# **AIF Cross Location Training**

# **28th to 30th July 2020**

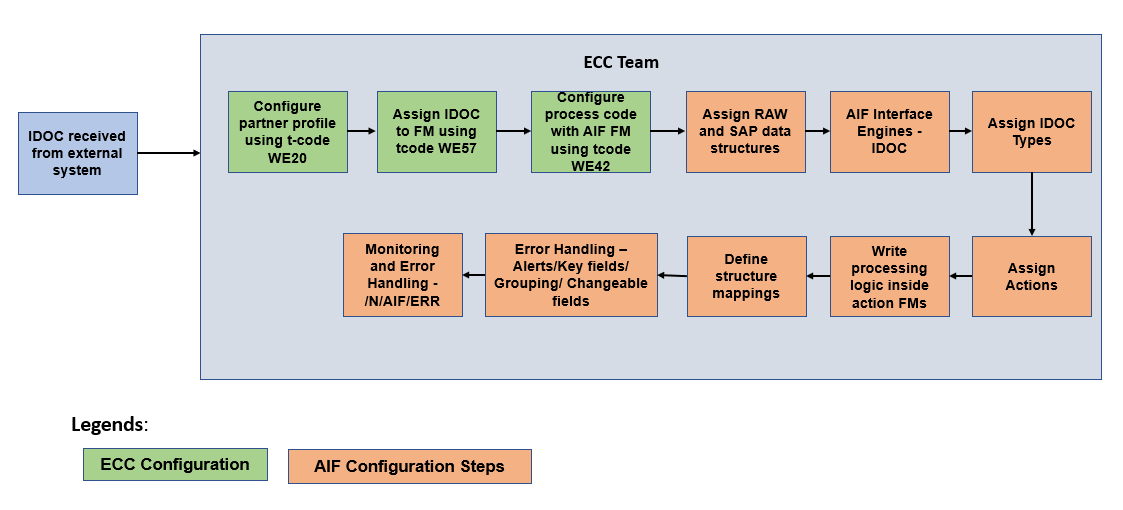
**How to call AIF interface from another interface**

**Scenario Description:**

We will create a scenario where a MATMAS05 IDOC will be an input to the ECC system which will contain multiple segments with details for individual IDOCs.

The input IDOC will trigger our first interface and create individual IDOCs by triggering the second interface multiple times to process each IDOC.

The interface development process flow is shown below:



**Calling AIF within AIF**

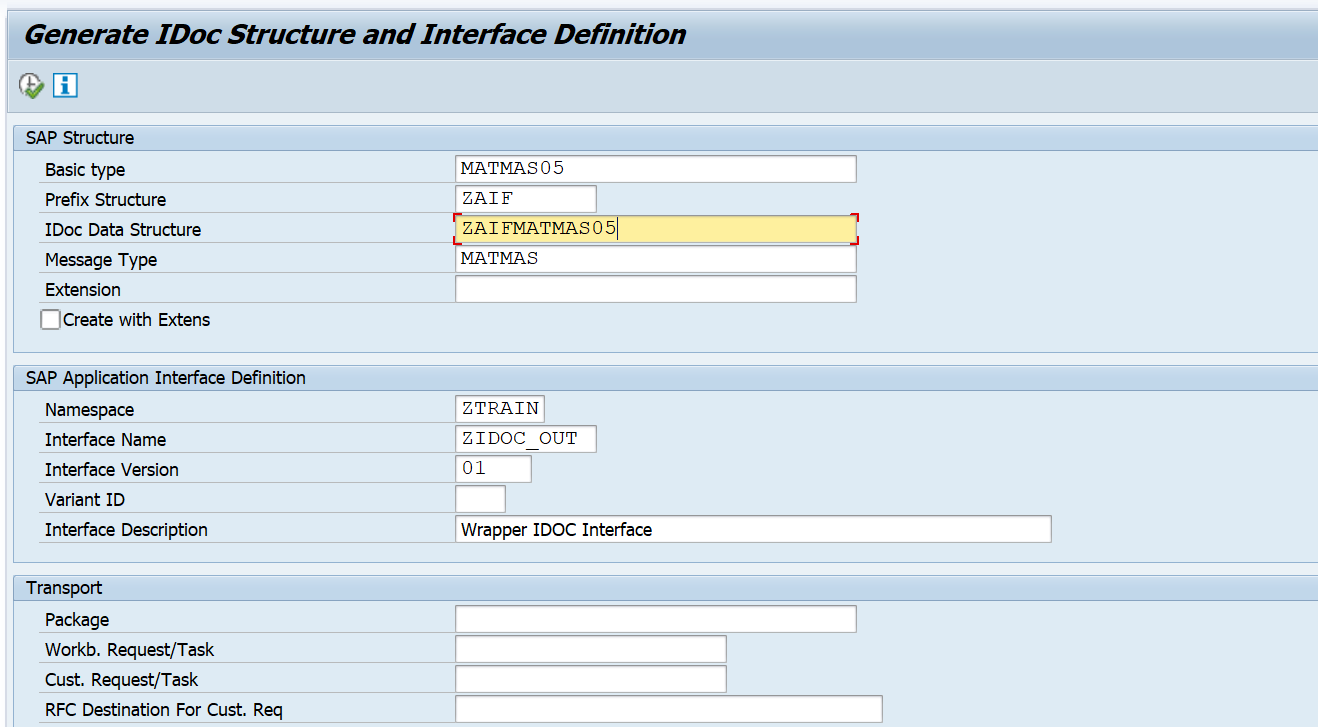
Action: An action can be assigned to an interface. An action is an interface building block that contains the business logic of an SAP Application Interface Framework interface. It consists of at least one action ‘function module’.

Manually trigger AIF interface: The super-interface will be triggered by the IDOC itself. But the second interface will be called within Action Function module of super-interface by triggering an IDOC which will internally trigger second interface.

