SAP Application Interface Framework (AIF) Capabilities in IDOC



Contents

[Scope 3](#_Toc46761568)

[Business Scenario 3](#_Toc46761569)

[Implementation 3](#_Toc46761570)

[1. Structure Generation 3](#_Toc46761571)

[2. AIF Customizing 6](#_Toc46761572)

[3. Namespace Definition 7](#_Toc46761573)

[4. Interface Definition 8](#_Toc46761574)

[5. Interface Engines 9](#_Toc46761575)

[8. Assign Actions 15](#_Toc46761576)

[9. Define Value Mapping 16](#_Toc46761577)

[10. Define Fix Values 17](#_Toc46761578)

[11. Define Actions 18](#_Toc46761579)

[12. Define Functions 18](#_Toc46761580)

[13. Changeable Fields for Forward Navigation 19](#_Toc46761581)

[14. Partner Profile Configuration 21](#_Toc46761582)

[Error Handling 23](#_Toc46761583)

[Conclusion 24](#_Toc46761584)

# Scope

The objective of this how to guide is to provide an insight on Application Interface Framework (AIF) and its capabilities. This document describes how to develop Outbound IDOC interfaces with the SAP Application Interface Framework and can be used as a detailed guide for implementation.



# Business Scenario

This document covers a business scenario where SAP Application Interface Framework triggers an outbound IDOC and performs various business transformation logics.

# Implementation

1. Structure Generation

For structure generation we use transaction /N/AIF/IDOC\_GEN. The structure will be used as SAP and RAW Data structure during Interface creation in further steps. We need to provide Basic Type, Prefix Structure and Message Type along with package and transport details.

Basic Type – IDOC Basic Type

Message Type – IDOC Message Type

Prefix Structure – The prefix with which we want the structure name to start with.

