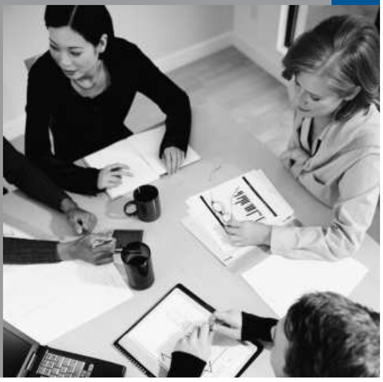
SAP Functions in Detail



THE SAP® Business One SOFTWARE DEVELOPMENT KIT

POWER TOOLS FOR SMALL AND MIDSIZE BUSINESSES

THE BEST-RUN BUSINESSES RUN SAP™

CONTENTS

INTRODUCTION

In today’s rapidly changing world, businesses need software that can easily adapt to meet new needs as they arise. SAP® Business One is an affordable, easy-to-use solution that allows small and midsize businesses to streamline and integrate a broad range of operational and managerial tasks. SAP Business One was built upon a fundamental design principle: adaptability. Simple, easy-to-use features allow users to add new tables, fields, and business objects; create custom lookup lists; establish new validation and field-default rules; and design custom alerts – all without writing a single line of code.

For more complex enhancements, the SAP Business One Software Development Kit (SDK) provides a comprehensive set of tools for professional software developers. The SDK maximizes your IT investment by allowing you to tailor SAP Business One to your particular needs, streamline and automate many processes, and integrate external applications. Built upon a standards-based platform that is compatible with Microsoft .NET, Microsoft Component Object Model (COM), and Java technologies, the SDK lets you leverage your development team’s existing skills and quickly build solutions that bring tangible value to your organization.

KEY BENEFITS

The SAP Business One SDK is a toolkit that enables software developers to add new functionality to SAP Business One, interface with external applications, make minor functional modifications, or simply change the look and feel of SAP Business One. It includes application programming interfaces (APIs), sample code, documentation, and development tools for SAP Business One.

In keeping with the "simple yet powerful" theme of SAP Business One, the SDK is designed to be easy to learn and use, while providing a level of flexibility unmatched by any other business management application in its class.

EASY TO LEARN

The SAP Business One SDK provides interfaces for the most popular technologies on the market today – including Microsoft .NET (C#, ASP.NET, Visual Basic.NET, and so on), Microsoft COM (C++, Visual Basic, ASP, and so on), and Java. There are no proprietary languages, tools, or arcane development environments to learn, neither extensive training nor lengthy ramp-up time is needed.

EASY TO USE

The SAP Business One SDK provides a set of tools to make software development faster and easier and to make deployment of add-on applications simple and efficient.

POWERFUL INTEGRATION

The SAP Business One SDK provides a robust set of programmable business objects for interfacing external applications to SAP Business One. The tiered architecture of SAP Business One maintains a clear separation between business logic and user interface (presentation layer) components, allowing software developers to work directly with the same business logic used by the SAP Business One client application. This approach ensures that integration is not only simple and straightforward, but also powerful.

POWERFUL CUSTOMIZATION

The SAP Business One SDK provides a rich application programming interface for working directly with the SAP Business One client application, allowing programmers to build unique functionality into the user interface: adding new windows, modifying the look and feel of existing windows, or making functional modifications to SAP Business One.

UPGRADE-FRIENDLY

The SAP Business One SDK is designed to be upgrade-friendly. Backward compatibility of interfaces makes it easy to upgrade to the latest release of SAP Business One with little or no modification to custom code.

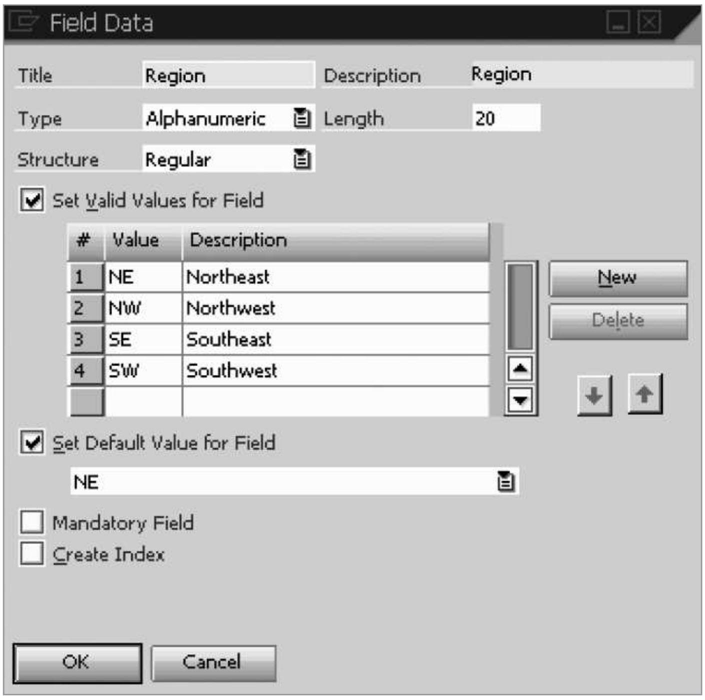
SAP BUSINESS ONE – BUILT FOR ADAPTABILITY

SAP Business One contains numerous features that make it easy for end users to tailor the software to a wide range of specific needs. Although these features are not technically part of the SAP Business One SDK, they are an important part of what makes SAP Business One so flexible and adaptable, and developers interested in working with the SAP Business One SDK should be familiar with them. In many cases, these features provide the fastest, easiest way to modify SAP Business One to suit an organization's specific needs.