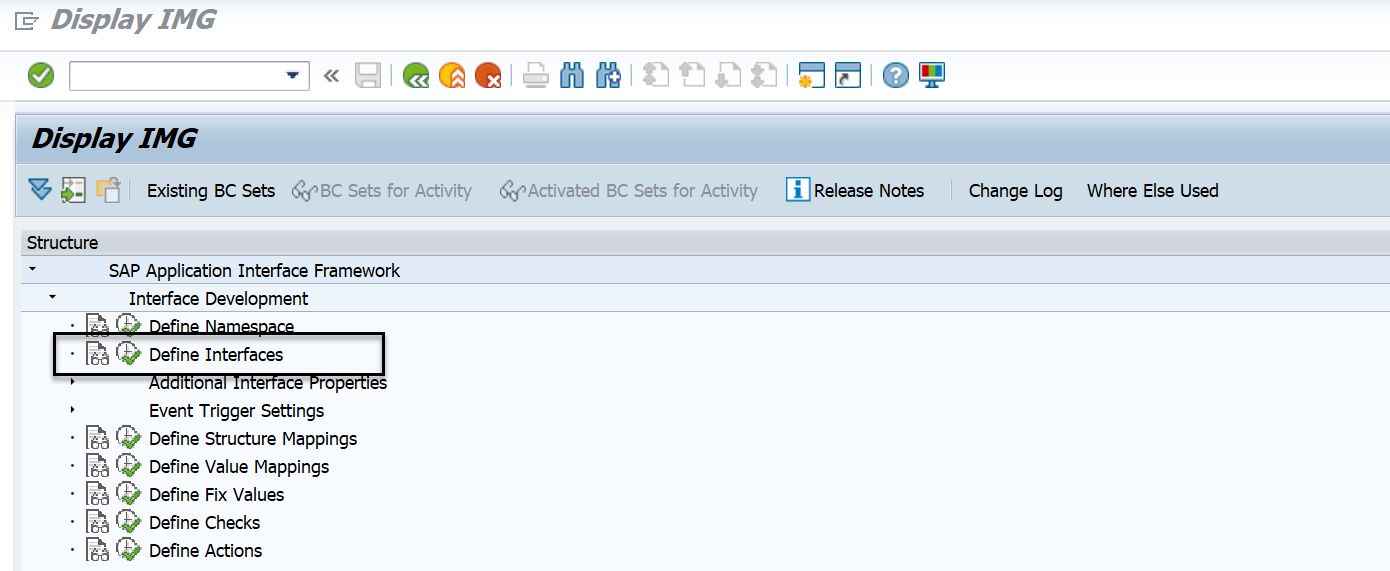
# **Step-by-step Guide on How to call AIF interface from another AIF interface**

We will configure interfaces and call second interface within Action FM of first interface

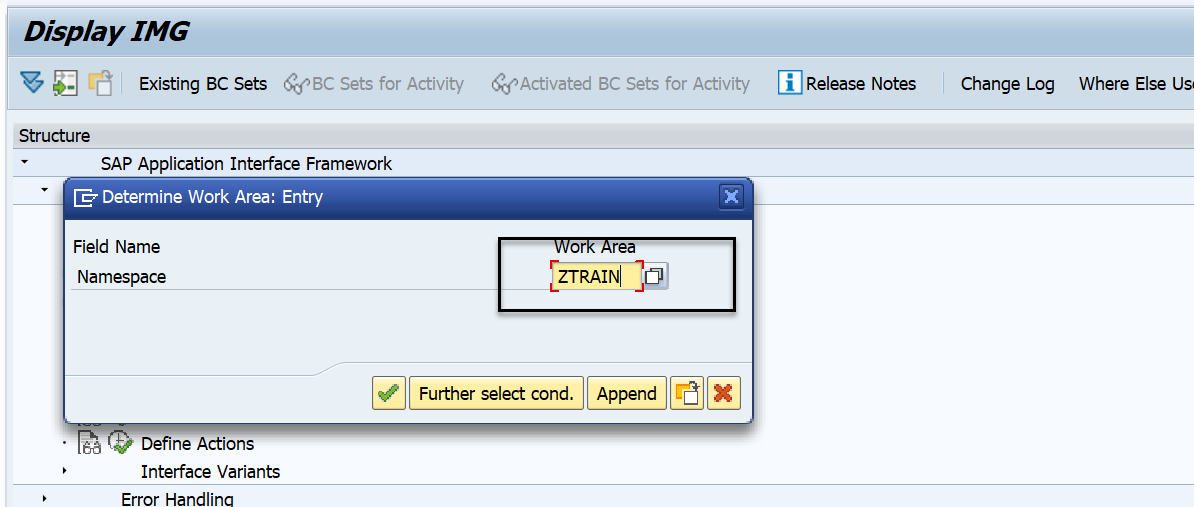
* First, we need to generate a new IDOC structure which will be used in AIF. We can do this in transaction – /AIF/IDOC\_GEN



