In this chapter, authors Puneet Asthana and David Haslam review ABAP Workbench usage, including the repository browser, the ABAP Editor, the transport organizer, and more. Once you’re comfortable with these key concepts, test your knowledge with practice questions designed to help your ace your upcoming ABAP certification exam!
Chapter 4
ABAP Workbench
Usage

Techniques You’ll Master:
- Describe the features and capabilities of the ABAP Workbench
- Explain the structure of the Object Navigator
- Use the Repository Browser to edit repository objects
- Access the Repository Information System to search for repository objects
- Understand the Enhancement Information System and its use
- Work with the ABAP Workbench tools
- Identify the Front-End Editor and the various settings to improve productivity
- Determine the concept of the Development Package and the Transport Organizer
The ABAP Workbench is the development environment for the SAP NetWeaver Application Server ABAP and is available in every ABAP-based SAP system. The ABAP Workbench provides the development tool needed for application development. The Object Navigator, the central entry point into the ABAP Workbench, contains the most commonly used development tools and is the recommended development environment.

In this chapter, we will cover the ABAP Workbench tool and the Object Navigator in detail. We will explain the features and capabilities of the ABAP Workbench, discuss the structure of the Object Navigator, and discuss the various browsers available in the Object Navigator. We will cover the Front-End Editor, the Repository Information System, the Enhancement Information System, and ABAP Workbench tools such as the ABAP Editor, Screen Painter, Menu Painter, ABAP Dictionary, Class Builder, Function Builder, and Web Application Builder. Finally, we will discuss packages and their attributes and the Transport Organizer.

Real-World Scenario
You have started on a customer project as an SAP development lead and have been asked to explain SAP development tools and the various features of the ABAP Workbench and the Transport Organizer to the project team.

As a development lead, you have to organize the technical development for the project. Your job is to define the development standard and the tools to be used for the various developments. You also have to define the development project, package, system landscape, and transport strategy for all of the developments in the project.

Objectives of this Portion of the Test
One of the objectives of the ABAP certification examination is to verify your knowledge of the ABAP Workbench and the various tools associated with it. You need to understand the following to be a successful ABAP developer:

- ABAP Workbench and development tools
- ABAP Workbench settings
- The use of various browsers in the ABAP Workbench

Key Concepts Refresher
The ABAP Workbench is the integrated development environment for the application developer. The Workbench has most of the tools a developer needs to develop an application. Some of the most commonly used Workbench tools are:

- ABAP Editor
- Screen Painter
- Menu Painter
- ABAP Dictionary
- Web Dynpro development tools
- Package Builder and Transport Organizer
- Repository Browser
- Repository Information System
- Enhancement Information System

ABAP Workbench
ABAP stands for Advanced Business Application Programming and is the language in which most of the SAP business applications are written. The ABAP Workbench is the development environment for the ABAP developer. The ABAP Workbench consists of the development tools required for creating repository objects and can be started via Transaction SE80. The Object Navigator is the main entry point to the ABAP Workbench and can be accessed from the menu path Overview • Object Navigator.

The main screen of the ABAP Workbench is divided into the navigation area and the tool area (see Figure 4.1). The navigation area displays the object list as a hierarchy tree. The object list consists of all of the objects within an application area, package, program, global class, module pool, function group, and so on. The benefit of editing or displaying the object within the Object Navigator is that you can view all of the related objects for the program, class, module pool, or package as a tree structure and can access them from the navigation area by just double-clicking on the object. If you work with the repository object with the individual tool,
then you only have the option to work with one type of object at a time with one tool.

You can create your own worklist within the ABAP Workbench. A worklist is useful to help you manage the development objects on which you need to work. To create your worklist, display or edit the repository object first in the tool area of the ABAP Workbench. Then select Utilities • Worklist • Insert Current Object from the menu. This inserts the object in the worklist. You can display the worklist via the menu path Utilities • Worklist • Display, and an additional window for the worklist will be displayed under the tool area of the ABAP Workbench.

You can access the Reuse Library from the workbench menu path Environment • Reuse Library. The Reuse Library provides you with a set of reusable objects with example code and documentation. You can execute the code to display the result and copy the reusable code for your application development. The Reuse Library is displayed in the browser, and you can display the individual development objects within the Reuse Library by double-clicking on the product and reading the example code. You can also access the Reuse Library via Transaction SE83.

Similar to the Reuse Library, you can access the ABAP examples via the menu path Environment • Example • ABAP Example. The ABAP example programs are displayed in the browser, and you can display individual programs by double-clicking on the example program in the navigation area. The actual program is displayed in the tool area of the Object Navigator. ABAP examples demonstrate most of the keywords and syntax based on the ABAP documentation. You can also access the ABAP documentation from the ABAP examples screen. ABAP examples can also be displayed by executing Transaction ABAPDOCU.

You can display the context menu for an object within both the navigation area and the tool area by right-clicking on the object. The context menu offers only the functions that are relevant for the selected repository objects. You have the option to add your development objects to your favorites list within the Object Navigator by clicking on the Favorites icon in the navigation window.

You can access the individual tools to create the repository objects using individual transaction codes or access the most commonly used tools from the Object Navigator. The initial screen of the Object Navigator displays the Browser list on the top-left section of the screen. You can use for various development needs. You have the option to navigate to the relevant tool based on the selected browser from the browser list (see Figure 4.2). You can add or remove browsers from the list via the menu path Utilities • Settings.
The Object Navigator list has the following browser selection options in the object navigation:

- **Web Dynpro Text Browser**  
  You use the Web Dynpro Text Browser to edit the text of the Web Dynpro UI elements that are created or managed in the Online Text Repository (OTR). This tool allows you to change OTR texts associated with Web Dynpro views that have been stored in the OTR or inserted from the OTR in the Web Dynpro application.

- **MIME Repository Browser**  
  You use the MIME (Multipurpose Internet Mail Extensions) Repository Browser to browse or import MIME objects such as style sheets, icons, graphics, and so on from the ABAP Workbench. The MIME Repository stores these MIME objects in the SAP system. The MIME Repository Browser automatically creates a folder for BSP applications. You can import MIME objects from the Repository Browser or view the MIME objects by double-clicking on them.

- **Tag Browser**  
  The Tag Browser provides the documentation of the user interface elements that can be used for web page development for BSP applications and Internet Transaction Server (ITS) templates. You have the option to filter the tags for the BSP application or the tags for the ITS templates. The Tag Browser provides documentation for HTMLB, BSP extension, BSP Directives, HTML, WML, and XHTML.

- **Test Repository Browser**  
  You use the Test Repository Browser to manage test cases. You can display eCATT test scripts, manual test cases, and external test cases. You also have the option to create test cases within the Test Repository Browser. It’s a good tool for you to manage the test cases within the Test Repository.

- **ABAP Unit Browser**  
  This ABAP Unit testing tool for developers is included in SAP NetWeaver 7.0 EHP 2 and is integrated with the Object Navigator. The ABAP Unit tool provides the ABAP developer with an easy-to-use software verification and validation mechanism. It is highly integrated with the ABAP Workbench, which makes it easier for an ABAP developer to get accustomed to an ABAP Unit. Using the service method of class `cl_unit_assert`, you can check many conditions; the system diverges to the ABAP Unit result display if some condition is not fulfilled. In the result display, you can navigate to the position where the error occurred that caused the incorrect result. The tests are implemented in plain ABAP Objects as methods of local classes.

  The test classes are part of the program being tested. This can be an executables program, a module pool, a function group, or a class pool. The program being tested can include many test classes, and the test class can contain many methods. All unit tests belonging to a program being tested are organized in a test task. You can test whether the test condition or the modularization unit works properly by using `ASSERT` methods. For functions and methods with returning, exporting, or changing parameters, you simply test with the relevant parameter to determine if the modularization unit returns the expected value.