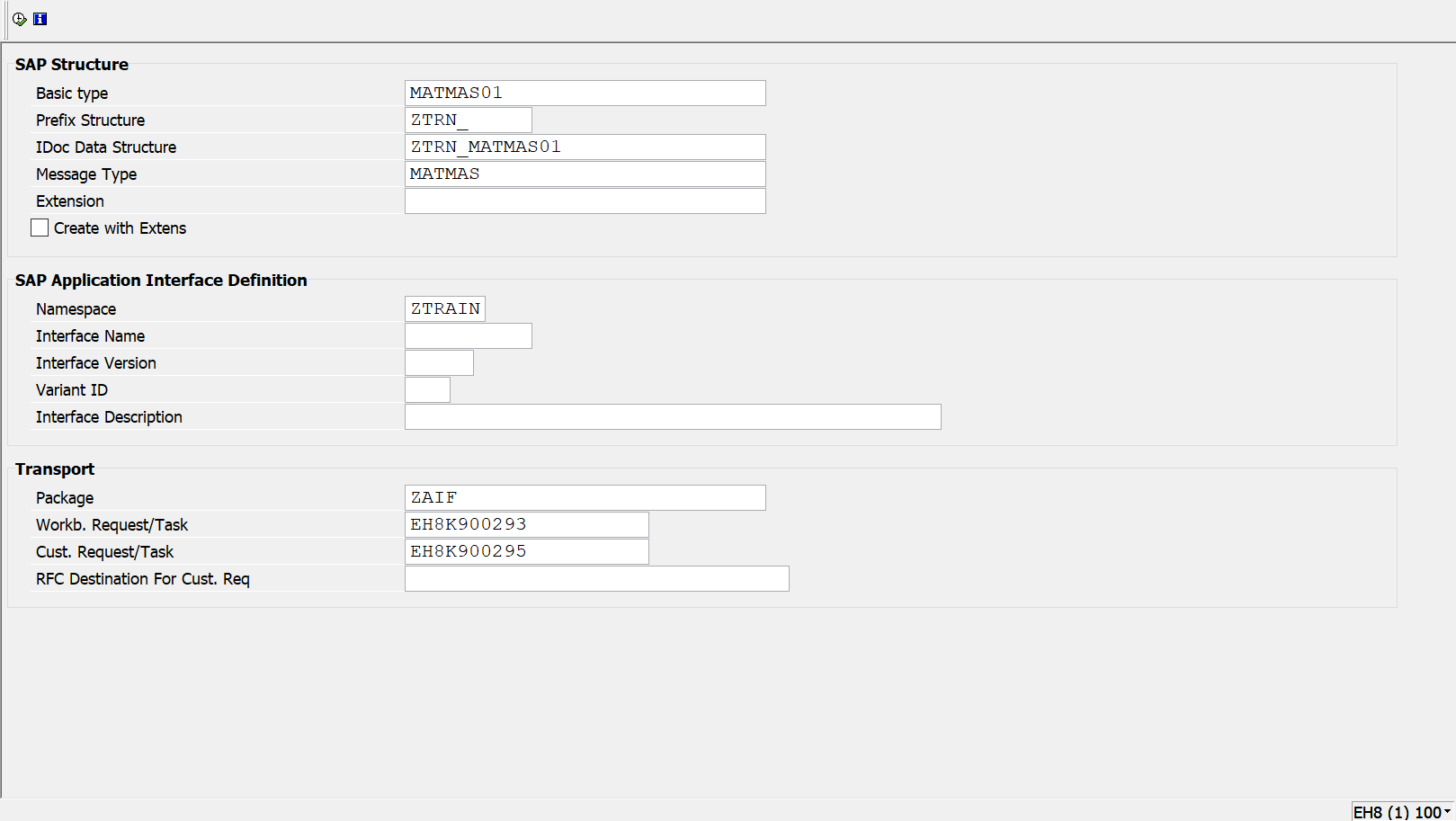


Figure 2: /N/AIF/IDOC\_GEN

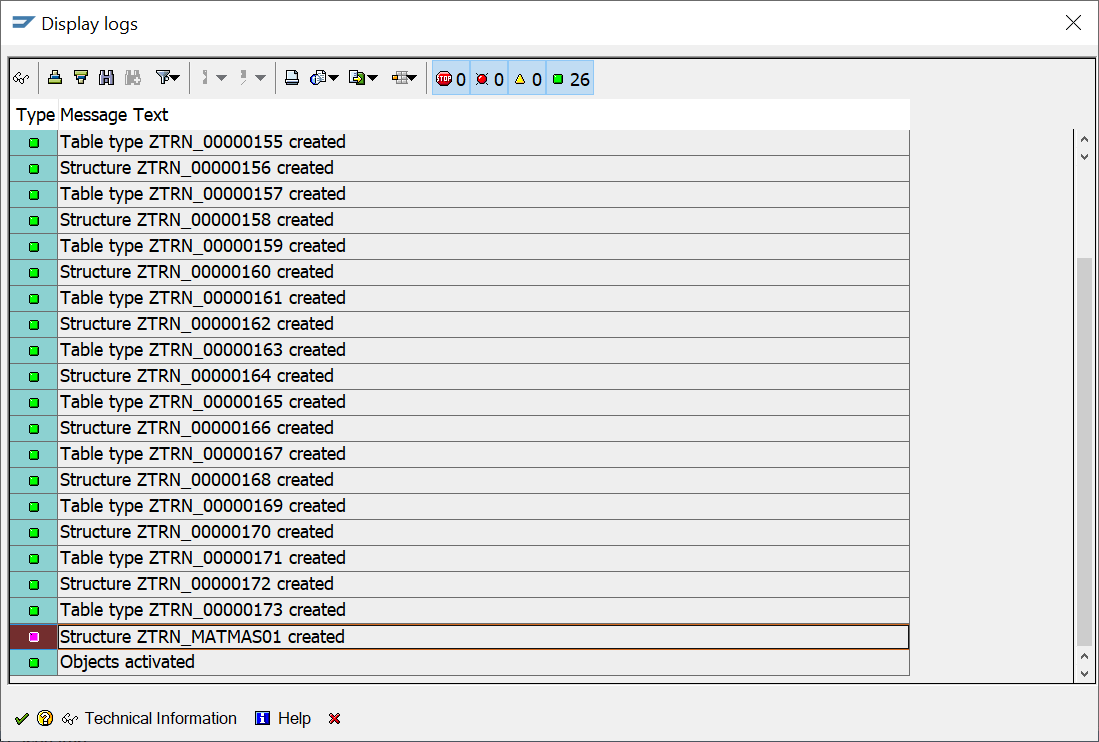
Structure Generated –

Figure 3: Structure Generated

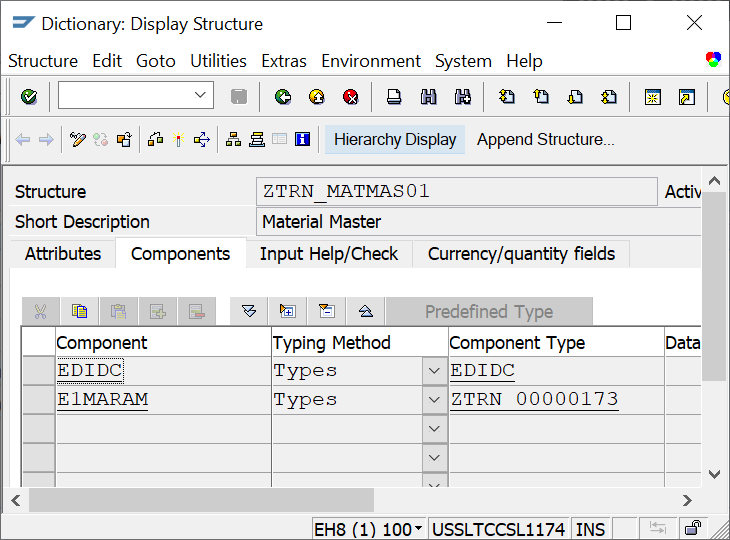


Figure 4: Structure Generated

If a structure is already present for a set of Basic Type and Message type, the transaction will give the following error and will stop the generation of duplicate structure for same basic type and message type.

1. AIF Customizing Now, we start customizing our AIF interface using the above Raw Data Structure and perform all the business logics. Go to the transaction “/N/AIF/CUST”.

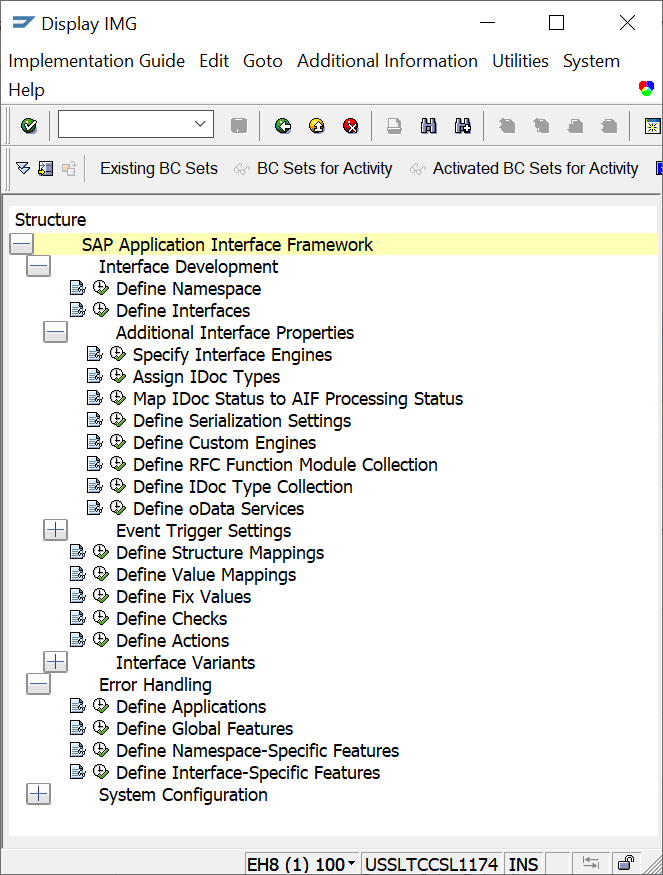


Figure 5: Transaction /n/AIF/CUST

1. Namespace Definition Execute the option “Define Namespace”. Select “New Entries” and define a New Namespace as per the business requirements.

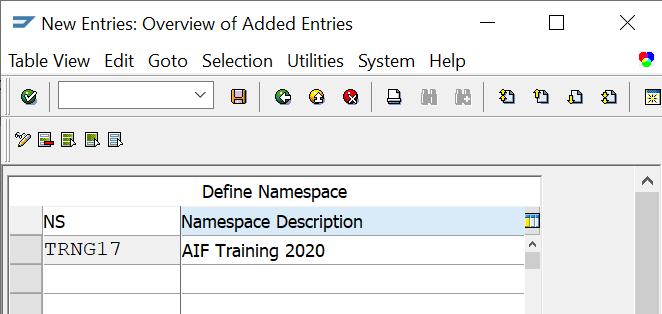


Figure 6: Namespace Definition

1. Interface Definition Come back one step and execute the option “Define Interfaces” next. And input the name of above created namespace.

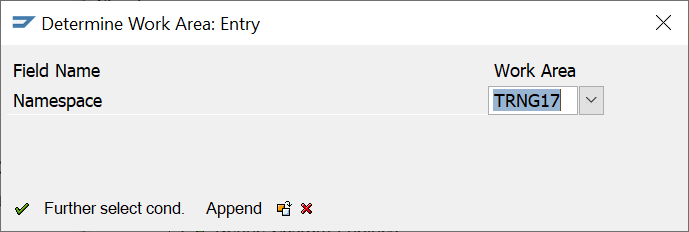


Figure 7: Interface Definition

Select “New Entries” at the top of the window.



