

SAP R/3 Document : Introduction To BAPI'S

Introduction to BAPIs

Definition

The SAP Business Objects held in the [Business Object Repository](#) (BOR) encapsulate their data and processes. External access to the data and processes is only possible by means of specific methods - BAPIs (Business Application Program Interfaces).

A BAPI is defined as a **method** of a SAP Business Object.

For example, the functionality that is implemented with the SAP Business Object type "Material" includes a check for the material's availability. Thus, the Business Object type "Material" offers a BAPI called "Material.CheckAvailability".

Use

To use a BAPI method, an application program only needs to know how to call the method; that is, it needs to know the method's interface definition. Therefore, when including a BAPI invocation in your application program, you only need to supply the appropriate interface information. A BAPI interface is defined by:

- Import parameters, which contain data to be transferred from the calling program to the BAPI
- Export parameters, which contain data to be transferred from the BAPI back to the calling program
- Import/export (table) parameters for both importing and exporting data

Structure

The BAPIs in the R/3 System are currently implemented as **function modules**, all of which are held in the Function Builder. Each function module underlying a BAPI:

- Supports the Remote Function Call (RFC) protocol
- Has been assigned as a method to an SAP Business Object in the BOR
- Is processed without returning any screen dialogs to the calling application

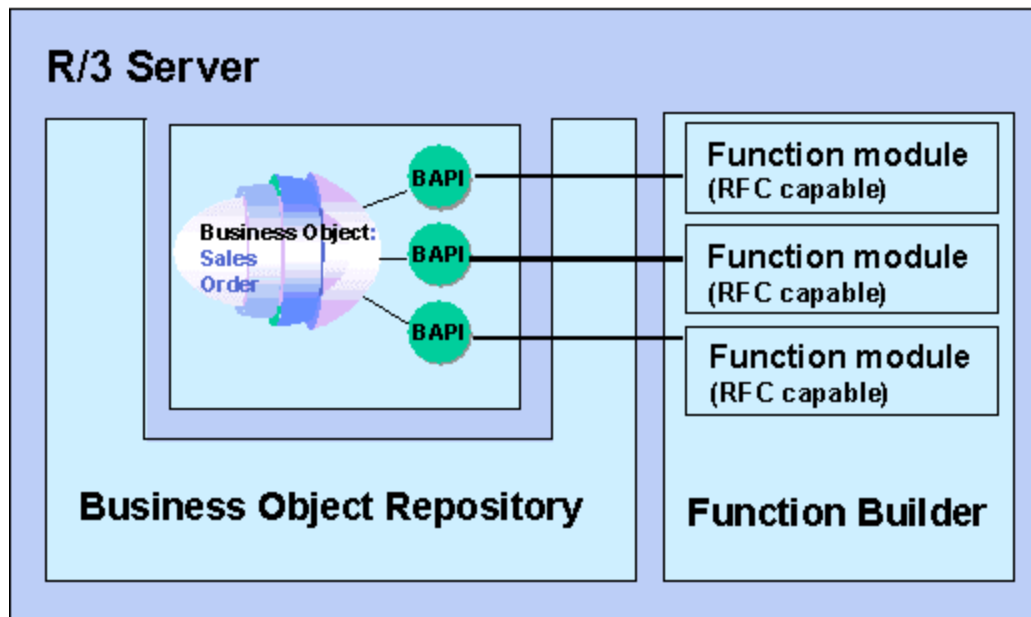
Compiled By : Seetharam Maddali

SAP R/3 Document : Introduction To BAPI'S

Integration

The following figure illustrates the relationship between an SAP Business Object type, its BAPIs, and function modules.

Business Object, with BAPIs and Associated Function Modules



This architecture enables SAP to change the details of a BAPI's implementation without affecting external applications, which are using the BAPI.

Advantages of Using BAPIs

Use

BAPIs are standardized methods of SAP Business Objects that enable customers and third parties to integrate their software components with the R/3 System and the Business Framework. This section lists some of the advantages of using BAPIs as a means of accessing the SAP Business Objects.

Compiled By : Seetharam Maddali

SAP R/3 Document : Introduction To BAPI'S

Business Standard

SAP Business Objects and their BAPIs provide a business content standard, rather than a technical interoperability standard; that is, they enable the integration of R/3 and other software components on a business level, not on a technical level.

