



# ABAP Course

## Chapter 2: Development Environment

### Content

The second chapter explains the main tools when developing ABAP programs. You will learn how to use the pretty printer, the version feature, the ABAP help, the patterns and the debugger.

### Prerequisites

Before starting the exercises you should successfully proceed through chapter 1.

### Motivation

This chapter explains how to use the main tools in the development environment to make developers' life easier. The shown tools are useful and helpful in the daily work.

### Lecture notes

The fundamental understanding of the ABAP development in the SAP system is a prerequisite for the students. Students can go on with their account from chapter 1.

- **Product:** All
- **Level:** Beginner
- **Focus:** Programming
- **Version:** 1.0
- **Author:** UCC Technische Universität München

**Task 1: Login into the SAP system**

**Short description:** Use SAPGui to login into the SAP system with your username and password

Start the SAPGui and login into the development system using the provided account and password. Please refer to chapter 1 for your username and your password.

*Login***Task 2: Use the pretty printer**

**Short description:** Use the pretty printer to enhance the readability of your programs

Please start the Object Navigator from the SAP Easy Access Menu by using the following path:

**Tools • ABAP Workbench • Overview • Object Navigator.**

*Menu path*


You may also use the transaction code **SE80** for direct access.

Open your program ZY\_##\_HELLOWORLD by double clicking on the program name in the left tree. The SAP system opens the program now in the display mode.

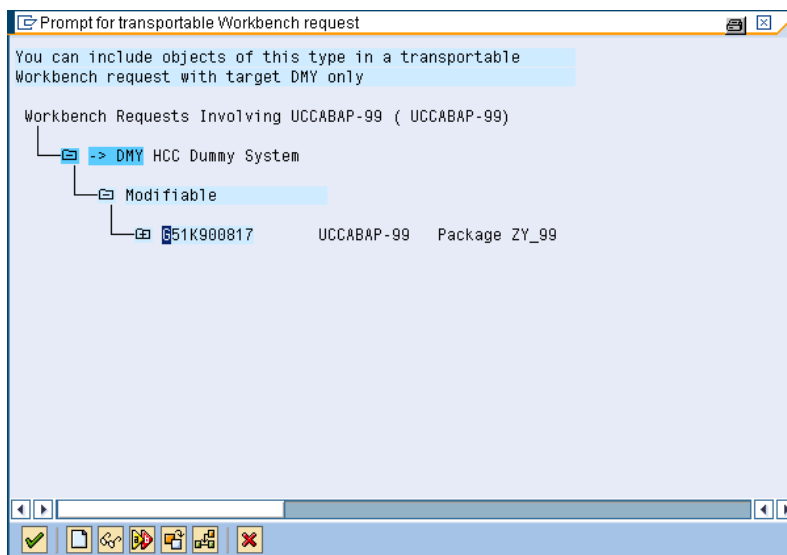
**Hint:**

Generally there are two modes: **display mode** for only having a look at a program and the **change mode** for changing a program. You can switch between both modes

*Useful hint*

by using the little pencil/glass symbol: .

Please switch to the change mode of your program and save your program (without making changes to it). The SAP system asks you for a new transport request. This is because you had already released the original transport request. For saving the program the SAP systems needs a transport request. From the pop-up click the '**Own Requests**' button to get a list with all of your transport requests and select the transport request which you used for the creation of your package ZY\_##.

*Change mode and save program*

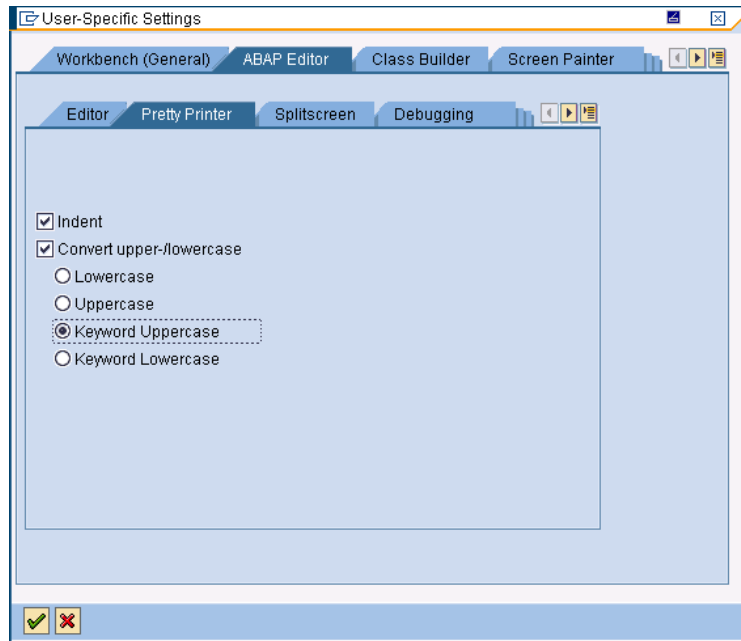
Now you want to configure the Pretty Printer. Therefore please call the menu via:

**Utilities • Settings**

*Menu path*

Switch to the '**ABAP Editor**' tab and navigate to the '**Pretty Printer**' tab. If it is not selected activate the '**Convert upper-/lowercase**' checkbox. As we want to see all keywords in uppercase select the '**Keyword Uppercase**' button.

