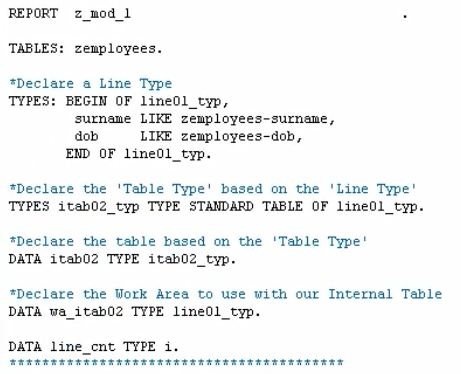
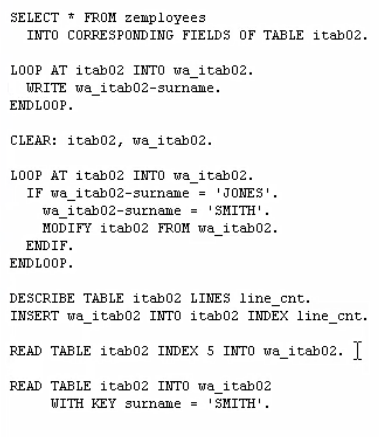
Source Code Modulars. Now when we talk about modularizing our program, all we mean is we are going to take a sequence of above statements in our program and place them in a separate module. Then in our normal ABAP program, we just call this module.





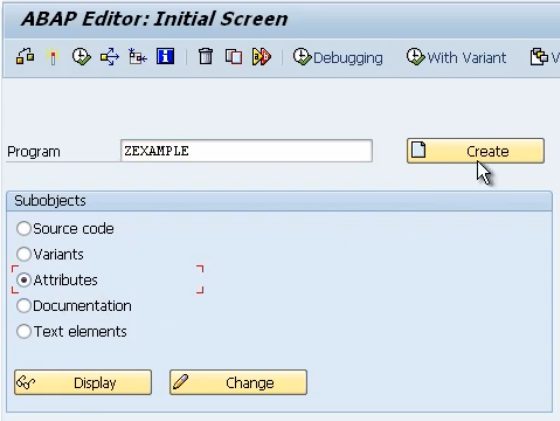
First, we are going to look at using what is called Include Programs. Now Include Programs, are made available globally within an SAP system, and their sole purpose is just for modularizing code. They are very simple to define and accept no parameters.

So first, just under the report statements at the very top of our program, let's fill in the statement to declare an Include.

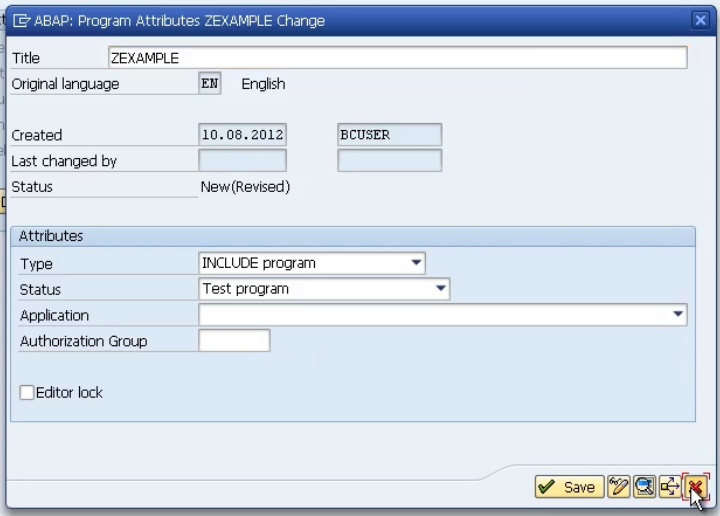


All we need to do is type, Include, and then we just define the name. So, for this example, let's do Z\_Employee\_Definitions, with a full stop at the end. we're telling the program here, is to include our Include Program that we've just named. Into our main program. And you probably worked out, that we haven't created this include yet.

You have two ways of creating an Include File. The first one, I open a new session and bring up SE38 again.

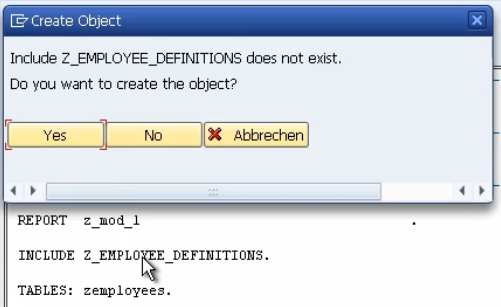


You can key in your include name, but I'm not going to do it here, I'll just Z example. But instead of saying Source Code, and you create your program, you're going to create Program Attributes.

