

THE EXPERT'S VOICE® IN SAP

SAP ABAP

Hands -On Test Projects with Business Scenarios

Sushil Markandeya and Kaushik Roy

Apress®

For your convenience Apress has placed some of the front matter material after the index. Please use the Bookmarks and Contents at a Glance links to access them.



Apress®

Contents at a Glance

About the Authors	xxv
About the Technical Reviewer	xxvii
Acknowledgments	xxix
Introduction	xxx
■ Chapter 1: ERP and SAP Overview	1
■ Chapter 2: ABAP Dictionary/Data Dictionary/DDIC – 1	21
■ Chapter 3: ABAP Dictionary/Data Dictionary/DDIC – 2	81
■ Chapter 4: ABAP Language Basics	141
■ Chapter 5: WRITE Statement (Classical Reporting)	197
■ Chapter 6: Internal Tables	245
■ Chapter 7: Modularization	323
■ Chapter 8: Open SQL Data Retrieval	391
■ Chapter 9: SELECTION-SCREENS	447
■ Chapter 10: Interactive Lists	513
■ Chapter 11: ABAP OOP	575
■ Chapter 12: ABAP List Viewer Outputs—Part 1	647
■ Chapter 13: ABAP List Viewer Output—Part 2	707
■ Chapter 14: Screen Programming	735
■ Appendix A: Description of the SAP-Delivered Tables Used in the Hands-On Exercises	829
Index	845

Introduction

ABAP (Advanced Business Application Programming) is the programming language and the platform on which the entire SAP software has been developed. Originally, a procedure oriented language, OOP constructs and features were added to it more than a decade back. Though, its core constructs remain the same, it underwent many enhancements over the last decade. The ABAP platform is used during both the customization and implementation phases of the SAP software in the enterprise.

Target Audience

The book addresses people who want to learn ABAP afresh and people working with ABAP for a few years (0-4 years). People with experience in ABAP will find this book useful as a reference for hands-on examples, from the context of doing meaningful things.

Target Audience Pre-requisites

The target audience must have a fair exposure to OOP language/s like C++, Java, etc. They must be exposed to the basic features of OOP including encapsulation, inheritance, polymorphism, etc.

They must also have a good exposure to RDBMS basics like RDBMS tables, table columns, relationships, primary key, foreign key, cardinality, entity, attribute and table metadata. An exposure to RDBMS SQL, especially the data retrieval statement, is required.

Book's Approach

The whole of the book's thrust is on the doing something meaningful part: to be able to create programs and objects required as per a defined context or scenario.

The book uses a scenario oriented presentation style. Concepts, commands and statements are communicated through illustrative examples and scenarios. Wherever possible, small business scenarios are used to communicate concepts, commands and statements.

Shorn of weighty theoretical treatment and preoccupations with language syntax, the book is a completely practical approach, demonstrating and conveying the language's commands, features through hands-on examples.

Definitions, descriptions, syntax, etc. are introduced even as you set about implementing a scenario.

The presentation of the features is scenario oriented: most of the features are demonstrated in terms of small business scenarios. The book's scenario descriptions along with source program resource containing the ABAP program source enable the reader to create all objects related to the scenario and run or execute them. The underlying concepts of a feature/command are totally conveyed through execution of these hands-on programs.

The demonstrating/illustrating objects, including the programs, rely on some of the SAP functional module tables being populated (the term functional module is explained in Chapter 1). This is assured if the reader is logged on to an IDES (internet demonstration & evaluation server) server or system. The IDES server is briefly described in Chapter 1. An IDES server is now a de facto system for all SAP training related activities. Specifically SAP functional module tables used for illustration and hands-on exercises in the book are the basic tables of sales and purchase. Most people with nil or little to no exposure to the business, and commercial world relate to the business areas of sales and purchase.

All of the hands-on exercises in the book are performed using SAP sales and purchase functional module tables. The IDES server is used ensuring substantial data in the SAP functional module tables. Details of the SAP functional module tables used in the hands-on exercises are to be found in Appendix A. The authors strongly advise you to go through the appendix describing the functional module tables used in the hands-on exercises of the book. You can visit this Appendix A before you commence the reading of a chapter from Chapter 5 onwards (when you start using the SAP functional module tables).

The book also uses a “carry forward” approach for the hands-on exercises. Where ever possible, a hands-on exercise performed in earlier chapters is carried forward to subsequent chapters, with additions and enhancements carried out to it. This involves re-usability of created objects again and again.

Chronology and order of presentation of topics and sub topics is related to pre requisites required for the presented topics and sub topics. At some places, topics and sub topics are introduced on a preview basis to make the current hands-on exercise more meaningful. Subsequently the topics and sub topics introduced on a preview basis are formally presented.

The authors strongly insist that you perform the hands-on exercises as you read a chapter and encounter the hands-on exercises—do not defer performing them until after reading a chapter or the book. It should not be just reading, but reading and simultaneously performing the hands-on exercises.

Resources

The book is supplemented with an E-resource or source program resource.

The E-resource or source program resource contain the hands-on exercises source programs. Most source programs in the E-resource are also listed in the book. Some source programs of the E-resource are partially listed in the book and some are not listed in the book. This is indicated in the respective chapters. The E-resource is downloadable and each of the hands-on exercise source programs can be opened as a separate file in Windows Notepad. While performing the hands-on exercises, you can copy and paste an entire source program from the E-resource, or manually enter the source program by referring to the source program lines of the E-resource. Manual entry of source programs will get you a feel of the syntax of statements. If you are copying and pasting, remember to replace the ‘REPORT. .’ statement line from the E-resource program into the created ABAP program in the ABAP workbench environment. You will be introduced to the ABAP workbench environment in Chapter 1.

In addition to the book and the supplementary E-resource, you will use the following additional resources:

You must have connectivity to SAP, preferably a SAP IDES server. Your log-in user id must be able to create, edit and delete objects in the ABAP workbench environment. Your log-in user id must be able to access data from SAP functional module tables. Apart from book reading, some extra reading is required such as:

You will need to read extra theory and description not expounded in the book. You need to refer to some detailed information not provided in the book.

You will need to read up the detailed tutorial or documentation related operations in the ABAP workbench environment not described in the book.

You will likely refer to and read the detailed properties, methods and events, etc. of some SAP supplied classes.

The material for most of the above mentioned readings is available in freely downloadable PDF documents from the following link <http://www.easymarketplace.de/online-pdfs.php>. You will not violate any copyright law downloading the documents from this link. These are PDF documents of SAP version 4.6C. Though the documents are older versions; they will largely serve your purposes.

Download the following PDF documents from the link: <http://www.easymarketplace.de/online-pdfs.php>.

ABAP Programming (BC-ABA)	BCABA.pdf
ALV Grid Control (BC-SRV-ALE)	BCSRVALV.pdf
BC - ABAP Dictionary	BCDWBDIC.pdf
BC ABAP Workbench Tools	BCDWBTOO.pdf
BC ABAP Workbench Tutorial	BCDWBTUT.pdf
SAP List Viewer (ALV): Classic	CAGTFLV.pdf

You will need to refer (not read) to these documents for information during your reading of this book. Preferably, you won't read these documents during your chapter reading. Read these documents between chapters. At what stage, which of these documents are to be read or referred is indicated in the book's chapters.

So, on to stimulating reading. Not just readings though, the hands-on exercises should be performed simultaneously with your reading. The documents mentioned above will also be available for download on <http://www.apress.com/9781430248033>.

CHAPTER 1



ERP and SAP Overview

ERP Overview

Enterprise Resource Planning (ERP) software products are business application, packaged software. They are used to run large- to medium-sized business enterprises. The ERP software suppliers claim that their software can cater to every business activity of every category of business enterprise. In most cases, an enterprise relies on multiple databases to maintain its operations: to facilitate the migration, the ERP software supports multiple databases for interaction and input but relies primarily on a centralized database for storing all function module data. The ERPs integrate the different business activities of a business enterprise. They support multiple currency handling, an essential feature for business enterprises operating globally. The ERPs also have the feature of consolidation, which facilitates the consolidation or accumulation of the accounting numbers of transnational business enterprises having a large number of subsidiaries around the world. The ERP software suppliers claim that all advances in hardware and software technologies are incorporated in their newer upgrades or versions. Quite a few phrases and terms have been used in this attempt at defining an ERP. Explanations of these terms and phrases follow.

Business Enterprises

A language dictionary meaning of business enterprise does not help in the present context. A definition is being attempted from the point of view of law or statute. The users of ERP software are large- and medium-sized business enterprises. Such business enterprises have to be registered or incorporated companies. Any group of people wanting to start a business enterprise should apply for registration or incorporation. Business enterprises are also called companies, enterprises, firms, or organizations. Any of these synonymous terms will be used. In the ERP world, the registered companies are also called 'legal entities'. Once a company is registered, it can commence business operations. The registered company must, as per law, publish its audited financial results once a year, and unaudited financial results every three months or quarter. Audited means the financial results generated by the company would be cross-checked and verified for its accuracy, or authenticity, by an independent audit company. These are statutory financial results as required by law distinct from financial results the company would produce for its own internal purposes. The statutory financial results must be reported in the currency of the country where the company is registered. Hence, companies registered in the United States would produce their financial results in U.S. dollars. This currency, in which a company has to produce financial results, as per law, could be termed 'operating currency of the company'.