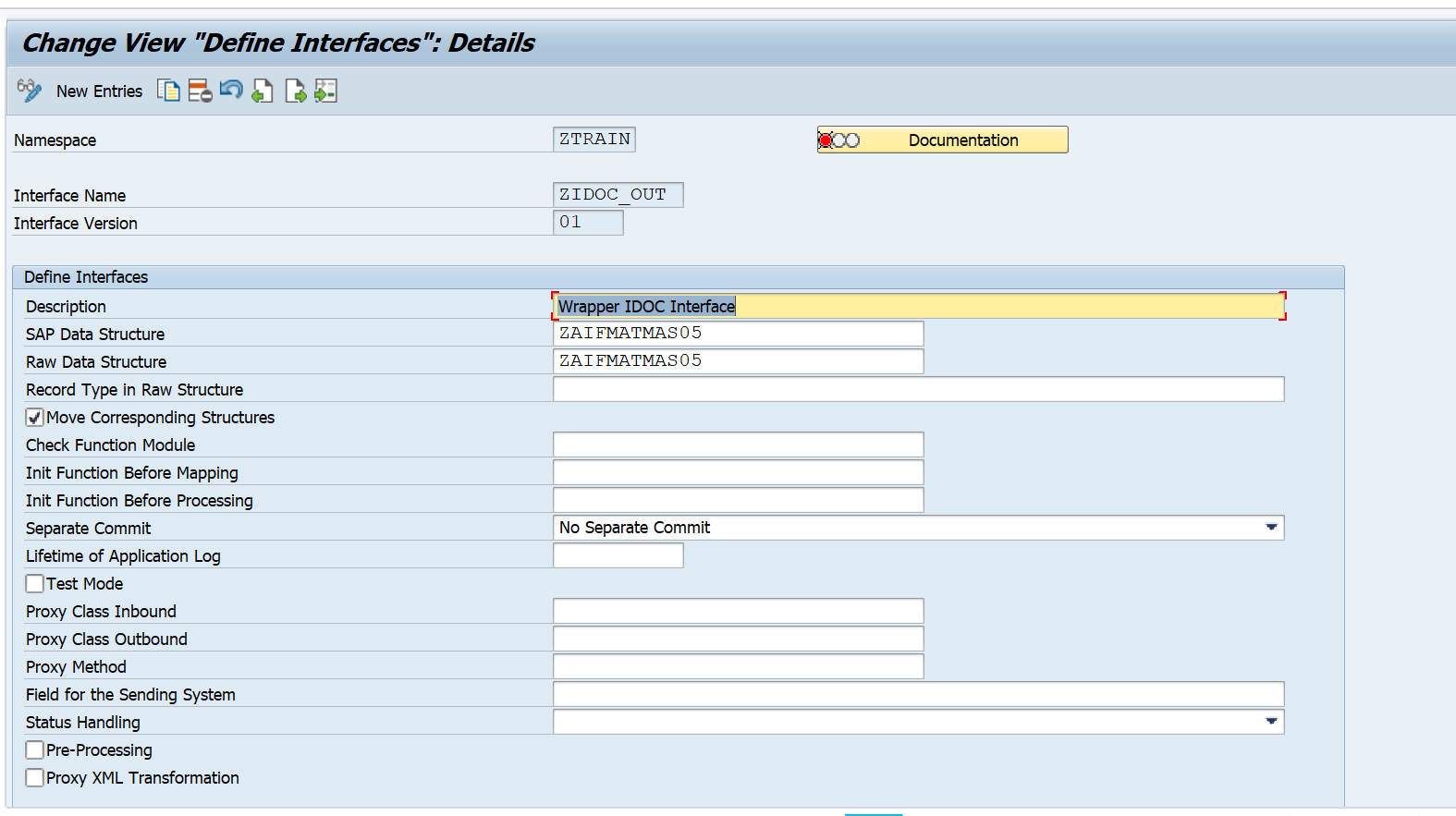
* Go to tcode /N/AIF/CUST and select Define Interfaces under Interfaces Development



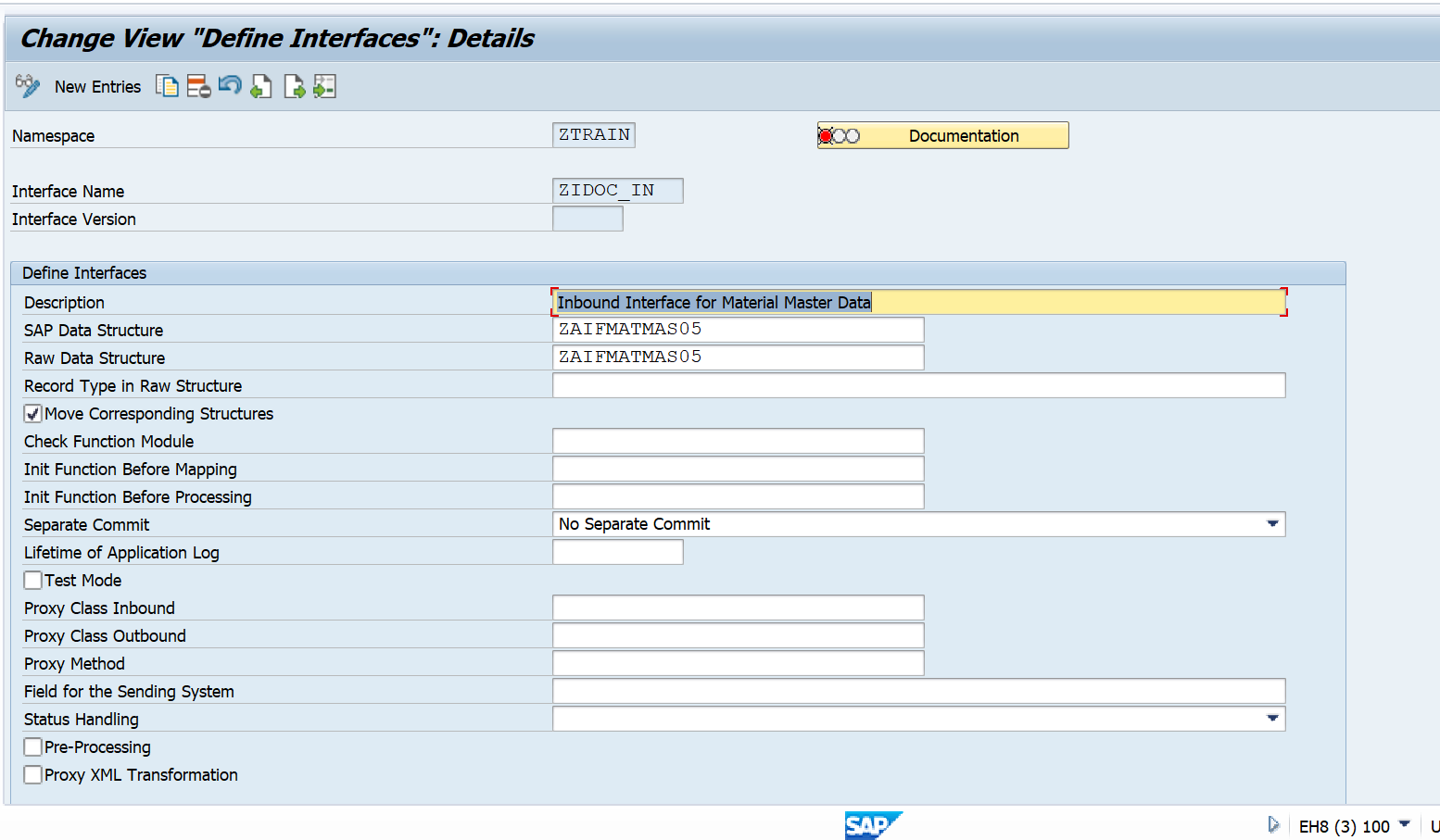
* Select the namespace ‘ZTRAIN’.