USER-DEFINED FIELDS AND TABLES

Users can quickly and easily add new fields to existing tables or add new tables to the SAP Business One solution. User-defined fields can be added to virtually any of the business objects within SAP Business One, including master records (customers or inventory items, for example), sales and purchasing documents (sales orders, invoices, or purchase orders, for example), document rows (invoice line items and purchase order line items, for example), financial transactions (journal entries or journal entry rows, for example), service calls, service contracts, bills of material, and many other objects.

When a user-defined field is added to the system, the underlying database is automatically modified to include the new field. Users can configure the attributes of a new field, including the data type, length, format mask, list of valid values, and default value. Users may also specify whether or not a field is required, can create indexes, and can easily create links to user-defined tables.



Title	Region	Description	Region
Type	Alphanumeric	Length	20
Structure	Regular		

☑ Set Valid Values for Field

#	Value	Description
1	NE	Northeast
2	NW	Northwest
3	SE	Southeast
4	SW	Southwest

☑ Set Default Value for Field

NE

□ Mandatory Field

□ Create Index

Figure 1: Setting Field Data

When a user-defined field is added, it automatically becomes available to the reporting and query tools within SAP Business One and within its user interface.

Users can also add user-defined tables to SAP Business One very quickly and easily. This ability enables users to manage virtually any kind of information within SAP Business One and to create add-on applications to manage that information alongside all their other accounting, logistics, and CRM data.

USER-DEFINED OBJECTS

The user-defined object is a unique feature. It allows end users to link two user-defined tables quickly and easily to create a new business object in SAP Business One. The application automatically creates a new window for the object.

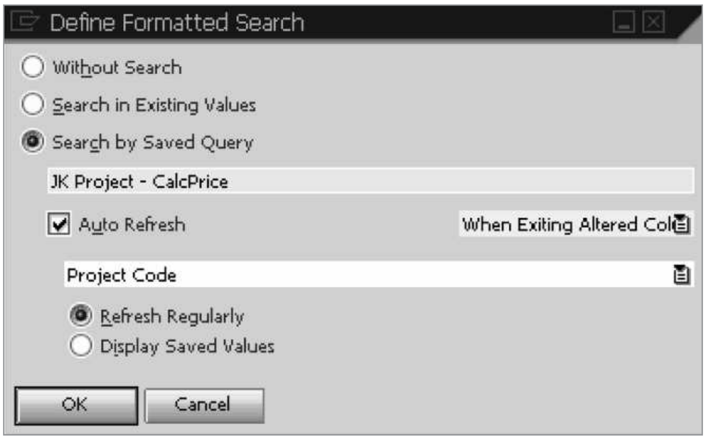
An example best illustrates the concept of a user-defined object. Imagine that your business sells its product through a channel of distribution partners. As part of your effort to ensure that the partners represent your product effectively, you provide a sales training and certification program. You want to maintain a list of certified account representatives in SAP Business One. For each person, you want to keep track of training classes attended, exam results, instructor comments, and a scanned copy of the signed attendance record for each class.

This information can be managed very easily in SAP Business One by adding a user-defined object. First, you set up a user-defined table to store information for each account representative. Next, you create a second user-defined table to store information on the certification classes attended. Finally, you start a simple, wizard-driven process to create a user-defined object to link the two tables to each other. SAP Business One can create a new window for data entry and it automatically handles insert, update, find, and delete operations. SAP Business One maintains the linkage between parent records (in the table containing the list of certified account representatives) and child records (in the table containing the list of training classes the representatives have attended). The user can also choose from a range of optional services, including automatic document numbering, history logging, and year-end transfer.

FORMATTED SEARCH

Formatted search is a powerful feature that allows users to create custom look-up lists and add unique business logic without writing any custom code. Formatted searches can be based upon a list of static, user-defined values, or upon a query that returns one or more values based upon other values in an SAP Business One window.

For example, one company that uses SAP Business One charges its customers different prices for the same item, depending upon the project with which the sale is associated. One customer, for example, pays €100 for an item if it is sold as part of the "World Cup" project, but pays $135 for the same item if it is sold in connection with the "Brady" project. Pricing therefore depends upon three dimensions: customer, project, and item.



Define Formatted Search

○ Without Search
○ Search in Existing Values
● Search by Saved Query

JK Project - CalcPrice

☑ Auto Refresh   When Exiting Altered Col

Project Code

● Refresh Regularly
○ Display Saved Values

Figure 2: Creating a Formatted Search

In this case, a custom pricing solution was created by maintaining a user-defined table to store project-based pricing information. The formatted search feature was used to assign a query to the unit price field within each line of the sales order. The query is simply an SQL select statement that looks for a price in the user-defined table, based upon the current customer, project, and item. The formatted search query is set to refresh the unit price automatically whenever the user changes the project code associated with the line in the sales order.

CUSTOM QUERIES AND REPORTS

SAP Business One provides several different mechanisms to create custom queries based upon the data in SAP Business One, including user-defined tables and fields. Queries provide a simple, intuitive way to view reports in real time and to export data to Microsoft Excel or other applications. Queries can also be used to create custom alerts based upon user-specified conditions.

SAP Business One includes a simple, intuitive report writer for creating or modifying printed reports.

ALERTS

Online alerts in SAP Business One allow users to create real-time notification of important events and conditions that require attention. Users can quickly and easily specify the information they want and can determine how and when they want to be alerted. For example, a sales manager might want to receive an e-mail notification of all sales opportunities that exceed a specified amount each morning at 10:00. Such alerts can be set up in just a few minutes.