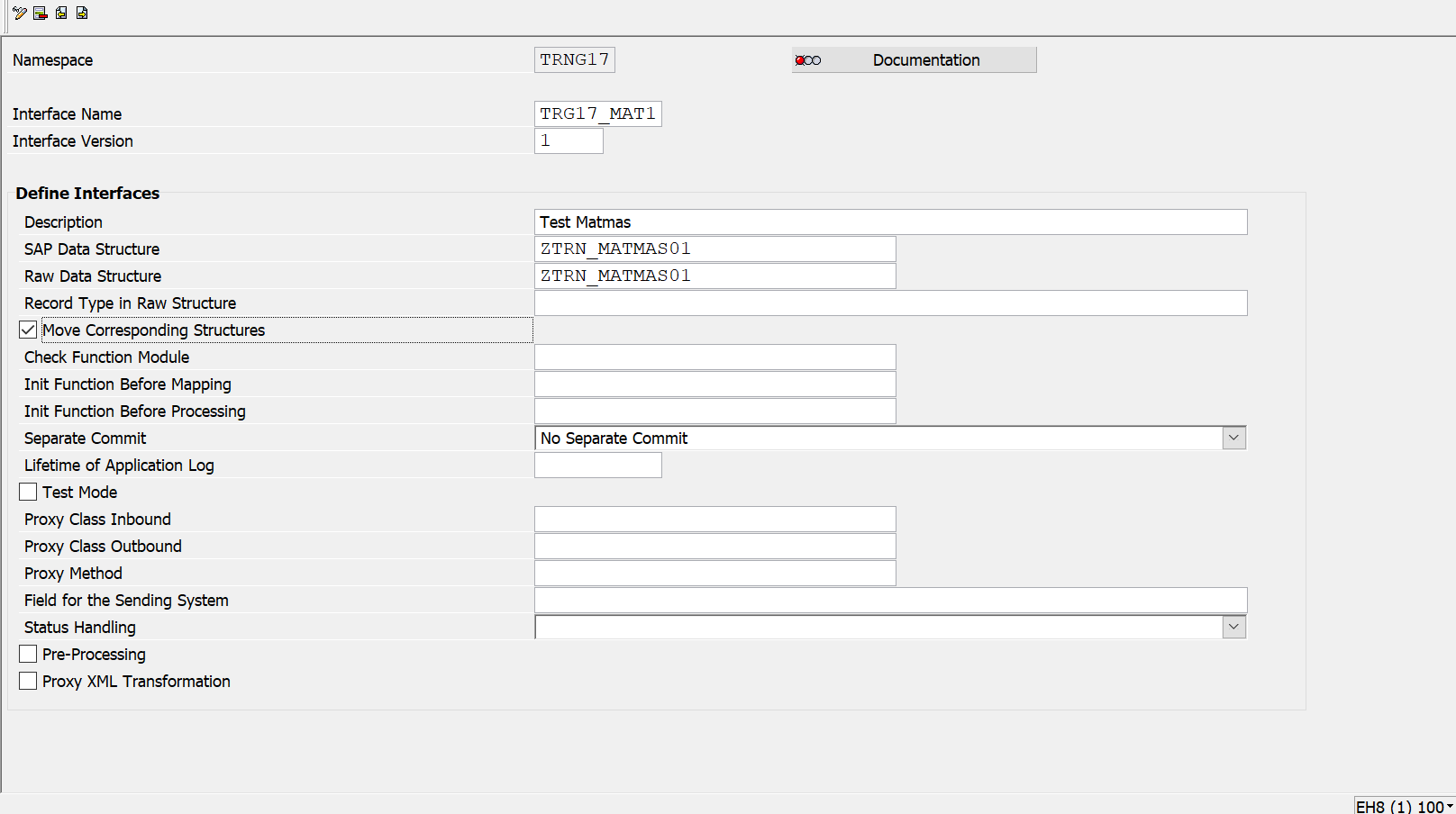
Provide the below details.

Figure 8: Interface Definition

The name of SAP and RAW Data Structure is obtained from the STEP 1.

1. Interface Engines Come back to customizing screen again and execute the option “Specify Interface Engines” under “Additional Interface properties”.

Provide the namespace that we have used and specify the Engines.

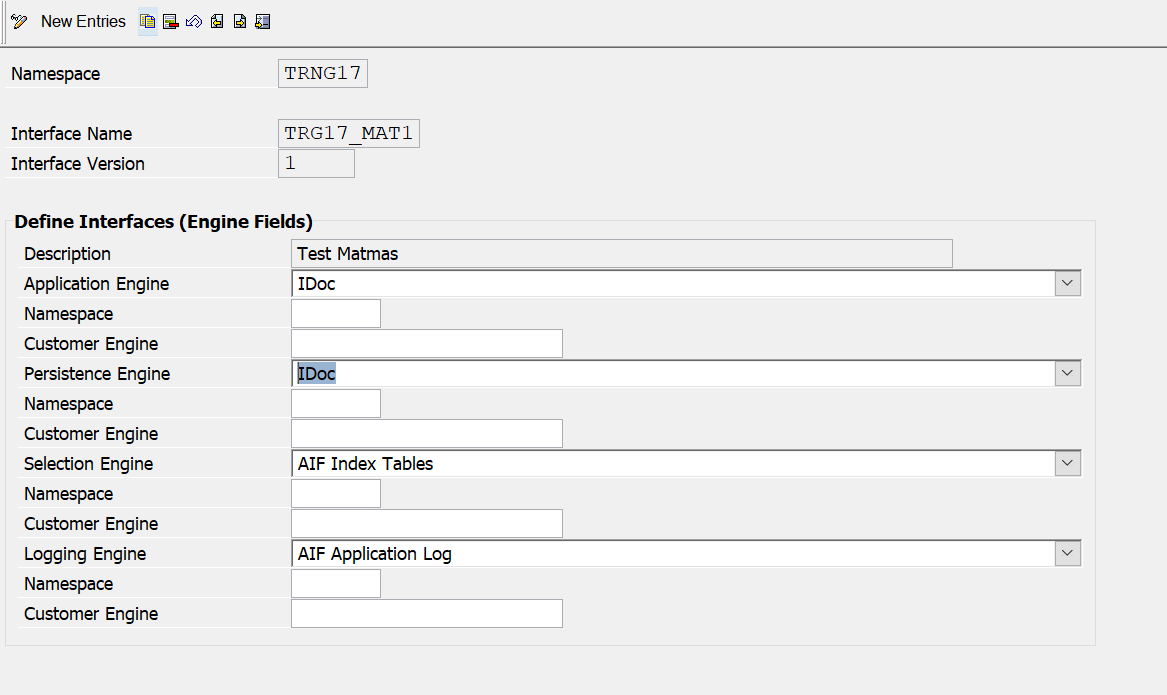


Figure 9: Interface Engines

Depending on the interface technology you have to define the engines that should be used in *Monitoring and Error Handling* in order to handle data messages.

There are four different engine types:

***Application Engine*:** Defines how messages should be handled in *Monitoring and Error Handling*. The application engine is, for example responsible for restarting and canceling messages.

***Persistence Engine*:** Different interface technologies use different persistence layers to store the data received or sent. The persistence engine is responsible for selecting, updating and locking the data content of a message.

***Selection Engine*:** The selection engine defines which messages should be displayed in *Monitoring and Error Handling*. The selection engine will select the messages based on the selection criteria of the selection screen.

***Logging Engine*:** The logging engine is responsible for handling the log messages written during message processing.

1. **Assign IDOC Types** Come back to customizing screen again and execute the option “Define Interfaces (IDOC fields)” under “Additional Interface properties”.

