23 | EKKO, EKPO |
| Goods Receipts (GR) | MIGO | EKKO, EKPO |
| Invoice Verification | MIRO | EKKO, EKPO |

## 23) Useful tools in projects?

**Ans: The below 3rd party security gateway tools are used to connect into client desktop/SAP system**

- **VPN/Cisco VPN – Virtual Private Network is used to connect into client SAP systems.**
- **Citrix – It is used to connect into client system desktop**

## 24) Useful tools for Ticket Handling?

**Ans: These tools are used in support & maintenance project to maintain Change requests/Tickets.**

- **Remedy**
- **SAP IMS (Incident management System)**

- **Lotus – IBM**
- **SMS - Accenture**

**Note : The Data migration toll 'Panaya' is used for migration/up gradation projects**

- **The task/work is assigned through mails (M.S Outlook) by Team Lead/Functional Consultant.**

**25) What are the performance tools do you perform before the object is released (or) once the development is completed?**

**Ans :- They are used to release the objects without errors , warnings and with better performance.**

**1. Extended program check (SLIN or program -->Check -->Extended program check).**

**This tool is used to list out the performance related errors, warnings& messages in the program & rectify them.**

**Steps to perform Extended program check:**

- **Execute the t-code – SLIN/ through Menu path**
- **Enter the required program Ex: ZALV_REPORTS**
- **Check the check of 'Character String'**
- **Click on F8 icon then all the corresponding errors, warnings, messages are listed.**

**2.Code Inspector (SCI or program -->Check -->Code Inspector).**

**This tool is used to perform different types of checks such as Performance checks, Security checks, and Syntax checks.**

**Steps to perform Code Inspector:**

- **Enter the required program name in SE38**
- **Click on program menu item -> Check -> Code inspector**
- **Expand appropriate checks such as Syntax checks/Performance checks/Security checks.**
- **Read & understand the system given messages & rectify them**

**3. Runtime Analysis (SE30 or System-->Utilities-->Runtime Analysis).**

**This tool is used to know Execution time or run time of the program/T code. It can also be used to know performance tips & tricks.**

**Steps to perform Runtime Analysis:**

- **Execute the T-code :SE30**
- **Enter program/T-code/Function Module**
- **Click on execute button**
- **Fill the required fields**
- **Click on execute icon**
- **Click on 'Back'**
- **Click on "Evaluate" button in runtime Analysis screen**

**4. SQL Tracer (ST05 or System → Utilities → Performance trace).**

**This tool is used to know the indexes either primary/secondary which are consider while extracting the data.**