*Configure  
Pretty Printer*



Go back to the Object Navigator and test the Pretty Printer by clicking on the '**Pretty Printer**' button. You will see the result of the Pretty Printer work instantly. The Pretty Printer has changed the write statement from lowercase to uppercase. The Pretty Printer also checks the syntax of the program and reports errors if there are any.

*Run Pretty  
Printer*

### Task 3: Use the version feature

**Short description:** Use the version feature to create a new version of your program


At first have a look at the versions of your program. Switch to the version management by navigating the following path:

**Utilities • Versions • Version Management**

*Menu path*

Your screen should look similar to this:

Version	Cat	Flg	SAP	Rel.	Arch	Request	Project	Date	Time	Author
Version(s) in the development database:										
modif			640			651K900018		26.09.2008	14:12:57	UCCABAP-99
activ			640			651K900018		25.09.2008	09:58:38	UCCABAP-99
Version(s) in the version database:										
000001			640			651K900015		25.09.2008	11:16:15	UCCABAP-99

By selecting a version you can have a look at it by clicking the glass button: . Furthermore you can compare two versions and retrieve a version from the version database.

Hint:

The modif version is the version you create when modifying a program. Beside the modif version there is always an active version which remains active unless you activate your modified version successfully. This means the SAP system always keeps an active version of a program to ensure the functionality of the system.

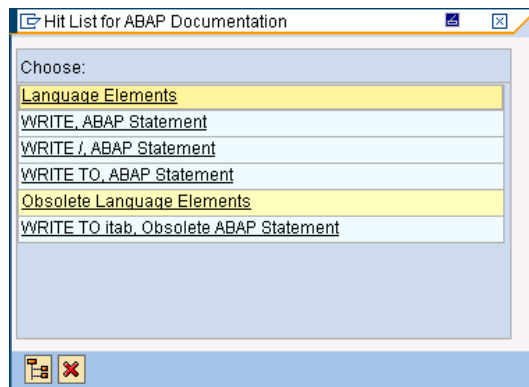
*Useful hint*

#### Task 4: ABAP help and patterns

**Short description:** Use the ABAP help to get hints about usage of an instruction and use a pattern to save work time

In this task we want to use the ABAP help to gain knowledge about the write instruction. Let's imagine we are not very familiar with the statement. Therefore we use the ABAP help. The help is activated by positioning the cursor on the write statement in your program. Then press the 'F1' button. The system is now searching for all related entries in the ABAP help and will come up with the results.

*Press F1*



You see the different derivatives of the write statement. Please select the first item by clicking on it. The SAP system now opens the ABAP help and jumps directly to the right entry in the ABAP help. Please navigate to the node '**WRITE – list\_elements**' by double clicking on it.

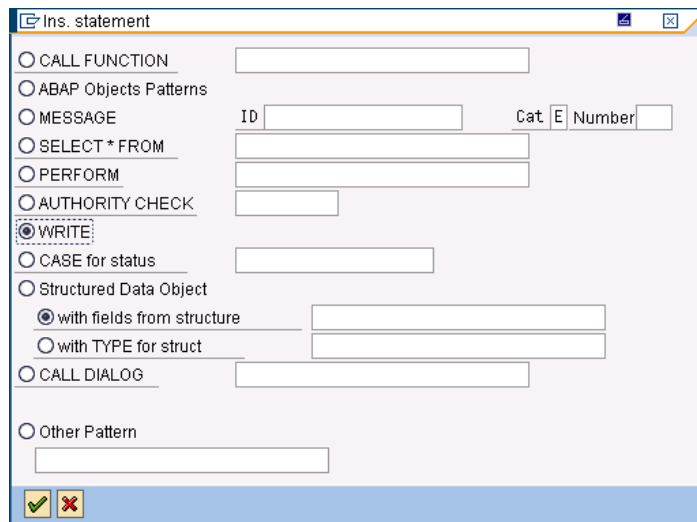
*Click on  
WRITE*



This write derivate can be used to write text into a report and automatically insert a checkbox, icon, symbol or line into the report too.

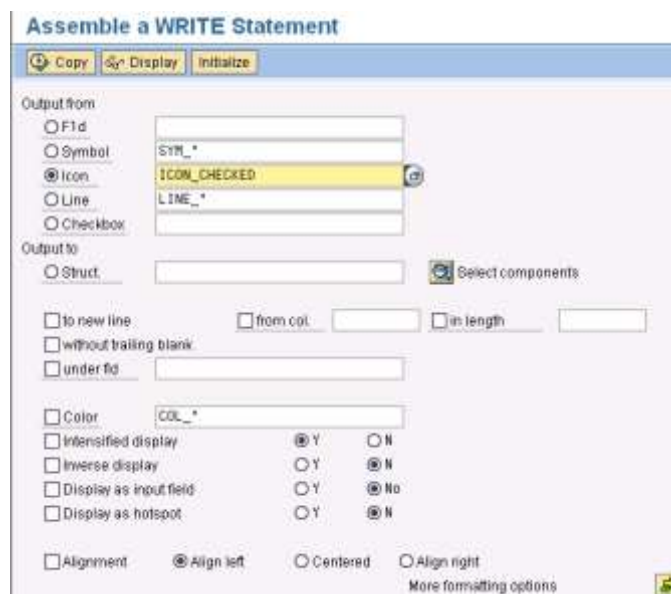
In the next step we want to use the new statement now. Therefore you take advantage of the **'Pattern'** button. At first close the ABAP help and go back to your program. Click on the **'Pattern'** button. A new window comes up. Please select the **'WRITE'** entry from the list and click on the green check mark.