- **ATC Result Browser**  
  ABAP Test Cockpit is a new integrated tool for quality checking of the ABAP repository objects and is available as of SAP NetWeaver 7.0 EHP 2 Support Package 2. The ATC Result Browser displays the findings of the ABAP Test Cockpit quality check run. The ATC Result Browser is integrated in Object Navigator, and it is used by the developer to resolve the issues reported in the ATC findings. The active ATC result is marked with a lightbulb icon and should be used by the developer to resolve the issues with the ABAP code. The analysis and the correction features of the ATC findings are displayed in the browser. The ATC browser opens with a **Quick Filter** option, and you can specify the filter criteria to filter the ATC findings. You can filter the objects for which you are the responsible developer and work on the list to resolve the issues with the code; you also have the option to specify the **Advance Filter** option and can switch between the Advance and Quick Filter options by clicking on the **Switch Filter** push button.

  You can double-click on the message in the browser to display the details and edit the program to resolve the reported issues or apply for exemption.
test system. The ATC can be set up to check the content of the transport request automatically upon release of the transport request.

The ABAP Test Cockpit is a good tool for ABAP quality management and can be used by the quality assurance team to check the quality of the ABAP programs in the ABAP consolidation system or the test or quality system. Typically, your consolidation or quality system is the ATC Master system, in which you perform testing and quality checks. The ATC central check run can be performed from the ATC administration, Transaction ATC in the ATC Master system.

The quality manager schedules the regular central ATC check run in the ATC Master system to catch the problems with the ABAP repository objects. The quality manager then designates the ATC result as the active result and publishes it to the development system. The published result from central ATC quality check run will appear in the ATC Result Browser, and the developer should work on the active result to resolve the problem with the ABAP repository objects or request for exemptions.

ABAP Test Cockpit is also used to check functional correctness and performance optimization potential for SAP HANA migration.

Other browsers such as the Repository Browser, the Repository Information System, the Enhancement Information System, and the Transport Organizer are discussed in detail in the following sections.

Repository Browser

The Repository Browser is one of the menu options in the Object Navigator and is started by default when you execute the Object Navigator (Transaction SE80). You can create and edit repository objects in the Repository Browser. The Repository Browser is the central and most commonly used tool for managing your development objects within the Object Navigator.

When we talk about repository objects, we mean all of the SAP-delivered objects and customer-developed objects. These repository objects consist of programs, classes, ABAP Dictionary objects, function modules, screens, menus, and so on.

The object list displayed in the Repository Browser is a hierarchical tree structure. You can navigate to a repository object by the application hierarchy, whereby you can list the objects within each of the SAP application components. The objects can also be listed in a hierarchical tree structure within a package, program, class or interface, BSP application, Web Dynpro application, local object, and so on.

You can display the object from the navigation area by double-clicking on it. The object will be displayed in the tool area of the object browser in the editor that has been used to create the object. The ABAP development objects are displayed in the ABAP Editor, whereas the ABAP Dictionary objects are displayed in the ABAP Dictionary tool. Similarly, you display the screens and the menus in the tool area in the Screen Painter or Menu Painter by double-clicking on the screen or menu in the navigation area. You can use the context menus in the navigation area to create or edit the object in the application area or package (see Figure 4.3).

Figure 4.3 Repository Browser Navigation Area

You can create or edit repository objects only if you have the appropriate developer authorization and your user ID has been registered in the SAP Service Marketplace as a developer. You need to provide your developer access key the first time you create any repository object. You only have to provide the access key once because the system stores it for you in the table DEVACCESS.

Some of the common context menu functions for repository objects are to check repository objects, activate an object, copy or rename objects, and change the
package assignment of an object or write the object to the change request. You also have the where-used list function of the object from the context menu.

**Repository Information System**

The Repository Information System Browser is used to search for repository objects in the SAP system. The initial screen of the Repository Information System displays a hierarchical list of the categories of repository objects in the SAP system (see Figure 4.4).

The Repository Information System has a selection screen to provide the search criteria for the repository objects. You can search for repository objects within a package or search for ABAP Dictionary objects or programs by specifying the search criteria on the selection screen. Figure 4.5 displays the Repository selection screen for database tables within ABAP Dictionary objects.

You can customize the selection screen of the Repository Information System by selecting Edit • Settings from the initial screen of the Repository Information System Browser. You can specify in the customization if you want to see the default selection screen or the screen with all of the available selection criteria. You can also display all of the selection criteria for the object by clicking on the icon on the application toolbar.

You can specify the search criteria for the repository object in the selection screen and execute. The system displays the search results in the tool area of the Object Navigator. You can also select the object from the search result and display the where-used list of the object if required, and you can double-click on it to look at the object itself in the appropriate tool.

The Repository Information System is a useful tool to display the available enhancements within the SAP system. You can search for both the definition and
the implementations of the enhancement with this tool. You can filter your selection of enhancements by application component or package. You can search for business add-ins, customer exits, enhancement implementations, and enhancement definitions. For a detailed explanation of enhancements and modifications, refer to Chapter 18.

**Workbench Settings**

You can configure the ABAP Workbench to change its look and functionality. The workbench is configured from the menu path **Utilities • Settings** (see Figure 4.6). The workbench settings are user-specific and therefore do not affect anyone else in the system.

The Object Navigator can display only eight browser lists at a time. You can unselect the browser selection checkbox for the browsers that you are not going to use very often.

Within the Workbench settings, you also have the option to configure the ABAP Editor settings. You can select the new Front-End Editor, the old Front-End Editor, or the standard Back-End Editor and customize its settings. You have the option to set your default ABAP Debugger to either the classic or new ABAP Debugger. You can change your pretty printer settings, split screen settings, and the pattern settings. You can maintain the settings for the Class Builder, Screen Painter, Menu Painter, Function Builder, Repository Information System, Data Browser, Internet Transaction Server, Business Server Pages, Web Dynpro, Transport Organizer, SAPscript, eCATT, and proxy generation from the Workbench settings as well.

**ABAP Editor and Workbench Settings**

There are three different modes for the ABAP Editor:
- Front-End Editor (source code mode—new)
- Front-End Editor (plain text mode—old)
- Back-End Editor (line-based mode)

The three editors are fully compatible and interchangeable. The source code created in one editor can be viewed by all other modes.

The choice of the editor is based on the user-specific settings made in the ABAP Workbench. The editor can be configured within the ABAP Editor via the menu path **Utilities • Settings • ABAP Editor** (see Figure 4.7).
The Front-End Editor (New) option provides the latest editor and comes with SAP GUI for Windows 7.0. The new editor is an ActiveX control and is fully integrated into the SAP NetWeaver 7.0 environment.

The new editor has all of the modern code editing features:

- The left margin of the main screen of the editor displays any bookmarks and breakpoints. Breakpoints are displayed with a red stop sign, and bookmarks are displayed as a blue flag in the editor margin. You can set up to nine numbered bookmarks and an unlimited number of bookmarks that are not numbered on the editor for fast navigation within the code.
- The editor has a line number margin next to the editor margin, where the line number is displayed.
- The code changes are marked with a red triangle against the line number.
- The vertical scroll tip provides information about the current scroll position within the code, current function, class, or method.
- You can split the code editor screen horizontally by double-clicking on the splitter line on the vertical scroll bar. You can also just drag the splitter line to split the editor screen horizontally.
- You can collapse or expand blocks of code such as IF-ENDIF or CASE-ENDCASE.
- You can change the status bar of the editor displays the current status of (CapsLock) and (NumLock) and the line number of the cursor position. (CapsLock) and (NumLock) can be changed by double-clicking on them. Double-clicking on the line number displays the Go to Line dialog.

The Front-End Editor has two types of context menu; the context menu options depend on the area selected for the context. The margin context menu is displayed by right-clicking on the left margin and has the option to set breakpoints, delete breakpoints, set bookmarks, clear bookmarks, or navigate to a bookmark (see Figure 4.8).

