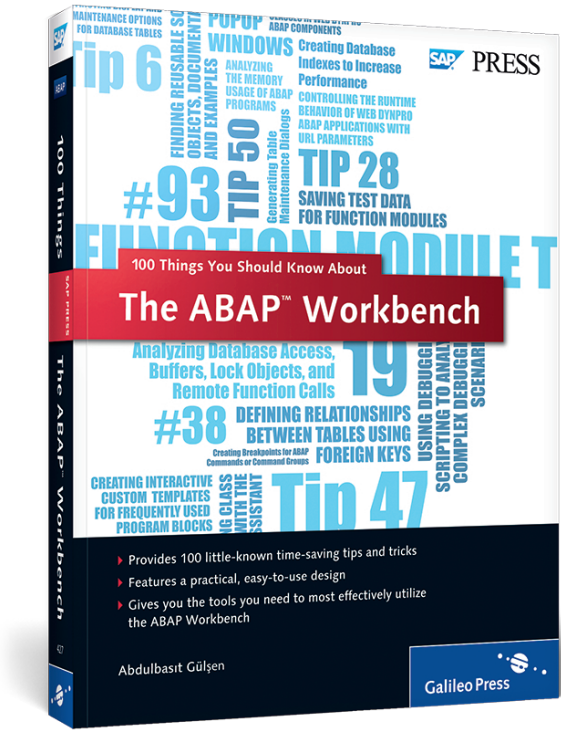
Abdulbasıt Gülşen

100 Things You Should Know About
ABAP® Workbench



Galileo Press

Bonn • Boston

Contents at a Glance

Contents

Introduction

SAP is continuously adding new products and solutions into the product portfolio to keep the platform up to date and meet the latest technology trends. Although some of these new products need to be developed in different programming languages, ABAP® still stands as a major programming language for software development in the SAP platform. It is a simple but powerful language for developers to build enterprise applications.

Several tools are available in the ABAP Workbench that make the development process easy and efficient. In this book, you'll find 100 little-known tips and tricks that teach you quick and practical techniques to solve your problems and increase your productivity while using the ABAP Workbench tools. This book neither gives detailed tutorials explaining how to use these ABAP Workbench tools nor teaches the ABAP programming language. Each tip has only 3-5 pages that focus on the specific problem or topic that you can immediately use in your daily work. If you want to get more background detail about any tool explained in the book, we recommend searching the SAP PRESS catalog for another book focused on the tool or refer to the SAP Community Network (*http://scn.sap.com*) or official documentation provided by SAP (*http://help.sap.com*).

The ABAP Workbench has many tools and features that cannot be limited to 100 tips. Tips that are included in the book are selected from the wide range of topics mostly to give you workarounds to perform your daily tasks in a more efficient way. Some tips in the book require expert-level knowledge, and some can be very simple for ABAP experts, but generally they are selected to address both beginner and advanced ABAP programmers. You can put this book into your library and consult it when you want to read tips about specific tools, or you can read from start to end. I was excited to learn many of these tips, and I hope you'll feel the same when you read tips that will save you time and effort on a daily basis.

Tips in the book are divided into nine different parts. In the first four parts, you'll find tips that mostly help you to improve your ABAP development capabilities by providing practical, alternative workaround solutions while you're using the ABAP Workbench tools to develop an ABAP program, such as the ABAP Editor, the Function Builder, and the Class Builder.

In Part 5, we move into the analysis phase of ABAP programming. In this part, you'll find very useful tips on the ABAP Debugger to help you find bugs in your ABAP programs. You'll see how the new ABAP Debugger is improved after it's launched. You'll also notice the latest improvements of the ABAP Debugger in SAP NetWeaver 7.3. We've added a footnote about the version restrictions in the individual tip title in all that apply.

You might also see some differences in the step-by-step procedures or screenshots because of the version difference. We've used the SAP NetWeaver 7.3 system to describe the step-by-step procedures and provide the screenshots. If you're using a different version, please note that there may be small differences on the tools.

Part 6 is also focused on problem analysis in ABAP programs. You'll find tips for various analysis tools in the ABAP Workbench. You'll learn practical ways of improving the code quality and find performance bottlenecks arising from wrong database or memory usages. There are also useful tips that will help you to analyze SAP Business Workflow problems.

In Part 7, you'll learn practical ways of using the ABAP Data Dictionary tools. The ABAP Data Dictionary has very broad features that can't be limited to a small part in a book. This part lists the most useful and practical tips on using the ABAP Data Dictionary tools.

Enhancement tips listed in Part 8 are included to give you practical ways of using the Enhancement Framework, which is also one of the most useful parts of the ABAP Workbench. In this part, you'll see different usages of the Enhancement Framework that will save a lot of time during patches or system upgrades.

Finally, in Part 9, you'll see some tips on Web Dynpro ABAP. Of course, it's impossible to fit tips about Web Dynpro ABAP in a small part of this book. We've added some tips about Web Dynpro ABAP that are mostly related to the other ABAP Workbench tools. If you want more details, we highly recommend the books *Getting Started with Web Dynpro ABAP* (SAP PRESS, 2010) and *Web Dynpro ABAP—The Comprehensive Guide* (SAP PRESS, 2013).

Part 5

ABAP Debugger

Things You'll Learn in this Section

Debuggers allow developers to locate and fix any bugs or problems that exist in the source code. You have two options available to find these problems:

1. Go through the logical steps of the program to see the state of variables step by step.
2. Use the debugger to analyze and extract the logic of the program.