The built-in customization features of SAP Business One are available to all SAP Business One customers at no additional cost.

ARCHITECTURE OVERVIEW – COMPONENTS

The SAP Business One SDK provides APIs that help you interface with and enhance SAP Business One in several ways. From simple screen modifications to sophisticated integration projects, the SDK provides a flexible set of tools that includes:

- **The Data Interface API**, which lets you build real-time links between SAP Business One and external applications. By providing a programmatic interface to SAP Business One through a set of fully functional business objects, the Data Interface API gives you the power to connect applications and automate processes. The Data Interface API exposes the business-logic layer of SAP Business One, and is independent of the SAP Business One client application.
- **The User Interface API**, which lets you modify SAP Business One by creating new windows, redesigning existing windows, and adding menus or dialog boxes. Modifications can be as simple as hiding or disabling a field, or as complex as adding a new transaction. The User Interface API enhances and extends the SAP Business One client application.

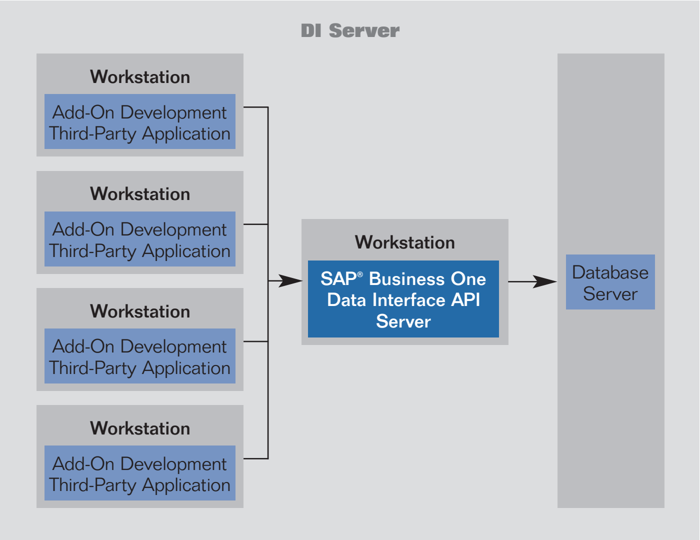

DI Server
Workstation
Add-On Development Third-Party Application
Workstation
Add-On Development Third-Party Application
Workstation
Add-On Development Third-Party Application
Workstation
Add-On Development Third-Party Application
Workstation
SAP® Business One Data Interface API Server
Database Server

Figure 3: The DI Server

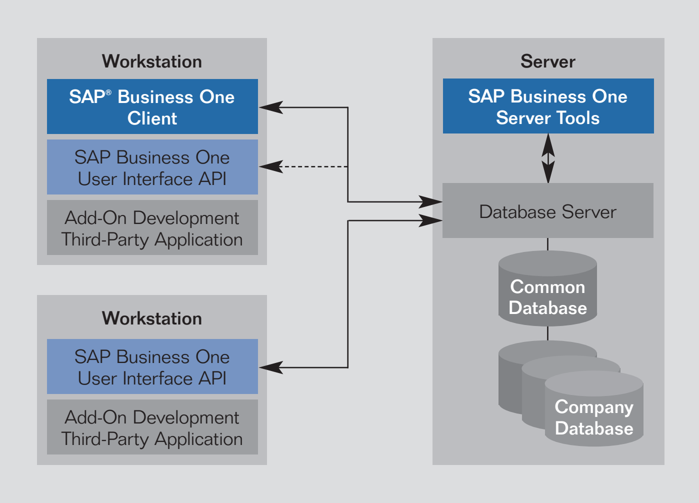

Workstation
SAP® Business One Client
SAP Business One User Interface API
Add-On Development Third-Party Application
Workstation
SAP Business One User Interface API
Add-On Development Third-Party Application
Server
SAP Business One Server Tools
Database Server
Common Database
Company Database

Figure 4: SAP Business One SDK Architecture

- **The DI Server**, which provides an XML interface based upon SOAP standards and is designed for high-volume, Web-services integration between SAP Business One and external applications.
- **The Java Connector**, which gives developers all the power and flexibility of the Data Interface API in a Java development environment.

The source code of SAP Business One is not publicly available and cannot be viewed or modified by third parties. This approach guarantees a uniform code base for all SAP Business One customers, which ensures stability, reduces risk, and facilitates smooth upgrades.

THE DATA INTERFACE API (DI-API)

The Data Interface API (DI-API) provides a programmatic interface to SAP Business One through a set of fully functional business objects. The DI-API gives developers access to the business logic and data-access layers of SAP Business One, enabling them to create real-time links between SAP Business One and external applications. The DI-API is independent of the SAP Business One client application.

The DI-API automatically handles communication with the underlying database, ensures full data validation, and automatically populates default field values based upon the business rules of SAP Business One. These features allow software developers quickly and easily to interface applications with SAP Business One without requiring a detailed understanding of the database or business rules of SAP Business One.

The DI-API is implemented as a DLL based upon Microsoft COM. It is commonly used with Microsoft Visual Studio or Visual Studio.NET (including Visual Basic 6, Visual Basic.NET, C#, C++, ASP, and ASP.NET), but may be used with any COM-compliant or .NET development tool. The DI-API is typically loaded by a client application that needs to read and write data to SAP Business One.

The DI-API contains four categories of objects: master data objects, transactional objects, metadata objects, and general purpose objects.