*Use Pattern  
button and  
select  
WRITE*



You see that the screen changes now and gives you some input fields. Type in **'ICON\_CHECKED'** into the icon input field and press the **'COPY'** button. This will copy the statement into your program code.

*ICON\_CHECKED*



Before activating the new program you have to add one additional instruction to your program: **INCLUDE <list>**. This ensures the SAP system knows about the icon you used in the write statement. Your program should look similar to this:

*INCLUDE*

```

*&-----*
*& Report  ZY_99_HELLOWORLD
*&
*&-----*
*&
*&
*&-----*

REPORT  zy_99_helloworld.

INCLUDE <list>|.

WRITE 'Hello World!'.
WRITE icon_checked AS ICON .

```

Now save, activate and test your new program. You will see a small icon next to 'Hello World!'.

### Task 5: Debugging

**Short description:** Use the debugger to debug programs and get knowledge about problems

In this task we want to use the debugger to get knowledge about problems during the program execution. The debugger is a very helpful and useful utility in the SAP system.

First start your program ZY\_##\_HELLOWORLD in the debug mode using the menu path:

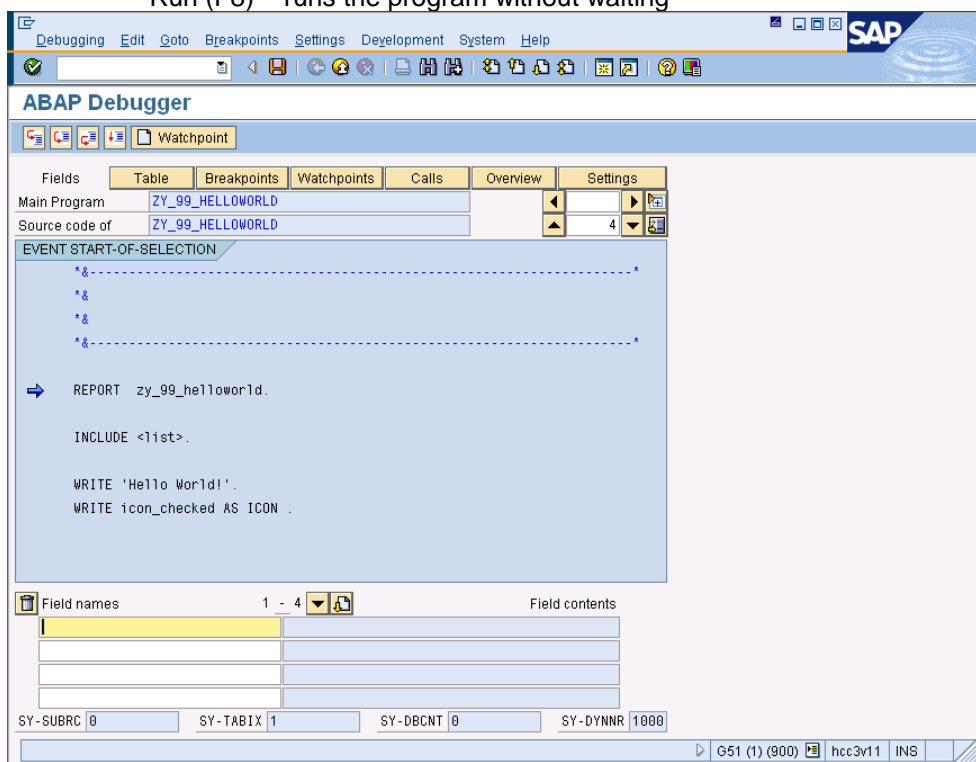
#### Program • Test • Debugging

*Menu path*

This will automatically start the program in debug mode and you will switch to the debug mode. You can control the debugger using four buttons:

- Single step (F5) – executes one step and waits for your input
- Execute (F6) – executes a bunch of steps and waits
- Return (F7) – runs the program until a return
- Run (F8) – runs the program without waiting

*Debugger controls*



The blue arrow always indicates at which step you are. Please keep in mind that the debugger also follows program paths. This means when calling external programs, modules or services the debugger will also debug the external components.

Beside the sequential program execution, the debugger provides you with some other features e.g. the variable allocation or table allocation. When working with tables later on you will wonder which data is in the table. This can be checked in the debugger during runtime. Moreover you see the current returncode of the program. It is written into the input field **SY-SUBRC**.