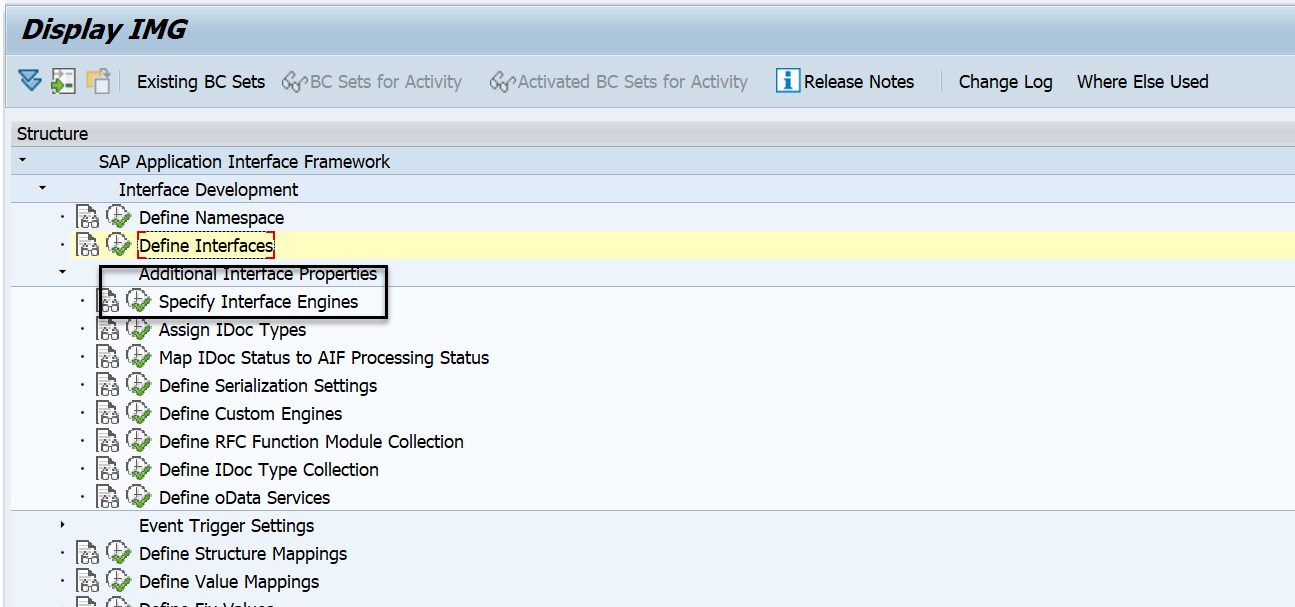
* Select New Entries to create a New Interface. Enter the below details and SAVE



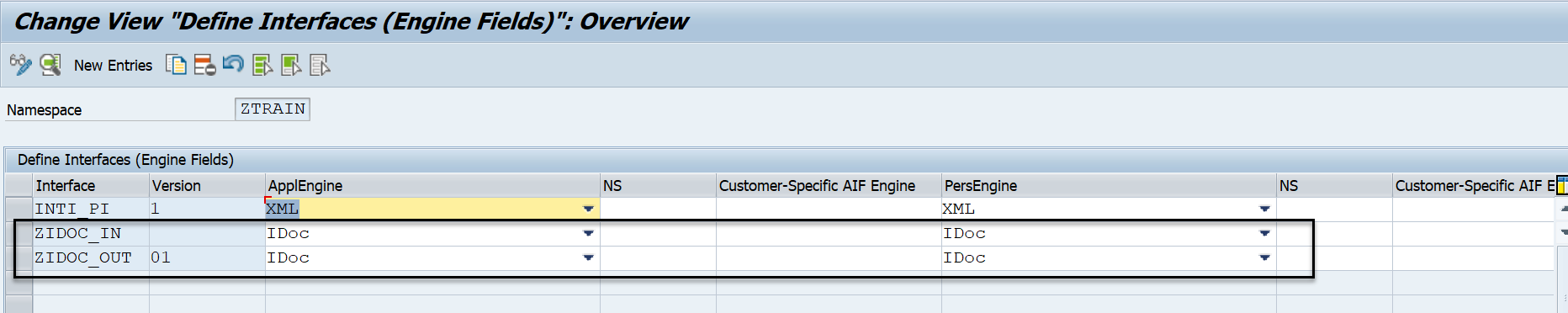
* Create another entry for the second interface



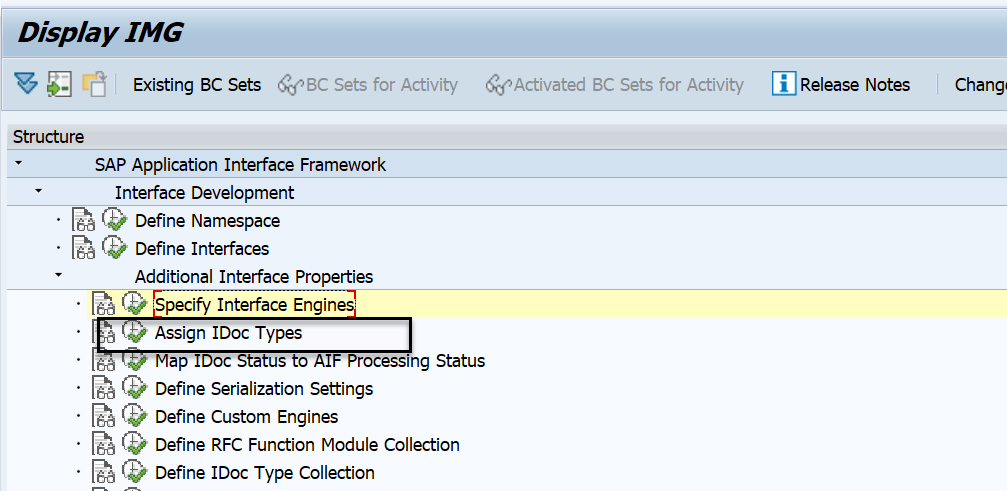
* Now, go back to /AIF/CUST and select Specify Interface Engines