Provide message Type, Basic Type, Extension for Basic Type (If required) and Message function (If required). Message function should be used if we have two different interfaces for the same set of message type and basic type. The same message function will be used in partner profile configuration to determine which interface will be called in case there are two entries for same message type and basic type.

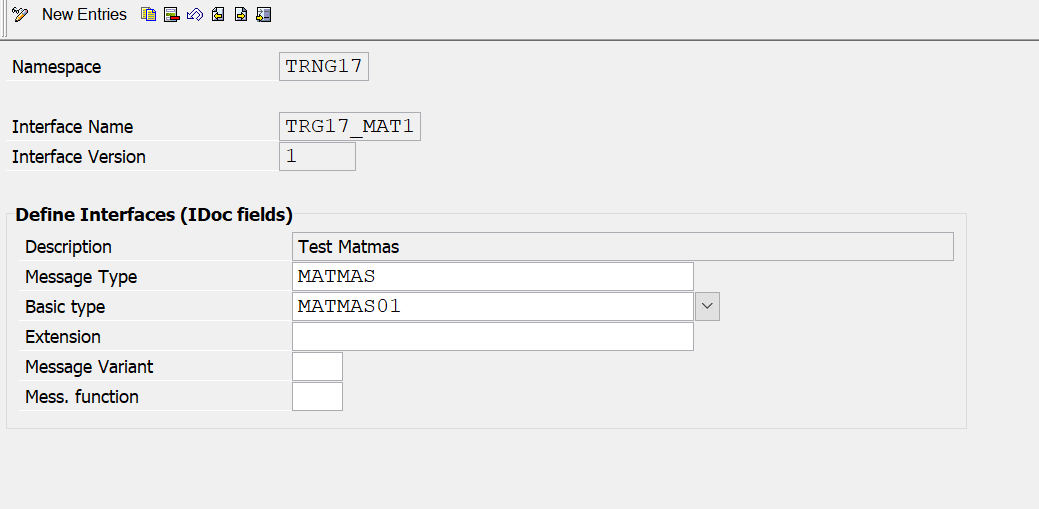


Figure 10: Assign IDOC type

1. **Define Structure Mapping** Come back to customizing screen again and execute the option “Define Structure Mapping”. Enter the Namespace, Interface Name and Version and Continue. Create an entry for structure from the source structure for which mapping is required.

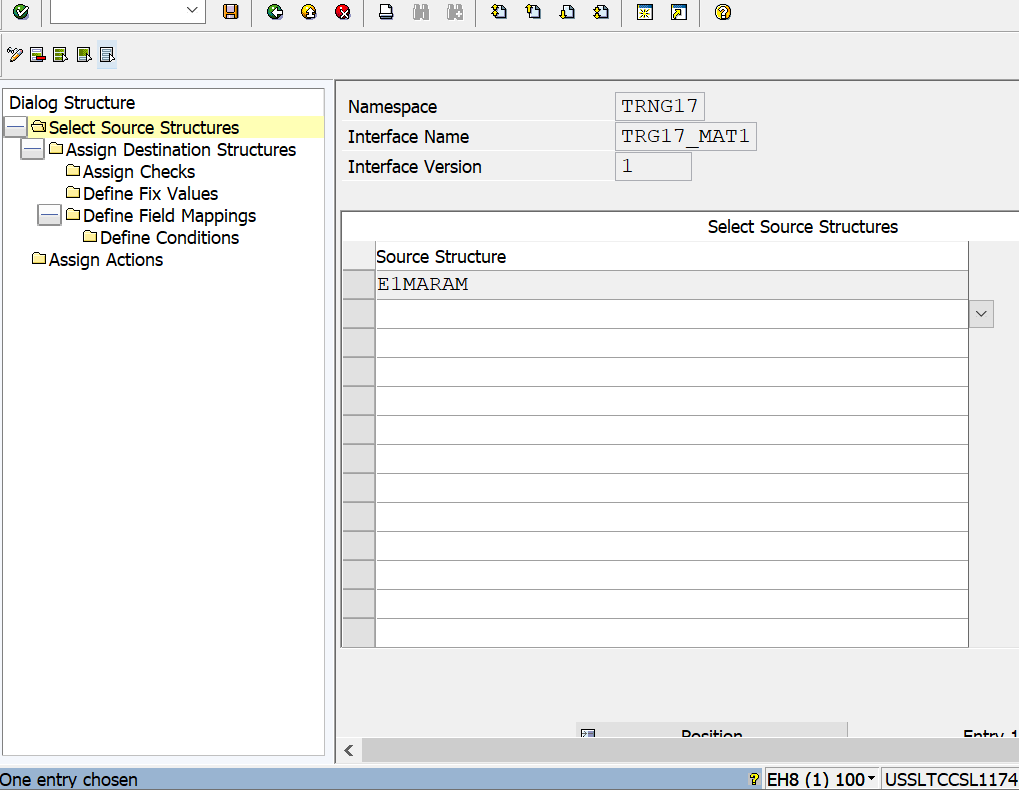


Figure 11: Structure Mapping

In Structure mapping the fields of the source structure are mapped to the fields of the destination structure.

Select the entry created for Source structure and then select “Assign Destination Structure. Enter the structure from Destination structure which we want to map the source structure entry.

