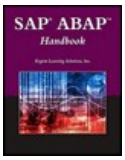

SAP ABAP Handbook

by Kogent Learning Solutions, Inc.

Chapter 1: A Gateway to SAP Systems

Overview

Systems Applications and Products in Data Processing (SAP) is business software that integrates all applications running in an organization. These applications represent various modules on the basis of business areas, such as finance, production and planning, and sales and distribution, and are jointly executed to accomplish the overall business logic. SAP integrates these modules by creating a centralized database for all the applications running in an organization. You can customize SAP according to your requirements by using the Advanced Business Application Programming (ABAP) language. ABAP, also referred to as ABAP/4, is the technical module of SAP and the fourth-generation programming language used to create applications related to SAP.

The SAP system was introduced as an Enterprise Resource Planning (ERP) software designed to coordinate all the resources, information, and activities to automate business processes, such as order fulfillment or billing. Nowadays, the SAP system also helps you to know about the flow of information among all the processes of an organization's supply chain, from purchases to sales, including accounting and human resources.

Integration of different business modules is a key factor that separates SAP from other enterprise applications. Integration of business modules helps to connect various business modules, such as finance, human resources, manufacturing, and sales and distribution, so that the data of these modules can be easily accessed, shared, and maintained across an enterprise. Integration also ensures that a change made in one module is reflected automatically on the other modules, thereby keeping the data updated at all times.

In this chapter, you learn about SAP and its need in today's businesses. This chapter also deals with the importance of ERP and its implementation in SAP. The chapter provides a comprehensive history of SAP, focusing on the circumstances that necessitated its development, and, finally, on how its introduction helped to improve system performance and business efficiency. In addition, this chapter describes the need for the ABAP/4 language in SAP and also explains the architecture of the SAP system, including its three views: logical, software-oriented, and user-oriented. It also explores the various components of the application servers that are used in SAP, such as work processes, the dispatcher, and the gateway, and describes the structure and types of work processes. You also learn how to dispatch a dialog step, a procedure that helps a user to navigate from one screen to another in the SAP system, as well as two important concepts: user context and roll area, which are memory areas that play an integral role in dispatching dialog steps and in implementing a work process. This chapter also explains the client-dependency feature of SAP. The chapter concludes with a brief discussion on the integrated environment of SAP.

Explaining the Concept of an ERP System

A system that automates and integrates all modules of business areas is known as an ERP system or simply ERP. An ERP system is used to integrate several data sources and processes, such as manufacturing, control, and distribution of goods in an organization. This integration is achieved by using various hardware and software components. An ERP system is primarily module-based, which implies that it consists of various modular software applications or modules. A software module in an ERP system automates a specific business area or module of an enterprise, such as finance or sales and distribution. These software modules of an ERP system are linked to each other by a centralized database.

A centralized database is used to store data related to all the modules of the business areas. Using a centralized database ensures that the data can be accessed, shared, and maintained easily. Combined with the module-based implementation, an ERP system improves the performance and efficiency of business processing.

Before the advent of the ERP system, each department of a company had its own customized automation mechanism. As a result, the business modules were not interconnected or integrated, and updating and sharing data across the business modules was a big problem. Let's use an example to understand this concept better. Suppose the finance and sales and distribution modules of an enterprise have their respective customized automation mechanisms. In such a setup, if a sale is closed, its status would be updated automatically in the sales and distribution module. However, the updated status of the sale of an item would not be updated in the finance module automatically. Consequently, the revenue generated from the sale of an item would need to be updated manually in the finance module, resulting in a greater probability of errors and an asynchronous business process. The problem was fixed with the help of the integration feature built into the ERP system.

Another benefit of the ERP system is that it helps synchronize data and keep it updated. Ideally, an ERP system uses only a single, common database to store information related to various modules of an organization, such as sales and distribution, production planning, and material management.

Despite the benefits of the ERP system, the system has certain drawbacks. Some of the major drawbacks of the ERP system are:

- Customization of ERP software is restricted because you cannot easily adapt ERP systems to a specific workflow or business process of a company.
- Once an ERP system is established, switching to another ERP system is very costly.
- Some large organizations may have multiple departments with separate, independent resources, missions, chains-of-command, etc., and consolidation into a single enterprise may yield limited benefits.

SAP was introduced to overcome the drawbacks of the contemporary ERP systems. The introduction of SAP systems not only removed the preceding bottlenecks but also led to improved system performance and business efficiency by integrating individual applications. In other words, an SAP system ensures data consistency throughout the system, in addition to removing the drawbacks of the contemporary ERP systems.

Next, let's explain why and how an SAP system is introduced in business processing.

History of SAP Systems

SAP is a translation of the German term Systeme, Anwendungen, und Produkte in der Datenverarbeitung. It was developed by SAP AG, Germany. The basic idea behind developing SAP was the need for standard application software that helps in real-time business processing. The development process began in 1972 with five IBM employees: Dietmar Hopp, Hans-Werner Hector, Hasso Plattner, Klaus Tschira, and Claus Wellenreuther in Mannheim, Germany. A year later, the first financial and accounting software was developed; it formed the basis for continuous development of other software components, which later came to be known as the SAP R/1 system. Here, R stands for real-time data processing and 1 indicates single-tier architecture, which means that the three networking layers, Presentation, Application, and Database, on which the architecture of SAP depends, are implemented on a single system. SAP ensures efficient and synchronous communication among different business modules, such as sales and distribution, production planning, and material management, within an organization. These modules communicate with each other so that any change made in one module is communicated instantly to the other modules, thereby ensuring effective transfer of information.

The SAP R/2 system was introduced in 1980. SAP R/2 was a packaged software application on a mainframe computer, which used the time-sharing feature to integrate the functions or business areas of an enterprise, such as accounting, manufacturing processes, supply chain logistics, and human resources. The SAP R/2 system was based on a two-tier client-server architecture, where an SAP client connects to an SAP server to access the data stored in the SAP database. SAP R/2 was implemented on the mainframe databases, such as DB/2, IMS, and Adabas. SAP R/2 was particularly popular with large European multinational companies that required real-time business applications, with built-in multicurrency and multilanguage capabilities. Keeping in mind that SAP customers belong to different nations and regions, the SAP R/2 system was designed to handle different languages and currencies. The SAP R/2 system delivered a higher level of stability compared to the earlier version.