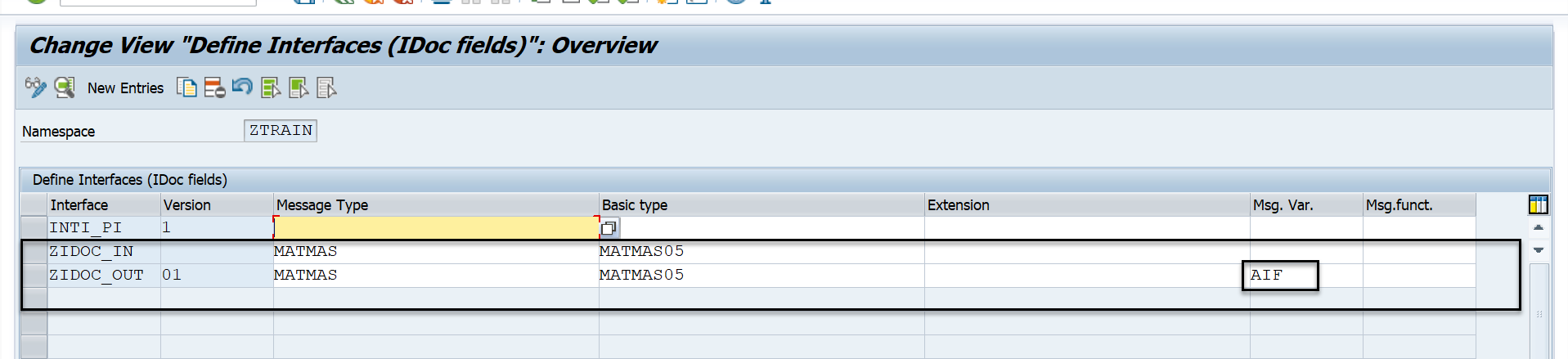
* Change the application and persistence engine to IDOC for both the interfaces



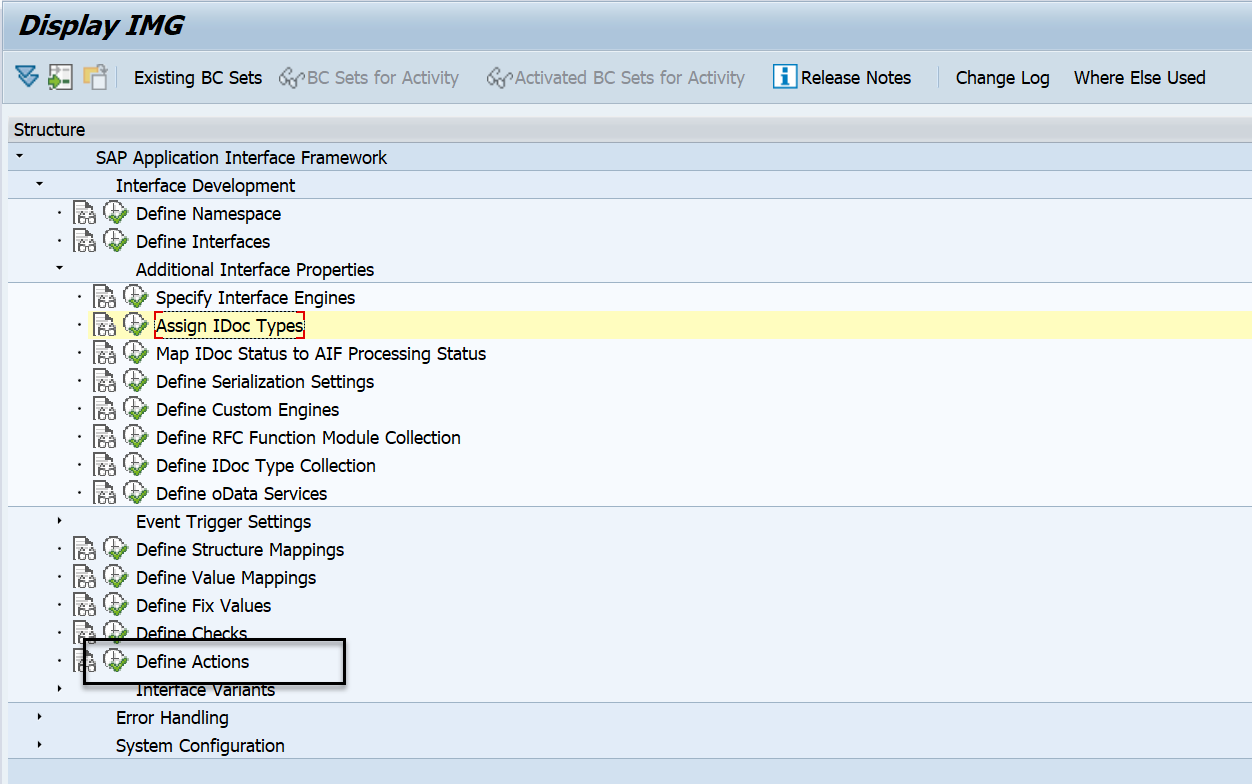
* Go back and select Assign IDOC types



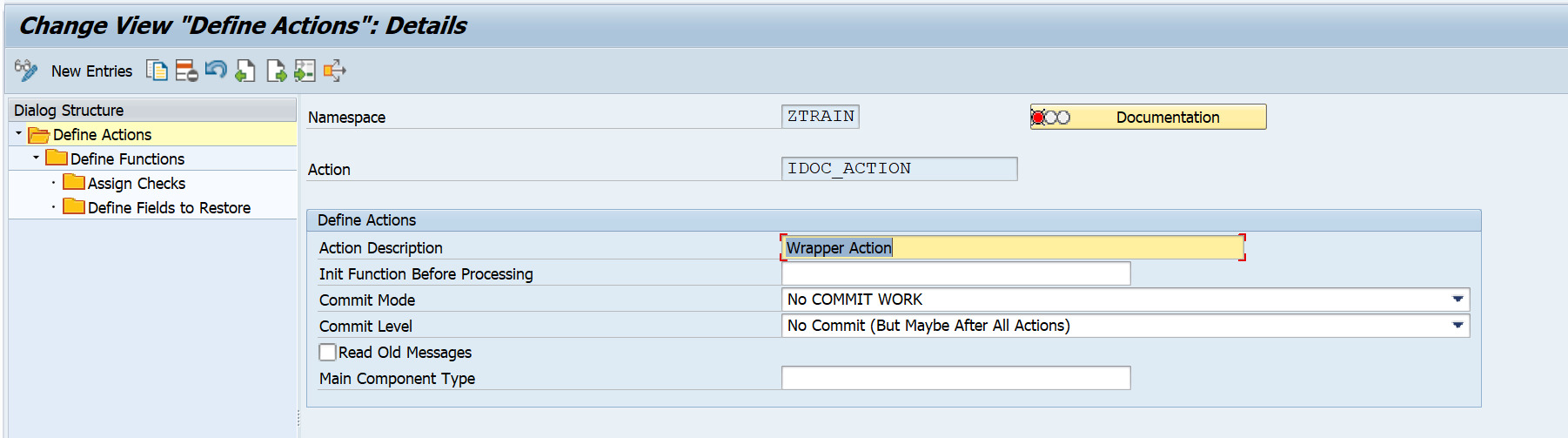
* Give the message type and basic type for both the interfaces. The message variant for the outer interface is to be given as ‘AIF’