The context menu in the editing area displays the menu options for the ABAP code. The editing area context menu has various formatting, editing, and navigation options (see Figure 4.9).

- The editor provides code hints at runtime as you type by suggesting keyword hints, block templates, and so on. You can accept the code hint by pressing the Tab key or insert a block template by pressing Ctrl + Enter. Furthermore, the editor supports WYSIWYG export functionality and exports to HTML, PDF, and RTF formats.
- The new ABAP Editor is a fully integrated development environment (IDE) for ABAP programming. It supports syntax highlighting, outlining language structures, real-time code hints, and auto-completion of language structures.
- With the new code editor you can customize the highlighting for the keyword, strings, and comments. You can customize the font, color, and size for ABAP keywords and comments. Similarly, you can customize font, color, and size for strings, breakpoints, and other display items in the code editor (see Figure 4.10).
You can customize the display settings of your editor. Figure 4.11 shows the display customization screen. The display and the word wrap options can be switched on or off according to your preference.

- Code completion has been added to the ABAP Front-End edition in SAP NetWeaver 7.0 EHP 2. The tool proposes appropriate ABAP words and operands for where you are in your code. To insert the proposed text, press `Ctrl` + `Space`.

- The other feature that’s new in SAP NetWeaver 7.0 EHP 2 is that with the new code-based view in the ABAP Editor, you can toggle between the classic form-based view in Class Builder and the new code-based view. This new view allows you to see a global class’s whole code, such as a program in the ABAP Editor, and you can edit the code. When you save, the modified source code is then split among the corresponding include programs.

- Finally, you can customize the code completion options for the editor. This option allows you to complete the available keyword from the dictionary or complete the class, method, or variable name within the scope of the visibility.

**ABAP Workbench Tools in Detail**

With the Object Navigator, you work directly with the repository object. The relevant tool for the repository object is automatically selected when you double-click on the repository object. The SAPscript Editor, Smart Forms Editor, and Customer Enhancement Projects are some of the development tools that are not available within the Object Navigator. The following are the most commonly used ABAP Workbench development tools:

- **ABAP Editor**
  The ABAP Editor is used to develop programs and can be executed via Transaction SE38. You use the editor to write ABAP programs.

- **Class Builder**
  The Class Builder allows you to create ABAP classes within the ABAP Workbench. You can create ABAP classes and interfaces, implement inheritance relationships, and create attributes, methods, and events for the global classes and interfaces that you build. You can access the Class Builder via Transaction SE24.

- **Function Builder**
  You use the Function Builder to write function modules and define function groups within the ABAP Workbench. You use the Function Builder to create, change, display, and test function modules. You can access the Function Builder directly via Transaction SE37.
- **Screen Painter**
  The Screen Painter is an ABAP Workbench tool used to create screens for SAP GUI transactions. You use the Screen Painter to create screens and write the flow logic for screens. You can access the Screen Painter via Transaction SE51. The Screen Painter layout editor used to design the screen has two versions: the alphanumeric layout editor and the graphic layout editor. By default, the Screen Painter layout editor is alphanumeric, but the mode can be changed to the graphical layout editor by changing the Screen Painter setting by following the menu path Utilities • Screen Painter and selecting Graphical Layout Editor.
  You define the screen attributes and the screen field attributes for the screen fields and the flow logic using the Screen Painter. You can create tab strips, table controls, subscreens, custom containers, and other screen elements using the Screen Painter.

- **Menu Painter**
  The Menu Painter is an ABAP Workbench tool to create the user interface for your program or transaction. You can create the menu bar, standard toolbar, application toolbar, and GUI title using the Menu Painter. You also assign the function keys to the functions you create in the Menu Painter. You can execute the Menu Painter outside the ABAP Workbench via Transaction SE41.

- **ABAP Dictionary**
  The ABAP Dictionary tool is an integral part of the ABAP Workbench. The ABAP Dictionary is used to create, change, and display transparent tables, pooled and cluster tables, views, types, domains, data elements, structures, table types, search helps, and lock objects. The database utility is also an integral part of the ABAP Dictionary tool. You can directly access the ABAP Dictionary tool outside the ABAP Workbench via Transaction SE11.

- **Web Application Builder**
  You can build web applications within the ABAP Workbench. The ABAP Workbench delivers the tool to develop ITS web applications or BSP applications.
  - ITS was the first approach by SAP to extend business applications to web browsers by converting the SAP Dynpro screen into HTML format, making it possible to access SAP transactions via web browsers. For ITS applications you can create Internet services for existing SAP transactions, HTML templates for the transaction screens, and MIME objects to add icons and graphics to the web screen layout.
  - You can also develop BSP applications with the Web Application Builder. You can design your web page for the BSP application using HTML and ABAP or Java Scripting language. You can define the page flow and implement the event handler with ABAP with the Web Application Builder for BSP. The Web Application Builder also allows you to define the theme for your layout and integrate the MIME objects into the web application.

- **Web Dynpro Explorer**
  The workbench has a tool for development of Web Dynpro ABAP applications (as of SAP NetWeaver 7.0). The tool consists of the runtime environment and the graphical tool to design the Web Dynpro views within the ABAP development environment.

---

**Enhancement Information System**

The Enhancement Information System provides an overview of the defined enhancements and the enhancement implementations within the Enhancement Framework. The Enhancement Framework is the technical basis for SAP’s enhancement concept (for detailed information, refer to Chapter 18). It enables you to see the enhancement definitions and implementations in the system.

The Enhancement Information System is a tree display structure for enhancement definitions and enhancement implementations. The enhancement element definition is displayed under the enhancement spot definition, which in turn is displayed under the composite enhancement definition. Similarly, the enhancement element implementation is displayed under the simple enhancement implementation, which in turn is displayed under the composite enhancement implementation. The enhancement implementation node is not displayed if the enhancement has not been implemented. Also, it is not necessary that all enhancement spot definitions or implementations are assigned to a composite enhancement definition. Enhancement definitions and implementations are discussed in detail in Chapter 18.

You can filter the display of the enhancements within the Enhancement Information System. You have the option to display any or all of the following: the composite enhancement, enhancement spot, enhancement implementation, or composite enhancement implementation within the Enhancement Information System (see Figure 4.12).
The enhancement spot definition can be displayed in the ABAP Workbench tool area by double-clicking on the enhancement spot definition in the navigation area. Figure 4.13 displays the Enhancement Spot definition screen in the tool area.

Similarly, you can display the Enhancement Implementation screen in the tool area of the ABAP Workbench by double-clicking on the enhancement spot implementation (see Figure 4.14).

### Packages and Their Attributes

Prior to release 4.6C, all ABAP development objects were assigned to development classes that were used to group related development objects. Packages were assigned to the application components. There can be multiple packages for an application component. As of release 4.6C and beyond, all ABAP development objects are to be assigned to the package. Packages are containers for the development objects within the transport layer that allow the objects to be transported. You can create or edit packages in the Package Builder in the Repository Browser. You can display all of the objects assigned to the package in the Repository Browser by selecting a package for the object list type. The SAP system has a predefined package ($TMP) to which all of the local development objects are assigned. Local objects assigned to the $TMP package cannot be transported. Nontransportable package names start with $.

The Package Builder can also be accessed via Transaction SE21 or SPACKAGE. The Package Builder is used to create and assign attributes to packages.