Note Time-sharing implies that multiple users can access an application concurrently; however, each user is unaware that the operating system is being accessed by other users.

SAP R/3, based on a client-server model, was officially launched on July 6, 1992. This version is compatible with multiple platforms and operating systems, such as UNIX and Microsoft Windows. SAP R/3 introduced a new era of business software—from mainframe computing architecture to a three-tier architecture consisting of the Database layer, the Application layer (business logic), and the Presentation layer. The three-tier architecture of the client-server model is preferred to the mainframe computing architecture as the standard in business software because a user can make changes or scale a particular layer without making changes in the entire system.

The SAP R/3 system is a customized software with predefined features that you can turn on or off according to your requirements. The SAP R/3 system contains various standard tables to execute various types of processes, such as reading data from the tables or processing the entries stored in a table. You can configure the settings of these tables according to your requirements. The data related to these tables are managed with the help of the dictionary of the SAP R/3 system, which is stored in an SAP database and can be accessed by all the application programs of SAP.

The SAP R/3 system integrates all the business modules of a company so that the information, once entered, can be shared across these modules. The SAP R/3 system is a highly generic and comprehensive business application system, especially designed for companies of various organizational structures and different lines of business.

The SAP R/3 system runs on various platforms, such as Windows and UNIX. It also supports various relational databases of different database management systems, such as Oracle, Adabas, Informix, and Microsoft SQL Server. The SAP R/3 system uses these databases to handle the queries of the users.

With the passage of time, a business suite that would run on a single database was required. This led to the introduction of the mySAP ERP application as a follow-up product to the SAP R/3 system. The mySAP ERP application is one of the applications within the mySAP Business Suite. This suite includes mySAP ERP, mySAP Supply Chain Management (SCM), mySAP Customer Relationship Management (CRM), mySAP Supplier Relationship Management (SRM), and mySAP Product Lifestyle Management (PLM). The latest release of the mySAP ERP application is SAP ERP Central Component (ECC6.0). The mySAP ERP categorizes the applications into the following three core functional areas:

- Logistics
- Financial
- Human resources

Note The book focuses on the latest release of the mySAP ERP application, i.e., ECC6.0.

As stated earlier, the runtime environment and integrated suite of application programs within the SAP R/3 system are written in a fourth-generation language, ABAP/4.

Need for ABAP

ABAP, or ABAP/4, is a fourth-generation programming language first developed in the 1980s. It was used originally to prepare reports, which enabled large corporations to build mainframe business applications for material management and financial and management accounting.

ABAP is one of the first programming languages to include the concept of logical databases, which provides a high level of abstraction from the centralized database of the SAP system. Apart from the concept of logical databases, you can also use Structured Query Language (SQL) statements to retrieve and manipulate data from the centralized database. To learn more about working with databases with the help of the SQL statements, refer to Chapter 8.

The ABAP programming language was used originally to develop the SAP R/3 system. That is, the runtime environment and application programs in the SAP R/3 system are written in the ABAP language. The SAP R/3 system provides the following set of applications, also known as functional modules, functional areas, or application areas:

- Financial Accounting (FI)
- Production Planning (PP)
- Material Management (MM)
- Sales and Distribution (SD)
- Controlling (CO)
- Asset Management (AM)
- Human Resources (HR)
- Project System (PS)
- Industry Solutions (IS)
- Plant Maintenance (PM)
- Quality Management (QM)
- Workflow (WF)

These functional modules are written in the ABAP language. In addition, you can use the ABAP language to enhance the applications that you create in the mySAP ERP system. For instance, besides the available reports and interfaces in the mySAP ERP system, you can create your own custom reports and interfaces.

The ABAP language environment, which includes syntax checking, code generation, and the runtime system, is a part of SAP Basis. SAP Basis, a component of an SAP system, acts as a technological platform that supports the entire range of SAP applications, now typically implemented in the framework of the SAP Web Application Server. In other words, the SAP Basis component acts as an operating system on which SAP applications run.

Similar to any other operating system, the SAP Basis component contains both low-level services, such as memory management and database communication, and high-level tools, such as SAP Smart Forms and log viewers, for end-users and administrators. You learn more about these concepts later in this book.

The ABAP language provides the following features:

- **Data sharing—** Enables you to store data in memory at a central location. Different users and programs can then access the data without copying it.
- **Exception handling—** Helps define a special control flow for a specific error situation and provide information about the error.
- **Data persistency—** Enables you to store data permanently in relational database tables of the SAP R/3 system.
- **Making enhancements—** Enables you to enhance the functionality of programs, function modules, and global classes, without modifying or replacing the existing code.

Exploring the Architecture of SAP R/3

As stated earlier, the SAP R/3 system evolved from the SAP R/2 system, which was a mainframe. The SAP R/3 system is based on the three-tier architecture of the client-server model. Figure 1.1 shows the three-tier architecture of the SAP R/3 system:

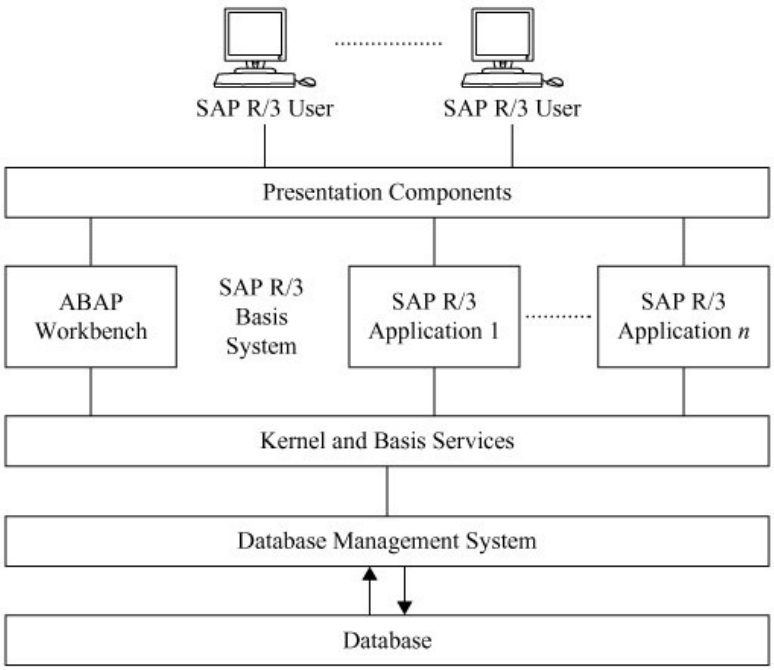

Figure 1.1: SAP R/3 architecture

Figure 1.1 shows how the R/3 Basis system forms a central platform within the R/3 system. The architecture of the SAP R/3 system distributes the workload to multiple R/3 systems. The link between these systems is established with the help of a network. The SAP R/3 system is implemented in such a way that the Presentation, Application, and Database layers are distributed among individual computers in the SAP R/3 architecture.