MASTER DATA OBJECTS

Master data objects represent relatively static entities, such as inventory items, general ledger accounts, business partners, bills of material, and employees. Master data objects are frequently organized hierarchically; an object may have parent-child relationships with other business objects. For example, the BusinessPartners object represents customers, vendors, and leads. A business partner may have many addresses, which are represented by the BPAddresses object. Likewise, a business partner may have many ContactEmployees, and may contain any number of user-defined fields, which can be accessed through UserFields – a child object of the BusinessPartners object.

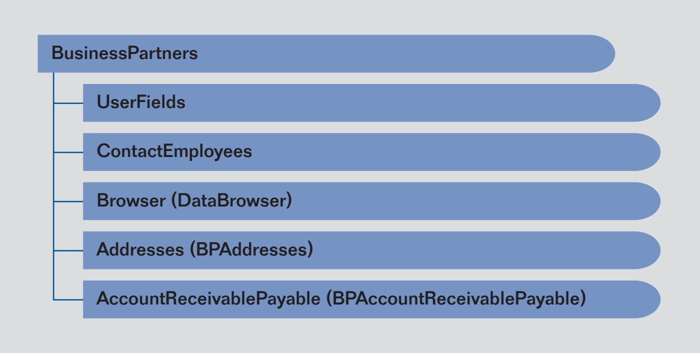


Figure 5: An Example of Master Data Objects

Master record objects typically include the following methods:

- **GetByKey** is used to retrieve a specific instance of the object from the SAP Business One database. For example, to update data for an existing business partner, you would create a new instance of a BusinessPartners object and include the business partner code as a parameter of the GetByKey method. Doing so would retrieve the specified record from the SAP Business One database for reading or writing data.

- **Add** is used to add new records to SAP Business One.
- **Update** is used to make changes to an existing record in SAP Business One.
- **Remove** is used to delete a record from SAP Business One, if such an operation is allowed. In many cases, the business rules of SAP Business One prohibit the removal of records. For example, a customer record may not be removed if transactions have been created for that customer.
- **SaveXML** can be used to save a business object to an XML file.

The following sample of Visual Basic code shows how the DI-API can be used to create a customer record in SAP Business One:

```
Dim oCustomer As SAPbobsCOM.BusinessPartners
Set oCustomer = oCompany.GetBusinessObject(oBusinessPartners)

oCustomer.CardCode = "OfficeEquipCorp"
oCustomer.CardName = "Office Equipment Corporation, Inc."
oCustomer.CardType = cCustomer

oCustomer.Add
```

Likewise, the same record can be retrieved later and updated as follows:

```
Dim oCustomer As SAPbobsCOM.BusinessPartners
Set oCustomer = oCompany.GetBusinessObject(oBusinessPartners)

oCustomer.GetByKey("OfficeEquipCorp")
oCustomer.CreditLimit = 10000

oCustomer.Update
```

TRANSACTION OBJECTS

Transaction objects typically represent events or business documents. In other words, transaction objects normally represent one or more stages in a business process. Transactions generally have financial implications, although some CRM transactions do not necessarily have any direct impact upon general ledger or inventory. Examples of transaction objects include journal entries, sales orders, accounts payable invoices, purchase orders, goods transfers, and work orders.

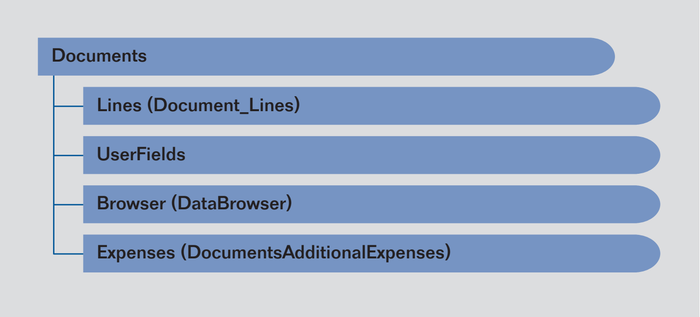


Figure 6: An Example of Transaction Objects

Transaction objects typically include the following methods:

- **GetByKey** is used to retrieve a specific instance of the object from the SAP Business One database. For example, to update a sales quotation, you would create a new instance of a Documents (quotation) object and retrieve the specific document from SAP Business One by including the document key in the GetByKey method.
- **Add** is used to add new records to SAP Business One.
- **Update** is used to make changes to an existing record in SAP Business One.

- **Remove** is used to delete a record from SAP Business One, if such an operation is allowed. In many cases, the business rules of SAP Business One prohibit the removal of records. For example, a sales invoice may not be removed.
- **Cancel** is used to cancel a document (for example, a sales quotation).
- **Close** is used to close a document (for example, a purchase order)
- **SaveXML** can be used to save a transaction object to an XML file.

The following sample Visual Basic code shows how the DI-API can be used to create a sales order in SAP Business One:

```
Dim oOrder As SAPbobsCOM.Documents
Set oOrder = oCompany.GetBusinessObject(oOrders)

oOrder.CardCode = "C40001"          'Specify the customer code
oOrder.DocDate = Date               'Specify the document date

'Add a line
oOrderLine.ItemCode = "A00001"
oOrderLine.Quantity = 5
oOrder.Add
```

Information not specified in the code is handled automatically by the DI-API, using the standard business logic of SAP Business One. For example, because the item price is not specified in the sample code shown above, SAP Business One calculates it automatically. Item prices may vary, depending upon the specific customer, price list, quantity, currency, and effective date. However, software developers using the DI-API do not need to be concerned with such details. They simply need to provide the same few pieces of information that a typical user would provide during normal data entry; the DI-API will handle the rest.