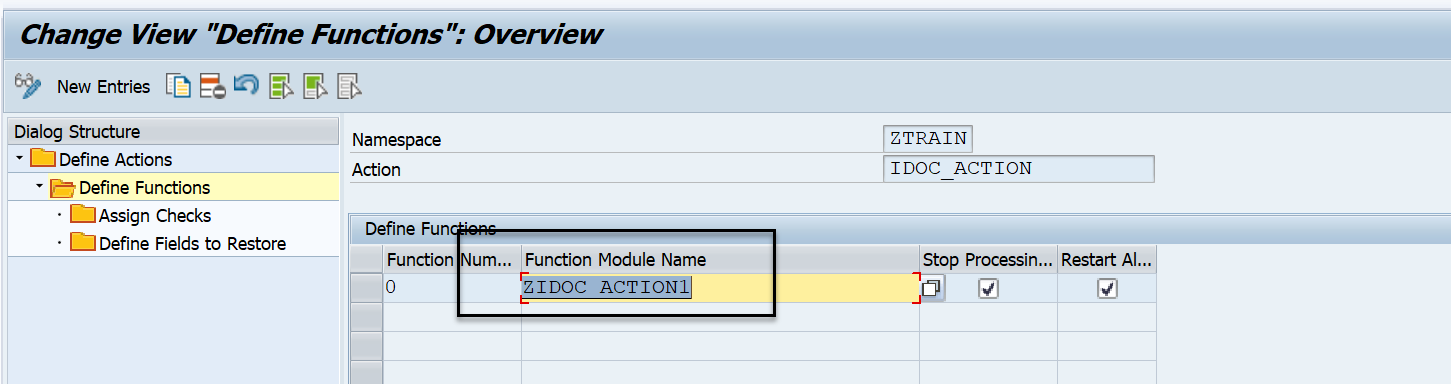
* Go back and select ‘Define Actions’.



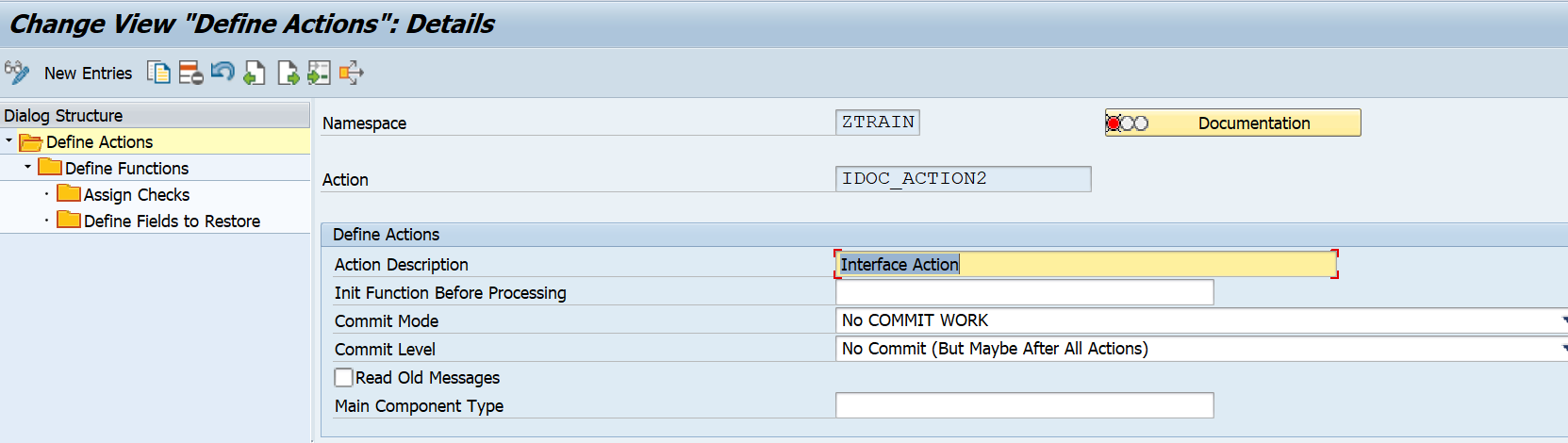
* Create new entry for Interface 1 and enter the action name