The SAP R/3 system consists of the following three types of views:

- Logical view
- Software-oriented view

- User-oriented view

The Logical View

The logical view represents the functionality of the SAP system. In this context, the R/3 Basis component controls the functionality and proper functioning of the SAP system. Therefore, in the logical view of the SAP R/3 system, we describe the services provided by the R/3 Basis component that help to execute SAP applications.

The following is a description of the various services provided by the R/3 Basis component:

- **Kernel and Basis services**—Provide a runtime environment for all R/3 applications. The runtime environment may be specific to the hardware, operating system, or database. The runtime environment is written mainly in either C or C++, though some parts are also written in the ABAP programming language. The tasks of the Kernel and Basis services are as follows:
  - Executing all R/3 applications on software processors (virtual machines).
  - Handling multiple users and administrative tasks in the SAP R/3 system, which is a multiuser environment. When users log on to the SAP system and run applications within it, they are not connected directly to the host operating system, since the R/3 Basis component is the actual user of the host operating system.
  - Accessing the database in the SAP R/3 system. The SAP R/3 Basis system is connected to a database management system (DBMS) and the database itself. R/3 applications do not communicate with the database directly; rather, these applications communicate with the database through the administration services provided by the R/3 Basis system.
  - Facilitating communication of SAP R/3 applications with other SAP R/3 systems and with non-SAP systems. You can access SAP R/3 applications from an external system by using the Business Application Programming Interfaces (BAPI) interface.
  - Monitoring and controlling the SAP R/3 system when the system is running.
- **ABAP Workbench service**—Provides a programming environment to create ABAP programs by using various tools, such as the ABAP Dictionary, ABAP Editor, and Screen Painter.
- **Presentation Components service**—Helps users to interact with SAP R/3 applications by using the presentation components (interfaces) of these applications.

The Software-Oriented View

The software-oriented view displays various types of software components that collectively constitute the SAP R/3 system. It consists of SAP Graphical User Interface (GUI) components and Application servers, as well as a Message server, which make up the SAP R/3 system. Since the SAP R/3 system is a multitier client-server system, the individual software components are arranged in tiers. These components act as either clients or servers, based on their position and role in a network. Figure 1.2 shows the software—oriented view of the SAP R/3 architecture:

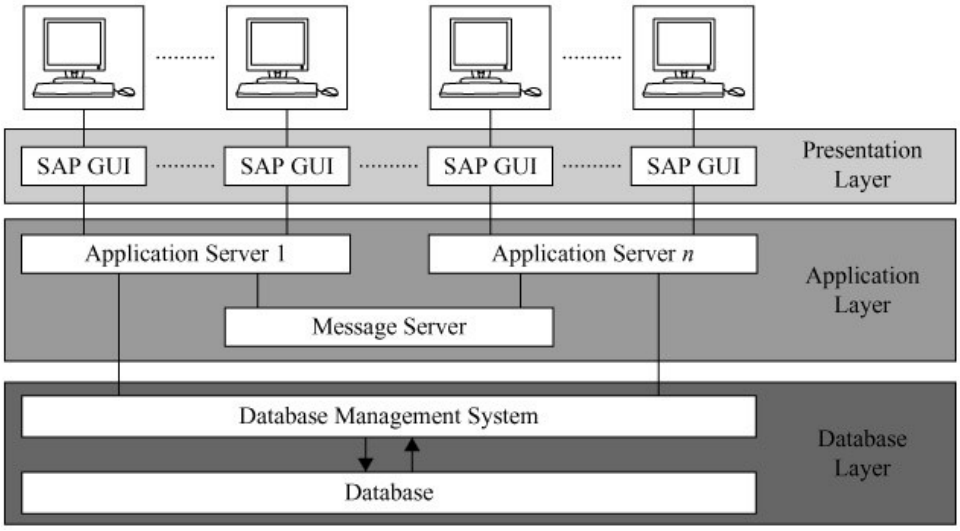

Figure 1.2: Software-oriented view

As shown in Figure 1.2, the software-oriented view of the SAP R/3 system consists of the following three layers:

- Presentation layer
- Application layer
- Database layer

Presentation Layer

The Presentation layer consists of one or more servers that act as an interface between the SAP R/3 system and its users, who interact with the system with the help of well-defined SAP GUI components. For example, using these components, users can enter a request, to display the contents of a database table. The Presentation layer then passes the request to the Application server, which processes the request and returns a result, which is then displayed to the user in the Presentation layer. While an SAP GUI component is running, it is also connected to a user's SAP session in the R/3 Basis system.

Note The servers in the Presentation layer have been referred to as Presentation servers in this chapter.

Application Layer

The Application layer executes the application logic in the SAP R/3 architecture. This layer consists of one or more Application servers and Message servers. Application servers are used to send user requests from the Presentation server to the Database server and retrieve information from the Database server as a response to these requests. Application servers are connected to Database servers with the help of the local area network. An Application server provides a set of services, such as processing the flow logic of screens and updating data in the database of the SAP R/3 system. However, a single Application server cannot handle the entire workload of the business logic on its own. Therefore, the workload is distributed among multiple Application servers. Figure 1.3 shows the location of the Application server between the Database and Presentation servers:

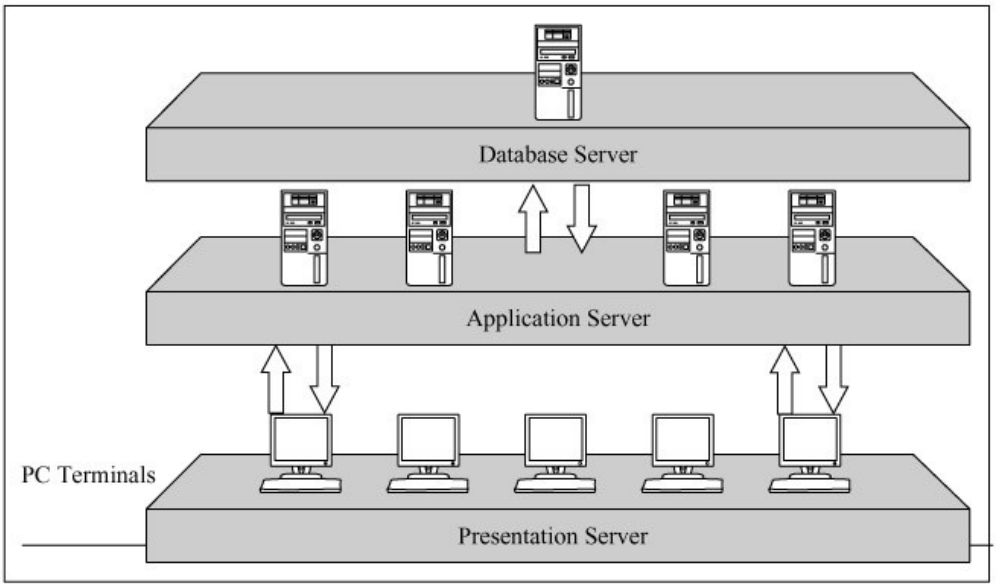

Figure 1.3: Application server

The Message server component of the Application layer (shown in Figure 1.2) is responsible for communicating between the Application servers. This component also contains information about Application servers and the distribution of load among these servers. It uses this information to select an appropriate server when a user sends a request for processing.

The separation of the three layers of the SAP R/3 system makes the system highly scalable, with the load being distributed among the layers. This distribution of load enables the SAP R/3 system to handle multiple requests simultaneously. The control of a program moves back and forth among the three layers when a user interacts with the program. When the control of the program is in the Presentation layer, the program is ready to accept input from the user, and during this time the Application layer becomes inactive for the specific program. That is, any other application can use the Application layer during this time. As soon as the user enters the input on the screen, the control of the program shifts to the Application layer to process the input and the Presentation layer becomes inactive, which means that the SAP GUI (the user interface of the SAP R/3 system) cannot accept any kind of input. In other words, until the Application layer completes processing the input and calls a new screen, the SAP GUI does not become active. The procedure in which a new screen is presented before the user is known as a dialog step. Dialog steps are processed in the Application layer, as shown in Figure 1.4:

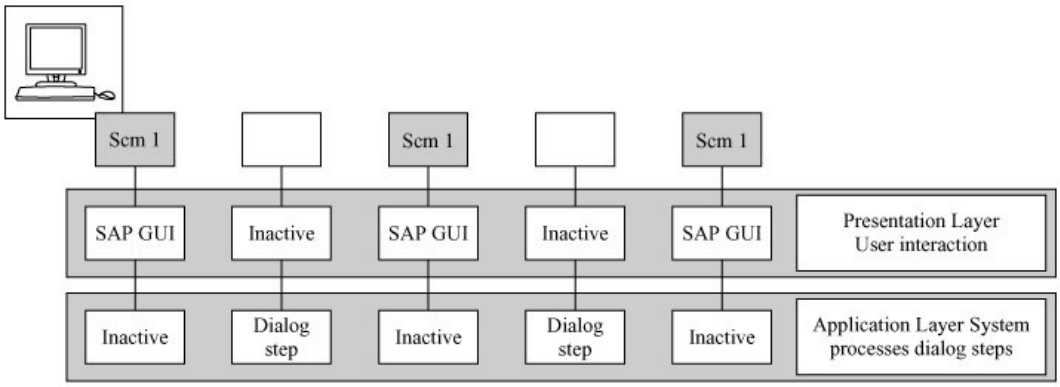

Figure 1.4: A dialog step

Database Layer

The Database layer of the SAP R/3 architecture comprises the central database system. The central database system has two components, DBMS and the database itself. The SAP R/3 system supports various databases, such as Adabas D, DB2/400 (on AS/400), DB2/Common Server, DB2/MVS, Informix, Microsoft SQL Server, Oracle, and Oracle Parallel Server.

The database in the SAP R/3 system stores the entire information of the system, except the master and transaction data. Apart from this, the components of ABAP application programs, such as screen definitions, menus, and function modules, are stored in a special section of the database, known as Repository, also known as Repository Objects. The database

also stores control and customized data, which govern how the SAP R/3 system functions. Distributed databases are not used in the SAP R/3 system because the system does not support them.

Note Master data is the core data, which is essential to execute the business logic. Data about customers, products, employees, materials, and suppliers are examples of master data. Transaction data refers to information about an event in a business process, such as generating orders, invoices, and payments.

The User-Oriented View

The user-oriented view displays the GUI of the R/3 system in the form of windows on the screen. These windows are created by the Presentation layer. To view these windows, the user has to start the SAP GUI utility, called the SAP Logon program, or simply SAP Logon. After starting the SAP Logon program, the user selects an SAP R/3 system from the SAP Logon screen. The SAP Logon program then connects to the Message server of the R/3 Basis system in the selected SAP R/3 system and retrieves the address of a suitable Application server; i.e., the Application server with the lightest load. The SAP Logon program then starts the SAP GUI connected to the Application server.

The SAP GUI starts the logon screen. After the user successfully logs on, the initial screen of the R/3 system appears. This initial screen starts the first session of the SAP R/3 system. Figure 1.5 shows the user-oriented view of the SAP R/3 system:

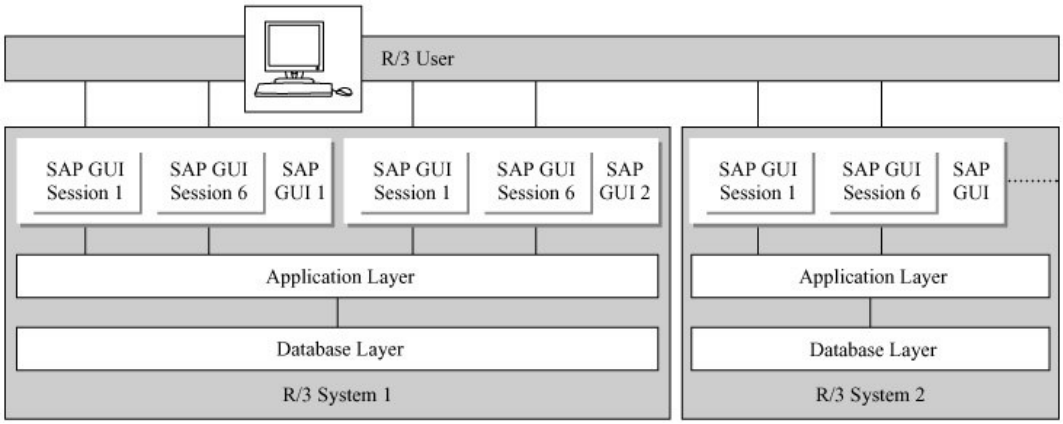

Figure 1.5: User-oriented view

A user can open a maximum of six sessions within a single SAP GUI. Each session acts as an independent SAP GUI. You can simultaneously run different applications on multiple open R/3 sessions. The processing in an opened R/3 session is independent of the other opened R/3 sessions.