A company registered in one country, generally, will not directly operate a full-scale business in another country. A company 'ABC U.S.' registered in one country, say the United States, wanting to conduct a full-scale business in another country, say the United Kingdom, will register a local company in the United Kingdom, such as 'ABC U.K.'. This company in the United Kingdom, 'ABC U.K.', is designated as a subsidiary of the parent company 'ABC U.S.'. A parent company can own 100% of a subsidiary or less. A parent company can have any number of subsidiaries around the world. A subsidiary company can, in turn, have many subsidiaries.

Business enterprises are categorized based on their prime activity. Figure 1-1 shows the categories of business enterprises.

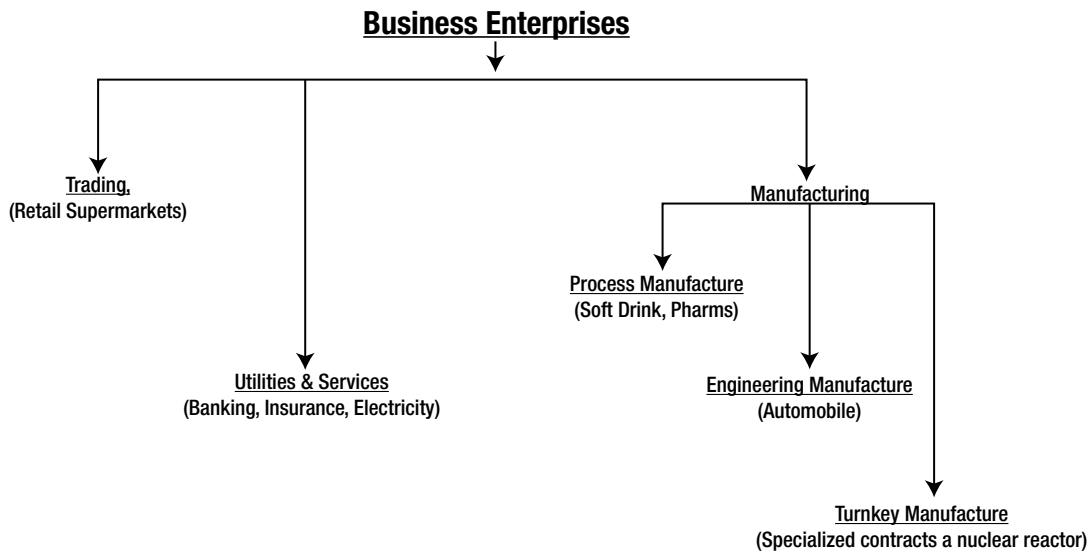


Figure 1-1. *Business Enterprise Categories*

The first category of business enterprise, ‘Trading,’ does not involve producing or manufacturing any goods. Such enterprises buy goods and sell them. A very common example of this subcategory is ‘Retail Trading,’ such as supermarkets. The second category of business enterprise is ‘Utilities & Services’ (‘Electric Energy Distribution,’ ‘Banking’) and do not do any manufacturing. Though electricity is produced, it is not a good in the normal sense of the word, so it is categorized under ‘Utilities.’

The third category of business enterprise is ‘Manufacturing.’ This category is further subcategorized into ‘Process,’ ‘Engineering,’ and ‘Turnkey.’ The subcategories reflect the production or manufacturing characteristic. In the first subcategory ‘Process,’ ingredients pass through a process, and the product at the end of the process is frequently packaged in different modes. Once ingredients and processes are fixed, this is repetitive in nature. The most common examples of ‘Process’ manufacturing are the soft drink companies and the pharmaceutical companies. One other example of the subcategory ‘Process’ would be an oil refinery company. The crude oil passes from one fractional distillation process to another, producing various petroleum products at each stage of fractional distillation.

In the second subcategory of manufacture, ‘Engineering,’ components or parts are produced and/or procured in the first stage. Then these components are assembled to produce the final product. Before the assembly stage, all the required components should be available if the assembly is to be carried out successfully. So in the ‘Engineering Manufacture,’ to produce a certain number of finished products, components required before the assembly stage are to be ascertained. This is called ‘Requirement Planning,’ a major exercise to be performed in ‘Engineering Manufacture.’ In fact, a set of software called ‘Material Requirement Planning’ – MRP – were predecessor to the ERP. The most common example of this subcategory is a business enterprise manufacturing automobiles. It is to be noted that once the specifications of, for example, a car model are finalized, the process of production, procurement, and assembly is repeated over and over again, a kind of routine process to produce that model of the car. Components could be manufactured within the business enterprise: for example, an engine is manufactured within the business enterprise from some raw materials. Ready-made components could be procured (outsourced) components: for example, headlights and tail lights are procured from other business enterprises. Therefore, in the assembly stage, there could be a mix of produced components and procured components.

The third subcategory of ‘Manufacture’ is ‘Turnkey Manufacture.’ This subcategory is similar to the subcategory ‘Engineering Manufacture’ to the extent that components are produced first and then subsequently assembled. But in ‘Turnkey Manufacture,’ unlike the ‘Engineering Manufacture,’ there is a degree of non-repetitiveness in the process. Some parts of the process could be repetitive, but some parts would be non-repetitive.

Though each of the business enterprise categories is complex in its own way, as shown in Figure 1-1, the complexity of business increases in relative terms as you move from left to right. So ‘Services’ are perceived as relatively more complex than ‘Trading,’ ‘Manufacture’ more complex than ‘Services,’ ‘Engineering Manufacture’ more complex than ‘Process Manufacture,’ and so on.

There could also be more categories of business enterprises. Try to think whether packaged software companies such as SAP, Oracle, and Microsoft would fit into these categories or require a new one. Similarly, think about whether ‘Real Estate’ business enterprise (selling houses) fits into an existing category in the diagram or not.

The categories of business enterprises in the diagram might not be complete. The subcategories of manufacturing business enterprises might not pass muster from a fastidious production planning professional. But the objective here of describing the categories and subcategories is to give a feel for the complexities of business enterprises that ERP software has to address to people having their first exposure to business in a commercial world.

Business Activities

Within a business enterprise, different business activities are performed. These business activities are ‘Accounting’ or ‘Finance,’ ‘Sales,’ ‘Material Accounting’ or ‘Material Management,’ ‘Production Planning,’ ‘HR,’ etc. Some of these business activities could be common to categories of business enterprises such as ‘Finance’ (‘Finance’ is the heart of every business, and every business enterprise must perform this business activity); ‘Sales’; ‘HR’; or ‘Material Management.’ Some of the business activities could be specific to certain categories of business enterprises. The business activity ‘Material Management’ is relevant to the categories of ‘Manufacturing’ and ‘Trade’ but not to the category ‘Services.’ Similarly, the business activity ‘Production Planning’ is not relevant to the categories ‘Trade,’ ‘Services,’ and so on. These business activities could be viewed as departments performing a specific function in a business enterprise. In ERP parlance, these departments are termed business functions or functional modules, and each functional module caters to a specific business function. Henceforth, you will refer to the business activities as functional modules.

ERP Concept, Its Evolution

Pre ERP Business Application Software Scenario: Before the advent of the ERP concept — in the mid-1970s and earlier (a long time ago) — business application software used to be created around each business function. The accounting or finance functional module would have its set of programs or software and data; the sales would have their own software and data to run the application. The data of the different functional modules was maintained separately, not in a centralized database. The software of different functional modules was running in a segregated manner (i.e., there was no integration – a major hallmark of the ERP software). Moreover, the software was the so-called tailor-made software. At that time there was no concept of packaged, off-the-shelf software that is available today. Business application software used to be created much like the tailor-made shirt. Every time you want a shirt, you go to the tailor with the requisite cloth, give measurements, and specify the shirt’s features (for example, how many pockets?). Are the pockets flapped? Are they full or half sleeved? In an analogous fashion, to create new business application software for a functional module of a business enterprise, personnel designated as ‘systems engineers’ or ‘systems analysts’ would interact with the functional modules’ personnel of the business enterprise, study the functional module’s processes and systems, do a conceptual design and a data design, write programs, test programs, and implement and maintain the programs. Next time that software for the same functional module is to be produced for a different business enterprise, the same gamut of activities would be completed, maybe involving repetition of some or most of these activities.

Pre ERP Technology Scenario: In the 1970s, the computer industry was in its nascent stage. The level and scale of hardware and software features available were nowhere close to what is available today.

To give a bare idea of what was conspicuously not available in hardware, consider this list: (i) No microprocessors. (ii) No distributed processing, only centralized processing represented by the mainframes and mini computers. (Mini computers subsequently vanished.) (iii) No semiconductor RAM. RAM sizes were of the order of kilobytes and a few megabytes. Disk sizes of the order of 10s of megabytes. (v) Very little networking, almost to the extent of nonexistence (i.e., no connectivity).