Standards Conformance

BAPIs are being developed as part of the SAP joint initiative with customers, partners, and leading standards organizations. BAPIs are becoming a communication standard between business systems.

You can access SAP Business Objects and their BAPIs by using object-oriented interfacing technologies such as Microsoft's COM/DCOM (Component Object Model/Distributed Component Object Model).

The SAP Business Objects already comply with the Open Applications Group (OAG) specifications, and, in conjunction with ObjectBridge from VisualEdge, conform to the Object Management Group's CORBA (Common Object Request Broker Architecture) guidelines.

Stability and Downward Compatibility

Once a BAPI is implemented and released by SAP, its interface definition and parameters remain stable in the long term, thus ensuring that your application program remains unaffected by any changes to the underlying R/3 software and data.

SAP can make any necessary extensions to the BAPIs, for example, additional optional parameters, without destabilizing the operation of existing applications and, at the same time, can offer the enhanced functionality to new applications.

Object Orientation

As methods of the SAP Business Objects, BAPIs provide access to R/3 data and processes following an object-oriented programming model. BAPIs can be called using object-oriented interfacing technologies, such as COM/DCOM, thus enabling software components from SAP and third parties to interact freely.

Openness

You can access BAPIs from all development platforms that support the SAP Remote Function Call (RFC) protocol.

Compiled By : Seetharam Maddali

SAP R/3 Document : Introduction To BAPI'S

BAPI

Definition

A Business Application Programming Interface (BAPI) is a precisely defined interface providing access to processes and data in business application systems such as R/3. BAPIs are defined as API methods of SAP Business Objects. These business objects and their BAPIs are described and stored in the Business Object Repository (BOR).

A BAPI is implemented, however, as a function module, that is stored and described in the Function Builder.

Integration

BAPIs can be called within the R/3 System from external application systems and other programs. BAPIs are the communication standard for business applications. BAPI interface technology forms the basis for the following developments:

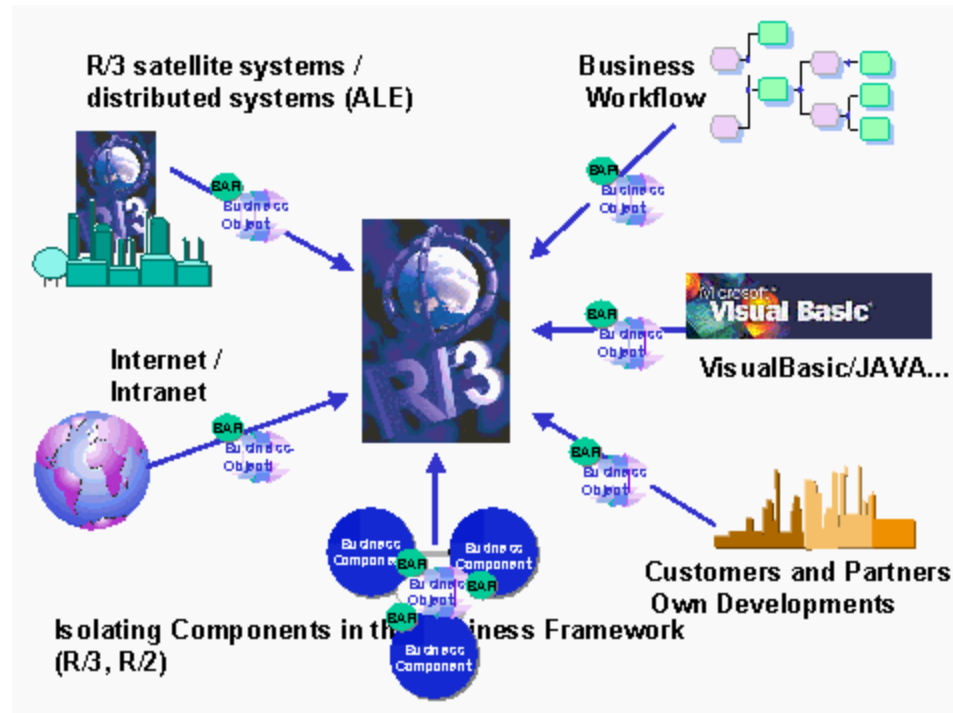
- R/3 satellite systems
- Isolating components within the R/3 System in the context of Business Framework
- Distributed R/3 scenarios using Application Link Enabling (ALE)
- Connecting R/3 Systems to the Internet using Internet Application Components (IACs)
- Visual Basic programs as frontends to R/3 Systems
- Workflow applications that extend beyond system boundaries
- Customers' and partners' own developments
- Connections to non-SAP software
- Connections to legacy systems