To create a package for your development project, you can create a main package and the subpackage, specify the package hierarchy, define the package interface, add elements to the package interface, and then define the use access for the package user. You create main packages only if you want to create a package hierarchy. Normally you create a package and specify the package type as Not a Main Package. You create the package from the Repository Browser (see Figure 4.15).
You specify the package name, short description, application component, software component, transport layer, and package type and then click on the Create Package icon. Customer package names should start with the letter Z or Y. The software component for a customer package is always HOME. The transport layer determines if the object assigned to the package is local or transportable. Normally, you carry out all of your development activity in the development system and then move your objects to the quality assurance system and the production system. Your system administration team will set up a transport layer for your development system, for which the transport route is defined to move the object from one system to another. Based on the transport layer, the objects assigned to the package are moved to the different systems defined in the transport route for the transport layer.

You can specify the package properties, use access, interface, and package hierarchy in the Package Builder screen in change mode. However, you are not required to specify the package interface or use access unless you want to protect the object assigned to the package from being used arbitrarily by other packages. Figure 4.16 displays the Package Builder screen where you can specify the package attributes. Packages have attributes such as nesting, package interfaces, visibility, and use access.

- Nesting allows you to embed packages in other packages, thus allowing you to split larger units of repository objects and structure them in a hierarchy.
- Visibility is a property of the package element. Elements within the package are visible to all other elements within the same package and are always invisible to the elements outside of the subpackage.

- Elements within a package can be made visible outside the package by defining a package interface.
- Use access is the right of one package to use the visible elements of the other package interface.

Figure 4.15: Create Package Screen

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Package</td>
<td>Z5001</td>
</tr>
<tr>
<td>Short Description</td>
<td>Custom SD Application Development Package</td>
</tr>
<tr>
<td>Application</td>
<td>SD</td>
</tr>
<tr>
<td>Software Component</td>
<td>HOME</td>
</tr>
</tbody>
</table>

Figure 4.16: Package Builder Screen for Package Attributes

<table>
<thead>
<tr>
<th>Package</th>
<th>Z5001</th>
<th>Saved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td>Use access</td>
<td>Package interfaces</td>
</tr>
<tr>
<td>Short Description</td>
<td>Custom SD Application Development Package</td>
<td></td>
</tr>
<tr>
<td>Person responsible</td>
<td>BOUSER</td>
<td></td>
</tr>
<tr>
<td>Created By</td>
<td>BOUSER</td>
<td></td>
</tr>
<tr>
<td>Last changed by</td>
<td>BOUSER</td>
<td></td>
</tr>
<tr>
<td>Application Component</td>
<td>SD</td>
<td></td>
</tr>
<tr>
<td>Software Component</td>
<td>HOME</td>
<td></td>
</tr>
<tr>
<td>Transport Layer</td>
<td>ZFLC</td>
<td></td>
</tr>
</tbody>
</table>

The system checks that your package complies with these rules based on the entry in the table PAKPARAM. The package check is switched off by default in the customer system. The entry GLOBAL_SWITCH for the key field NAME in the table PAKPARAM controls the behavior of the package check. By default, the GLOBAL_SWITCH key is set to OFF in the table PARPARAM. To switch on package check, set the GLOBAL_SWITCH key to RESTRICTED or ENTERPRISE.
Table 4.1 displays the values for `GLOBAL_SWITCH` and their effects on package check.

<table>
<thead>
<tr>
<th>Value</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESTRICTED</td>
<td>The package check is only performed if you have selected the Package Check as Client or Package Check as Server Attribute in the package.</td>
</tr>
<tr>
<td>R3ENTERPRISE</td>
<td>The same checks that are performed for entry RESTRICTED are performed here. With objects belonging to structure packages, an additional check is performed to determine whether use access has been defined between them. Note that in this case an additional entry has to be made in the table PAXPARAM with the key field NAME equal to <code>SAP_DEV_SYSTEM</code> and VALUE set to X.</td>
</tr>
<tr>
<td>OFF</td>
<td>Switches off package check.</td>
</tr>
</tbody>
</table>

Table 4.1 Valid Values for Package Check

The Change and Transport System (CTS) provides you with a tool to organize your ABAP Workbench, cross-client customization, and customization work and then transport the changes through your system landscape. The CTS records all of the changes in the change request. The change request is also referred to as the transport request. You can release your task and the change request once you have completed and tested your development. You have to ensure that your development object is syntactically correct and active before you release the change request. The change request is then used to transport the changes to other systems or clients based on the transport route.

Repository objects and cross-client customizations are assigned to a workbench request, whereas the client-specific customization objects are assigned to a Customizing request. Each change request can have one or multiple tasks, and the repository objects or the customization objects are assigned to one of these tasks. The task is assigned to a user, and only the owner of the task can record his changes to the task.

There are two types of workbench tasks: development/correction and repair tasks. The repository object changes are recorded in the development/correction task if the current system is the original system of the object. The object is recorded in the repair task if the current system is not the original system of the object. All SAP standard object modifications are recorded in the repair task of the workbench request. Figure 4.18 displays the change request structure. The top level is the change request, and the lower-level number is the task of the change request.

Figure 4.17 Package Structure in Object Navigator of ABAP Workbench

![Package Structure in Object Navigator of ABAP Workbench](image)

Transport Organizer

Each repository object is assigned to a package, and the package is assigned to the transport layer. If the route for the transport layer is defined in the Transport Management System (TMS), then the object recorded in the transport task is transportable; otherwise, the task belongs to the local change request.

The system administration team creates the transport layer in the TMS to transport the objects in your system landscape. The transport layer is assigned to the
development system. The administration team needs to set up a transport route after creating the transport layer.

There are two types of transport routes: Your administration team will set up the consolidation route for each transport layer and then the delivery route. The development system is the source system for the consolidation route, and the quality assurance system is the target system. You define the delivery route to transport the objects from the quality system to the other target system, which can be production and other training systems. So once the object is imported to the quality system, you need delivery routes to transport the objects from the quality system to other systems.

The Transport Organizer is integrated with the Object Navigator or can be accessed via Transaction SE10. You can display the transport request and the associated tasks of the request from the workbench by double-clicking on the request in the navigation area of the workbench. The request will be displayed in the tool area, whereby you can display the properties, objects, and documentation of the request. The request editor displays all of the recorded changes within the request (see Figure 4.19).

You can create change requests by creating the repository object. When you are creating the repository object, a dialog window appears to allow you to create a new request or assign the object to an existing request. The transport attributes are populated automatically based on the attributes of the repository object. Figure 4.20 displays the initial transport request screen.

You can also create the transport request from the transport organizer, but you have to populate the transport attributes on your own. You release all of the tasks within the change request and then the change request itself to transport the objects through your system landscape. The system administrator imports the objects to the target system. You can display the transport log by selecting the transport log menu option.

The Transport Organizer tools are integrated with the Transport Organizer. They are a set of programs that help you work with the Transport Organizer. You can start the tool from the initial screen of the Transport Organizer via the menu path Goto • Transport Organizer Tools (see Figure 4.21).
The tools can be used to search for objects in transport requests or include objects in the transport requests. You can search the request or task based on different search criteria. Similarly, you can display all of the modifications in the customer system using the modification browser.

**Practice Questions**

These practice questions will help you evaluate your understanding of the topic. The questions shown are similar in nature to those found on the certification examination. Although none of these questions will be found on the exam itself, they allow you to review your knowledge of the subject. Select the correct answers and then check the completeness of your answers in the following solution section. Remember that you must select all correct answers and only correct answers on the exam to receive credit for the question.

1. The Object Navigator incorporates a total of 11 browsers.
   - [ ] A. True
   - [ ] B. False

2. The Repository Browser is started by default when you execute Transaction SE80 for the Object Navigator.
   - [ ] A. True
   - [ ] B. False

3. You can list a maximum of six browsers in the Object Navigator.
   - [ ] A. True
   - [ ] B. False

4. You can maintain SAPscript forms and SAP Smart Forms within the ABAP Workbench.
   - [ ] A. True
   - [ ] B. False

5. The Repository Information System is a useful tool to search for customer exits/function exits and BAdIs in the SAP system.
   - [ ] A. True
   - [ ] B. False

6. Which of the following statements about the Object Navigator are true? Select all that apply.
   - [ ] A. ABAP programs can be displayed and edited in the Object Navigator.
   - [ ] B. Screens can be displayed and edited in the Object Navigator.
   - [ ] C. Menus can be displayed and edited in the Object Navigator.
   - [ ] D. You can create BAdI implementations in the Object Navigator.
   - [ ] E. You can create customer projects (Transaction CMOD) in the Object Navigator.
   - [ ] F. The ABAP Dictionary can be maintained in the Object Navigator.