Here is now an idea of what was conspicuously not available in software: (1) No Graphical User Interface – GUI. (2) No networking operating systems. (3) No OOPS technology. (4) The operating systems were rudimentary; even a task like disk and file space management was performed by the end user. (5) RDBMS was in a very nascent stage.

And all of this cost lot of money, millions of dollars. The computers could only be afforded by the Fortune 500 companies!

The ‘Pre ERP Business Application Software Scenario’ and ‘Pre ERP Technology Scenario’ descriptions have been included because ERP software carries a legacy relating to these scenarios, and exposure to this helps relate to the legacies in the ERP software.

Major Attributes of ERP Software

Centralized Database

In ERP software, all the data relating every functional module is stored in one centralized database. The next attribute, integration, is possible only with a centralized database.

Integration

The term ‘integration’ has been introduced in the context of ERP software. Let this integration be illustrated by a simple business need. Consider the instance of business enterprise payroll processing. Every week/month (depending on the payment period), an enterprise has to generate the pay sheet of its employees. For each employee, a pay sheet has to be created detailing the various earnings – basic salary, house allowance, conveyance allowance, etc.; and the various deductions – provident fund, health insurance, etc. Each employee will receive his or her respective pay sheets through an e-mail or by some other mode. The functional module responsible for the pay sheet (perhaps HR) will maintain the pay sheet data.

Now, since the pay sheet involves money payment, the finance functional module should be notified. This is a basic need of any business. Any activity involving payment or receipt of money will invariably involve this notification to the finance functional module. But the finance functional module requires the pay sheet information in a different fashion. They do not want the detailed information of every employee. Instead, they want summary information for each functional module: the total of every type of earning and deduction. Tables 1-1 and 1-2 elaborate this scenario.

Table 1-1. *Employee Pay Sheet*

Functional Module/ Department	Employee No.	Basic	House Allowance	Conveyance Allowance	Provident Fund	Health Insurance
HR	500001	20000	4000	2000	1600	1000
HR	500010	30000	6000	3000	2400	1500
HR	500015	25000	5000	2500	2000	1250
Total - HR		75000	15000	7500	6000	3750
Sales	500030	50000	10000	5000	4000	2500
Sales	500035	40000	8000	4000	3200	2000
Total – Sales		90000	18000	9000	7200	4500

Table 1-2. *Functional Module Wise Pay Sheet Summary*

Functional Module Department	Basic	House Allowance	Conveyance Allowance	Provident Fund	Health Insurance
Total - HR	75000	15000	7500	6000	3750
Total - Sales	90000	18000	9000	7200	4500

For illustrative purposes, only three employees for the first functional module have been considered, and only two employees for the second functional module. In a real-life scenario, there could be hundreds and thousands of employees in a functional module, and there would be quite a few functional modules, not just two. What is to be understood is the way the functional module of finance should receive the pay sheet data.

In the ERP software, this facility of summarizing the data and sending the data from HR to finance is built in. The sending and receiving of data in technical terms obviously means transfer of data from HR tables to finance tables of the centralized ERP database.

Multiple Currency Handling

Business enterprises conduct business around the globe. They buy goods and services from suppliers in countries other than the country of their registration, so they may have to pay in the currencies of the countries where the suppliers are located. Similarly, they sell goods and services to customers located in countries other than the country of their registration, so they may raise bills in the currencies of the countries where the customers are located. But as per the law of the land, ultimately everything has to be converted into the currency of the country where the business enterprise is registered. So the element of conversion from one currency to another comes into play. The conversion rate to convert from one currency to another is called the exchange rate. The issue of conversion is complicated by requirements of the laws. Most countries' laws require that different exchange rates be adopted in different contexts. This entails maintenance of a large number of exchange rates to be used and assigned in different contexts as prescribed by laws. Let these concepts be again explained with a simple business scenario.

Assume that there is a software services company 'ABC Ltd.' operating in India; its operating currency is Indian rupees. 'ABC Ltd.' has provided services (probably implemented SAP) to a customer company located in the United States. For the services provided, 'ABC Ltd.' presents a bill of 5,000,000 U.S. dollars. As per the law, this figure needs to be converted to Indian rupees to incorporate this into the book of accounts of 'ABC Ltd.' The issue is what exchange rate to adopt. This is a context of sales (i.e., software services have been sold). Suppose the law states that in the context of sales, the exchange rate (U.S. dollar vis-à-vis Indian rupee) prevailing on the last trading day of the month on the money market in which the sales bill was raised should be adopted. Suppose this rate was 1 U.S. dollar = 55.00 Indian rupees. [There is a money market, where money of different currencies is sold and purchased like a commodity]. Hence the figure of 5,000,000 is taken into 'ABC Ltd.' account books as 275,000,000. This is the end of this transaction. This was a simple scenario of currency conversion and multiple currency handling.

The issue of multiple currency handling is a complex one. The ERP software has built-in features to enable the handling of this issue.

Consolidation

This feature is required by business enterprises having subsidiaries around the globe. When the financial statements are prepared for these business enterprises (parent companies) having a number of subsidiaries around the world, the financial statements must incorporate the numbers of all the subsidiaries as well. This flow of numbers from the subsidiaries to the parent is called consolidation. The process of incorporating subsidiaries' numbers is complicated by two factors: (a) When the numbers of the subsidiaries flow to the parent company, they should be converted to the currency of the parent company, as the parent company and its subsidiaries will be operating in different currencies.

The multiple currencies factor is again operating. (b) The parent company's ownership is not direct but through other subsidiaries, and ownership might be less than 100%. The ERP software addresses these issues and has built-in features to enable consolidation.

Figure 1-2 attempts to graphically convey the consolidation of a simple scenario.

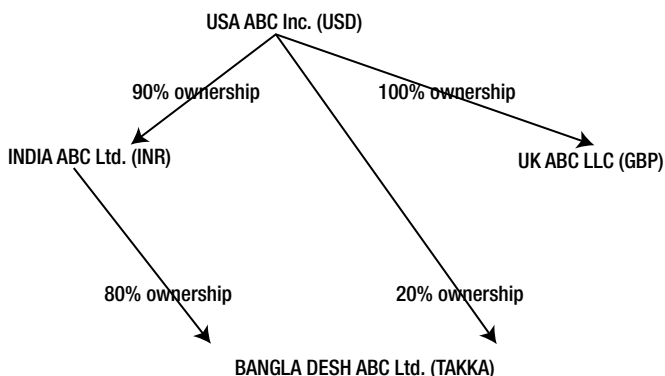


Figure 1-2. Parent & Subsidiary Companies

In the graphic hierarchal representation, the parent company USA ABC Inc. has subsidiaries INDIA ABC LTD. (90% ownership), UK ABC LLC (100% ownership), and BANGLA DESH ABC Ltd. (20% ownership). USA ABC Inc. also owns BANGLA DESH ABC Ltd. through subsidiary INDIA ABC Ltd. (80% ownership). The currencies of these countries are indicated in parentheses.

When figures of parent company USA ABC Inc. are prepared, the numbers of its subsidiaries should flow in the manner shown in Table 1-3.

Table 1-3. Flow of Numbers from Subsidiary Companies to Parent Company

From Company	Currency Conversion	% Ownership	To Company
INDIA ABC Ltd.	INR to USD	90	USA ABC Inc.
UK ABC LLC	GBP to USD	100	USA ABC Inc.
BANGLA DESH ABC Ltd.	TAKKA to USD	20	USA ABC Inc.
BANGLA DESH ABC Ltd.	TAKKA to INR	80	INDIA ABC Ltd.
INDIA ABC Ltd.	INR (from TAKKA) to USD		USA ABC Inc.

This is a small illustrative scenario. A real-life scenario would involve more subsidiaries and more complex parent vis-à-vis subsidiary relationships. The ERP software is equipped to deal with the consolidation process.

Technology Advances

Every major ERP supplier/vendor has a technical tie-up with every other hardware and software supplier/vendor. These major ERP suppliers claim they introduce and incorporate features of technology advances into their newer release/versions.

Multiple Lingual Support

Business enterprises conduct business in their respective native language: business enterprises located in China conduct business in Mandarin, business enterprises located in Germany conduct business in German, and so on. The ERP software supports multiple languages. At the user login stage or a process of setup, a language is input. In the ERP operating environment, the messages, labels, and texts, etc. will then appear in the logged language.

SAP Overview

SAP - *Systems, Applications, and Products in Data Processing* – is the name of the company as well as the product. The company was later renamed as SAP A.G. It is headquartered in Germany and was started by five engineers who were then employed with IBM. At IBM they performed the task of creating tailor-made business application software. At that time every single computer supplier supplied everything related to computers such as all hardware (processors and memory devices), operating systems, development tools such as language compilers, and tailor-made business application software. When the five engineers left IBM and started their own company, they persuaded IBM to subcontract the *business-application-software* part to their company, probably the first instance of outsourcing in the IT industry. Wherever IBM delivered mainframe hardware and accompanying deliverables, except the business application software, SAP created this business application software.