METADATA OBJECTS

Metadata is data about data. Metadata objects in SAP Business One are used to add, remove, or update user-defined fields, user-defined tables, and user-defined objects. Metadata objects include the UserTablesMD object (user-defined tables metadata), the UserFieldsMD object (user-defined fields metadata), the UserKeysMD object (user keys metadata), UserObjectMD (user-defined object metadata), and the ValidValues object, which applies to any list of valid values (typically seen as a drop-down list in SAP Business One).

GENERAL-PURPOSE OBJECTS

General-purpose objects provide connectivity, rapid data access, and miscellaneous functions. They comprise the smallest set of business objects in the DI-API and include:

- **The Company object** is used to create a connection to an individual SAP Business One company.
- **The RecordSet object** is used for rapid, read-only access to high volumes of SAP Business One data.
- **The DataBrowser object** provides a convenient mechanism to navigate through a set of business objects (through a filtered subset of customers, for example).
- **The SBOBob object** includes a set of miscellaneous functions for quickly and easily setting and retrieving commonly used information from SAP Business One. Examples include GetItemPrice, GetCurrencyRate, SetCurrencyRate, and Format_DateToString. The SBOBob object includes approximately 20 miscellaneous functions.

THE USER INTERFACE API (UI-API)

The User Interface API (UI-API) provides programmatic access to windows, menus, and events in SAP Business One. The UI-API allows software developers to create new windows, modify existing windows, add new business logic, add or modify application menus, and more. Modifications may be as simple as hiding or disabling a field or as complex as adding a new type of transaction or new module to the application.

The UI-API is implemented as a DCOM executable that runs on a client machine alongside SAP Business One. The UI-API exposes the interface components of SAP Business One (windows, controls, and menus) and various events that occur within the SAP Business One client application.

OBJECTS IN THE UI-API

The UI-API contains several categories of objects: application/connection objects, menu objects, forms, items (controls), and data sources.

Application/Connection Objects

Application/connection objects provide a means to create a connection to a specific instance of SAP Business One and manage that application after a connection has been established.

Connecting to an instance of SAP Business One is simple and straightforward. The following sample Visual Basic code demonstrates how to create a connection:

```
Dim oApps As SAPbouiCOM.SboGuiApi
Dim oApplication As SAPbouiCOM.Application

Set oApps = New SboGuiApi
oApps.Connect Command

Set oApplication = oApps.GetApplication
```

The command parameter shown in the example above is used by SAP Business One's auto-start mechanism for add-on applications created with the UI-API. The auto-start mechanism is a configurable feature – it ensures that add-on applications are running when required by SAP Business One. Auto-start is described below in the section on add-on administration.

The UI-API also includes a mechanism for creating a single-sign-on connection to the DI-API. In other words, when an add-on application uses the UI-API and DI-API together, the single-sign-on feature allows developers to create a connection to the DI-API using the credentials of the user logged into SAP Business One.

The application object also includes some commonly used features that are useful for most applications, including:

- **MessageBox** displays a dialog box in SAP Business One. A message box may present several options to the user (for example: "Do you want to save this record?"… "Yes", "No", or "Cancel"), or it may simply present information to the user.
- **SetStatusBarMessage** creates a message that appears in the SAP Business One application status bar.
- **SendKeys** is a simple method for sending keystrokes to the application.

The following sample Visual Basic code shows how to present a dialog box to a user with the MessageBox method:

```
iResponse = oApplication.MessageBox("Save this record?", 3,
            "Yes", "No", "Cancel")
```

The desktop object allows developers to determine the state of the SAP Business One desktop programmatically (whether it is minimized, maximized, or neither, for example), or set desktop properties (the size of the desktop or desktop wallpaper, for example).

Menu Objects

Menu objects allow software developers to add, remove, or modify menus programmatically in SAP Business One. Developers can add new items to the command center in SAP Business One, add new menus or submenus, enable or disable menu items, or modify menu text.

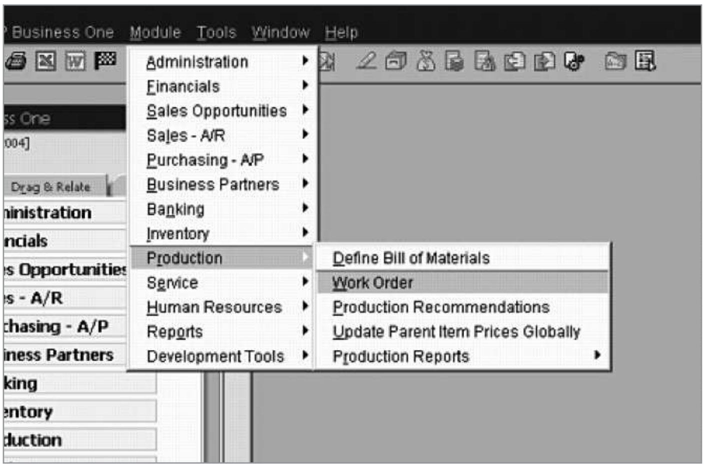


Figure 7: SAP Business One Menus

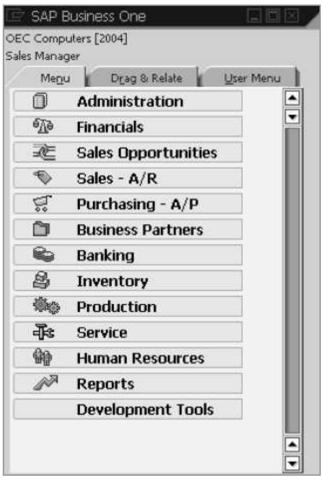


Figure 8: The SAP Business One Command Center

Forms

Forms collection and the form object provide a means of adding new forms (windows) to SAP Business One or modifying existing forms.

