

**BOR & BAPI**

# Agenda

---

**1 Introduction to Business Framework**

**2 Business Objects & BOR**

**3 BAPI**

**4 Advantages of BAPI**

**5 Characteristics of BAPI**

# Agenda (Contd.).

---

**6 Standard BAPI**

**7 Sample Program**

**8 Exercise**

**9 Customized BAPI**

**10 Exercise**

# **Introduction to Business Framework**

# Objectives

---

- On Completing this course, participants will be able to
  - Understand about Business Framework
  - Understand of Business Objects & BAPI
  - Understand the advantages & characteristics of BAPI
  - Work with standard BAPI
  - Work with custom BAPI

# Business Framework

---

- The SAP R/3 Business Framework provides a structure of R/3 functionality based on application components (business components) and object models.
- The Business Framework enables customers and partners to link their own components to the R/3 system.
- The basic components of SAP R/3 Business Framework are:
  - Business Components
  - Business Objects
  - Business Application Programming Interface (BAPI)
  - Integration Service, Application Link Enabling (ALE)
  - Communication Services rfc

# Basic Components of Business Framework

---

- **Business Components:** SAP Business Component provide autonomous business functions and consist of business objects.
- **Business Objects:** Business Objects encapsulate business data and functionality and define the functional scope and boundaries of a Business Component.
- **Business Application Programming Interface:** BAPI along with business objects define and document the interface standard at the business level.

# Basic Components of Business Framework

---

- **Application Link Enabling:** The ALE Integration service enables the integration of business processes that are carried out in different R/3 and non-sap systems.
- **Communication Services:** These are communication technologies like Remote function call (RFC) ,which use business framework to access BAPI.



# **Business Objects & BOR**

# Business Objects

---

- Business Objects encapsulate business data and functionality and define the functional scope and boundaries of a Business Component.
- SAP Business Objects and their BAPI provides an object orientation R/3 Business functionality.      BAPI\_    SE37 FM    RFC.
- SAP has introduced object oriented technologies to make the R/3 process and data is available in the form of business objects.
- Business objects are used to break the SAP system down into smaller ,disjunctive units, improving its structure and reducing its complexity.



# Business Objects

---

- SAP Business Objects encapsulate R/3 data and business processes, thus hiding the details of the structure and implementation of the underlying data.
- The set of methods (BAPI) that is associated with a business object represents the object behavior.
- **Object types** : Each individual business object belongs to a specific object class, depending on nature and general characteristics of the object. These object classes are called object types.
- **Object Instance**: Each individual SAP Business object is a representation, or instance of its object type.

# SAP Business Object Types

---

- **Object Type** : The object type describes the features common to all instances of that object type.
- **Key Fields** : The key fields determine the structure of an identifying key, which allows an application to access a specific instance of the object type.
- **Method** : A method is an operation that can be performed on a business object and that provides access to the object data.
- **Attributes** : An attribute contains data about a business object, thus describing a particular object property.
- **Events**: An event indicates the occurrence of a status change of a business object.