SAP came out with product SAP R/1 in 1972. In this product specification, ‘R’ signified real time. Real time was then a hot technology feature like cloud computing is today. The simplest example of real time is when your airline tickets are booked, the seat availability for booked flight/s of the specific day/s are updated instantly. The ‘1’ signified one or single-tier architecture; the database, application, and presentation resided on a single system.

In 1979, SAP released the multiuser SAP R/2 with the following features:

- Supported on IBM Mainframe
- ‘R’ indicated ‘Real Time’
- ‘2’ indicated 2-tier architecture: (1) Database+Applications layer (2) Presentation layer
- Most of the common modules, such as Finance, Sales, Material Management, Manufacturing, and HR were offered

Until mid-1992, when SAP R/3 was released, the number of SAP installations ≈ 300.

In mid-1992, SAP released the SAP R/3 product with the following features:

- ‘3’ indicated a 3-tier architecture: (1) Database (2) Applications layer (3) Presentation layer
- Supported on (1) popular flavors of UNIX: IBM, HP, Sun Solaris; (2) OS/2, Windows
- Supported the following databases: (1) ADABAS now called SAP MaxDB (2) Oracle (3) Informix (4) IBM DB2 (5) MS SQL Server (when it was released)
- SAP as of now supports the following additional databases: (1) Sybase ASE (2) HANA

No other ERP suppliers or vendors could match this multi-platform offering of SAP at that time. Because of a drastic reduction in hardware and software licensing costs in following years (more and companies could afford it), SAP witnessed an exponential growth in these years. Its installed base crossed 100,000 in 2010. It is now the third largest software company.

SAP followed up the SAP R/3 product with MySAP Business Suite (with integration through the Internet), E.C.C. 5.0 (E.C.C. – ERP Core Component), and E.C.C. 6.0. The latest version at this time of writing is E.C.C. 6.0 EHP6 (Enhancement package 6).

Client Server Architecture

At this stage, it is sufficient to have only a basic idea about the client server architecture, which is the software view of SAP architecture. An SAP installation consists of a database server, typically on a UNIX-based machine/system; one or more application server/s, each typically on a UNIX-based machine/system (maybe a maximum of six); and many window-based presentation servers or systems. The database server is meant to provide the services of data maintenance and data access. Programs are loaded into the application server/s and executed in the application server/s' RAM. Presentation servers are for user access, operation, interaction, and dialogue.

In typical window-based SAP training environments, the database server and application servers are located on a single system/machine. If you install SAP software on a laptop or a desktop and work on it, all three servers (i.e. database, application, and presentation) are located on the same system or machine.

SAP NetWeaver

SAP NetWeaver is an integration framework. It was designed to operate in *service-oriented architecture (SOA)*. It constitutes cooperative technologies that integrate SAP functional modules and also provide connectivity to external systems. The NetWeaver is a successor to SAP R/3. The first NetWeaver version 6.2 was released in 2003. The latest NetWeaver version is 7.3 at the time of this writing. The NetWeaver version is distinct from The SAP software version (E.C.C. 6.0). The SAP NetWeaver has components, tools, and applications.

The application server is a component of the SAP NetWeaver and also part of the software view of SAP architecture. The application server has a complex architecture. The application server is called the NetWeaver Application Server (SAP NW AS). The NetWeaver application server has a five-tiered architecture: presentation, business, integration, connectivity, and persistence. The NetWeaver application server consists of an application server ABAP (AS ABAP) and an application server Java (AS JAVA). You can install either one or both.

Other components of NetWeaver: business warehouse, business process management, business rules management, process integration, master data management, mobile, portal, Auto-ID Infrastructure, identity management, and information life-cycle management.

SAP Functionalities

The SAP business suite consists of five components:

- Enterprise Resource Planning – ERP
- Customer Relationship Management – CRM
- Supply Chain Management – SCM
- Supplier Relationship Management – SRM
- Product Life-Cycle Maintenance – PLM

The ERP component, in turn, consists of the various functional modules. The functional modules are referred by the short forms: FI for finance, SD for sales and distribution, and so on.

A partial list of functional modules with their names and their short forms:

- | | |
|-------------------------|----|
| 1. Accounting/Finance | FI |
| 2. Control(Cost) | CO |
| 3. Sales & Distribution | SD |
| 4. Material Management | MM |
| 5. Production Planning | PP |

6.	Human Resources	HR
7.	Quality Maintenance	QM
8.	Plant Maintenance	PM
9.	Project System	PS
10.	Treasury	TR

SAP Implementation Overview

The SAP implementation is a substantial exercise and involves large high-skill manpower resources and time resources. The cost of SAP implementation exceeds the combined cost of hardware and software licensing. Generally the company planning to use the SAP software does not implement the software; they would not have the requisite know-how. The implementation for the most part is done by third parties. These third parties are SAP-authorized implementation partners. The SAP implementers form implementation teams at the commencement of implementation. There is a team from the implementer side and another team from the company that will use the SAP software. A company planning to use SAP will not implement all the SAP functional modules. They will only implement functional modules relevant or required by them. Even the functional modules to be implemented by a company need not be implemented in one phase. The implementation could be done in phases, with a time gap between the phases. A typical SAP implementation time period is 18 months.

A SAP implementation involves configuration, customization, enhancement, and modification. Modification involves modifying SAP-provided objects - not advisable except in rare situations.

A typical configuration activity will involve creation of a company code: assigning a chart of accounts and business area to the company code, etc.

A simple example of customization: on a SAP-provided data entry screen, disabling fields not required by the enterprise.

An example of enhancement: addition of fields required by the enterprise to a SAP-delivered table.

Implementation Team Composition

An implementer side team will have a hierarchy such as project managers, project leads, team leaders, team members, and so on. The team from the implementer side consists of three main categories of consultants. They are:

- I. The Basis consultants
- II. The Functional consultants
- III. The ABAP/Technical consultants

Implementation Environments

The SAP implementation is carried out in well-defined controlled environments. Different stages of implementation are carried out in different environments segregated from each other. The typical implementation environments are (i) development and (ii) testing and quality assurance. The outcome of these two environments will flow to the third environment, which is the live environment (a typical implementation has at least three stages, which are discussed here).

End-user training is a very critical activity of SAP implementation that commences after configuration, customization, and enhancement stages are complete. It deals with training of staff at the customer site in usage of software to carry out their day-to-day tasks.

Objects would be created by an ABAP consultant in the development environment. These would be moved - or to use the SAP terminology - transported to the testing and quality assurance environment. Once they pass testing and quality assurance, they will be transported to the live environment.

Similarly, a functional consultant will create functional components (like company codes, charts of accounts, etc.) in the development environment. These would be transported to testing and quality assurance. Once they pass testing and quality assurance, they will be transported to the live environment.

Suppose, in another scenario, testing and quality assurance are two separate environments. Then, objects/components would be transported from development to testing, from testing to quality assurance, and from quality assurance to live environment.

Now how an object/component travels is determined by the transport layer that gets associated with the object/component. The transport layer and transportation of objects/components are part of a feature, 'Central Transport System', which is the purview of the Basis consultant.

In today's hardware and software scenario, you can have different systems for different environments (a three-system 'Landscape'). You can have one Windows-based system assigned to the development environment. (Windows can support a few hundred users without any performance or response bottlenecks.) Similarly another Windows-based system is assigned to testing and quality assurance. The end-user training usually resides on one of these two systems. On the other hand, a UNIX-based system is assigned to a live environment. The systems are connected to each other through network for transporting objects and data from one environment to another as per transport layer specification. The objects and data in these environments are segregated from each other because they are residing physically on different systems. This is called the three-system landscape. This description is represented in Figure 1-3.

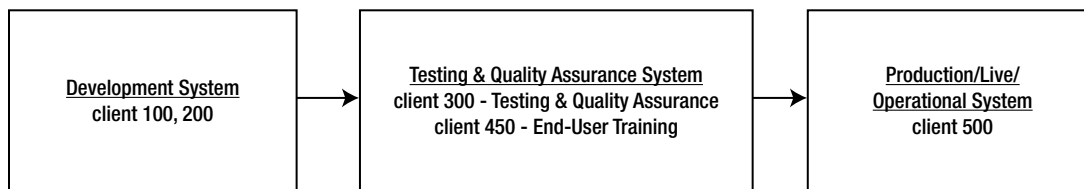


Figure 1-3. *Three-System Landscape*

The 'Clients' are explained in the section entitled 'Client Code Perspectives'.

Now, consider this case in the SAP R/2 mainframe system. A separate mainframe could not have been assigned for each environment (development, testing and quality assurance, live). All the environments had to reside on a single mainframe system. Yet these environments needed to exist in a segregated manner. The segregation was obviously implemented through a logical means. The means of logical segregation is a legacy carried by the subsequent and latest version of SAP software.