The form object represents a unique instance of a form in the SAP Business One application. The following sample Visual Basic code shows how a new form can be created.

```
Private oForm As SAPbouiCOM.Form

'Create a fixed-size form with the unique ID "MyForm"
Set oForm = oApplication.Forms.Add("MyForm", ft_Fixed)

'Set some of the form's properties:
   oForm.Title = "My New Form"
   oForm.Left = 400
   oForm.Width = 329
   oForm.Top = 100
   oForm.Height = 100

'Add a button:
   Dim oItem As SAPbouiCOM.Item
   Set oItem = oForm.Items.Add("MyButton", it_BUTTON)
   oItem.Specific.Caption = "&OK"

'Make the form visible:
   oForm.Visible = True
```

Note that this sample shows a somewhat manual method for creating a new form (that is, using code). This method is not the only one possible. In fact, the preferred method is to design a form using the graphical screen design tool (the screen painter) in the SAP Business One SDK and load it using a single method call. This issue is discussed below in the section on the screen painter.

Items (Controls)

The UI-API provides access to the native, user-interface controls in SAP Business One: buttons, checkboxes, grids, and text boxes, for example. The UI-API refers to these controls as "items."

The following lists item types:

- Button
- Static Text (Label)
- Text Box
- Extended Text Box (for multiple lines of text)
- Pipe
- Frame Rectangle
- Combo Box
- Link Arrow
- Picture
- Check Box
- Option Button
- Matrix (Grid)
- Pane Combo Box

Each type of item (control) has a unique set of methods and properties. For example, a ComboBox item contains a drop-down list of valid values, which are represented in by the item's ValidValues property. A button has a Caption property, a CheckBox item has a Checked property, and so on.

Some items may contain child objects. For example, a matrix item represents a grid. It contains a columns collection, which in turn contains individual column objects, each containing one or more cell objects.

Data Sources

Data sources are used to bind data from SAP Business One to items (controls), providing rapid access to data and a simple means of populating items (such as a grid) with that data. Data sources act as containers for retrieving and managing data within a form.

EVENTS IN THE UI-API

Events drive most of the code within a UI-API-based application. In other words, code within an add-on application will usually be triggered by events within the user interface of SAP Business One. There are three kinds of events: application events, menu events, and item events. Item events can further be subdivided into general item events and form item events.

Application Events

Application events provide notification when a user closes the SAP Business One application, changes companies, or changes the display language in SAP Business One. These events are typically used to change the display language of or to shut down an add-on application.

Menu Events

The menu click event is used to perform operations in response to a user selecting a menu item from SAP Business One or clicking on a toolbar icon. This event is typically used to open a window, but other operations may be performed as well.

General Item Events

Item events are the most commonly used events in SAP Business One. Item events apply to user interface items (controls) and forms. Some examples of item events include:

- GotFocus
- LostFocus
- Click
- Double Click
- Combo Select
- Key Down
- Item Pressed

Form Item Events

Forms are a special kind of item, so that some item events pertain uniquely to forms:

- Form Load
- Form Unload
- Form Activate
- Form De-Activate
- Form Close
- Form Resize
- Form KeyDown
- Form MenuHighlight

The following sample Visual Basic code shows how to respond to an event in the UI-API. In this example, the code responds to a button click event by presenting the user with a message box:

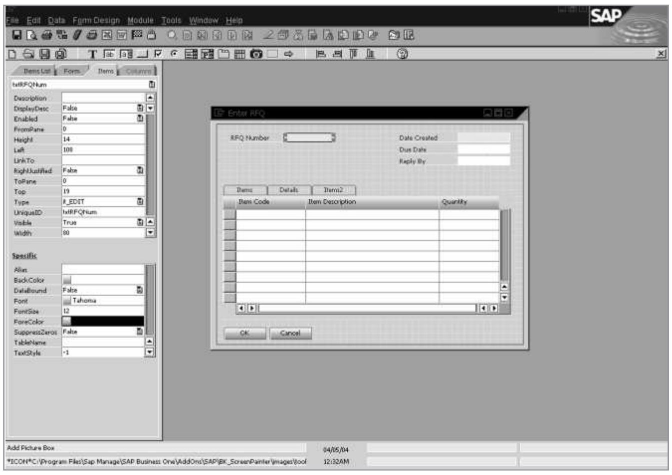
```
'If the event is occurring on form #60002 and the user clicked the item
'called "MyButton"... then pop up a message box:

If pVal.FormType = 60002 And pVal.ItemUID = "MyButton" And _
pVal.EventType = et_CLICK Then
Set oForm = oApp.Forms.Item(FormUID) oApp.MessageBox "Hello World!"
End If
```

THE SCREEN PAINTER

The screen painter is a graphical screen-design tool that enables you to create new forms (windows) for SAP Business One quickly and easily.

The screen painter is an add-on application. It installs separately and is then started automatically when SAP Business One is launched.

