

Dr. Boris Rubarth

First Steps in ABAP®

- ▶ Step-by-Step Instructions for Beginners
- ▶ Comprehensive Descriptions and Code Examples
- ▶ Guide to Create Your First ABAP Application
- ▶ Various Tutorials Provide Answers for Programming Questions

Table of Contents

Preface	5
1 Getting Started with ABAP	7
1.1 Hello ABAP	7
1.2 Using the ABAP Editor	8
1.3 Starting the report	13
1.4 Using ABAP keyword documentation	15
1.5 Viewing existing code	16
1.6 Versioning your report	18
1.7 Conclusion	18
2 Using ABAP Statements	21
2.1 Using parameters as input variables	21
2.2 Using conditional branches	25
2.3 Using texts that can be translated	26
2.4 Working with string variables	28
2.5 Using elementary data types	30
2.6 Debugging your report	33
2.7 Conclusion	36
3 Using the ABAP Workbench	37
3.1 Starting the ABAP Workbench	37
3.2 Configuring ABAP Workbench options	39
3.3 Working with packages and transport requests	41

3.4	Transaction code for a report	46
4	Using data types from ABAP Data Dictionary (DDIC)	51
4.1	Getting started with complex data types	51
4.2	Working with the ABAP Data Dictionary (DDIC)	55
4.3	Working with internal tables	60
5	Modularization and reusing functionality	67
5.1	Using form routines	67
5.2	Using function modules	71
5.3	Using ABAP classes	90
5.4	Exception concept for classes	101
6	Accessing the database	105
6.1	Using ABAP statements to access the database	105
7	ABAP User Interface Technologies	115
7.1	Working with messages	115
7.2	Working with list processing	117
7.3	Working with screens and DynPros	127
7.4	Short overview on SAP UI technologies	144
8	Conclusion	151
A	The Author	154
B	Disclaimer	155

2 Using ABAP Statements

Now let's take a closer look at ABAP statements, and expand our view to variables. In this chapter, readers will learn how to define variables and enable the user of a report to specify values for variables.

Debugging an application allows developers to follow the execution flow step by step in order to check the application. I will also introduce the ABAP debugger in this unit.

2.1 Using parameters as input variables

The statement PARAMETERS (always plural) is a simple way to allow users to provide input to a report that is stored in a variable. Specify the name of the parameter, its type, and if you would like a default value:

```
PARAMETERS pa_name type c length 12 DEFAULT 'HUGO'.
```

(In this example, we already used the type *c* for character which will be explained shortly, but yes: it is the type you expect it to be).

When you start the report, the system will display a simple input field with the given default value that you can overwrite, see Figure 2.1.



Figure 2.1: A parameter as an input variable

The parameter serves as an input field and as a variable.

Now let's use the input variable to make our welcome greeting a bit more personal, introducing chain statements as well:

```
WRITE: 'Hello', pa_name.
```

Chain statements



A chain statement combines several statements that use the same keyword into one statement that will be applied to several data objects. It requires a colon after the keyword (here: WRITE), and a comma between the data objects that the keyword applies to.

Did you try your own name with some lower case letters and find that they were displayed as upper case letters? You can use the *keyword addition* “lower case” for the PARAMETERS keyword to account for lower case letters. Note that this is not a *conversion* to lower case, but the permission for lower case letters (omit the conversion to upper case).

Still, the user interface (UI) in Figure 2.1 shows the name of the input variable, which is not always understandable to the user. If you consider that the length of a parameter name is restricted to 8 characters, you may want to include more meaningful text.

You can provide additional text for this variable by using the menu GOTO • TEXT ELEMENTS • SELECTION TEXTS. Enter “Your name” for the *selection text* of PA_NAME, as shown in Figure 2.2.

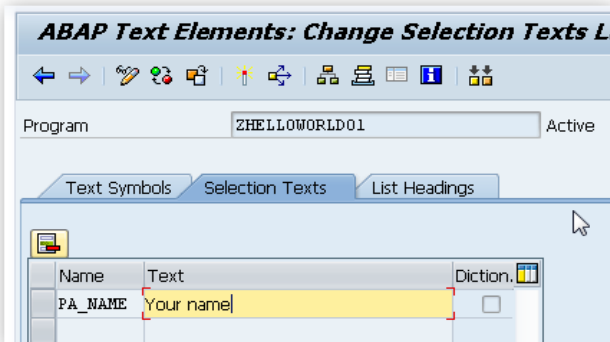


Figure 2.2: Maintenance of text for a parameter variable

It is important to note that you have to activate the selection text in this view before returning to the coding of your report (by using F3).

In the case that you have more than one object in an inactive state, the system will display these objects in a list. In this manner, you can selectively activate some of the objects, or all of them. In the following example, the report is inactive. When you activate the selection text both objects are part of the displayed list: object REPS (S for source), and object REPT (T for text), see Figure 2.3.

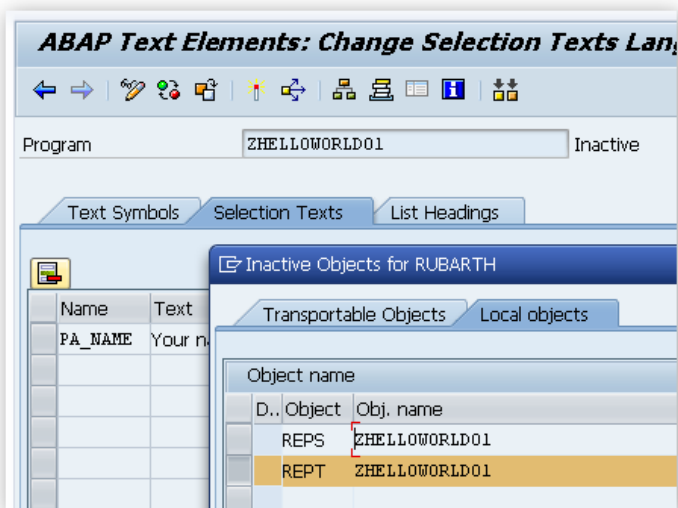


Figure 2.3: List of inactive objects

List of inactive objects



At a later stage of your ABAP development practice, you may come across a situation where the list of inactive objects is also shown, although it looks as if you have just one inactive object.

Check the tab “Transportable Objects” to look for additional inactive objects. The *transport* of developed objects will be introduced and explained later.

The advantage of a selection text is that this text can be translated into other languages. If you choose a different language during log on, your report will show the text for the parameter in that language (assuming that the text was translated into that language).

Selection texts and selection screen



The parameter text is called “selection text” because it specifies the data selection. An advanced report that reads data from the database will have the following sequence: first the selection criteria have to be specified by the user (parameters statement), then the database access uses the selection criteria, and then the selected data are displayed to the user.

The display of the selection texts is called the selection screen. The selection screen is assigned to the internal screen number 1000 and will be covered in more detail later.

To summarize:

- ▶ A PARAMETERS statement declares a local variable.
- ▶ At the beginning of the report, the selection screen displays an input field, so that the user can specify the value of the variable.
- ▶ The input field on the selection screen has a text label associated with it that can be translated.

2.2 Using conditional branches

Now let's look at using an IF statement to determine if the user has changed the default value for a parameter. Using an IF statement is easy. Remember – don't use