The segregation is essential. Whatever is being done in the development environment should not be visible and available in other environments and vice versa. There is a separate environment for end-user training. To train an end user in the procedure and process of creation of customer data, dummy customers are created in the end-user training. These dummy customers must not appear in the live environment or other environments. There would normally be no transport of objects and data from end-user training environment to other environments.

Overview of SAP Login, SAP GUI, ABAP Workbench

You are advised to use the SAP IDES server. (The IDES server has been elaborated in the section entitled 'SAP IDES Server and use of SAP IDES Server for Trainings.) All the hands-on exercises in this book have been performed on the IDES server.

It is assumed you have access to a SAP server, preferably to an IDES server. The front-end software has been installed, connectivity established, and you are ready with the login pad.

SAP Login

When you click on the SAP login pad on your system, the SAP Login screen will look like the one in Figure 1-4.

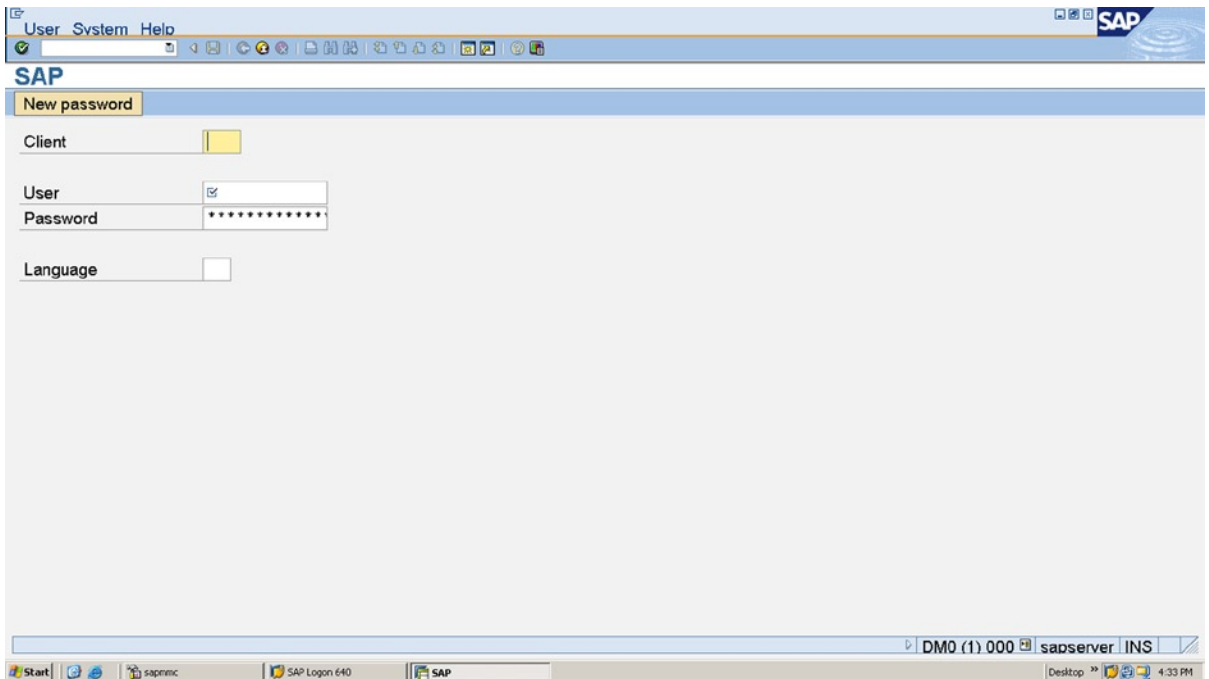


Figure 1-4. SAP Login

The SAP login screen (Figure 1-4) prompts for four fields of information:

1. Client
2. User
3. Password
4. Language

1. The Client, a three-digit number, is one of the parameters through which environment segregation (development, testing & quality assurance, et al.) is implemented. In the section entitled ‘Client Code Perspectives’, this has been discussed.

2 & 3 User and Password – essential in any multiuser system.

4. A language code (‘EN’ for English, ‘DE’ for German et al.) is to be input here. Language code field in the SAP tables is a single character, but while presented on screen, appears as two characters. This is the typical situation prevailing in the SAP environment. The internal storage of data fields in the tables is different from the manner in which it is presented on the screen and printer. During SAP installation, you can specify a default language for the installation. You can also specify the language/s to be installed. (All the languages supported by SAP need not be installed.) The German language (‘DE’) gets installed mandatorily. You cannot omit its installation. If you leave the Language field blank on the login screen, login will occur with the default language.

The User and Language are case insensitive. The Password is case sensitive.

SAP GUI

After the entries on this screen have been accepted, the SAP opening screen appears called the SAP Easy Access. (Figure 1-5)

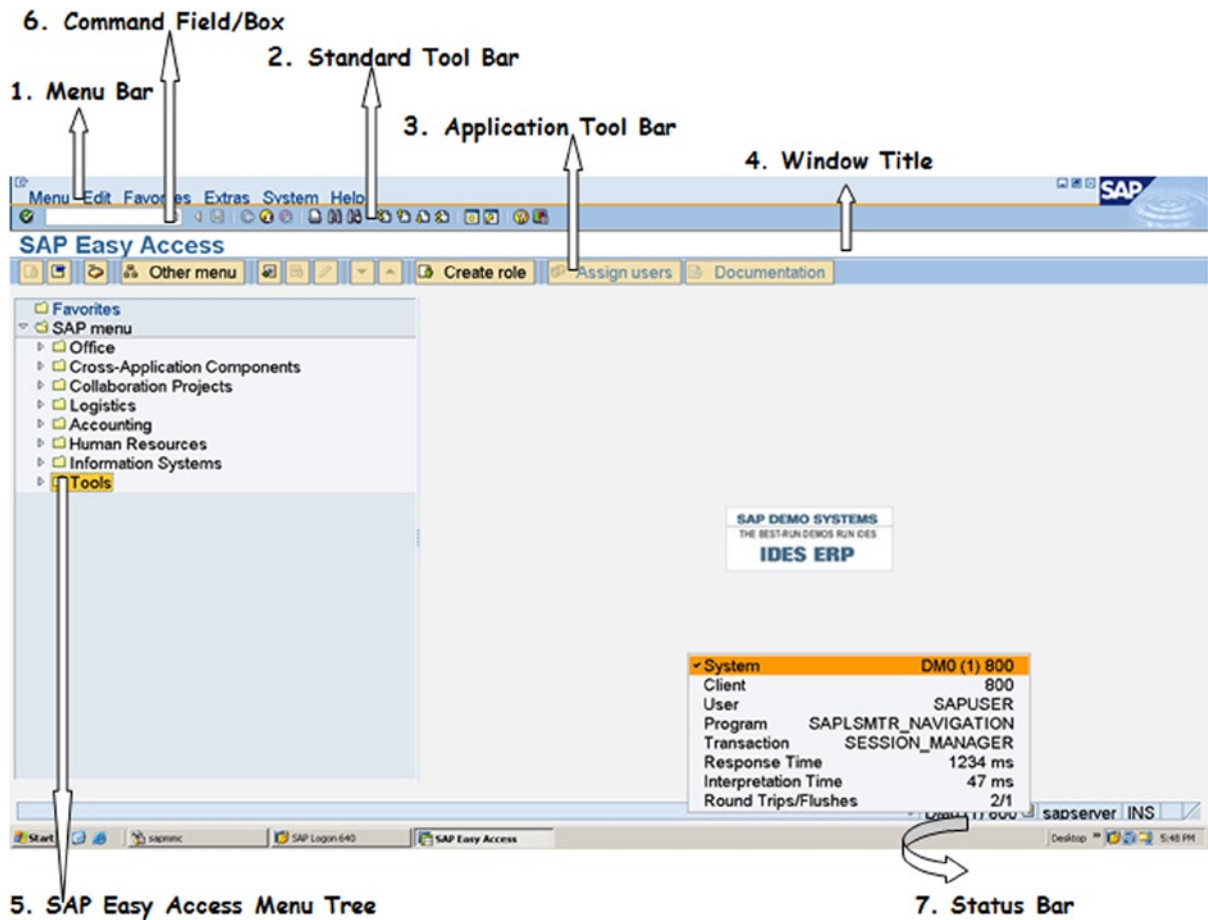
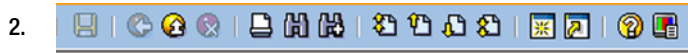



Figure 1-5. SAP Easy Access Screen

In the SAP Easy Access screen, the following areas, indicated by arrows in Figure 1-5, need to be recognized:

1. Menu Edit Favorites Extras System Help

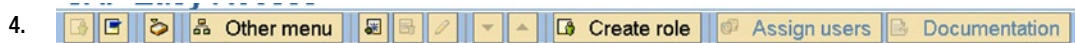
The SAP Menu Bar, Menu Options: This will vary screen to screen except for the last two bars, System and Help.