Using the screen painter is simple. To create a new form, simply click the new form icon and then select the items (controls) to place on the form. Items can easily be moved around on the form by dragging and dropping them in the desired location, and item properties can be set within the properties window (shown on the left side of the screen in Figure 9).



txtRFQNum

Description	
DisplayDesc	False
Enabled	False
FromPane	0
Height	14
Left	100
LinkTo	
RightJustified	False
ToPane	0
Top	19
Type	it_EDIT
UniqueID	txtRFQNum
Visible	True
Width	80

Specific

Alias	
BackColor	
DataBound	False
Font	Tahoma
FontSize	12
ForeColor	
SuppressZeros	False
TableName	
TextStyle	-1

Add Picture Box 04/05/04
*ICON*C:\Program Files\Sap Manage\SAP Business One\AddOns\SAP\BK_ScreenPainter\images\tool 12:32AM

Figure 9: The Screen Painter

The screen painter saves each form as an XML file, which can then be loaded with a single line of code (using the Application.LoadBatchActions method).

THE DI SERVER

The DI Server provides a SOAP interface to SAP Business One. The DI Server is an extension of the DI-API and supports all its objects. The DI Server is intended for high-volume data integration, where numerous client connections must be managed simultaneously and optimized for speed. It is also suitable where Web-services architecture is preferred (with remote clients, for example).

The DI Server supports a wider range of client technologies than the DI-API, and allows use of COM, CORBA, or TCP/IP to interface with SAP Business One using XML. The DI Server applies the same business logic (validation, default fields, and so on) as the client application does.

The DI Server provides automatic connection pooling for enhanced performance and scalability. It provides a highly efficient means for making calls to SAP Business One by allowing multiple functions and multiple objects within a single call, and can handle an unlimited number of cached objects per database.

THE JAVA CONNECTOR (JCO)

The SAP Business One Java Connector (JCo) provides a Java interface to SAP Business One. The JCo is a wrapper around the DI-API and supports all the objects, methods, and properties supplied by the DI-API

ADD-ON ADMINISTRATION

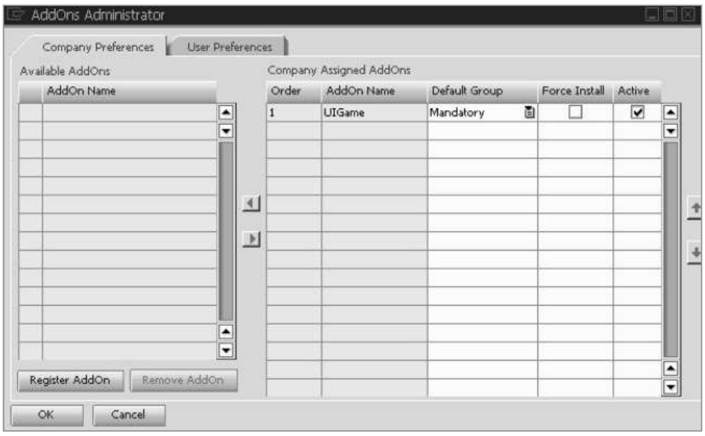
The add-on administration tool is designed to help administrators deploy and manage add-on applications on end-users' workstations easily.

The add-on administration tool allows you to:

- Register an add-on to run automatically when SAP Business One is launched
- Configure company-wide preferences
- Configure user preferences
- Configure auto-install or auto-upgrade for the add-on application

COMPANY PREFERENCES

The add-on administration tool lets you set various company-wide preferences for each add-on assigned to a particular company.



Order	AddOn Name	Default Group	Force Install	Active
1	UIGame	Mandatory	☐	☑

Figure 10: The Add-On Administration Tool

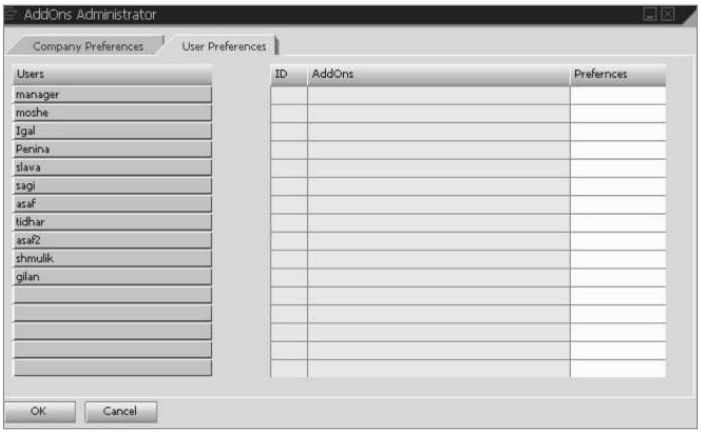
1. Start behavior : *Automatic, Manual*, and *Disabled*
   a. **Mandatory** – This setting is used when the add-on must be running to ensure data integrity. In this case, the add-on is launched automatically when SAP Business One is started, and SAP Business One shuts down if the add-on is terminated for any reason.
   b. **Automatic** – The add-on is launched automatically when SAP Business One is started. End users may terminate the add-on application without adversely affecting the SAP Business One solution. In this case, SAP Business One issues a warning that the add-on was closed.
   c. **Manual** – The add-on application is not started automatically by the SAP Business One solution. A user may run a manual add-on at any time (provided that the user has permissions to do so). When a manual add-on is terminated, SAP Business One issues a warning message that the add-on was closed.
   d. **Disabled** – The add-on is not available for this company, but can be executed manually by specific users with the appropriate permissions. This option is typically used for debugging purposes.

2. Event-receiving order
   With this setting, the administrator can specify the order in which add-on applications receive notification of events from SAP Business One.

3. Force install
   This setting can be used for automatic deployment and upgrade of add-on applications. To use this feature, the administrator sets a path to a self-extracting setup executable for the application. The add-on is automatically installed or upgraded when the user logs on to SAP Business One.