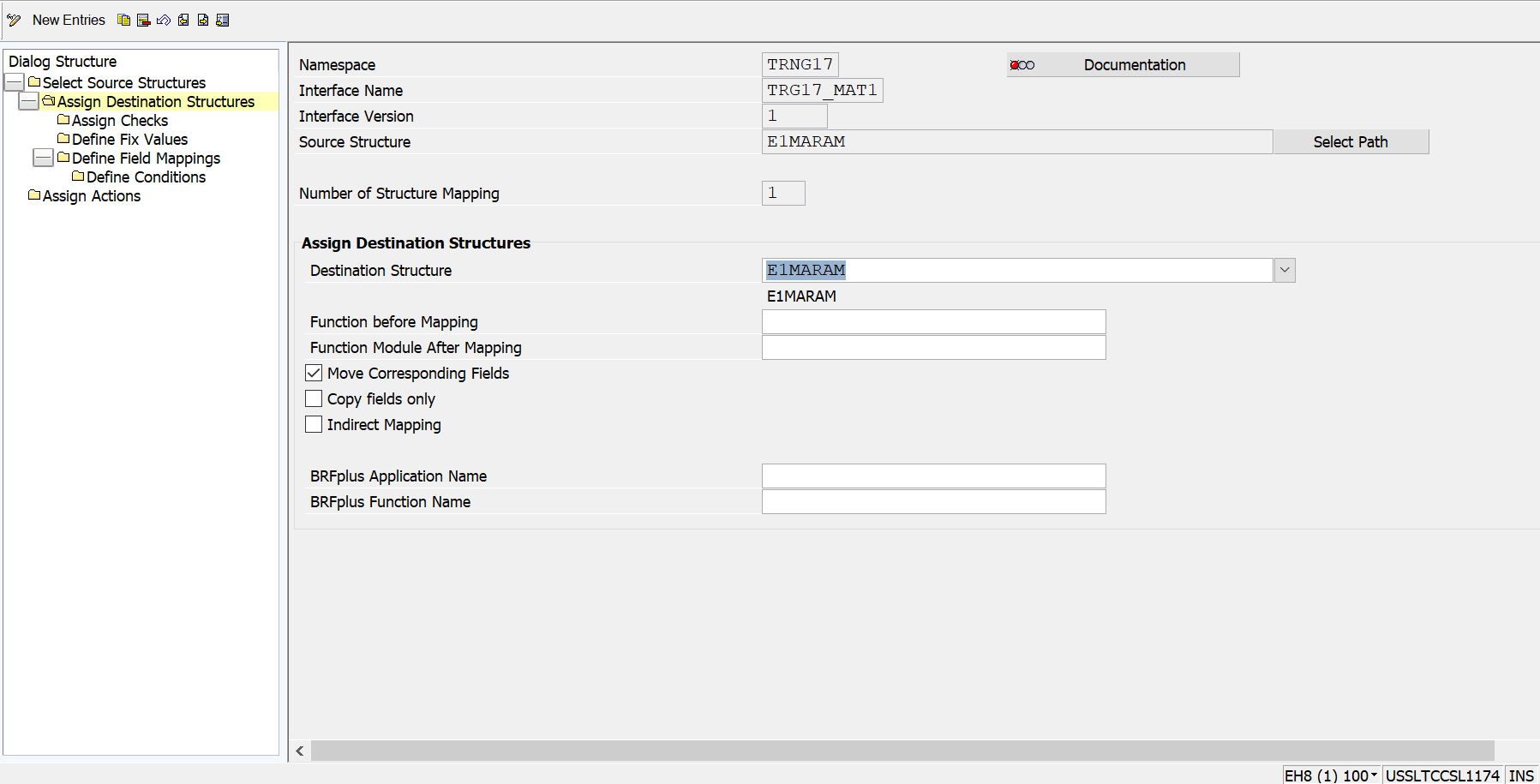


Figure 12: Destination Structure Assignment

We can also assign value mappings, fix values, checks, and actions.

**Define Field Mappings:** The fields of the source structure need to be mapped to the fields of the destination structure. This is done using field mappings.

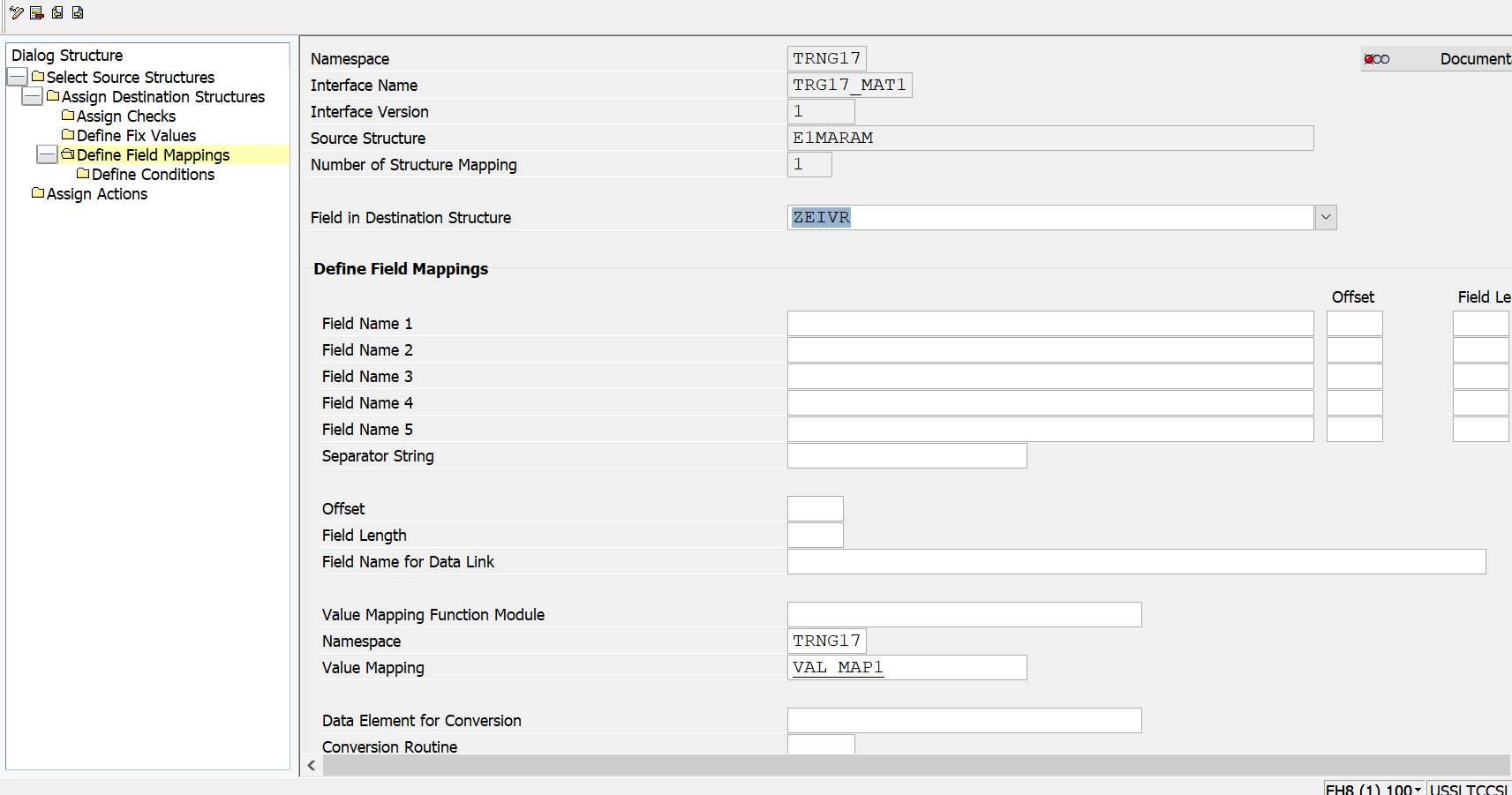


Figure 13: Define field mappings

**Assign Checks:** It is possible to validate fields from raw data or SAP data structures. Works based on “Where” condition.