The Standard Tool Bar: The buttons or icons appearing in this area are the same from screen to screen. Buttons or icons are enabled or disabled depending on the context. The  button is for saving. On the opening SAP Easy Access screen, this is in a disabled state, as there is nothing to save on this screen.

3. SAP Easy Access

The Window Title: This contains the title of the window.



The Application Tool Bar: The buttons or icons on the Application Tool Bar vary from screen to screen.

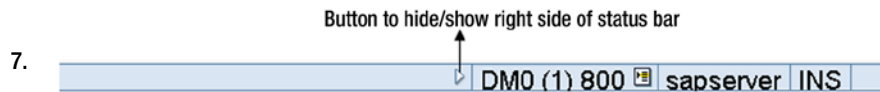


SAP Easy Access Menu Tree: In the *SAP GUI environment*, you need to navigate to different screens to carry out different tasks. One way of navigating to different screens is to use the menu tree. Clicking on or expanding the nodes of the menu tree will give access to subnodes and to the innermost subnode; clicking on this innermost subnode will enable you to navigate to different screens to carry out the various tasks. If you wanted to create an ABAP program, you will (1) click on the node ‘Tools’ (at the bottom), and subnodes will appear. (2) Click on the subnode ‘ABAP Workbench’, and further subnodes will appear. (3) Click on the subnode ‘Development’, and further subnodes will appear. (4) Click on the last level of subnode ‘ABAP Editor’. This will take you to the screen supporting the ABAP program maintenance. This is the ABAP editor screen. This sequence of nodes/subnodes will appear, provided nobody has customized the SAP Easy Access screen from its initial default. In this manner the user can navigate to various screens to carry out different tasks.

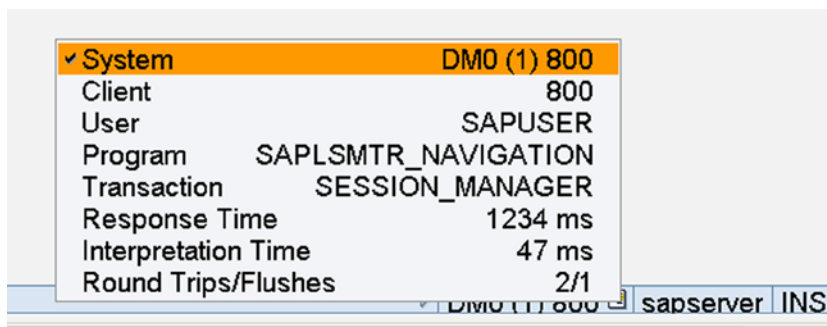


Command Field: Using the command field is another way of navigating to screens to carry out various tasks by the SAP user. One mode has been described under SAP Easy Access Menu Tree. The other shorter mode is to enter the transaction code in the command field and press the <Enter> key. For instance, again, you want to navigate to the ABAP editor screen. The transaction code to navigate to the ABAP editor screen is ‘SE38’. You enter SE38 in the command field and press <Enter>. If you want to navigate to the ABAP dictionary screen, the transaction code is ‘SE11’. These are predefined transaction codes provided by SAP. You can also create your own transaction codes.

You will create your own transaction code during in the Chapter14 entitled 'Screen Programming'. There are more than 100,000 predefined SAP provided transaction codes available in the *SAP GUI environment*. You will be familiar with around 15 transaction codes by the time you finish reading this book. The command field can be made to appear/disappear through the button at the right.



Status Bar: This is at the bottom of the screen. In the right side area of the status bar, system-related information such as 'system id,' 'system no,' log-in client, application server name and status of keyboard INS key on/off is presented. The left side of the status bar is used for notifications to the user through messages during program execution. If the button appearing on the status bar after 'system id,' 'system no' and log-in client is clicked; a pop-up like the one shown below appears. You get to know the program, transaction code getting executed, and login client et al.



Most of the icons in the SAP GUI environment are peculiar and specific to the SAP GUI environment. You will get familiar with them and what they signify as you operate in the SAP GUI environment.












Function keys can be used in the SAP Workbench environment to perform operations. A partial list is available in Table 1-4.

Table 1-4. Function Keys Operations


Function Key	Operation
F1	When the cursor is on a screen field, help text for the screen field pop-ups. When the cursor is on an ABAP program keyword, ABAP documentation for the ABAP program keyword in a new session pop-ups.
F2	Mouse Double Click.
F3	Goes to previous screen.
F4	Selection List pop-ups.
F5	To create a new object.
F6	To edit or change an existing object.
F7	To display an existing object.
F8	To execute a program.

A list of very commonly used icons or buttons (on the system and application tool bars) is available in Table 1-5.

Table 1-5. *Very Commonly Used Icons or Buttons*

Icon or Button	Description
s	Equivalent to pressing <enter> key. The <enter> key has a different meaning in the SAP GUI environment. When you are finished with a screen, you press the <enter> key. To navigate forward from one field to another, use the [tab] key; to navigate backwards between fields use the [shift]+ [tab] key combination.
	Universal Save
	Navigate to previous screen [F3]
	Exit screen
	Cancel operation
	Find
	Create session
	Execute [F8]
	Create object [F5]
	Display object [F7]
	Change or Edit object [F6]
	Delete

SAP External Sessions, Sessions' Control

In a standard tool bar, the  button is to create new external sessions. Alternatively, you can select the menu option 'System' and select the sub-option 'Create Session' to do the same. The system creates a new task or window. In this manner, a single SAP logged-in user can create a maximum of 16 windows, which are called external sessions. The Basis administrator can reduce this limit to less than 16. This limit of 16 is excluding the external sessions the system spawns when a nonclassical debugger is used and when the function key F1 is pressed on the keyword of an ABAP program. These are external sessions (simply different windows operating system tasks) and there is the concept of internal sessions. The difference between the two will be known when the concept of internal sessions is introduced in Chapter 7.

A developers' work involves handling of multiple objects like ABAP program, ABAP dictionary objects, screens, GUI status etc. One way of carrying out the developmental work involving multiple objects is to have multiple windows or external sessions. In one external session, the developer would be operating the ABAP editor; in a second, operating ABAP dictionary; in a third, the screen painter; in a fourth, the menu painter and so on.


The external sessions can be manipulated through the command box. Table 1-6 summarizes this:

Table 1-6. SAP Sessions' Handling

Command Box Content	Explanation
/NXXXX	This terminates the current transaction and starts Transaction XXXX (example: /NSE38).
/N	This terminates the transaction. This generally corresponds to pressing F15 to go back.
/NEND	This terminates all separate sessions and logs off (corresponds to System - Logoff).
/NEX	This terminates all separate sessions and logs off immediately (without any warning!).
/NOXXXX	This opens a new session and starts transaction XXXX in this session.
/O	This lists existing sessions and allows deletion or opening of a new session.
/I	This terminates the current session (corresponds to System End).

Perform each of the commands in the table as an exercise.

ABAP Workbench

ABAP – Advanced Business Application Programming is SAP's propriety language, earlier called ABAP/4: '4' signified a fourth-generation language and has been dropped now that most of the languages are fourth-generation languages. In the SAP environment, except for very few kernel programs, (like login screen) every other program is an ABAP program, and the entire SAP ERP software has been written in ABAP. When you click on the  button on the status bar of SAP GUI screen, the name of the ABAP program in execution is available. Refer to SAP GUI 7. ABAP is not just a language. It is a full-fledged developmental platform with a plethora of tools. The whole gamut of developmental tools such as ABAP editor, ABAP dictionary, Function module (general purpose, generic subroutines) builder, Class builder, Screen painter, Menu painter et al. constitute the ABAP Workbench. As the chapters of this book unfold, these workbench tools will be introduced: how to access them (transaction codes), how to operate them, and settings, etc. will be elaborated.

In this book, the focus is on the creation of workbench objects as per scenarios. The operations required to create these objects is described. For more elaborate descriptions of workbench operations, you can refer to the downloaded documents: BC ABAP Workbench Tools and BC ABAP Workbench Tutorial.

Client Code Perspectives

In the SAP login section, you have seen that the first field on the login screen is the client or client code. It is a three-digit number. When a new user is created by the Basis administrator, client code will be part of it; a user cannot be created without a client code. And the client associated with a user has to exist on the system. It was mentioned that the client code is one of the parameters used to implement the segregation of SAP environments: development, quality, testing and live. You will then get a clearer picture of the environment segregation. A functional consultant will view the client code very differently; a basis administrator will perceive a client differently than a functional consultant. You, as an ABAP consultant or an aspiring ABAP consultant, will view the client code from a technical and developer perspective.

When SAP software is installed, the SAP database gets created automatically as part of the installation process. This database (in version E.C.C. 6.0) contains more than 70,000 tables. Almost all of these tables will be empty after installation. A few of the tables envisaged contain the universal nature of data such as country codes and names; states/provinces/regions of countries; and currency codes and names will be populated with data after SAP installation.

Every developmental object that a developer creates inside SAP such as programs, screens, and ABAP dictionary objects, etc. are stored in tables of SAP database.