The graphic below shows how BAPI interfaces enable different types of applications to be linked together.

Compiled By : Seetharam Maddali

SAP R/3 Document : Introduction To BAPI'S

BAPIS - Interfaces to the R/3 System



Standardized BAPIS

Use

Some BAPIS and methods provide basic functions and can be used for most SAP Business Objects. Such BAPIS are known as "standardized" BAPIS.

Using the descriptions below as a guide, verify whether you can implement the BAPI as a standardized BAPI.

Features

BAPIS for Reading Data

The following BAPIS provide you with read-only access to data in the associated business object:

Compiled By : Seetharam Maddali

SAP R/3 Document : Introduction To BAPI'S

➤ **GetList**

With this BAPI you can select a range of object key values, for example, company codes and material numbers. To specify appropriate selection requirements the calling program must pass the relevant parameters to the interface.

The key values selected by the BAPI *GetList* are returned to the calling program in a table, together with other useful information, for example, short texts. The key values can then be passed on to another BAPI for further processing, for example, the BAPI *GetDetail*, as listed below.

For further information on programming GetList BAPIs see [Specifying Selection Criteria in the BAPI](#) .

➤ **GetDetail**

The BAPI *GetDetail* uses a key to retrieve details about an instance (s specific occurrence) of a business object and returns this data to the calling program.

➤ **GetStatus**

The BAPI *GetStatus* is used to query the status of an SAP Business Object, for example, to display the processing status of a sales order. This BAPI is used only for displaying the status of an object and does not retrieve full details like the BAPI *GetDetail*.

➤ **ExistenceCheck**

The BAPI *ExistenceCheck* checks whether an entry exists for an SAP Business Object, for example, whether the customer master has been created. You should implement this method as a workflow method and not as a BAPI (RFC capable function module).

The method *CompanyCode.ExistenceCheck* of the business object *CompanyCode* (BUS0002) is an example of this. This workflow method is indirectly invoked when the calling program instantiates an object, for example, by using `GetSAPObject("CompanyCode")` from within Visual Basic.

BAPIs for Creating or Changing Data

The following BAPIs can create, change or delete instances of a business object: If required, you can implement these BAPIs so that, several instances of a business object can be

Compiled By : Seetharam Maddali

SAP R/3 Document : Introduction To BAPI'S

created, deleted or modified simultaneously in the same call. In such cases "multiple" is added to the method name, for example ChangeMultiple.

BAPIS that can create, change or delete instances are:

➤ **Create or CreateFromData**

The BAPI *Create* or *CreateFromData* creates an instance of an SAP Business Object, for example, a sales order. *Create* is the preferred name for this BAPI. Use the name *CreateFromData* only when a workflow method called *Create* already exists for the business object in question.

➤ **Change**

The BAPI *Change* changes an existing instance of a SAP Business Object, for example, a sales order.

➤ **Delete**

The BAPI *Delete* deletes an instance of a SAP Business Object, for example, a sales order.

BAPIS for Replicating Business Object Instances

The BAPIS below can be implemented as methods of business objects that can be replicated. They enable specific instances of an object type to be copied to one or more different systems. These BAPIS are used mainly to transfer data between distributed systems within the context of Application Link Enabling (ALE).

The method below must be implemented for each business object to be replicated.

➤ **Replicate**

The BAPI *Replicate* is called in the system which contains the originals of the business object instances to be replicated. It is used to:

- Identify the business objects to be replicated and to organize the required data.
- Call the clone methods described below in the receiving system

Moreover, at least one of the clone methods below must be implemented for each business object to be replicated.