7. Enhancement definitions and implementations can be displayed in the Enhancement Information System.
   - [ ] A. True
   - [ ] B. False

8. Which of the following is a true statement? Select all that apply.
   - [ ] A. All customer repository objects have to be assigned to a package.
   - [ ] B. Packages use interfaces and visibility to make their elements visible to other packages.
   - [ ] C. The transport layer is a mandatory input field for the package.
   - [ ] D. A package can be nested.

9. The software component for a customer package can be...
   - [ ] A. HOME
   - [ ] B. Any SAP software component (i.e., SAP_APPL, SAP_BASIS, SAP_HR, etc.)
10. Which of the following is a true statement? Select all that apply.
- A. All transportable objects have to be assigned to a package.
- B. Local repository objects can be transported.
- C. Repository objects and cross-client customization objects are assigned to the workbench request.
- D. Client-specific customization objects are assigned to the customizing request.
- E. Inactive objects can be transported.

11. Which of the following is true? Select all that apply.
- A. The repository objects and cross-client customization objects are recorded in a task belonging to a local change request if there is no consolidation route leading from the current system defined in the Transport Management System for the transport layer.
- B. The repository objects and the cross-client customization are recorded in a task belonging to the transportable request if the consolidation route is defined in the Transport Management System.

12. There are ____ versions of the ABAP Editor.
- A. 3
- B. 4
- C. 2

13. Repository objects are client-specific.
- A. True
- B. False

14. ABAP Unit Test Browser is included in Object Navigator with SAP NetWeaver 7.0 EHP 2.
- A. True
- B. False

Practice Question Answers and Explanations

1. Correct answer: A
   The Object Navigator has a total of 11 browsers, as an additional ABAP Unit Browser has been added in SAP NetWeaver 7.0 EHP 2. In earlier releases, it was nine. The following are the object browsers available in the ABAP Workbench:
   - Repository Browser
   - Repository Information System
   - Transport Organizer
   - MIME Repository
   - Tag Browser
   - Test Repository
   - Enterprise Service Browser
   - Web Dynpro Text Browser
   - Enhancement Information System
   - ABAP Unit Browser
   - ABAP Test Cockpit

2. Correct answer: A
   The Repository Browser is the default browser for the ABAP Workbench.

3. Correct answer: B
   You can display up to eight browsers in the ABAP Workbench.

4. Correct answer: B
   You cannot maintain SAPscript forms or SAP Smart Forms in the ABAP Workbench.

5. Correct answer: A
   You can search for enhancements including business add-ins and customer exits with the Repository Information System.

   You can edit or display ABAP programs in the ABAP Workbench in the ABAP Editor. You can display or edit screens within the ABAP Workbench in the Screen Painter. You can display and edit menus within the ABAP Workbench in the Menu Painter. The BAdI implementation tool is not integrated within the Object Navigator, so it’s not possible to create BAdI implementations in the Object Navigator. The customer project (CMOD) is not integrated within the ABAP Workbench, so it’s not possible to create customer projects with the ABAP
Workbench. The ABAP Dictionary is integrated within the ABAP Workbench, so it can be maintained in ABAP Workbench.

7. Correct answer: A
Enhancement definitions and implementations can be displayed in the Enhancement Information System.

8. Correct answers: A, B, D
You can create a local object and assign it to package $TMP$, but you have to assign the object to a package (whose name does not start with ‘$’) if you want to transport the object from one system to another. A package has to define a package interface and visibility to make its elements visible to other packages. The transport layer is not a mandatory input field for the package. The transport layer is assigned to the package if it is defined for the system. A package can be nested.

9. Correct answer: A
The software component of the customer package should always be HOME.

10. Correct answers: A, C, D
The repository object has to be assigned to the package to transport the object to another system within the system landscape. You cannot transport a local repository object. Repository objects and cross-client customization objects are assigned to the workbench request. Client-specific customization objects are assigned to the customization request and are not assigned to the package. Inactive objects can be transported.

11. Correct answers: A, B
A local change request is created if the consolidation route is not defined; otherwise, a transportable change request is created if the consolidation route for the transport layer is defined.

12. Correct answer: A
There are three modes of ABAP Editor: the new Front-End Editor, the old Front-End Editor, and the Back-End Editor.

13. Correct answer: B
Repository objects are cross-client objects, and therefore they are system wide.

14. Correct answer: A
ABAP Unit Test Browser is the new tool in Object Navigator in SAP NetWeaver 7.0 EHP 2.

**Takeaway**
You should now understand the ABAP Workbench features and be able to navigate within it. You should be able to use the available browsers within the ABAP Workbench and complete your development in an efficient manner. You should know how to configure the ABAP Workbench and the various development tools.

To be a successful developer, you should know about all the available development tools and their uses. It is important to understand the features of the new ABAP Editor because it will be very helpful for your application development and will increase the productivity of your development team.

Finally, you should understand the package concept and the use of the Transport Organizer. You should understand the concept of the transport request and the different types of transport request and their use for the migration of development objects from the development environment to the production environment.

**Refresher**
Table 4.2 shows key concepts about the ABAP Workbench.

<table>
<thead>
<tr>
<th>Key Concept</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABAP Workbench</td>
<td>The ABAP Workbench is an integrated development environment for the ABAP developer.</td>
</tr>
<tr>
<td>Repository Browser</td>
<td>The Repository Browser is a tool within the Object Navigator and is used by the developer to access the repository objects and workbench tools to create repository objects.</td>
</tr>
<tr>
<td>Repository Information System Browser</td>
<td>The Repository Information System Browser is a search tool to search for repository objects.</td>
</tr>
<tr>
<td>Enhancement Information System</td>
<td>The Enhancement Information System is a tool to search for enhancement definitions and implementations in the SAP system.</td>
</tr>
<tr>
<td>Transport Organizer</td>
<td>The Transport Organizer is a tool to work with the change request and the objects within the change request. Using this tool, you can view the objects assigned to the change request and the transport log of the request.</td>
</tr>
</tbody>
</table>
In this chapter we covered the ABAP Workbench and its use in detail. We covered some of the most commonly used browsers such as the Repository Browser, Repository Information System, Enhancement Information System, and Transport Organizer. Furthermore, we discussed the ABAP Workbench, the new ABAP Editor, and the concepts of package and transport requests in detail. This knowledge will allow you to easily pass this topic on the certification examination.
Contents

Acknowledgments ....................................................................................................................................... 17
Preface ....................................................................................................................................................... 19

PART I General Introduction

1 ABAP Development Certification Track: Overview .......................................................................................... 29
   Certification Exam Scoring Changes .................................................................................................................. 30
   Certification Levels ........................................................................................................................................... 30
   Advanced Certification Levels ........................................................................................................................... 32
   Becoming an SAP Certified ABAP Development Associate: Overview .............................................................. 33
      Associate Examination Specifics ....................................................................................................................... 33
      Competency Areas .......................................................................................................................................... 34

2 Courses and Experience .................................................................................................................................... 37
   Training Courses for ABAP ................................................................................................................................. 38
   Sources of Information ....................................................................................................................................... 39
   Strategic Understanding ................................................................................................................................... 39
   SAP Examination Strategies ................................................................................................................................. 41
   General Examination Strategies .......................................................................................................................... 42