# BOR (Business Object Repository)

---

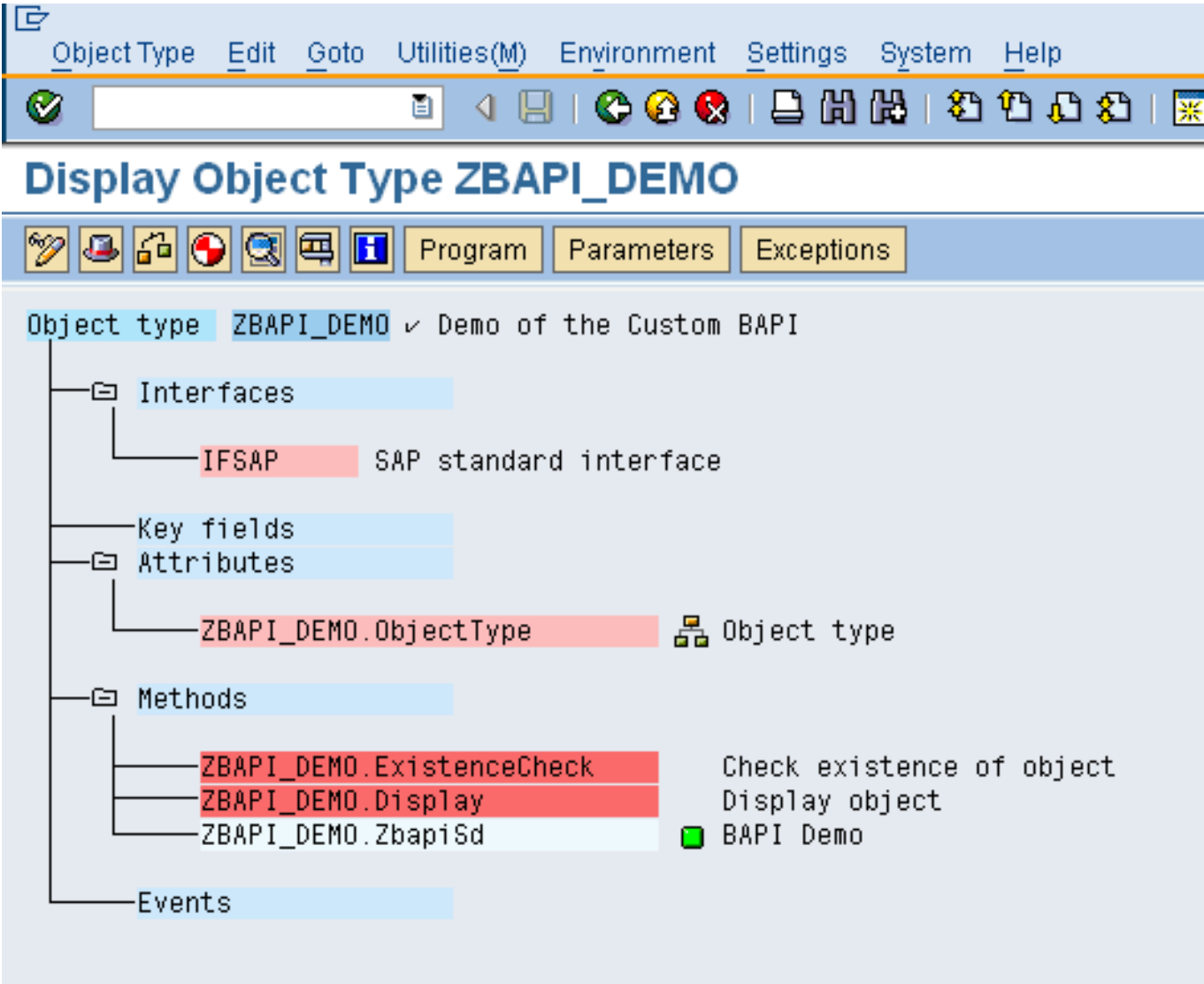
- Business Object Repository.
- Transaction code for BOR is SWO1.( O not Zero )
- It is the central access point for the SAP business object types and their BAPI.
- BOR allows an object-oriented view of all data and processes in an SAP system.

# BOR (Business Object Repository)

---

- This enables to find the required functions easily:
  - Storage of all relevant information
  - It ensures BAPI interface stability
  - It manages BAPI in release updates
  - It creates instances of SAP business objects
- The SAP Business objects held in the business object repository (BOR) encapsulate their data and processes

# BOR (Business Object Repository)





# **BAPI - Business Application Programming Interface**

# BAPI

---

- Business Application Programming Interfaces (BAPI) are standard SAP interfaces.
- The transaction code for BAPI is BAPI .A BAPI is implemented as an RFC enabled function module.
- BAPI are defined as methods for the business object types.
- They play an important role in the technical integration and in the exchange of business data between SAP components & between SAP & non SAP components.
- BAPI 's allow integration at the business level, not the technical level.
- External Applications can access SAP Business objects through standardized, platform independent interfaces called BAPI.

- External access to the data and processes is only possible by means of specific methods – BAPI (Business Application Programming Interface).
- **Definition:** A BAPI is a well defined Interface to process the data of a Business Application System, Implemented as a method of an object in Business Object Repository (BOR).
- The BAPIs in the R/3 system are currently implemented as Function Modules, which are created and managed in the Function Builder (SE37).
- The function modules within a BAPI should:
  - Supports the Remote Function Call (RFC) protocol.
  - Has been assigned as a method to an SAP Business Object in the BOR.
  - Is proceeded without returning any screen dialogs to calling application.