USER PREFERENCES

The User Preferences tab allows the administrator to configure the behavior of each add-on for each user. Options include: automatic, manual, disabled, and default. The default option uses the settings established at the company level.



Users
manager
moshe
Igal
Penina
slava
sagi
asaf
tidhar
asaf2
shmulik
gilan

ID	AddOns	Prefernces

Figure 11: The User Preferences Tab of the Add-On Administrator

BACKWARD COMPATIBILITY

The SAP Business One SDK is designed to be upgrade-friendly. SAP is committed to maintaining backward-compatible interfaces to the fullest extent possible. Although it is impossible to maintain 100% backward compatibility, SAP strives to change interfaces only when absolutely necessary.

Backward compatibility of interfaces means that few code changes (if any) should be required when upgrading to a new version of SAP Business One.

DEVELOPING WITH SAP

TRAINING AND CERTIFICATION

SAP provides several training programs for the SAP Business One SDK. Because it is based upon industry-standard technologies, most developers will feel comfortable using the APIs within a very short time. Training classes for the SAP Business One SDK are usually scheduled for four days, with a developer certification exam on the fifth day.

SAP DEVELOPER NETWORK

The SAP Developer Network (SDN) is SAP's collaborative portal for developers and integrators, serving the needs of SAP technical experts worldwide. Members of the SDN community can find how-to articles on the SAP Business One SDK, integration toolkit (ITK), and core application features. Customers, partners, and newcomers to the SAP ecosystem will find detailed information on evaluating, implementing, and using SAP technologies – all in one place.

To learn more about the SAP Developer Network, visit http://sdn.sap.com.

QUALIFICATION AND CERTIFICATION OF ADD-ON SOLUTIONS

SAP provides a qualification and certification program for independent software vendors who have developed add-on products for SAP Business One or who have integrated existing products with SAP Business One. As part of the SAP solution developer program, software partners have access to SAP's training and certification programs, partner events, and more.

NAMESPACES

Partner solutions developed using the SAP Business One SDK should utilize namespaces approved by SAP. The use of namespaces ensures that solutions developed by different partners do not conflict with each other because of table names, field names, form identifiers, and so on.

OTHER TOOLS

DATA TRANSFER WORKBENCH

The data transfer workbench offers an easy-to-use wizard that imports new data into and updates existing data in SAP Business One. The data transfer workbench is frequently used to migrate data from legacy systems to a new SAP Business One system. It can also be used for ongoing, batch-mode data integration.

The data transfer workbench enables users to import data from delimited text files quickly and easily, ensuring that all the business logic of SAP Business One is applied during the import process. Where an object contains child records (if a sales order contains multiple line records or a customer contains multiple address records, for example), the data transfer workbench automatically ensures the integrity of the related data with the SAP Business One database.

To simplify the preparation of the data for import, SAP provides predefined data file templates. Users may format the data in a spreadsheet program (such as Microsoft Excel) using the predefined templates provided or their own file format. The data transfer workbench contains a mapping engine, allowing users to create and save mappings from custom file formats to SAP Business One.

INTEGRATION TOOLKIT

The integration toolkit (ITK) provides scenarios for integrating one or more instances of SAP Business One with mySAP™ Business Suite (4.6C or higher). The ITK is designed to use SAP Exchange Infrastructure (XI) as an integration hub to handling, mapping, translation, transformation, routing, and security.

- **Central accounting** – The ITK supports a set of predefined central accounting scenarios that allow businesses to process transactions that span multiple entities, including trading partners and subsidiaries, along with consolidation, vendor-managed inventory, intercompany processing, and master data distribution. Businesses using the ITK can also build function modules for custom scenarios using the SAP Business One JCo.
- **Consolidated reporting** – Thanks to the integration between SAP Business One and SAP Business Information Warehouse (SAP BW), the head office can conduct consolidated reporting on all the group's activities. SAP BW standardizes the reporting data from multiple systems used by the group's subsidiaries (SAP R/3®, SAP Business One, and other ERP systems) and prepares it for compiling consolidated reports. You can then analyze the consolidated data with the Business Explorer tool or Web reporting.
- **Data migration** – SAP Business One gives businesses the power and freedom to grow. But as they grow, so do their requirements. If companies grow to need more sophisticated ERP functionality, they can simply migrate from SAP Business One to mySAP Business Suite. The ITK fully supports the migration process.

VERSIONS AND LICENSING

The SAP Business One SDK is a supplementary product to SAP Business One and must be purchased and installed separately.

The SAP Business One SDK includes:

- APIs
- Development tools
- Sample programs
- Documentation

Several versions of the SAP Business One SDK are available: **Runtime components** (with the exception of the DI Server) are available to all customers at no additional charge. Users accessing SAP Business One data using runtime components must be licensed (named) users of SAP Business One.

The **implementation version** includes only the UI-API. As such, it can be used to make very small functional modifications and to change the appearance and behavior of SAP Business One forms. Because it does not include the DI-API or other development tools, the implementation version is primarily intended for making cosmetic changes to SAP Business One. This version is available to all SAP Business One customers free of charge.

The **development version** includes the UI-API, DI-API, JCo, and the screen painter. This version does not include the DI Server. The development version is intended for SAP partners and customers who want to create add-on solutions for SAP Business One or who wish to interface external applications with SAP Business One.

The **DI Server** requires a separate license and is available only to partners or customers who already own the development version of the SAP Business One SDK. The DI Server is installed on a server and may be accessed by multiple clients. The DI Server is licensed on a per-CPU basis.

All versions are shipped on separate CDs and with individual license keys.

www.sap.com/contactsap

50 071 193 (04/11)