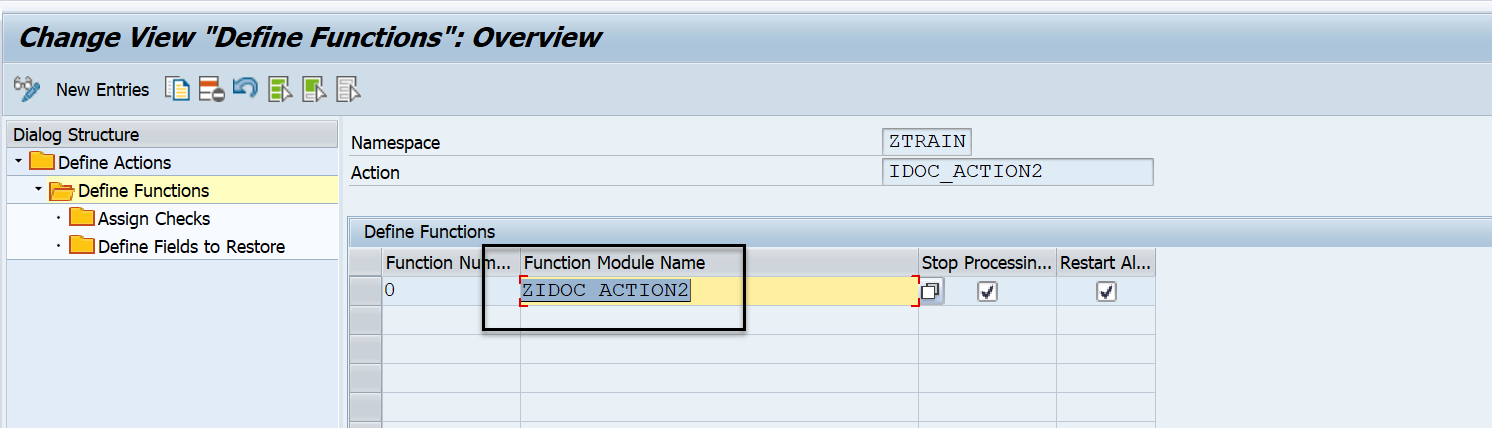


* Select ‘Define Functions’ and assign the action function module

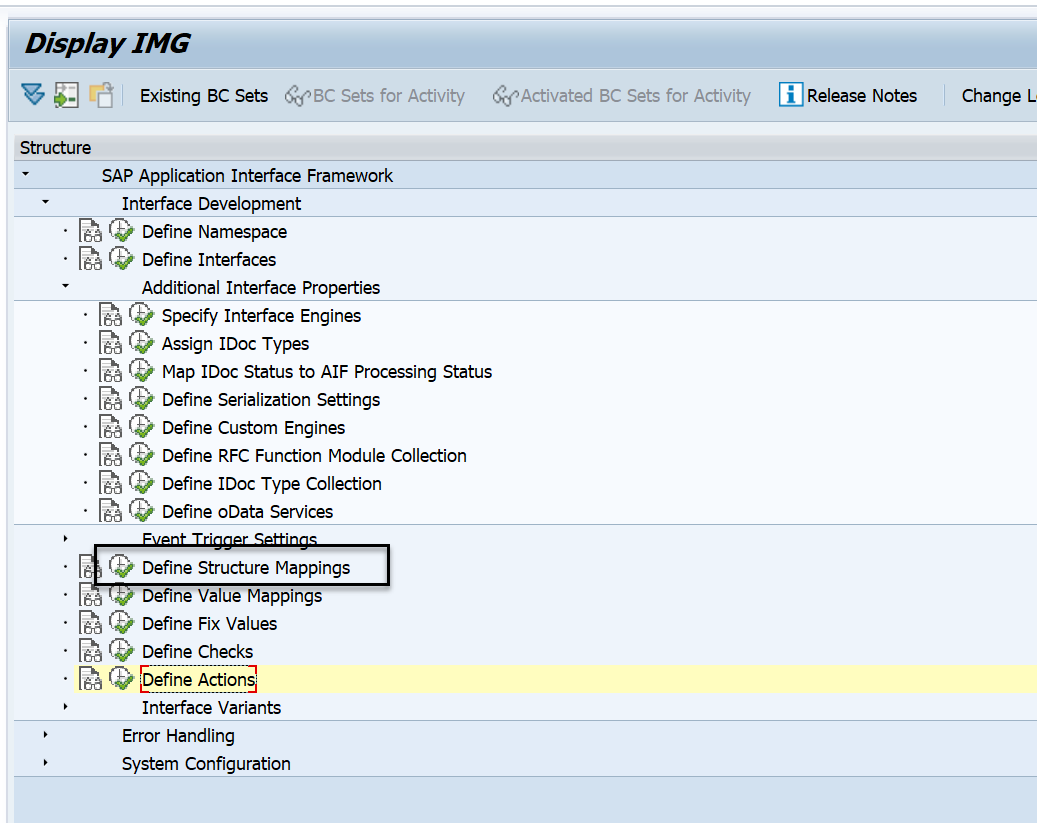


* This FM has to be created with the template /AIF/FILE\_TEMPL\_PROCESS
* Create a similar entry for Interface 2