➤ **Clone**

The BAPI *Clone* is used by a system to replicate one business object on another system or to modify one business object that has already been cloned.

➤ **CloneMultiple**

The BAPI *CloneMultiple* is used by a system to replicate **several** business objects on another system or to modify **several** business objects that have already been cloned.

Compiled By : Seetharam Maddali

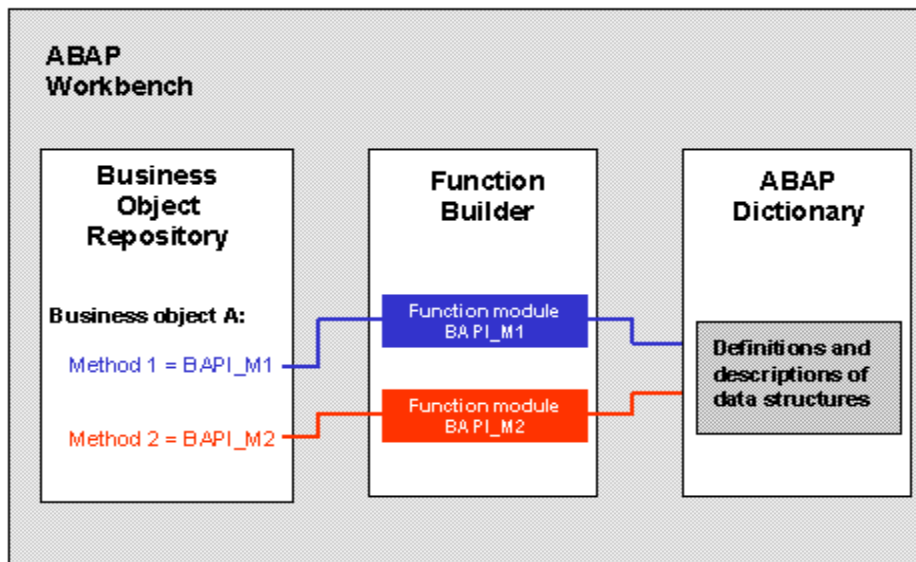
SAP R/3 Document : Introduction To BAPI'S

Defining and Implementing the BAPI

Purpose

Various components of the ABAP Workbench are used when you define and implement a BAPI. A BAPI is an API method of a business object and is defined as such in the Business Object Repository (BOR). However, a BAPI is implemented as an RFC capable function module, which is maintained in the Function Builder. The definitions and descriptions of the data structures used by the BAPI are stored in the ABAP Dictionary. The graphic below shows the components of the ABAP Workbench that are used when BAPIs are defined and implemented.

BOR, Function Builder and ABAP Dictionary



For function modules that implement BAPIs, certain standards and rules must be adhered to over and above the standard programming rules for function modules. For example, COMMIT WORK commands must not be used in the function modules that a BAPI is based on.

The following sections guide you through the steps involved in developing a BAPI. The sections contain information about the guidelines and conventions that you should adhere

Compiled By : Seetharam Maddali

SAP R/3 Document : Introduction To BAPI'S

to when defining and implementing a BAPI. When implementing BAPIs follow the requirements below to ensure you achieve consistent behavior and representation of BAPIs as object oriented methods of SAP Business Objects.

Process Flow

Before you start to develop your BAPI you should print out the [BAPI Development Checklist](#) :

The development of a BAPI involves the following steps:

| Task | For further information see: |
|---|--|
| Defining the scenario in which the BAPI will be used | <u>Defining the Scenario</u> |
| Reviewing the BAPI scenario and how the BAPIs interact with one another | <u>Reviewing the Scenario Concept</u> |
| Defining a BAPI and Its Interface | <u>Defining a BAPI and Its Interface</u> |
| Creating individual programming objects | <u>Creating Individual Programming Objects and Documentation</u> |
| Programming BAPIs | <u>General Programming Guidelines for BAPIs</u> |
| Testing the BAPI and its documentation | <u>Testing the BAPI and Its Documentation</u> |
| Releasing the BAPI | <u>Releasing and Freezing the BAPI</u> |
| Enhancing existing BAPIs, if required | <u>Enhancing Existing BAPIs</u> |

Compiled By : Seetharam Maddali