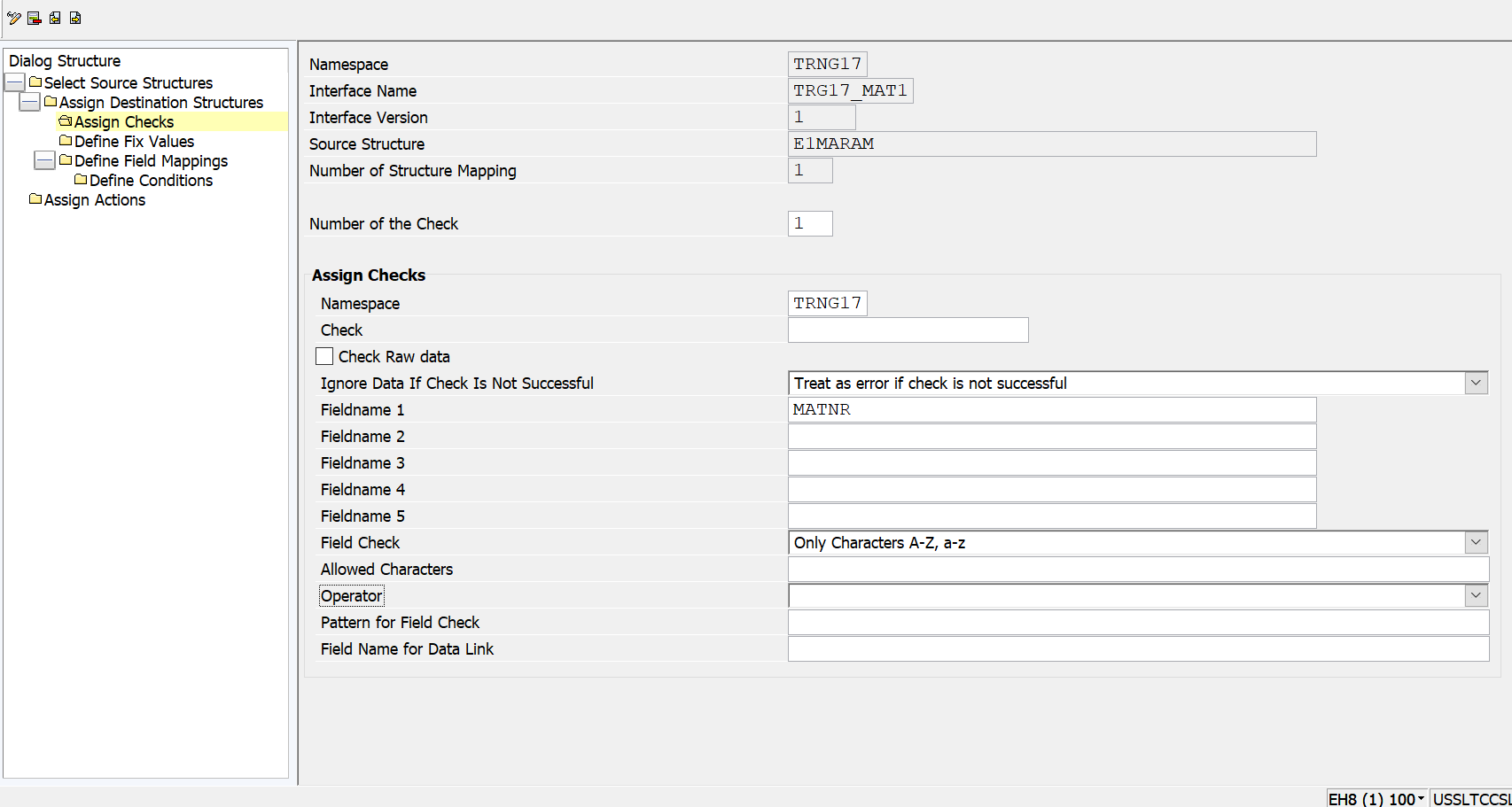


Figure 14: Define Checks

**Define Fix Values**: For a field that is supposed to have the same value every time, we can define a fix value. This can be done by creating a fix value and reusing it or by directly using some value.