A majority of the 70,000+ tables of SAP database will store the functional module data. To mention simple examples: data relating to customers; suppliers or vendors; business documents data such as customers' bills, purchase orders, etc. The tables storing the functional module data invariably contain in the table structure a field of ABAP dictionary type 'CLNT' as the first field. ABAP dictionary supports its own 20 odd data types; 'CLNT' is one of them. The next chapter (Chapter 2) will introduce these 20 odd data types. Most often the name of this first field of type 'CLNT' is 'MANDT'.

All tables containing in its structure the first field of type 'CLNT' are termed client dependent tables. Tables not containing this field as the first field in the structure are termed client independent tables. Shown below in Figure 1-6 is a partial structure of a functional module (client dependent) table and in Figure 1-7, a partial structure of a client independent table.

Transp. Table		KNA1		Active					
Short Description		General Data in Customer Master							
Attributes		Delivery and Maintenance		Fields		Entry help/check		Currency/Quantity Fields	
Field		Key	Initia	Data element	Data Ty	Length	Decim	Predefined Type	
MANDT		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT	3		0 Client	
KUNNR		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	KUNNR	CHAR	10		0 Customer Number 1	
LAND1		<input type="checkbox"/>	<input type="checkbox"/>	LAND1_GP	CHAR	3		0 Country Key	
NAME1		<input type="checkbox"/>	<input type="checkbox"/>	NAME1_GP	CHAR	35		0 Name 1	

First field - Client Code

TYPE - CLNT

Figure 1-6. Table KNA1 Structure - Client Dependent

Field	Key	Initia	Data element	Data Ty	Length	Decim	Short Description
<u>PGMID</u>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<u>PGMID</u>	CHAR	4	0	0 Program ID in Requests and Tasks
<u>OBJECT</u>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<u>TROBJTYPE</u>	CHAR	4	0	0 Object Type
<u>OBJ_NAME</u>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<u>SOBJ_NAME</u>	CHAR	40	0	0 Object Name in Object Directory
<u>KORRNUM</u>	<input type="checkbox"/>	<input type="checkbox"/>	<u>TRKORR_OLD</u>	CHAR	10	0	0 Request/task up to and including Re

↓
First field TYPE 'CHAR' not 'CLNT'

Figure 1-7. Table TADIR Structure – Client Independent, first field not TYPE CLNT

It has been mentioned earlier that SAP supports different database brands: Oracle, MS SQL server, Informix, ADABAS, and IBM DB/2. You can have any one of these installed. (Database brand can be specified during SAP software installation time. Installation of database software forms part of SAP software installation.) This creates an ABAP program portability problem. Suppose you have created an ABAP program using Oracle SQL statements. This program would run as long as the installed database of SAP installation is Oracle. If the installed database is other than Oracle, the source program needs to be modified. SAP got over this portability problem by introducing its own SQL called the ‘Open SQL’. If programs are written using Open SQL statements, they are portable across all the supported brands of databases. The ABAP language supports Open SQL as well as the SQL of the supported database brands. In practice, invariably the Open SQL is used; the use of database SQL is rare and infrequent.

When data is retrieved or updated through SELECT or UPDATE statements using Open SQL statement for any of the functional module tables (i.e., the client dependent tables), the system will retrieve or update only data belonging to the client code with which user has logged in. To explain in specific terms: A user has logged into client 800. Once logged in, user executes an ABAP program using Open SQL SELECT statement to retrieve data from a client dependent table, say TCURC (table containing the currency codes, data of universal nature); only rows having MANDT = 800 are retrieved. It is as if an implicit ‘WHERE MANDT = 800’ is executed. The table will contain data belonging to template client code/s such as client code 000 and/or client code 001. (Template clients are explained in the following text in this section.) The table could contain data of non-template clients, if non-template clients exist.

When data is inserted through the INSERT statement of Open SQL statement for any client dependent tables, the system will automatically assign logged-in client values to the client code fields (i.e., a row will be inserted with a client code value equal to the value of the logged-in client code). Similar will be the case for the SQL DELETE statement; only rows belonging to the client in which the user is logged in will be operated upon by the DELETE statement of Open SQL.

With client dependent tables, Open SQL statements, in normal course, will operate only on data belonging to client in which the user is logged in.

The SQL commands INSERT, UPDATE, DELETE will not be used in an ABAP program by a developer in the SAP implementation or support team on the SAP delivered tables. The SAP delivered tables are strictly maintained by SAP provided programs for the most part.

You are seeing the effect of segregation. The SAP implementation environments are associated and assigned a specific client code. In an installation, the development environment would be one client, say 100; testing and quality would be a second client, say 200; live, a third client say 300; end-user training, a fourth client, say 400 and so on.

When SAP software is installed, as part of the installation process, template client/s gets created. The option is to either create 000 or 001 or both. These are classified as template clients. The template clients are not to be operated in. The template clients are copied into the environmental clients such as developmental, testing/quality, and live. For example, suppose you decide to have client 100 for developmental environment, 200 for testing/quality, 300 for live. Then these would be derived by making copies of 000/001 into 100, 200, and 300. The process of copying (template or even non-template) clients is a task carried out by the basis administrator.

The difference between the template clients 000 and 001 is that the client 001 contains the special functionalities of Euro sales tax VAT. (Value Added Tax). If Euro VAT functionalities are required (implementation or support is being done for a customer located in European Union), 001 needs to be copied to non-template/operational clients, or else 000 needs to be copied to non-template/operational clients. One can, of course, copy one non-template to another non-template client.

There is one other client that gets created as part of SAP software installation. This is client 066 called the 'early watch system'. When SAP is installed initially in an enterprise, it could face performance and response time issues. The SAP On-line Support System – OSS - offers its services to help fix these issues. For this, the SAP OSS requires login into the enterprise SAP system. They would log through client 066. After all, the SAP OSS should not access the enterprise data.

SAP IDES (Internet Demonstration & Evaluation System) Server and Usage of IDES Server for Trainings

When SAP software is installed, as has been mentioned earlier, except for the few tables containing data of universal nature, such as countries, states/provinces/regions, currencies et al., all other functional module tables will be empty. It has also been mentioned that the SAP functional module tables are populated through SAP provided ABAP programs that perform rigorous validation before data is accepted and inserted into the said tables.

Also, data can be inserted into functional module table, (through SAP provided ABAP programs) provided the functional modules have been configured and customized by the functional consultants. To give one scenario, if you want to create a new customer (i.e., insert data into customer tables), the SD (Sales and Distribution) or FI (Accounting/Finance) module should have been configured and customized by the respective functional consultants.

When SAP software is installed, no module is configured and customized. If this were so, there would be no need for the costly exercise of implementation through functional consultants.

Since the SAP functional module tables are maintained by SAP provided ABAP programs, ABAP consultants implementing and supporting SAP projects are absolved of this task of maintaining data of SAP tables. One of the major tasks required to be performed by ABAP consultants implementing and supporting SAP is retrieving data (SQL SELECT statements) and presenting data (reports).

To get good exposure to the data retrieval aspect, you need to do fairly complex exercises in data retrieval. With no data in the SAP functional module tables of bare SAP installed software, this would be impossible. Even if modules were configured and customized, it would require some skills to manually enter data. And manually you can't be creating too much of data: data running into thousands of rows. Not everybody can have access to SAP installation with substantial data in functional module tables.

The IDES server is preconfigured and customized for a number dummy business enterprises and contains lot of dummy data in functional module tables.

As its name suggests, SAP initially intended the IDES server for demonstration of software to prospective buyers. But now it is being used everywhere for training, especially for functional modules' trainings. To those undergoing training in the functional modules, the preconfigured dummy data of more than a dozen companies is a very handy reference.

Most ABAP trainings, including the certification training, use SAP provided tables of 'Flight Data System' for exercises. These tables have been specially created and provided for ABAP trainings. But these are small tables. (Small in terms of number of fields, small in terms of number of tables, less than a dozen.) These tables are available in all servers. They can be populated by data through executing an ABAP program. They are of hypothetical nature.

But why use hypothetical tables when the real-world ones are available? You will use a few tables of sales and purchase in your exercises. People not exposed to the business and commercial world can easily relate to the basic sales and purchase functionalities. Though these tables on an average contain more than 100 fields, you will be restricted to using a few fields, maybe 15-20. But you will get the feel and exposure of the SAP real-world tables while doing the exercises with these tables.

Typically, the IDES server installed with default and not custom options creates four clients: 800, 810, 811, and 812. The client 800 could be used as it contains substantial dummy data. Typically, it contains:

- Data of more than 15 companies' codes.
- More than 7,000 customers.
- More than 1,500 suppliers or vendors.
- More than 30,000 billing documents.
- More than 10,000 purchasing documents.

Terminology Clarification

The functional modules have been described. In the ABAP Workbench, there is an object 'Function Module.' The 'Function Modules' are general purpose routines existing in the library area. They can be called from multiple ABAP programs (Reusability). The function modules are the most powerful construct of the procedure-oriented ABAP. This will be covered in Chapter 7 entitled 'Modularization.' The two similar sounding terms – functional modules and function modules – should not be confused. Sometimes, functional consultants refer to functional modules as function modules!