Explaining the Architecture of the Application Server

One of the most important components of the SAP R/3 system is the Application server, where ABAP programs run. The Application server handles the business logic of all the applications in the SAP R/3 system. The Application layer consists of Application servers and Message servers. Application servers communicate with the Presentation and Database layers. They also communicate with each other through Message servers. Application servers consist of dispatchers and various work processes, discussed later in this chapter. Figure 1.6 shows the architecture of the Application server:

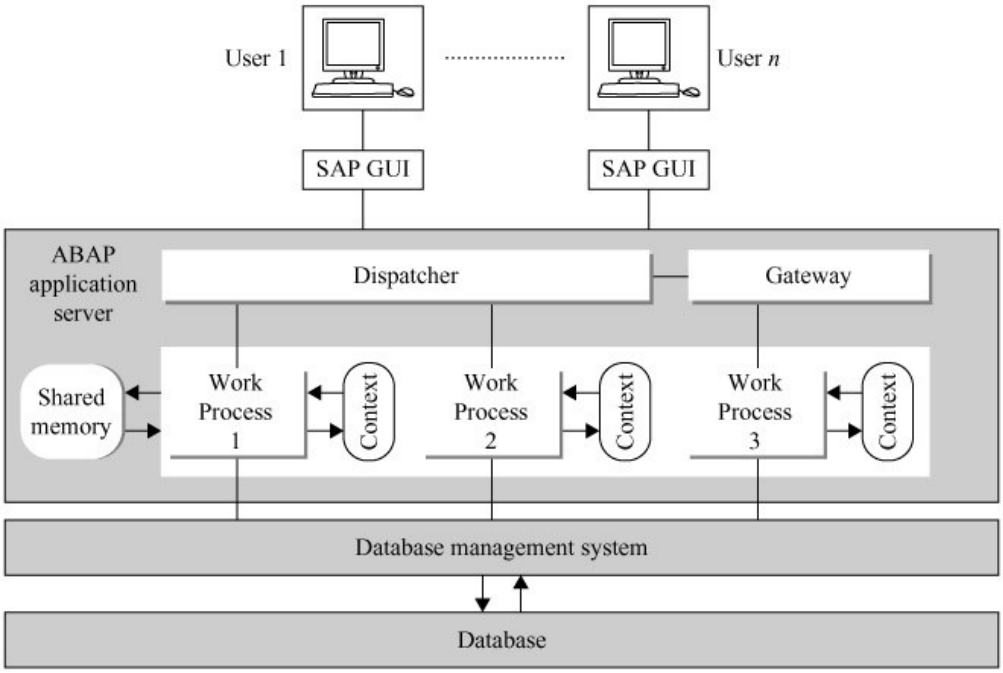

Figure 1.6: Architecture of the application server

Figure 1.6 shows the following components of the Application server:

- **Work processes—** Represents a process used to execute the user request. An Application server contains multiple work processes that are used to run an application. Each work process uses two memory areas, the user context and the roll area. The user context contains information regarding the user, and the roll area contains information about program execution.
- **Dispatcher—** Acts as a bridge to connect different work processes with the respective users logged on to the SAP R/3 system. The requests received by Application servers are directed first to the dispatcher, which enrolls them to a dispatcher queue. The dispatcher then retrieves the requests from the queue on a first-in, first-out basis and allocates them to a free work process.
- **Gateway—** Acts as an interface for R/3 communication protocols, such as a Remote Function Call (RFC). RFC is the standard SAP interface used to communicate between SAP systems.
- **Shared Memory—** Represents the common memory area in an Application server. All work processes running in an Application server use shared memory. This memory is used to save the contexts (data related to the current state of a running program) or buffer data. Shared memory is also used to store various types of resources that a work process uses, such as programs and table content.

Describing a Work Process

A work process is a component of the Application server that is used to run individual dialog steps used in an SAP R/3 application. Each work process contains two software processors, the Screen processor and the ABAP processor, and one database interface. A work process uses two special memory areas whenever it processes a user request. The first memory area is known as user context, which holds information regarding the user logged on to the SAP R/3 system. This information consists of user authorization as well as the names of currently running programs. The second memory area is known as the roll area, which holds information about the current program pointer (the location in which data of the program is stored), dynamic memory allocations, and the values of the variables needed to execute the program.

Exploring the Structure of a Work Process

In this section, we discuss the structure of a work process that is used in the R/3 system. Figure 1.7 shows the components of a work process:

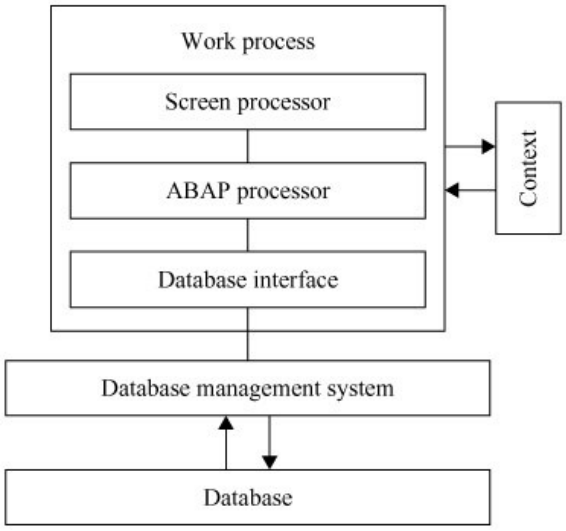

Figure 1.7: The components of a work process

As shown in Figure 1.7, the three components of a work process are:

- Screen processor
- ABAP processor
- Database interface

The Screen Processor

In R/3 application programming, user interaction and processing logic are different operations. From the programming point of view, user interaction is controlled by screens consisting of flow logic. The screen processor executes screen flow logic and also controls a large part of the user interaction. This flow logic helps a work process to communicate with the SAP GUI through a dispatcher. The screen flow logic also includes some modules, such as `PROCESS AFTER INPUT` (PAI) and `PROCESS BEFORE OUTPUT` (PBO), which explain the flow of data between the screens.