PART II Exam Preparation

3 SAP NetWeaver: Overview .................................................................................................................................... 47
   Objectives of This Portion of the Test ................................................................................................................... 48
   Key Concepts Refresher ..................................................................................................................................... 48
4 ABAP Workbench Usage

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objectives of this Portion of the Test</td>
<td>76</td>
</tr>
<tr>
<td>Key Concepts Refresher</td>
<td>77</td>
</tr>
<tr>
<td>ABAP Workbench</td>
<td>77</td>
</tr>
<tr>
<td>Repository Browser</td>
<td>82</td>
</tr>
<tr>
<td>Repository Information System</td>
<td>84</td>
</tr>
<tr>
<td>Workbench Settings</td>
<td>86</td>
</tr>
<tr>
<td>ABAP Editor and Workbench Settings</td>
<td>87</td>
</tr>
<tr>
<td>ABAP Workbench Tools in Detail</td>
<td>91</td>
</tr>
<tr>
<td>Enhancement Information System</td>
<td>93</td>
</tr>
<tr>
<td>Packages and Their Attributes</td>
<td>95</td>
</tr>
<tr>
<td>Transport Organizer</td>
<td>99</td>
</tr>
<tr>
<td>Practice Questions</td>
<td>102</td>
</tr>
<tr>
<td>Practice Question Answers and Explanations</td>
<td>105</td>
</tr>
<tr>
<td>Takeaway</td>
<td>107</td>
</tr>
<tr>
<td>Refresher</td>
<td>107</td>
</tr>
</tbody>
</table>

5 ABAP Debugger Program Usage

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objectives of this Portion of the Test</td>
<td>111</td>
</tr>
<tr>
<td>Key Concepts Refresher</td>
<td>111</td>
</tr>
<tr>
<td>New and Classic Debugger</td>
<td>112</td>
</tr>
<tr>
<td>New Debugger Tools and UI</td>
<td>116</td>
</tr>
<tr>
<td>Assertions and Breakpoints</td>
<td>125</td>
</tr>
<tr>
<td>New Debugger Customization and Settings</td>
<td>126</td>
</tr>
<tr>
<td>Important Terminology</td>
<td>132</td>
</tr>
<tr>
<td>Practice Questions</td>
<td>133</td>
</tr>
<tr>
<td>Practice Question Answers and Explanations</td>
<td>136</td>
</tr>
<tr>
<td>Takeaway</td>
<td>138</td>
</tr>
<tr>
<td>Refresher</td>
<td>139</td>
</tr>
<tr>
<td>Tips</td>
<td>139</td>
</tr>
</tbody>
</table>

6 ABAP Types and Data Objects

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objectives of This Portion of the Test</td>
<td>142</td>
</tr>
<tr>
<td>Key Concepts Refresher</td>
<td>143</td>
</tr>
<tr>
<td>ABAP Types and Data Objects</td>
<td>143</td>
</tr>
<tr>
<td>ABAP Data Types</td>
<td>151</td>
</tr>
<tr>
<td>Local Data Types</td>
<td>155</td>
</tr>
<tr>
<td>Global Data Types</td>
<td>158</td>
</tr>
<tr>
<td>Data Object Visibility</td>
<td>160</td>
</tr>
<tr>
<td>Important Terminology</td>
<td>160</td>
</tr>
<tr>
<td>Practice Questions</td>
<td>161</td>
</tr>
<tr>
<td>Practice Question Answers and Explanations</td>
<td>164</td>
</tr>
<tr>
<td>Takeaway</td>
<td>166</td>
</tr>
<tr>
<td>Refresher</td>
<td>166</td>
</tr>
</tbody>
</table>
7 Internal Table Definition and Use

Objectives of this Portion of the Test ............................................................................. 170
Key Concepts Refresher .................................................................................................. 171
Internal Table Definition and Use ................................................................................ 171
Defining ABAP Internal Tables .................................................................................... 175
Using ABAP Internal Tables ......................................................................................... 184
Important Terminology ................................................................................................. 201
Practice Questions ......................................................................................................... 201
Practice Question Answers and Explanations ............................................................ 204
Takeaway ....................................................................................................................... 206
Refresher ....................................................................................................................... 207

8 SQL Statements Including Update Strategies

Objectives of This Portion of the Test ............................................................................. 211
Key Concepts Refresher .................................................................................................. 211
Data Modeling ............................................................................................................... 211
Data Retrieval ............................................................................................................... 214
Arrangement of SELECT and FROM Clauses ............................................................... 217
Unions .......................................................................................................................... 217
Performance of Database Access ................................................................................ 218
Logical Units of Work .................................................................................................. 221
Enqueue and Dequeue .................................................................................................. 223
Inline Updates ............................................................................................................. 228
Perform on Commit ...................................................................................................... 229
Update Modules .......................................................................................................... 229
Open SQL ..................................................................................................................... 235
SQL Parser ................................................................................................................... 236
SQL Expressions .......................................................................................................... 237
SQL Expressions Expanded ......................................................................................... 238
SQL Functions Expanded ............................................................................................ 238
Access to CDS Entities ................................................................................................. 241
Practice Questions ......................................................................................................... 241

9 Basic ABAP Programs and Interface Creation

Objectives of this Portion of the Test ............................................................................. 233
Key Concepts Refresher .................................................................................................. 233
Organizing Development .............................................................................................. 234
ABAP Programming Overview ................................................................................... 235
ABAP Event Blocks ...................................................................................................... 238
Basic ABAP Statements ............................................................................................... 246
ABAP Subroutine ......................................................................................................... 249
ABAP Function Module ............................................................................................... 249
ABAP Classes and Methods ......................................................................................... 250
ABAP Selection Screen ................................................................................................. 250
Authorization Checks .................................................................................................... 254
ABAP Dynpros .............................................................................................................. 255
Dialog Messages ........................................................................................................... 259
Important Terminology ................................................................................................. 281
Practice Questions ......................................................................................................... 281
Practice Question Answers and Explanations ............................................................ 286
Takeaway ....................................................................................................................... 288
Refresher ....................................................................................................................... 288
Tips ................................................................................................................................. 288

10 ABAP Dictionary

Objectives of This Portion of the Test ............................................................................. 304
Key Concepts Refresher .................................................................................................. 304
Overview ....................................................................................................................... 304
# Index

## ABAP

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>basic statement</td>
<td>266</td>
</tr>
<tr>
<td>data object</td>
<td>143</td>
</tr>
<tr>
<td>data type</td>
<td>339</td>
</tr>
<tr>
<td>data type, domain</td>
<td>142, 151, 166</td>
</tr>
<tr>
<td>definition</td>
<td>255</td>
</tr>
<tr>
<td>event blocks</td>
<td>261</td>
</tr>
<tr>
<td>global declaration</td>
<td>256, 290</td>
</tr>
<tr>
<td>global definition</td>
<td>256</td>
</tr>
<tr>
<td>obsolete additions</td>
<td>256</td>
</tr>
<tr>
<td>procedural</td>
<td>256</td>
</tr>
<tr>
<td>processing block</td>
<td>258</td>
</tr>
<tr>
<td>processor</td>
<td>65</td>
</tr>
<tr>
<td>runtime environment</td>
<td>255, 261, 266, 344, 545</td>
</tr>
<tr>
<td>subroutine</td>
<td>266, 269</td>
</tr>
<tr>
<td>syntax rules</td>
<td>266</td>
</tr>
<tr>
<td>text translation</td>
<td>256</td>
</tr>
<tr>
<td>type</td>
<td>313</td>
</tr>
<tr>
<td>type of program</td>
<td>256</td>
</tr>
<tr>
<td>ABAP application server</td>
<td>48, 61, 63, 66, 214</td>
</tr>
<tr>
<td>resource optimization</td>
<td>67</td>
</tr>
<tr>
<td>ABAP class</td>
<td>443</td>
</tr>
<tr>
<td>attribute</td>
<td>451</td>
</tr>
<tr>
<td>class events</td>
<td>463</td>
</tr>
<tr>
<td>constructor</td>
<td>460</td>
</tr>
<tr>
<td>create object</td>
<td>451</td>
</tr>
<tr>
<td>event</td>
<td>463</td>
</tr>
<tr>
<td>global class</td>
<td>445</td>
</tr>
<tr>
<td>local class</td>
<td>443</td>
</tr>
<tr>
<td>method</td>
<td>453</td>
</tr>
<tr>
<td>private</td>
<td>444</td>
</tr>
<tr>
<td>private instantiation</td>
<td>449</td>
</tr>
<tr>
<td>private section</td>
<td>447</td>
</tr>
<tr>
<td>protected</td>
<td>444</td>
</tr>
<tr>
<td>protected instantiation</td>
<td>449</td>
</tr>
<tr>
<td>protected section</td>
<td>448</td>
</tr>
<tr>
<td>public</td>
<td>444</td>
</tr>
<tr>
<td>public section</td>
<td>447</td>
</tr>
<tr>
<td>raise exception</td>
<td>459</td>
</tr>
</tbody>
</table>