The word *company* refers to a business enterprise. In the SAP environment, the word company has a different meaning. In the SAP environment, the word company is used to refer to a business group. For instance, suppose there is a business group called 'Super Group.' This business group operates a number of registered companies or business enterprises such as 'Super Iron & Steel Co. Inc.', 'Super Motors Inc.', and 'Global Beverages Inc.'. The registered companies in the SAP environment are termed as company codes. It is important to know this terminology.

Conclusion

In this chapter, you learned the basics of ERP, SAP, and SAP GUI environments. You have also learned about client code and have been apprised of the suitability of SAP IDES server for SAP trainings. In the forthcoming chapters you will be learning technical aspects of the ABAP developmental platform. In the next two chapters, you will be learning about the *SAP ABAP dictionary* or *data dictionary* (DDIC) as it is popularly known.

CHAPTER 2



ABAP Dictionary/Data Dictionary/DDIC – 1

ABAP Dictionary, *data dictionary*, *dictionary*, and *DDIC* are synonymous terms. *DDIC* — a contraction of *data dictionary* — will be used throughout this book. The DDIC is a multifaceted facility that is a substantial feature of the ABAP Development Workbench environment.¹

Any developmental activity invariably uses DDIC objects. This chapter and the next one list and identify DDIC objects that will be covered in subsequent chapters. This chapter also describes the major attributes of the DDIC, introduces the DDIC Data Types, and teaches the reader to create Domains, Data Elements, and Transparent Tables through hands-on exercises framed in a simple business scenario.

■ **Note** The word *object* is used in this book in its natural language meaning of thing. *Object* will not be used in the context of OOP unless explicitly stated.

ABAP Dictionary: Initial Screen

To navigate to the screen for maintaining and viewing DDIC objects from the SAP Easy Access screen, enter the transaction code SE11 in the command field. Alternatively, you can always navigate from the SAP Easy Access menu tree. A screenshot of the ABAP Dictionary: Initial Screen is shown in Figure 2-1. (The term *ABAP Dictionary* is used here to match the window title in Figure 2-1. Throughout this book, the term *DDIC* will be used by default).

¹For a detailed theoretical treatment of the DDIC, download *BC – ABAP Dictionary* from <http://help.sap.com/printdocu/core/print46c/en/data/pdf/bcdwbdic/bcdwbdic.pdf>.

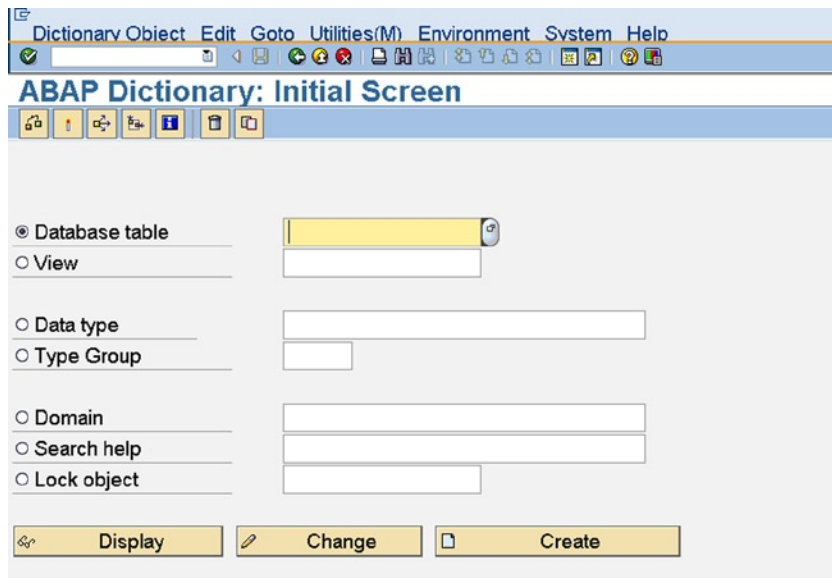



Figure 2-1. ABAP Dictionary: Initial screen

A complete list of DDIC objects follows. The non-struck items in the list are covered in this chapter and the next one. The struck-out items in the list will be covered in subsequent chapters as detailed below, after the reader has acquired prerequisite knowledge.

- Database table
 - Transparent
 - Pooled
 - Cluster
- View
 - Database
 - Projection
 - Maintenance (Chapter 7)
 - Help
- Data type
 - Data element
 - Structure (Chapter 6)
 - Table type (Chapter 6)
- Type group (Chapter 4)
- Domain
- Search help
- Lock object (Chapter 14)

View Table Structure Definitions of the SAP Delivered VBRK and VBRP Tables

You will be using the VBRK and VBRP tables frequently in the hands-on programming exercises, so let's view the structure of these table definitions as a first exposure to the DDIC object database table. To view the VBRK structure definitions, click on the Database Table radio button, enter the table name VBRK, and click on the  Display button or function key F7. The structure definitions of the VBRK database table are displayed as shown in Figure 2-2.

Data Type

Table Field names **number of fields in the structure**

Field	Data element	Data Type	Reference table	Ref. field	Short Description
HANDT	HANDT	CLNT			Client
VBELN	VBELN_VF	CHAR			Billing Document
FKART	FKART	CHAR			Billing Type
FKTYP	FKTYP	CHAR			Billing category
VBTYP	VBTYP	CHAR			SD document category
WAERK	WAERK	CUKY			SD Document Currency
YKORG	YKORG	CHAR			Sales Organization
VTWEG	VTWEG	CHAR			Distribution Channel
KALSH	KALSMASD	CHAR			Sales and Distribution: Pricing Procedure in Pricing
KNUMV	KNUMV	CHAR			Number of the document condition

Figure 2-2. Structure definitions of the VBRK database table

The VBRK table consists of 97 fields. The VBRP table consists of 190 fields. You can locate a field by its name by using the Find button. Functional module tables in the SAP environment typically consist of more than 100 fields.

These tables store data of customer bills. A single bill might consist of 1 or n number of items. The bill (called Billing Document in SAP parlance) information is located in two tables. One table, designated the *header table*, contains the header information of the bill, with one row for each bill. The second table, designated the *item table*, contains the item information of the bill, with as many rows as the number of items in each bill.

DDIC Attributes

The prime attributes of the DDIC are described in the following sections.

Metadata

The DDIC can be viewed as metadata (data about data) that resides in the SAP database along with the metadata maintained by the database (Figure 2-3).



Figure 2-3. The two distinct sets of metadata in the SAP database: Database metadata and DDIC metadata

Standardization and Uniformity

SAP ECC ERP Version 6.0 has more than 70,000 tables in its database. Initially, SAP R/2 versions had many fewer tables in their database — typically only a few thousand. Yet whenever the number of tables in a database runs into thousands, two issues need to be addressed in the design and maintenance of the database: standardization and uniformity.

Standardization: When a database has thousands of tables, a field — such as customer code — might occur in multiple tables. And in this manner, many more fields would be occurring in multiple tables. Any field occurring in two or more tables must have identical type and length in all its table occurrences. This enforcement of standardization on fields occurring in multiple tables is implemented through the DDIC object *Domain*, graphically represented in Figure 2-4.

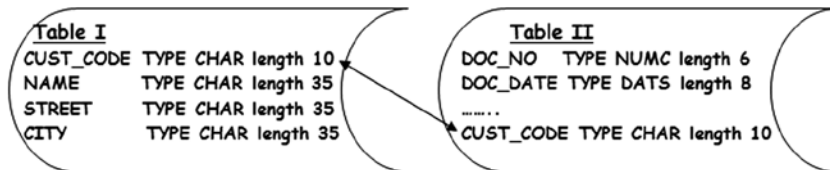


Figure 2-4. A field occurring in two tables should have identical data type and length in both

Uniformity: The name of a field in a table is the technical name for it used by technical personnel and developers. It is not expected to make sense to an end user. When a field appears on a screen, however, it must carry a field label that will make sense to the end user.

An example: The field name of a customer code in the SAP database table KNA1 is KUNNR, which would mean nothing to end users. For a user logged in English, the field KUNNR should appear on the screen with a label such as Customer Code — and similarly for users who are logged in in other languages. When a given field appears on multiple screens, care must be taken to ensure that the label is uniform across screens for a given login language. These requirements are implemented through the DDIC object *Data Element* (Figure 2-5).

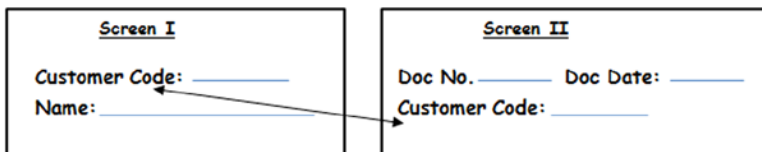


Figure 2-5. Field names should appear identically across different screens