The ABAP Processor

The ABAP processor executes the processing logic of an application program written in the ABAP language. The ABAP processor not only processes the logic but also communicates with the database interface to establish a connection between a work process and a database. The screen processor informs the ABAP processor of the module of the screen flow logic that will be processed. Figure 1.8 shows the communication between the screen processor and ABAP processor, when an application program is running:

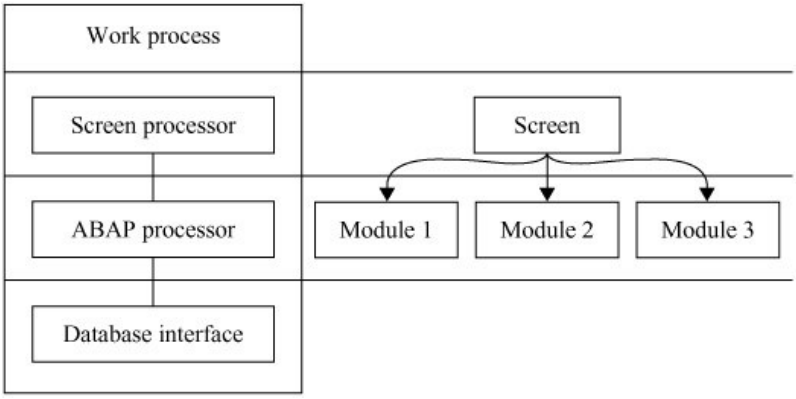

Figure 1.8: The screen processor and the ABAP processor at work

The Database Interface

The database interface performs the following tasks in a work process:

- Establishing or terminating the connection between the work process and the database
- Accessing database tables
- Accessing the R/3 Repository Objects, such as ABAP programs and screens
- Accessing catalog information (the ABAP Dictionary)
- Controlling transactions (commit and rollback handling)
- Managing table buffering on an Application server

Figure 1.9 shows the different components of the database interface:

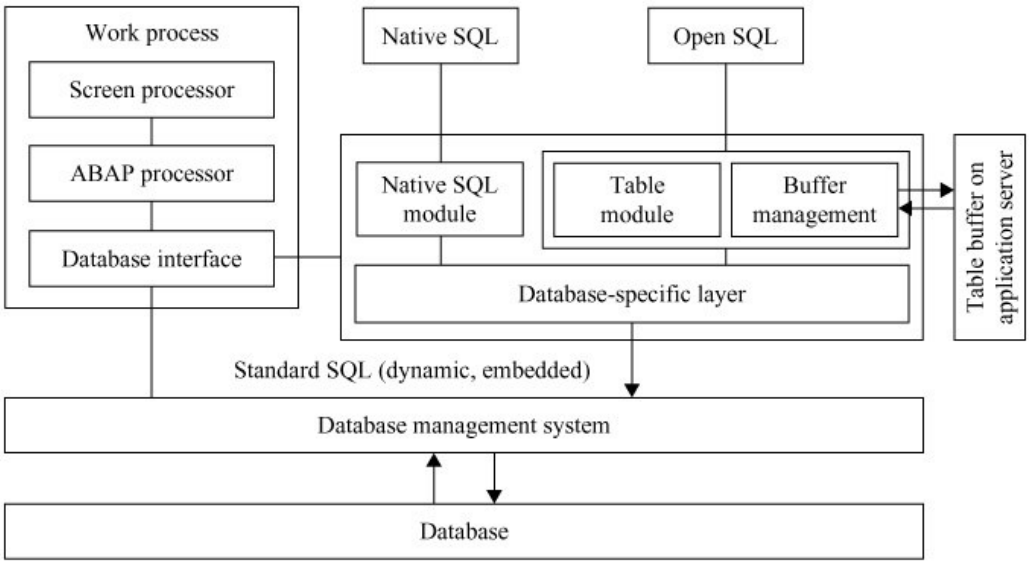

Figure 1.9: Components of the database interface

As shown in Figure 1.9, databases can be accessed in two ways: using `Open SQL` statements and using `Native SQL` statements.

Open SQL provides statements that, in conjunction with other ABAP constructions, can simplify or speed up database access. Native SQL statements, on the other hand, are a subset of standard SQL that is not integrated with the ABAP language code. To learn more about Open and Native SQL statements, refer to Chapter 8.

The Database-specific layer (Figure 1.9) hides the differences between database systems from the rest of the components of the database interface.

Now, let's describe the various types of work processes.

Types of Work Processes

All work processes can be categorized into five basic types on the basis of the tasks they perform: dialog, update, background, enqueue, and spool. In the Application server, the type of the work process determines the kind of task for which it is responsible. The dispatcher starts a work process, and depending on the type of work process, assigns tasks to it. This means that you can distribute work process types to optimize the use of resources in the Application servers. Figure 1.10 shows different types of work processes within an ABAP Application server:

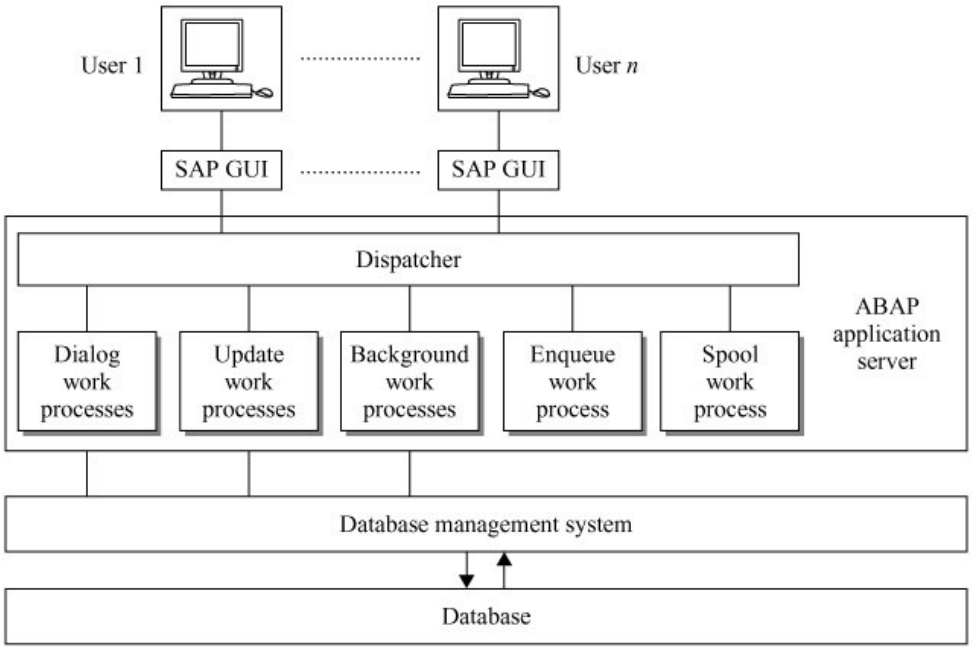

Figure 1.10: Types of work processes

In Figure 1.10, you see the different types of work processes, including the dialog work process, update work process, background work process, enqueue work process, and spool work process.

Table 1.1 describes the types of work processes:

Table 1.1: Different types of work processes

Work Process	Description	Settings
Dialog work process	Deals with requests to execute dialog steps triggered by an active user. The dialog work process is not used for requests that take long to execute and lead to high central processing unit (CPU) consumption. The default time for a dialog work process is 300 seconds. If the dialog work process does not respond in this time period, it is terminated.	The maximum response time of a dialog work process can be set by specifying the time in the rdisp/max_wprun_time parameter.
Update work process	Executes database update requests. There must be at least one update work process per SAP system, but there can be more than one update work process per dispatcher. An update work process is divided into two different modules, V1 and V2. The V1 module describes critical or primary changes, for example, creating an order or making changes to the material stock in the SAP R/3 system. The V2 module describes less critical secondary changes. These are pure statistical updates, for example, calculating the sum of the values of certain parameters. V1 modules have higher priority than the V2 modules.	The rdisp/wp_no_vb profile parameter is used to control the number of update work processes of V1 modules, and the `rdisp/wp_no_vb2` parameter is used to control the number of update work processes of V2 modules.
Background work process	Executes the programs that run without the involvement of the user, such as client copy and client transfer. There must be at least two background work processes per SAP system, but more than one background work process can be configured per dispatcher. Usually, background work processes are used to perform jobs that take a long time to execute.	The number of background work processes can be changed by specifying the value in the `rdisp/wp_no_btc` parameter.
Enqueue work process	Handles the lock mechanism. It administers the lock table, which is the main part of a Logical Unit of Work (LUW). The lock table stores the locks for logical databases in the SAP R/3 system. Only one enqueue work process is required for each SAP R/3 system.	The number of enqueue work processes can be specified in the `rdisp/wp_no_enq` parameter.
Spool work process	Passes sequential data flows on to printers. Every SAP system requires at least one spool work process. However, there can be more than one spool work process for a dispatcher.	The parameter to set the number of spool work processes is `rdisp/ wp_no_spo`.

Note In Table 1.1, all the parameters related to different types of work processes are specified in the Maintain Profile Parameters screen of the SAP system. You can access the Maintain Profile Parameters screen by entering the RZ11 transaction code in the Command field. To learn more about the Command field, refer to Chapter 3.

Now, let's discuss how dialog steps are executed by a work process.

Dispatching Dialog Steps

The dispatcher distributes the dialog steps among the various work processes on the Application server. A dialog step is a procedure in which a new screen appears in the SAP R/3 system for user interaction. Dispatching of dialog steps means navigating from one screen to another screen, where one screen accepts a request from the user and the other screen displays the result of the request.

It is very important for a programmer in SAP to understand how dialog steps are processed and dispatched, because the process is completely different from the processing involved in executing an ABAP program.

Note A dialog step is an SAP R/3 screen, which is represented by a dynamic program called a dynpro. The dynpro program consists of a screen and all the associated flow logic. It contains field definitions, screen layout, validation, and flow logic. A flow logic explains the sequence in which the screens are processed. When users navigate the SAP R/3 system from screen to screen, they are actually executing dialog steps. A set of dialog steps make up a transaction.

Often, the number of users logged on to an ABAP Application server is many times greater than the number of available work processes. In addition, each user can access several applications at a time. In this scenario, the dispatcher performs the important task of distributing all the dialog steps among the work processes on the ABAP Application server. Figure 1.11 shows an example of how dialog steps are dispatched in an ABAP Application server:

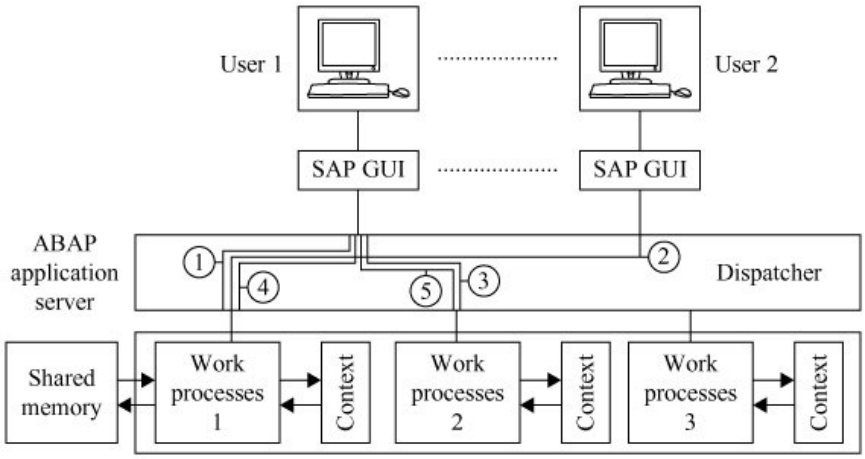

Figure 1.11: Dispatching dialog steps

Figure 1.11 shows two users, User 1 and User 2. The dispatcher receives a request to execute a dialog step from User 1 and directs it to work process 1, which is free. Work process 1 addresses the context of the application program (in shared memory), executes the dialog step, and becomes free again. Now, the dispatcher receives a request to execute a dialog step from User 2 and directs it to work process 1. Work process 1 executes the dialog step in the same way that it did in the case of User 1. However, while work process 1 is in progress, the dispatcher receives another request from User 1 and directs it to work process 2 because work process 1 is not free. After work processes 1 and 2 have finished processing their respective dialog steps, the dispatcher receives yet another request from User 1 and directs it to work process 1, which is now free. When work process 1 is in progress, the dispatcher receives another request from User 2 and directs it to work process 2, which is free. This process continues until all the requests of the users are processed.