## ABAP class (Cont.)

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>read-only</td>
<td>452</td>
</tr>
<tr>
<td>returning</td>
<td>459</td>
</tr>
<tr>
<td>static method</td>
<td>462</td>
</tr>
<tr>
<td>ABAP Debugger</td>
<td>86, 110, 112</td>
</tr>
<tr>
<td>architecture</td>
<td>111</td>
</tr>
<tr>
<td>classic</td>
<td>112, 113, 115, 117</td>
</tr>
<tr>
<td>flow control</td>
<td>127</td>
</tr>
<tr>
<td>new</td>
<td>113, 115–117, 126, 133</td>
</tr>
<tr>
<td>process call</td>
<td>268</td>
</tr>
<tr>
<td>start methods</td>
<td>114</td>
</tr>
<tr>
<td>starting</td>
<td>111, 113, 115</td>
</tr>
<tr>
<td>stopping</td>
<td>115</td>
</tr>
<tr>
<td>switching</td>
<td>115</td>
</tr>
<tr>
<td>work areas/desktops</td>
<td>117</td>
</tr>
<tr>
<td>ABAP Dictionary</td>
<td>66, 92, 303, 359, 525, 602, 611</td>
</tr>
<tr>
<td>activate table definition</td>
<td>212</td>
</tr>
<tr>
<td>append structure</td>
<td>359</td>
</tr>
<tr>
<td>check table</td>
<td>602–604, 606, 607, 610, 611</td>
</tr>
<tr>
<td>create view</td>
<td>611</td>
</tr>
<tr>
<td>customizing include</td>
<td>359</td>
</tr>
<tr>
<td>data element</td>
<td>283, 486, 491, 525, 602, 603, 611, 613</td>
</tr>
<tr>
<td>data element, individual field</td>
<td>213</td>
</tr>
<tr>
<td>data type</td>
<td>142, 162</td>
</tr>
<tr>
<td>domain</td>
<td>213, 603</td>
</tr>
<tr>
<td>fixed value</td>
<td>604, 606</td>
</tr>
<tr>
<td>foreign key</td>
<td>602, 606–608, 610</td>
</tr>
<tr>
<td>input field</td>
<td>277</td>
</tr>
<tr>
<td>interface parameters</td>
<td>271</td>
</tr>
<tr>
<td>key concepts</td>
<td>345</td>
</tr>
<tr>
<td>overview</td>
<td>305</td>
</tr>
<tr>
<td>program development support</td>
<td>603</td>
</tr>
<tr>
<td>search help</td>
<td>602–604, 611</td>
</tr>
<tr>
<td>search helps</td>
<td>328</td>
</tr>
<tr>
<td>structure enhancement</td>
<td>359</td>
</tr>
<tr>
<td>table</td>
<td>345</td>
</tr>
<tr>
<td>text table</td>
<td>609, 612</td>
</tr>
<tr>
<td>tool</td>
<td>307</td>
</tr>
<tr>
<td>transparent table</td>
<td>212</td>
</tr>
<tr>
<td>value table</td>
<td>602, 604, 606, 608</td>
</tr>
<tr>
<td>value table to check table</td>
<td>607</td>
</tr>
</tbody>
</table>
Data type .................................. 143, 144, 166, 305, 309
basic ........................................ 305
complex .................................... 155, 305, 307
declare incomplete ................. 154
define structure ......................... 156
elementary ................................ 142, 153, 158
generic ...................................... 141, 143
global ........................................ 158, 162, 166
LCHR ........................................ 310
local ......................................... 155, 166
RAW .......................................... 311
MESH ........................................ 157
parameters ................................ 144
predefined .................................. 143, 309
QUAN ......................................... 311
RAWSTRING ................................ 311
select-options ......................... 144
STRING ....................................... 312
STRING ....................................... 153
TIMS ......................................... 312
types ........................................ 144
XSTRING .................................... 312