In SAP, you can debug any ABAP program, whether it's custom developed or a standard program. This section provides tips, tricks, and techniques to improve your experience when using the ABAP Debugger and make it easier to analyze and find bugs in your ABAP programs.

Tip 37

Using SAP GUI Shortcuts to Debug Popup Windows

You can switch on debugging for popup windows by creating an SAP shortcut file for command /h and dragging this file onto a popup window.

Normally, if you know the place in the source code that you want to debug, you can easily place a breakpoint there, and a debugger will be triggered when you execute the program and the position that you marked is reached. You can also switch on debugging at any screen by entering the `/h` command into the command field on the SAP GUI toolbar. However, you can't use the `/h` command to stop popup windows because there is no command field. In this tip, we'll show you how to bypass this problem by using SAP GUI shortcuts to trigger the `/h` command on popup windows.

And Here's How ...

To switch on debugging for popup windows, you need to create an SAP GUI shortcut file. The SAP GUI shortcut is a file that can be used to execute transactions or system commands directly from a file.

You can use three different methods to create an SAP GUI shortcut, which we discuss in the following sections.

Method 1

Click the Generate a Shortcut button () on the SAP GUI toolbar. The window in Figure 1 appears; enter the details of the shortcut.

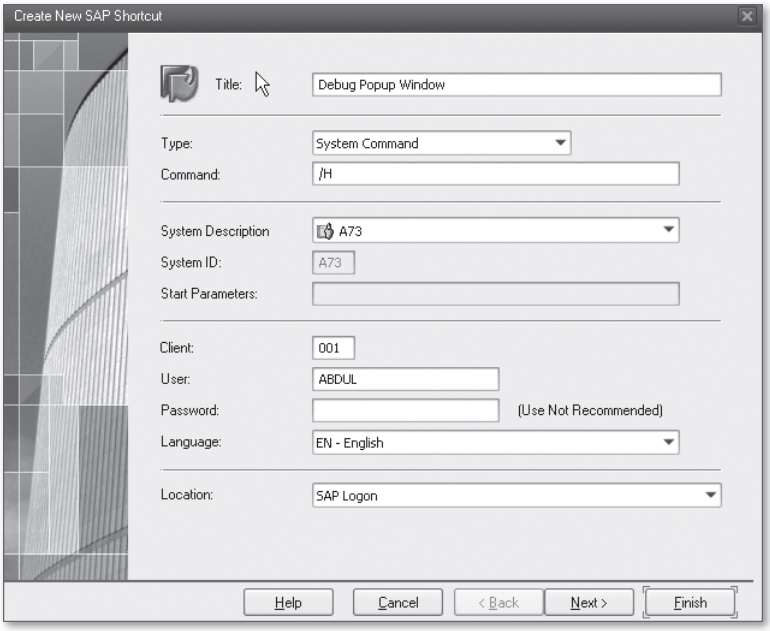


Figure 1 Creating an SAP GUI Shortcut

Follow these steps:

1. Enter a title in the TITLE field.
2. Select SYSTEM COMMAND from the TYPE dropdown.
3. Enter "/h" in the COMMAND field.
4. Click FINISH.

A shortcut file with an extension *.sap* is created on the desktop.

Method 2

1. Right-click on the Windows desktop.
2. Select NEW – SAP GUI SHORTCUT.
3. Enter the name of the file and press Enter.
4. Right-click on the file, and select EDIT from the context menu.
5. Select SYSTEM COMMAND from the TYPE dropdown.
6. Enter "/h" in the COMMAND field.
7. Click OK.

Method 3

Create a text file named *Debugger.sap,* open it in any text editor (e.g., Notepad), and write the following text into the file:

```
[Function]
Command=/H
Title=Debugger
Type=SystemCommand
```

Debug the Popup

Now you have created a SAP GUI shortcut file. Open the popup window that you want to debug, and drag the shortcut file onto the popup as shown in Figure 2.

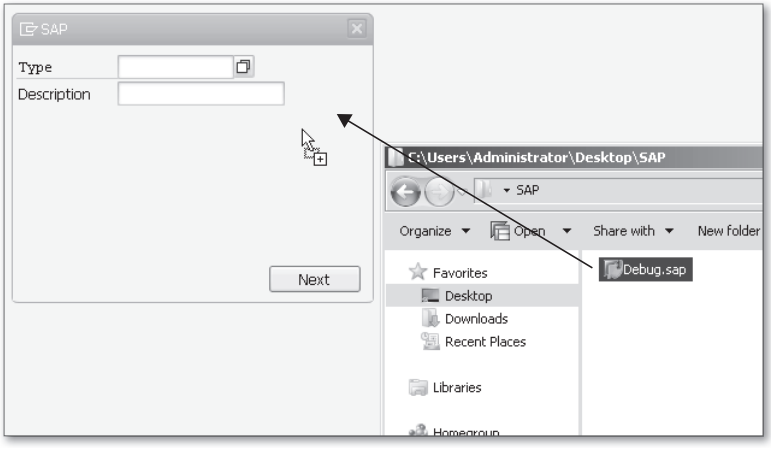


Figure 2 Dragging a Shortcut File onto a Popup Window

When you drop the file on the popup window, you'll see the DEBUGGING SWITCHED ON message on the status bar. Now you can start the debugger tool by pressing the Enter key.

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

X

Y

Z