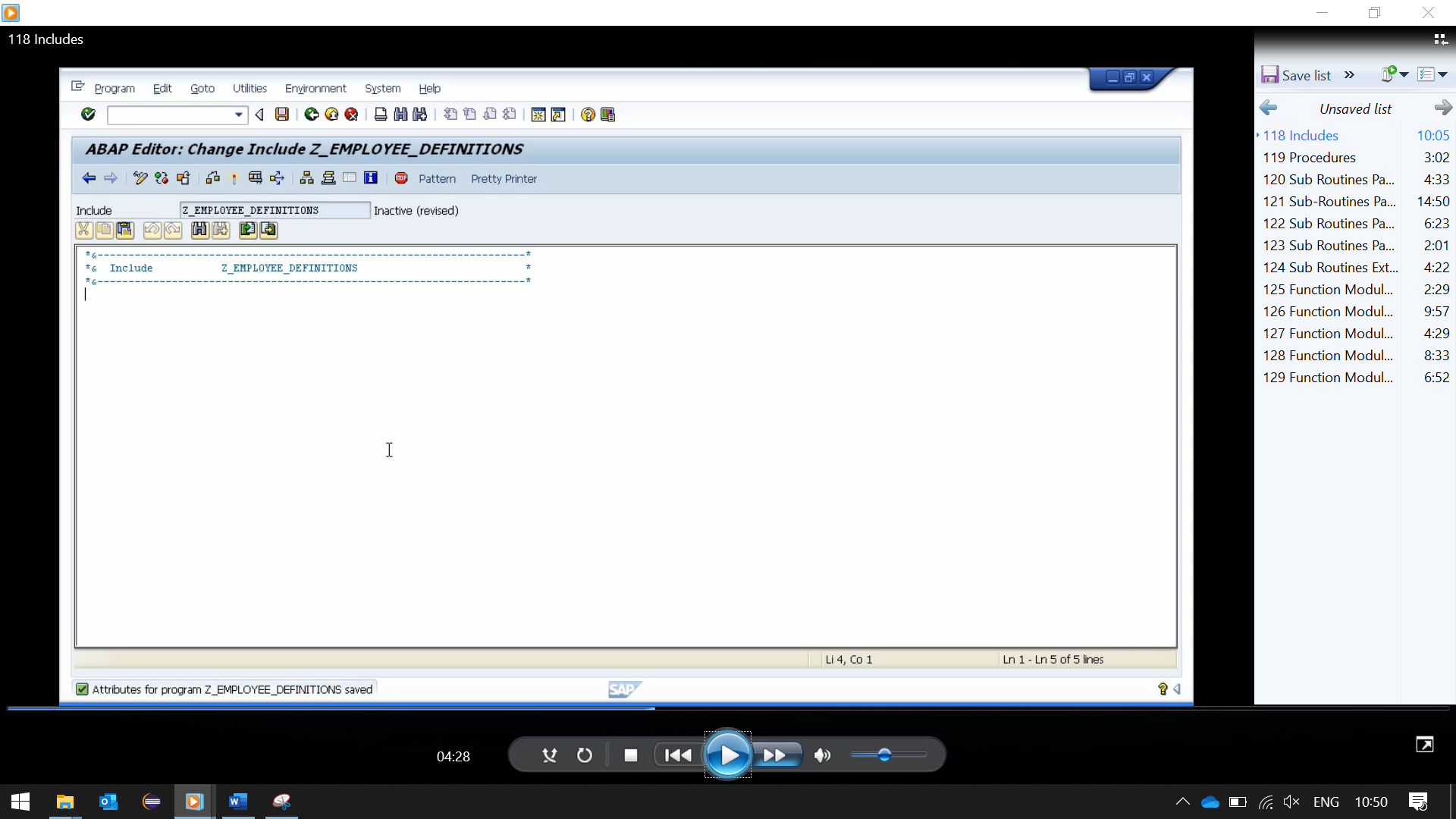


So here we just give it a title. When we come down to type, instead of choosing executable program, we will choose Include Program.

Now the second method, is something I'm sure you're familiar with already. And that's by using Forward Navigation.

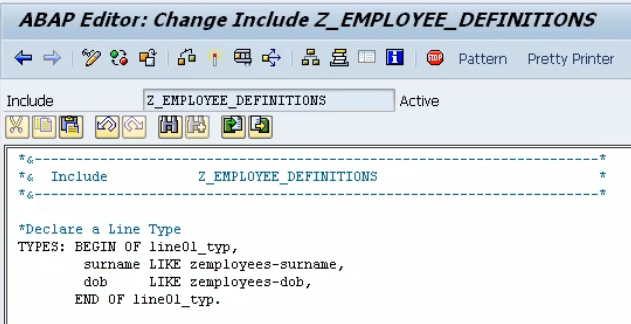


all you need to do, is once you type the include, just give it a double click, and we get the pop-up box saying Your Include Does Not Exist.



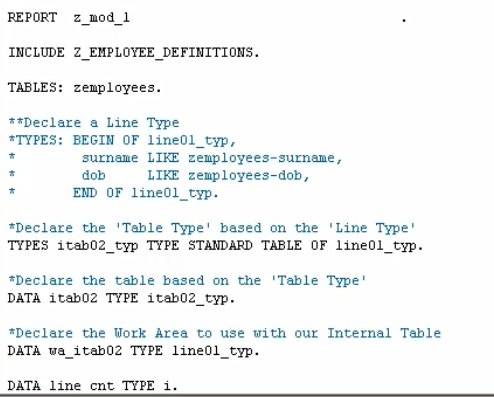
and it gives us a blank program where we can put our code.

we use the Go To menu, and have a look at the program attributes. We choose Yes to save. You can see it's the same screen. It's chose the Include Program, it's a test, it's assigned to basis, and there's our development class. Now remember, this Include File, or Program, is a separate file on the SAP system, and can be included in any other program. So, you must treat it as a separate program. And you'll see that it's got its own check, it needs activating, and so on.



our employee definition include already, so we just give it a green tick.

Whenever we activate this program, our main one. It will automatically check to make sure our include program has already been activated.



when we do syntax checks in our main program, it will also carry out that syntax check, on the inactive version of you include, so they work in tandem.