From the preceding example, we can conclude that a program assigns a single dialog step to a single work process for execution. The individual dialog steps of a program can be executed on different work processes, and the program context must be addressed for each new work process. Moreover, a work process can execute dialog steps of different programs from different users.

An ABAP program is always processed by work processes, which require the user context for processing. A user context represents the data specifically assigned to an SAP user. The information stored in the user context can be changed by using the roll area of the memory management system in SAP.

Describing the User Context and Roll Area in the SAP System

All user contexts are stored in a common memory area of the SAP system. The memory management system of SAP comprises the following three types of memory which can be assigned to a work process in SAP:

- **SAP Roll Area—** Specifies a memory area with a defined size that belongs to a work process. It is located in the heap of the virtual address space of the work process.
- **SAP Extended Memory—** Represents a reserved space in the virtual address space of an SAP work process for extended memory. The size of the extended memory can be set by using the `em/initial_size_MB profile` parameter of the `Maintain Profile Parameters` screen in the SAP system.
- **Private Memory—** Specifies a memory location that is used by a work process if a dialog work process has used up the roll area memory and extended memory assigned to it.

Roll area memory is used as the initial memory assigned to a user context. Roll area memory is allocated to a work process in two stages. In the first stage, memory is allocated by specifying the `ztta/roll_first` parameter in the `Maintain Profile Parameters` screen. However, if this memory is already in use by the work process, additional memory is allocated in the second stage. The size of the additional memory area is equal to the difference between the `ztta/roll_area` and `ztta/roll_ first` parameters. Here, the `ztta/roll_area` parameter specifies the total size of the roll area, in bytes. Figure 1.12 shows the structure of the roll area memory:

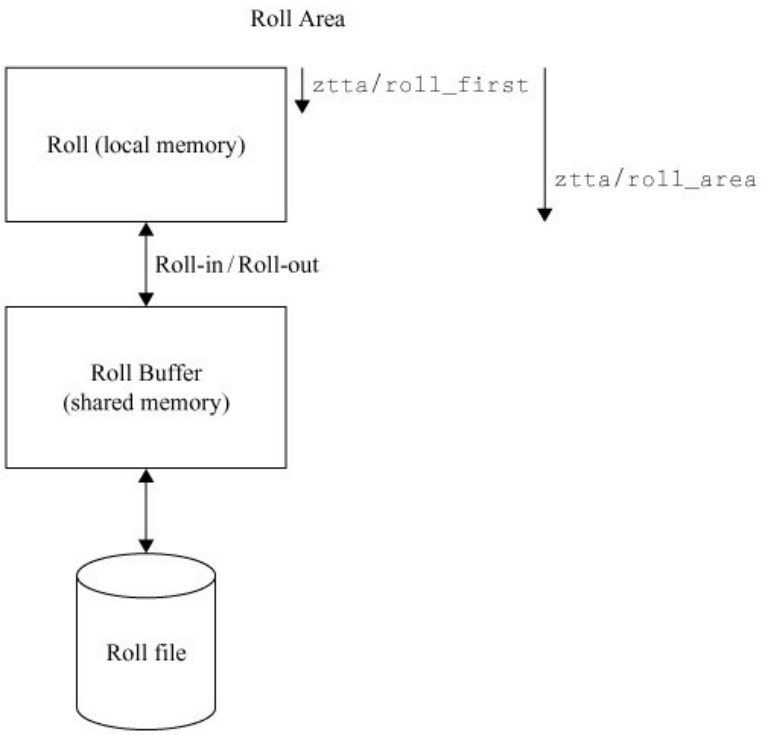

Figure 1.12: Structure of the roll area memory in SAP

As shown in Figure 1.12, whenever a dialog step is executed, a roll action occurs between the roll buffer in the shared memory and the roll local memory, which is allocated by the `ztta/roll_first` parameter. The area in the shared memory, which belongs to a user context, is then accessed. Note that when the context of a work process changes, its data is copied from the local roll area to a common resource called the roll file through the roll buffer (a shared memory).

As shown in Figure 1.12, the following roll processes are performed by the dispatcher:

- `Roll-in` —Copies user context from the roll buffer (in shared memory) to the roll local memory
- `Roll-out`—Copies user context from the roll local memory to the roll buffer

The Client-Dependency Feature

The SAP R/3 system provides an important feature called client-dependency, which means that a change made by a client in the SAP system is reflected on the other client. Let's take an example of R/3 database tables to illustrate this. Some

tables in the SAP R/3 system are client-dependent, while others are client-independent. A client-dependent table has its first field or column of the `CLNT` type. The length of this field is always of three characters, and by convention, this field is always named `MANDT` and contains the client number as its content. A client-independent table, on the other hand, does not have the `CLNT` type as its first field. Now, if any data is updated in the rows of a client-independent table, the change is not reflected on the other clients of the SAP R/3 system.

The client-dependency feature can also be explained in terms of SAP`script` forms and Smart Forms. An SAP script form is a template that simplifies the designing of business forms. On the other hand, `SAP Smart Forms` is a tool used to print or send business forms through e-mail, the Internet, and faxing. In the SAP R/3 system, SAP`script` forms are client-dependent, while the SAP Smart Forms are not.

Now, let's assume that a user generates two forms by using SAP`script` forms with two different client logins, client 800 and client 000. In this case, any changes made in client 800 will not be reflected in the form designed in client 000. On the other hand, in the case of Smart Forms, any changes made to one client will be reflected in the other client as well.

Note SAPscript and Smart Forms are described in detail in Chapter 12.

Summary

This chapter has explored the concept of SAP and its importance as leading business software. The chapter has also described the concept of ERP and its implementation in SAP. In addition, it has described the architecture of SAP R/3 system and the role and function of its three layers: Presentation, Application, and Database, and the various components of the Application server, such as work processes, the dispatcher, and the gateway. In addition to these topics, the text has explored memory management in SAP. Finally, the chapter concludes with a discussion on the client dependency feature of SAP.