## 19

Function Module   Edit   Goto   Utilities(M)   Environment   System   Help

Function Builder: Display BAPI\_PO\_CREATE1

Function module: BAPI\_PO\_CREATE1   Active

Attributes   Import   Export   Changing   Tables   Exceptions   Source code

**Classification**

Function Group	2012	BAPI in the Enjoy Purchase Order
Short Text	Create Purchase Order	

**Processing Type**

☐ Normal Function Module  
☒ Remote-Enabled Module  
☐ Update Module  
☒ Start immed.  
☐ Immediate Start, No Restart  
☐ Start Delayed  
☐ Coll.run

**General Data**

Person Responsible	SAP
Last Changed By	SAP
Changed on	07.11.2006
Package	ME
Program Name	SAPL2012
INCLUDE Name	L2012U02
Original Language	DE
Released on	06.05.2002

☐ Edit Lock  
☐ Global

## **Advantages of BAPI**

# Advantages of BAPI

---

- The advantages of using BAPI are
  - Business Standard
  - Easy to use and Understood
  - Object Oriented Access
  - Usable from different programming languages
  - Compatible with different Communication Technologies
  - Follows Industry Standards
  - Stability and downwards compatibility

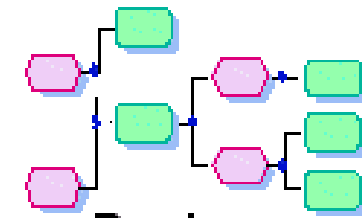
# Benefits of BAPI

---

- Can be used in diverse languages / Development Environments (ABAP, Visual Basic, Java, C++, etc.)
- Can be called from diverse platforms (COM, CORBA, Unix)
- Reduced development cost
- Reduced maintenance cost
- “Best-of-both-worlds” approach
- Rich functionality of the R/3 system
- User-specific front-ends

# Where BAPIs can be used

R/3 satellite systems /  
Distributed scenarios  
(ALE)



Business  
workflow

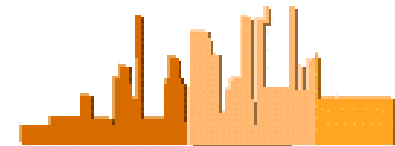
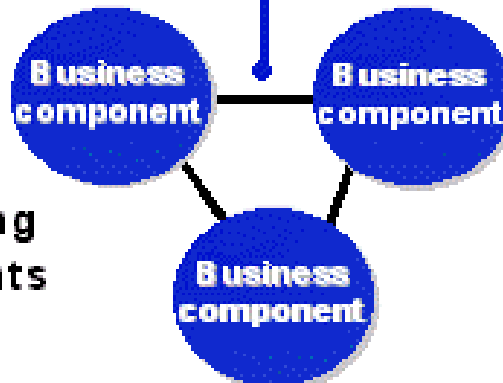


Visual Basic/JAVA...



Internet /  
Intranet

R/3 building  
components



Customer and partner  
development



# **Characteristics of BAPI**

# Characteristics of BAPI

---

- The following are the characteristics of BAPI :
- **Naming Conventions** : <business object>.<BAPI name>.
- **Database Consistency** : Each BAPI that creates an instance of an object or updates the data of an object is responsible for database consistency.
- **No Dialog Orientation** : BAPI do not return dialog screens from the R/3 server system to the calling program.
- **Authorization.** : Any interaction with the R/3 system requires the user to have a certain set of authorizations

# Standard BAPI

# Standard BAPI

---

- Some BAPI and methods provide basic functions and can be used for most SAP business objects. Such BAPI are known as '**Standardized BAPI**'.
- **Features** : With BAPI (methods) ,we can differentiate between instance methods and class methods.
  - Instance Method : Refer precisely to one instance
  - Class Method : Instance independent.
- BAPI for reading data :
  - Getlist (Class Method) : List of object key values.
  - Getdetail (Instance Method) : Uses a key to retrieve details about an instance of a business object and returns this data to the calling program.
  - Getstatus (Instance Method) : The BAPI Getstatus is used to query the status of an SAP Business object.
  - Existence Check (Instance Method) : This checks whether an entry exists for an SAP Business object.

# BAPI for creating and changing data

---

- The following BAPI can create, change or delete instances of a business object :
- Create or createfromdata (Instance Method) : The BAPI create or createfrom data creates an instance of an SAP Business Object.
- Change (Instance Method) : The BAPI changes an existing instance of a SAP Business Object.
- Delete and Undelete (Instance Method) : The BAPI deletes an instance of an SAP Business Objec from the database.
- **BAPI for Mass Processing** : When a BAPI is called ,several busines object instances are processed at the same time.
- With BAPI for Mass Processing 'Multiple' is added to the method name as
  - CreateMultiple
  - ChangeMultiple
  - DeleteMultiple

# Service BAPI for help functions

---

- Service BAPI provide the information or services for the BAPI from individual business components.
- **HelpValues.Getlist** : This method determines the allowed input values (F4 help).
- **BapiService.FieldHelpGetDocu** : This method reads the documentation (F1 help).
- **BapiService.InterfaceGetDocu** : This method reads the entire interface documentation of a bapi.