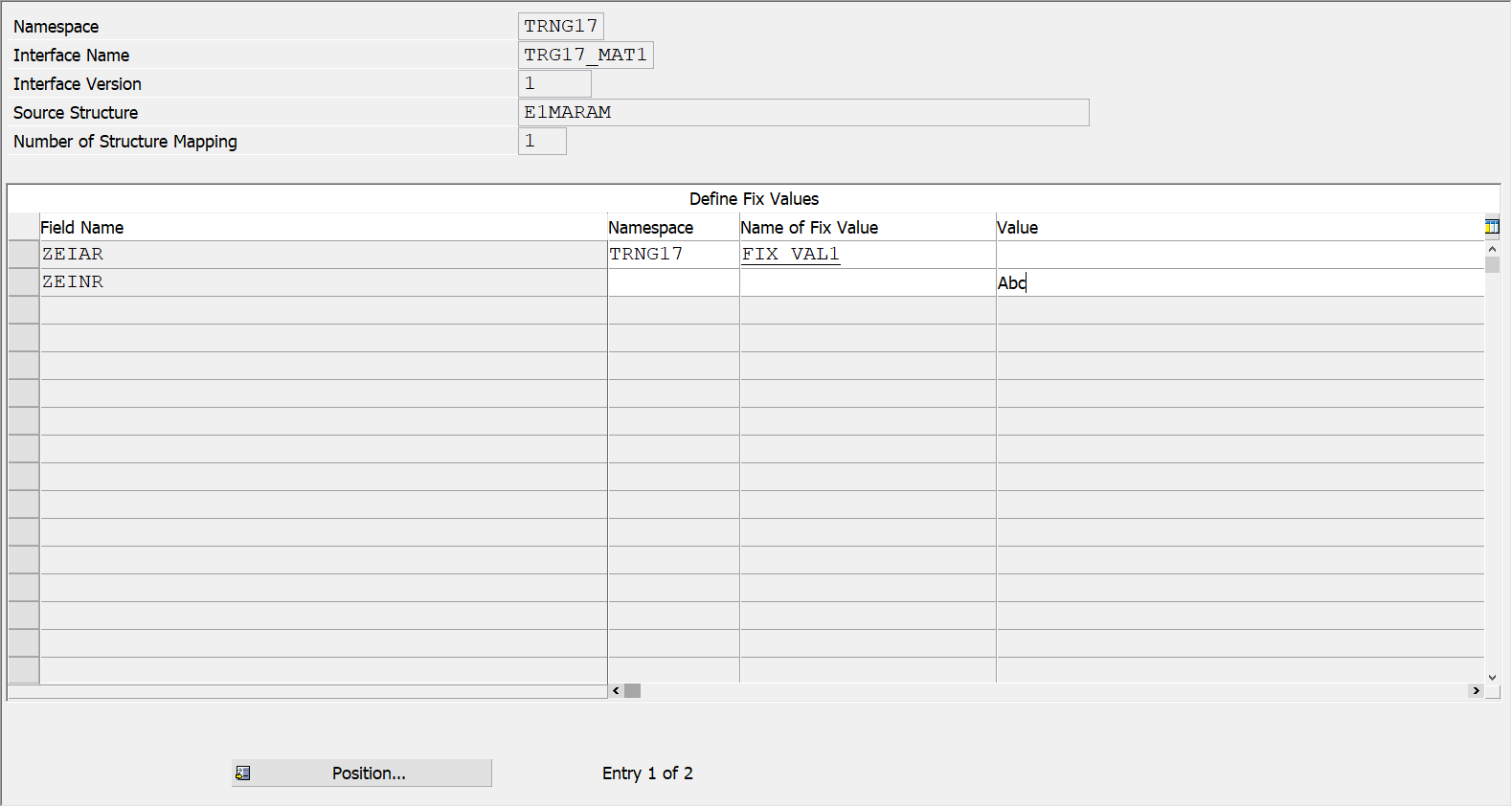


Figure 15: Define Fix values

1. Assign Actions Select Assign Actions and create a new action for the Interface

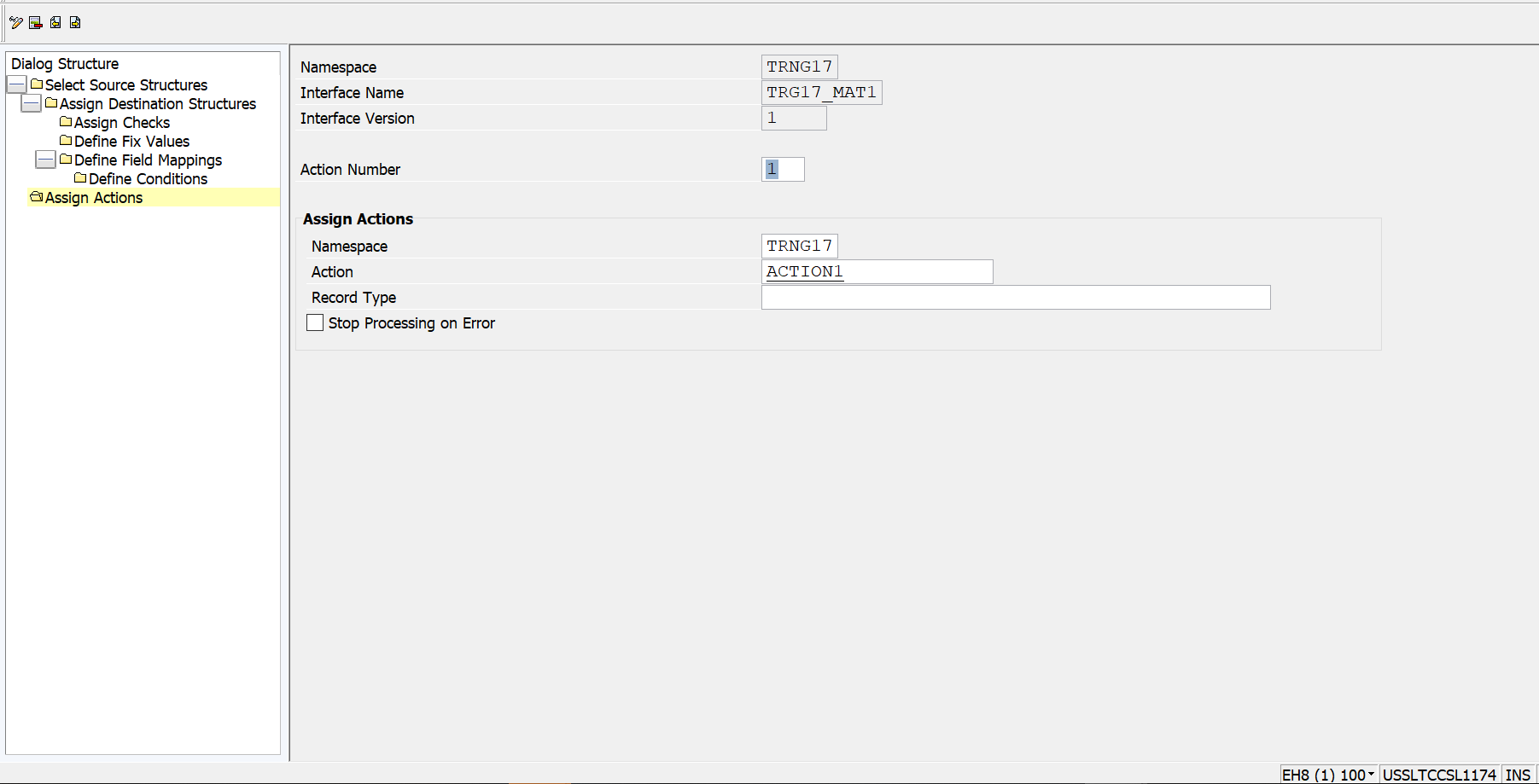


Figure 16: Assign Actions

1. Define Value Mapping Come back to /N/AIF/CUST screen and select the option “Define Value Mappings”. Provide the namespace and enter. Provide the value mapping name and field details. Data Element for EXT1- Data Element for EXT5 are the fields from the source structure based on which we have to find the value of the respective field from destination structure. This field in destination structure is defined in data Element for INT.

Number of External Values = No of fields used from Source structure to derive the respective value in destination structure.