Dequeue ..................................... 333, 338
Desktop ................................... 126, 128
tool ........................................... 117, 118
Dialog Message -> Message .... 370
Dialog programming concept ....... 371
Dialog screen ......................... 415
event ......................................... 413
Dialog step ............................. 63, 68, 223, 228
Diff tool ................................... 119, 128
Dispatcher .............................. 64
domain ...................................... 309, 305, 308, 345, 603
data element ....................... 315, 605
equality ..................................... 608
fixed values ............................. 613
format ....................................... 309
output property ...................... 309, 313
value table ............................. 606
Down cast ................................ 545
Dynpro ..................................... 281
Enhancement Framework (Cont.)
enhancement spot ................... 93, 590
new BADI ......................... 589
Enhancement Information System .. 75, 93, 107, 599
Enqueue .................................. 333, 338
enterprise extension ............... 50
Entity relationship model (ERM) ... 211
Event ......................................... 463, 473
declaration and implementation ... 464
sender ....................................... 463
Event block .............................. 252, 262, 290
at line-selection ..................... 265, 288
at selection-screen ................. 262, 265, 286, 287
at user-command .................... 266, 288
details ..................................... 262
deadend .................................. 265, 286
dead-of-selection ................. 264, 286
get node .................................. 262, 264, 286
initialization ......................... 263, 286
load-of-program ..................... 262, 286
start-of-selection ................. 262, 263, 286
VABC-of-page ......................... 265, 286, 288
Examination
Strategy .................................. 42
Examine processing logic ........ 111
result ....................................... 111
source code ........................... 111, 115, 118-120, 122, 123,
127, 128
Exclusive lock ....................... 334
Exclusive mode ..................... 116, 117, 133
External mode ....................... 113, 115, 133
External view ......................... 235
Field .......................................... 381
OK COD .................................. 485
Ref_field ............................... 485, 486
Ref_table ............................... 485, 486
File interface ......................... 361
open dataset .......................... 361
Fixed point arithmetic ............ 152
Flat structure ......................... 141, 166, 342
type ........................................ 155
Flow logic .................................. 281
Index 632
Menu Painter ........................................................... 92
create GUI status ................................................... 392
Mesches ................................................................. 157
Message ................................................................. 252, 259, 283
class ................................................................. 284
number ................................................................. 284
placeholder .......................................................... 284
server ................................................................. 58, 61–63, 68
type ................................................................. 284
Method ................................................................. 441, 453, 473, 546
changing .............................................................. 459
default ................................................................. 459
exporting .............................................................. 459
importing .............................................................. 459
instance ............................................................... 275, 537, 541
optional ................................................................. 459
static ................................................................. 537, 543
Model View Controller → MVC .................................. 591
Modification .......................................................... 591
Modification adjustment ......................................... 562
Modification Assistant ........................................... 591
Modification Browser ............................................ 592
Modify ................................................................. 195, 221
Modify table .......................................................... 177
Modularization unit .................................................. 252, 256, 267, 268, 290, 496, 536
function module ..................................................... 256, 258, 261, 268, 270, 274, 289, 536
method ............................................................... 256, 258, 261, 268, 274, 289, 536
subroutine ............................................................. 256, 258, 259, 261, 268, 269, 271, 274, 496, 536
Module ................................................................. 381
Module pool program, create .................................... 406
MOVE-CORRESPONDING statement ......................... 187
Multiple inheritance .............................................. 541, 546, 552
MVC ................................................................. 509
application structure ............................................. 510
controller ............................................................. 510
model ................................................................. 510
view ................................................................. 510
Non-cumulative lock ............................................... 334
Non-Unicode system ............................................. 351
Numeric literal ...................................................... 145
Numeric type ........................................................ 151
Object ................................................................. 451, 473
context menu ....................................................... 79
instantiation ......................................................... 275
Object Navigator .................................................. 78, 79
Object-oriented programming .................................... 440
concept ............................................................... 441
On ................................................................. 381
Open SQL buffering ................................................ 219
read database records ............................................ 215
SQL Parser statements .......................................... 220
Outer join ............................................................ 335
Overloading ......................................................... 542
Package ............................................................ 95
global_switch ...................................................... 97
interface ........................................................... 97
nesting ............................................................... 96
papkparan .......................................................... 97
visibility ............................................................ 96
Package Builder ................................................... 95, 96
PAI ................................................................. 371, 373
Parameter .......................................................... 144, 150, 259, 268
changing .......................................................... 258, 261, 269, 271, 274, 535
default .............................................................. 419
input ............................................................... 258, 261, 269, 271, 274, 487, 613
memory ID ........................................................ 419
obligatory .......................................................... 418, 419
output ............................................................. 258, 261, 269, 271, 274, 535, 613
parameter ID ....................................................... 317, 419
pass by reference ................................................... 259, 261, 269, 290
pass by value ....................................................... 259, 261, 269, 273, 275, 290
radiation group .................................................... 419
returning ............................................................ 258, 274, 277, 535
table ................................................................. 258, 269, 271
count check ........................................................ 420
Parameters statement ............................................ 418
PBO ................................................................. 371, 373
Personalization ..................................................... 561, 562
Physical lock ......................................................... 224
Plug ................................................................. 513, 514
POH ................................................................. 373, 414
Polymorphism ...................................................... 442, 544
Pooled table ........................................................ 343
POV ................................................................. 373, 414
Predefined function for internal table ......................... 200
Presentation layer ................................................ 61, 222, 223, 255
user view ........................................................... 62
Primary index ....................................................... 324
Private attribute ................................................... 452
Procedure ........................................................ 262
block ............................................................... 258, 270, 272, 274, 546
exception ......................................................... 258
parameter ........................................................ 258
Process After Input → PAI ....................................... 577
Process Before Output → PBO ................................ 577
Process On Help-Request → POH ......................... 577
Process On Value-Request → POV ......................... 577
Processing block .................................................. 256, 258, 286, 290
dialog module ..................................................... 258, 614
event block ......................................................... 258, 261
procedure ........................................................ 258
Professional certification ......................................... 31
Program attribute ................................................ 351
Program context .................................................. 223
Program exits ....................................................... 568
Program type ....................................................... 285
class ................................................................. 257, 274
executable .......................................................... 256, 282
function group .................................................... 256, 257, 270, 272, 282
interface ........................................................... 257
module pool ......................................................... 256, 282
type group ........................................................ 257
Projection view .................................................... 337, 341, 434
Public attribute ................................................... 452
Quality management ............................................. 82

Index 633
Radio button ......................................................... 420
Read lock ........................................................... 334
Read table .......................................................... 177, 189, 191, 193
Redefinition ........................................................ 541
Refactoring Assistant ............................................ 541
Reference data type ............................................... 343
Reference table .................................................... 322
Reference type ..................................................... 158
REFRESH .......................................................... 198
Relational operator ................................................ 360
Remote Function Call → RFC ................................... 541
Repository Browser .............................................. 82, 107
create package .................................................... 95
Repository Information System ................................ 75, 84, 107
customize selection screen ..................................... 85
Repository object ................................................ 60, 69, 74, 82, 254, 255
create/edit .......................................................... 83
search ............................................................... 84
Reuse component ................................................ 215, 267, 270, 510
Reuse Library ...................................................... 79
RFC ................................................................. 59, 222, 256
Root component .................................................. 519
SAP application server ......................................... 48, 223, 224
SAP applications ................................................ 48
SAP Business ByDesign ......................................... 49
SAP Business One ................................................ 49
SAP Business Suite .............................................. 49
SAP Certified ABAP Development Associate ............ 33
SAP Community Network (SCN) ............................ 39
SAP examination ................................................... 41
SAP GUI .......................................................... 59, 61, 64, 65
SAP HANA ......................................................... 61
as standard database .......................................... 235
primary database ................................................ 61
SQLScript ........................................................ 235
view ................................................................. 235
SAP Help Portal .................................................. 39
SAP List Viewer → ALV .......................................... 39
SAP Logon ......................................................... 62
| Tool (Cont.) | Value .................................................................  150 |
|            | Value help .................................................. 603, 610 |
|            | Variable ....................................................... 148 |
|            | inline declaration ......................................... 148 |
|            | STATIC'S keyword ............................................. 149 |
|            | View container ................................................ 520 |
|            | context ......................................................... 513, 515, 524 |
|            | controller ....................................................... 513, 515, 523, 524 |
|            | define ............................................................ 335 |
|            | layout ............................................................. 519, 520, 524 |
|            | type ................................................................. 335 |
|            | UI element ....................................................... 511, 517, 519, 520, 522, 524 |
|            | Visibility section ............................................. 542, 547 |
|            | private ............................................................ 538, 539, 542, 543 |
|            | protected .......................................................... 539, 543 |
|            | public ................................................................. 538, 539, 543, 547 |

| U            | Warning message ............................................. 386 |
|             | Watchpoint .................................................... 111, 123, 132 |
|             | Web Application Builder .................................... 92 |
|             | Web Dynpro ..................................................... 508 |
|             | application ....................................................... 518 |
|             | component .......................................................... 510, 514 |
|             | component interface .......................................... 513, 518 |
|             | controller ......................................................... 511, 512, 514, 515 |
|             | event handling .................................................... 513, 517 |
|             | Explorer ............................................................ 93 |
|             | Text Browser ....................................................... 80 |
|             | view ................................................................. 511, 513, 514 |
|             | window ................................................................. 511, 514 |
|             | WebDynpro .......................................................... 53, 62 |
|             | Widening cast → Up cast ........................................ 58, 63, 64, 223, 233 |
|             | Work process ...................................................... 67 |
|             | determine number of ............................................. 542, 67 |
|             | register ............................................................ 64 |
|             | Worklist .............................................................. 79 |
|             | Write lock .......................................................... 334 |

| X            | XSTRING .......................................................... 153 |
Puneet Asthana is a UX and mobility architect and a recognized expert in ABAP development; he has more than 18 years of development experience, and has been working for SAP for over 15 years. Apart from ABAP development, he is also an expert in SAP Fiori, mobility, SAP Cloud Platform, SAP Cloud Platform Integration, IDocs, ABAP on SAP HANA development, ALE, EDI, workflow, and process integration.

David Haslam is a recognized expert in ABAP development, having worked for SAP for more than 23 years. David has led or participated in more than seven full lifecycle implementations of SAP, which include several multiple-phase projects and four large development projects. He was awarded the prestigious status of SAP Platinum Consultant in 2001, and he enjoys helping others through workshops and white papers, and sharing his knowledge and experience.