# Service BAPI

---

- BAPI for interpreting error messages :
- **BapiService.MessageGetDetail** : This method displays the short and long texts of BAPI Error messages.
- **BapiService.ApplicationLogGetDetail** : This method reads the details of entries in the application log.
- BAPI for controlling COMMIT and ROLLBACK :
- **BapiService.TransactionCommit** : This method executes a COMMIT WORK command.
- **BapiService.TransactionRollback** : This method executes a ROLLBACK work command.
- BAPI for conversion between internal and external data formats:
- **BapiService.DataConversionInt2Ext** : This BAPI converts from internal format into the required external format.
- **BapiService.DataConvrsionExt2Int** : This BAPI converts from external format into the required internal format.

# Standardized Parameters

---

- There are some parameters that can be created for various BAPIs because they contain some equivalent data in all BAPIs. Such parameters are known as '**Standardized Parameters**'.
  
- The various Standardized Parameters are :
  - Change Parameters
  - Extension Parameters
  - Return Parameters
  - Selection Parameters
  - Test Run Parameters



# Prerequisites to work with Existing BAPI

---

- The information required to call the method (BAPI) :
- The name of the BAPI.
- Details of BAPI interface :
  - **Import Parameters** :  
Data to be transferred from the calling program to the BAPI.
  - **Export Parameters** :  
Data to be transferred from the BAPI back to the calling program.
  - **Import/Export (Table) Parameters** :  
Importing and Exporting data.

# PO BAPI

The screenshot displays the SAP BAPI Explorer interface. The top menu bar includes Explorer, Edit, Goto, Environment, System, and Help. Below the menu is a toolbar with various icons. The main window is titled "BAPI Explorer" and features a logo on the left. The left pane shows a hierarchical tree of BAPIs, with "PurchaseOrder" expanded and "CreateFromData1" selected. The right pane shows the details for the selected BAPI method, including its name, business object, short description, new in release version, function module, ALE message type, and status.

**BAPI Explorer**

**Method (BAPI)**

Method	CreateFromData1
Business object	PurchaseOrder
Short description	Create Enjoy Purchase Orders
New in Release	46C
Function module	BAPI_PO_CREATE1
ALE message type	PORDCR1
Instance-indepndt	<input type="checkbox"/> Dialog

**Status**

Release status	Released
----------------	----------

# **Sample Program**

# Sample Report

---

## ▪ Calling BAPI from ABAP :

```
DATA :MSGID LIKE BAPIRET2-ID,
      MSGNO LIKE BAPIRET2-NUMBER,
      MY_TEXTFORMAT LIKE BAPIRET2-TEXTFORMAT,
      MSGVI LIKE BAPIRET2-MESSAGE_VI,
      MY_MESSAGE LIKE BAPIRET2-MESSAGE,
      MY_RETURN LIKE BAPIRET2.

DATA : BEGIN OF MY_TEXT OCCURS 1.
      INCLUDE STRUCTURE BAPITGB.
      END OF MY_TEXT.

MOVE 'FI' TO MSGID.
MOVE '024' TO MSGNO.
MOVE 'ASC' TO MY_TEXTFORMAT.
MOVE '0001' TO MSGVI.
CALL FUNCTION 'BAPI_MESSAGE_GETDETAIL'
EXPORTING
ID = MSGID
NUMBER = MSGNO
LANGUAGE = SY-LANGU
TEXTFORMAT = MY_TEXTFORMAT
MESSAGE_VI = MSGVI
```

# Sample Report

---

```
IMPORTING
MESSAGE = MY_MESSAGE
RETURN = MY_ETURN
TABLES
  TEXT = MY_TEXT
IF MY_RETURN IS INITIAL.
WRITE: / 'INPUT'.
SKIP 5.
WRITE : / 'MESSAGE CLASS.....',MSGID.
WRITE; / 'MESSAGE NO.....',MSGNO.
Write:/ 'Variable Message...',MSGVI.
SKIP 1.
WRITE: / 'OUTPUT'.
WRITE;/ 'MESSAGE.....',MY_MESSAGE.
SKIP 2.
WRITE;/ 'TEXT OUTPUT'.
LOOP AT MY_TEXT.
WRIE:/ MY_TEXT.
ENDLOOP.
ELSE.
WRITE :/ MY_RETURN-MESSAGE.
ENDIF.
```

# Exercise

# Exercise

---

1. Call `BAPI_SALESORDER_GETSTATUS` to get status of the sales order.
2. Call `BAPI_MATERIAL_SAVE_DATA` to create the material master record.
3. Call `BAPI_BANK_CREATE` to create the bank master record.
4. Call `BAPI_TRANSACTION_COMMIT` to commit it in database.