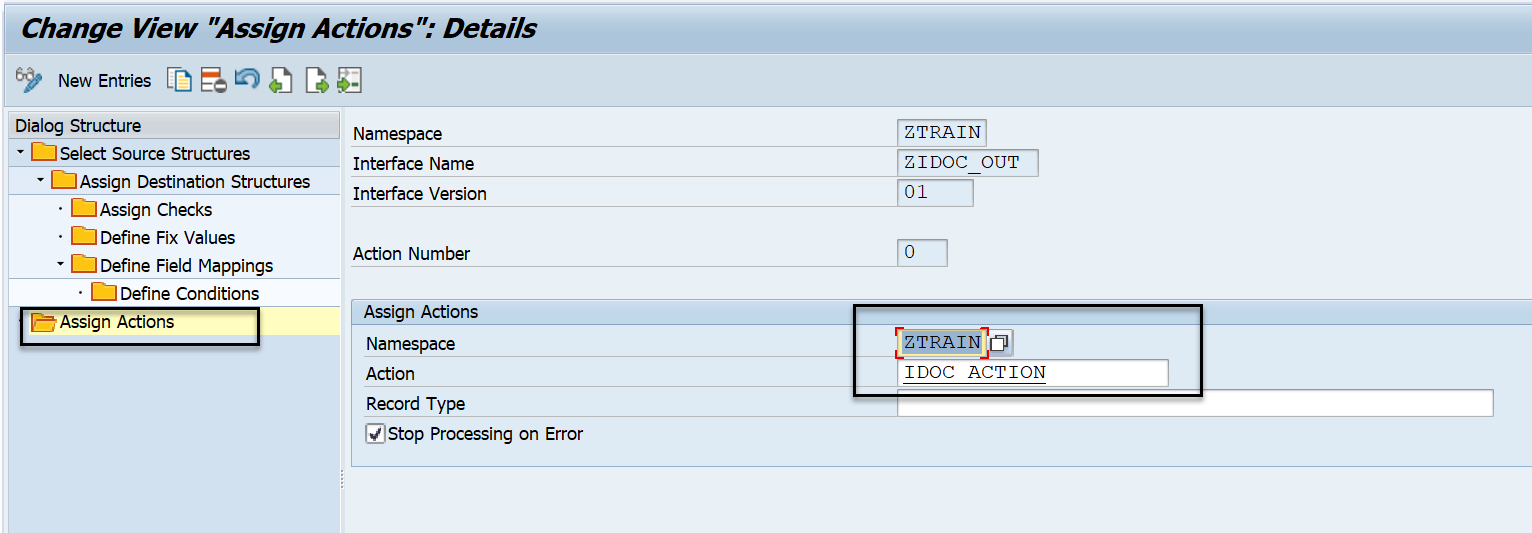




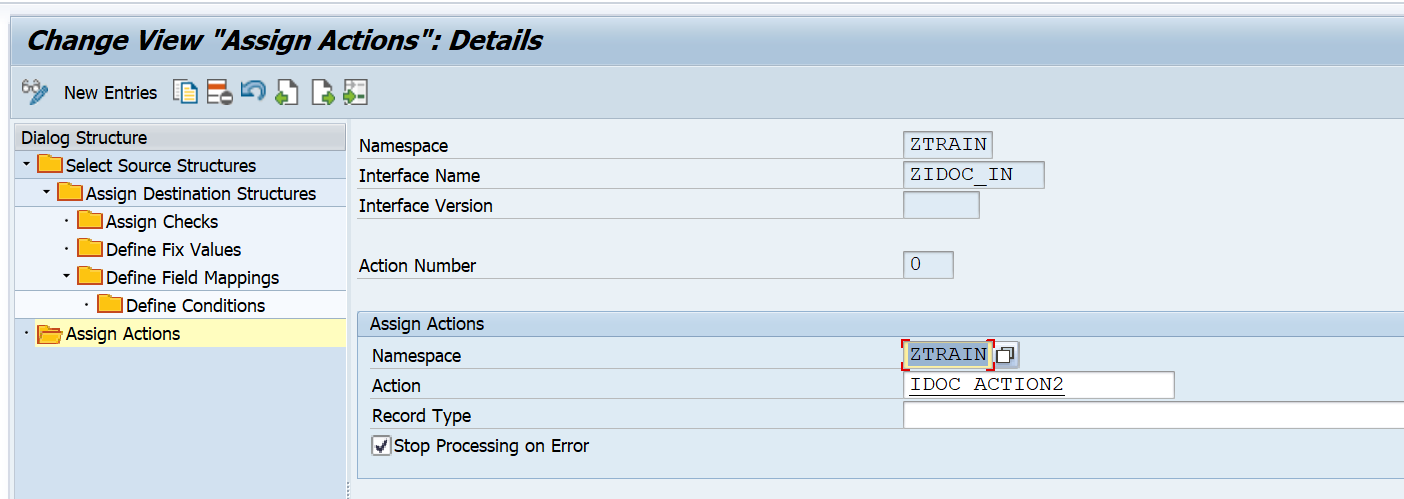
* Go back and select ‘Define Structure Mappings’



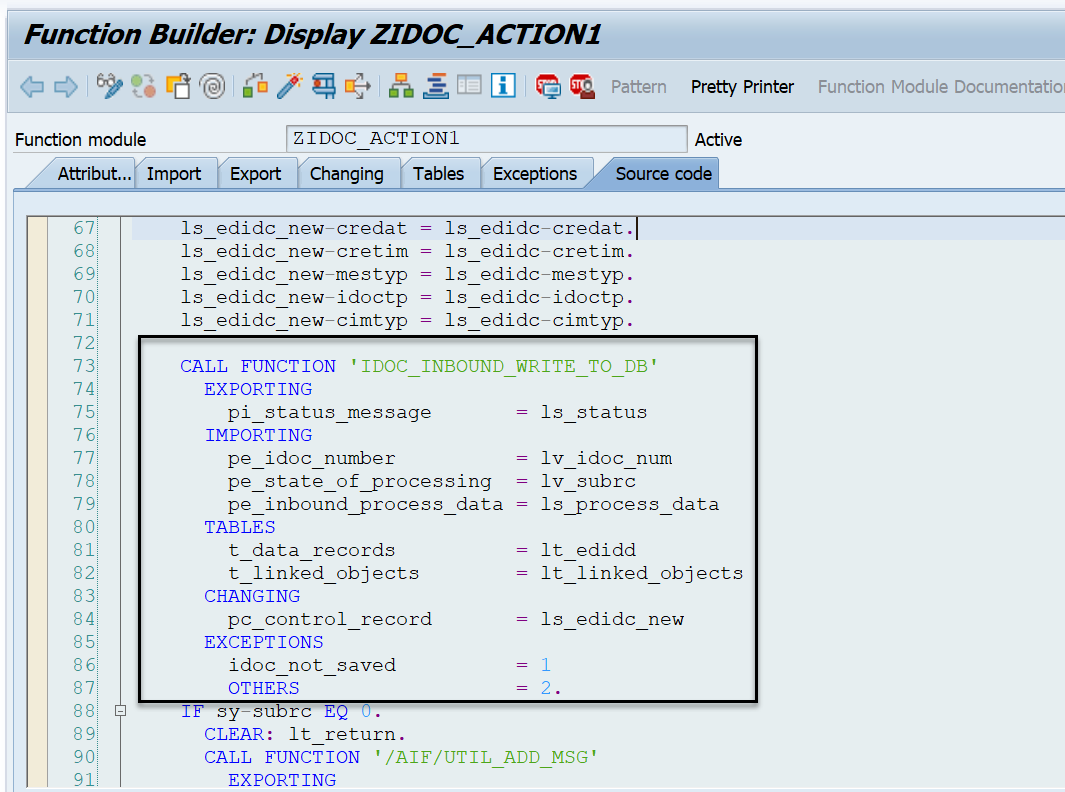
* Select the first interface and assign the action created



* Repeat the same for the second interface



* Now, we edit the action FM of the first interface to call the second interface from it.



* Once the second interface is triggered, it will reach its own Action FM and execute the desired logic. In this case, we have written the logic to process the data using standard FM '/AIF/IDOC\_ACTION\_FUNCTION'

