SAP Business Object Types

1. **Business Object Type - An Introduction**

   Business Object Repository is an object oriented approach used to model Business Objects and processes. It is a collection of Business Object types and Interface types and also their components like attributes, methods and events.

   Business Object type is a representation of any business entity in an SAP system. Example of entity could be an employee, purchase order, etc.

   A BO type includes both the data of the entity and the functions it performs. Data is stored in form of BO attributes and functions in form of BO methods. Thus we can understand BO as an object oriented approach to model business entities and processes.

2. **Role of Business Object types**

   Business Object has the following functions:
   a. BO reduces the complexity of the SAP system by dividing it into smaller units.
   b. At BO level, non-SAP systems can communicate with SAP system. This can be done using BAPIs.
   c. BO enables various business components to communicate with each other.
   d. BO types acts as entry points to data and functions for an SAP System.
   e. BO's can be used in business workflows, ALE and RFC.

3. **Business Object Properties**

   Each business object has the following properties:
   a. **Object Type ID** - It is used for internal identification of the business object.
   b. **Object Name** - It is used for external identification of the business object. Using this property external system can interact with SAP system.
   c. **Super Type** - It is the parent object from which all the components of the BO are inherited.
   d. **Name** - It is a descriptive name of the BO.
   e. **Description** - It is a brief description of the business object type.
   f. **Application** - The application component to which the BO belongs to.
   g. **Program** - Program that will contain the ABAP code of the BO.

4. **Components of a Business Object**

   There are 5 components of a business object:
   a. **Key Fields**
      - Can be a combination of one or more fields
      - Is a specification of unique object key - to access a specific instance of a BO
      - Is a reference to underlying application table
      - Must be character based
      - Value set when creating a BO instance or by a method
      - Can be read from outside. Cannot be changed from outside
      - Maximum length of concatenated key field can be 70 characters
   b. **Attributes**
      - Properties and characteristics of a BO
      - Can be read from outside. Cannot be changed from outside
      - Can be single-line or multi-line
      - Can be instance independent (static) or instance dependent
   c. **Events**
      - Indicates the occurrence of a status change of a BO
      - Can have parameters and exceptions like methods
      - Have listeners outside the BO. E.g. Workflow
   d. **Methods**
      - These are functions that encapsulates the functionalities of a BO
      - Can be synchronous or asynchronous
      - Can be instance independent or instance dependent
      - Can have parameters (mandatory/optional, single/multi-line, import/export/both) and exceptions
      - The ABAP implementation can be:
        - A function module
        - A BAPI
        - A transaction
• A dialog module (Obsolete)
• A report
• Other ABAP forms (Direct implementation, ABAP class methods, system program etc.)

Types of methods:
• Synchronous - Result is returned directly to the calling program.
• Asynchronous - Result can be returned to the calling program only through an event.

e. Interface Types
Ø Like models for BO Types
Ø Do not have any key fields
Ø Cannot be instantiated
Ø Do not have development statuses
Ø Can be inherited - substitute for multiple inheritance
Ø Can have implementation
Ø A BO Type can inherit one or more interfaces
Ø Methods defined in an interface can be redefined in a BO Type
Ø The interface IFSAP and its methods ExistenceCheck and Display are inherited by all BO Types by default

5. Business Object Statutes

The following are the different development statuses of a BO/BO Component:
a. Modeled - Only defined (the method and its signature)
b. Implemented - ABAP implementation is done, can be used for testing or internal use
c. Released - Method ready to be used
d. Obsolete - Attribute / Method / Event not to be used anymore

You can check the status of the Business Object by clicking on the 'Basic data' button (cap sign). It lists the details of the created / open BO with the details of the super type, status, name, description and transport data, etc.

6. Creating a Business Object

Business Object Builder is the tool used for creating, implementing and testing business object types. The menu path to access it is:
Transaction 'SWO1' can also be used for the same.
a. Go to transaction SWO1 (Business Object Builder).

b. Input the name of the Z-business object and click on 'Create'.
c. Select the 'Supertype'.

d. Enter details of properties of the business object like object type, object name, description, program name and application.
e. Click on the Check Button.
f. BO would get created and the default structure would be as below:

[Image of a default structure]

g. Go to the 'Basic Data' button or click Ctrl+Shift+F1 to view the basic data of the BO.

[h. In the 'Methods' tab, create a new method by clicking 'New' button or F5.]
i. Click on 'Parameters' button or click Ctrl+Shift+F7 to add parameters to the method.

j. Click on 'Program' button or click Ctrl+Shift+F3 to add code to the program. A template of the method would be generated. Add the source code in the method.

k. Save the method and set its release status to 'Implemented'.
l. Change the status of the BO to 'Implemented'.

m. Generate the BO.

n. Change the status of the BO to 'Released' to make it ready for use.
o. To see the details of the exceptions, click on 'Exceptions' button or click Ctrl+Shift+F8.

p. To see the list of errors navigate to Goto -> Error list.