# Custom BAPI

# Custom BAPI

---

- Custom BAPI are developed when there is no Standard BAPI to meet the requirements.
- **Process Flow**
- The BAPI process flow can be divided into four phases :
  - Analysis
  - Design
  - Implementation
  - Test and Release



# Analysis in Detail

---

- The analysis phase involves determining which business process is to be implemented using the BAPI.
- The business objects types used in the scenario and their BAPI need to be identified.
- The BAPI required for the scenario are simply identified, and their functions and the data they require are defined.
- The analysis can be divided into three phases:
  - Describing the scenario
  - Defining the scenario in the BAPI Explorer
  - Reviewing the scenario

# Analysis Phase in detail

---

- **Describing the scenario**
  - A scenario is a computerized implementation of a business process and Business process consists of a series of individual business functions.
  
- **Defining the scenario**
  - Procedure to define a project form in the BAPI Explorer :
    - 1.Start the BAPI Explorer using TC BAPI.
    - 2.Select tab page Project and create a project to implement new BAPI.
    - 3.In the first section the basic data can be created for the defined scenario.
  
- **Reviewing the scenario**
  - Before the scenario can be converted and started with a concrete definition and implementation of the BAPI, the scenario

# Designing Phase in Detail

---

- After the scenario and the business object types used in it have been identified in the analysis of the scenario, the BAPI signature is defined conceptionally within the design phase.
- The following conventions need to be adhered for BAPI methods:
  - Check whether instance method or class method is used.
  - The method name must be maximum of 30 characters.
  - The individual components of the BAPI names are marked by initial letters. Underscores are not permitted in BAPI names.
  - Every parameter must have a return parameter that is either an export parameter or an export table.

# Implementing a BAPI

---

- The Implementation phase can be divided into three successive stages:
- Defining the Data Structures in the ABAP dictionary (SE11).
- Implementing the function module in the function builder as Remote Enabled (SE37).
- Defining the business object type and its methods in the BOR (sw01).

# Testing and Releasing

---

- Testing
  - Testing the function module in the function builder.
  
- Releasing
  - Once all the tests have been completed successfully, the BAPI and all related development objects must be released in order to make it available to the customer.



# Sample Report

---

- BAPI (BAPI\_COURSE\_GETDETAIL) CALL FROM ABAP PROGRAM
- After creating ZBAPI\_COURSE\_GETDETAIL and finishing the process flow

Parameter : p\_cid type zgcid,  
                    p\_name type yname.

Data : wa\_return type bapiret2,  
            wa\_trainer\_details type ztriner\_det,  
            it\_course\_details type table of zcourses\_at\_emax,  
            wa\_course\_details type zcourses\_at\_emax.

Call function 'ZBAPI\_COURSE\_DETAIL'

EXPORTING

cid = p\_cid

Name = p\_name

IMPORTING

Return = wa\_return

Wa\_return\_det = wa\_trainer\_details

Tables

It\_details = it\_course\_details

If not wa\_return is initial.

Write:/ wa\_return-message.

Else.

Write:/ 'the course details are'

Uline.

# Sample Report

---

```
Loop at it_course_details into wa_course_details.
Write:/ 'course id', wa_course_details-cid,
    / 'course name', wa_course_details-name,
    / 'Trainer', wa_course_details-trainer,
    / 'Duration',wa_course_details-duration, 'Days',
    / 'Fee', wa_course_details-fee left-justified,
    / 'Demo on', wa_course_details-demo_on,
    / 'Demo at' , wa_course_details-demo_at.
Endloop.
Skip 2.
Write:/ 'The trainer details are'.
Uline.
Write:/ 'trainer is', wa_trainer_details-trainer,
    / wa_trainer_details-text1,
    / wa_trainer_details-text2,
    / wa_trainer_details-text3,
    / wa_trainer_details-text4.
Endif.
```

# Exercise

# Exercise

---

- Call a customized BAPI to create customer master record.
- Call a customized BAPI to display material master record.

# QUESTIONS?

# Summary

---

- By end of this course , participants know
  - The Structure of Business Framework
  - Understand Business objects and BOR
  - Understand the advantages and characteristics of BAPI
  - To work with standard & custom BAPI

**Thank You**