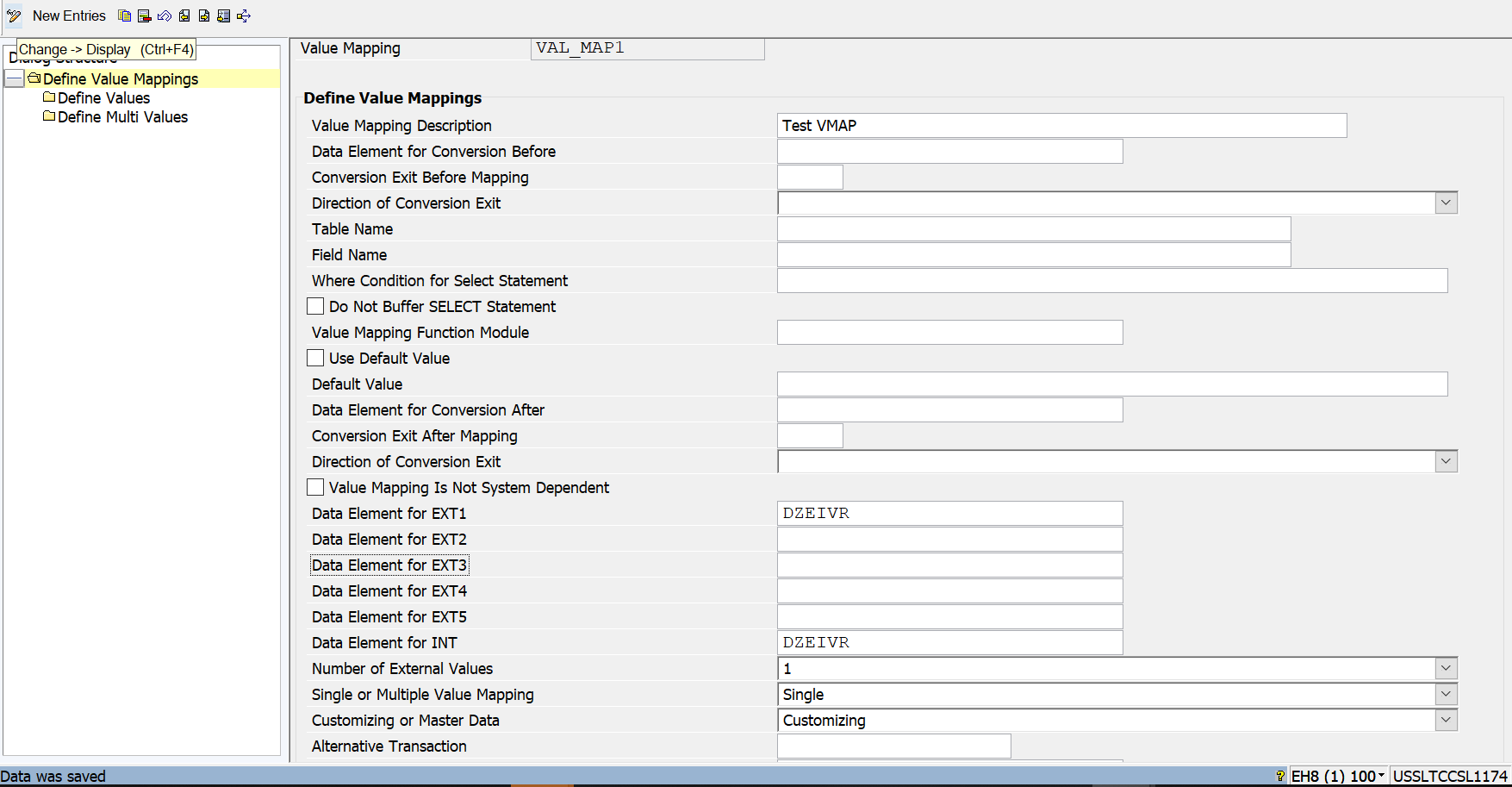


Figure 17: Define Value Mapping

1. Define Fix Values Come back to the AIF Customizing screen and execute the option “Define Fix Values”. Provide the namespace and enter. Provide the Name for fix value, description and the actual value.

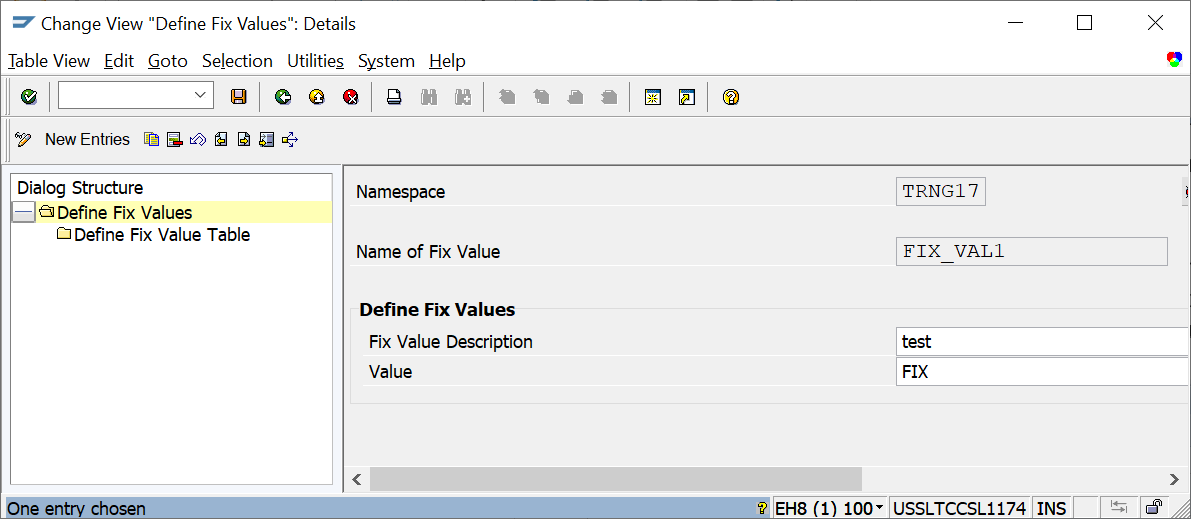


Figure 18: Define Fix Value

1. Define Actions Go back to transaction “/n/AIF/CUST”. Execute the option “Define Actions”. The action name you created in Step 8 will appear. Provide the Description and Commit Details. Here Init Function Module before processing can also be provided. It is used for processing the logic before the Action Function Module is called.

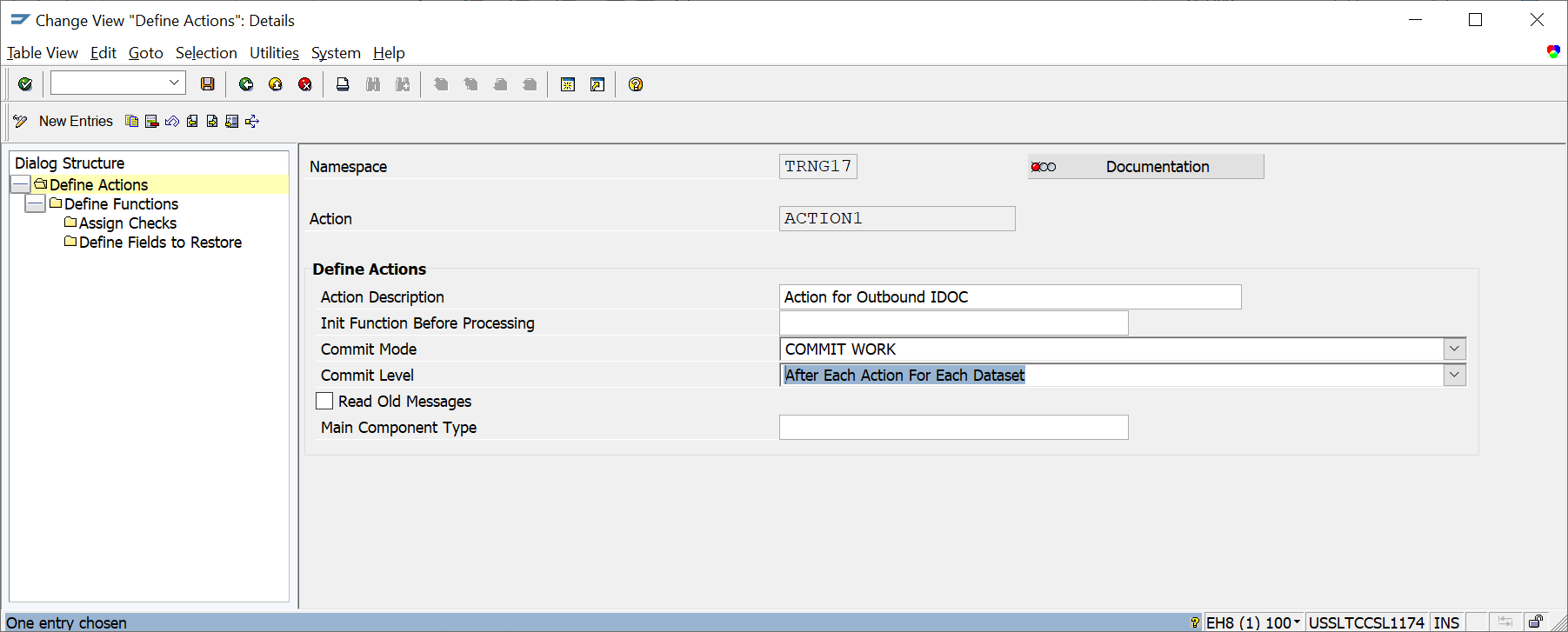


Figure 19: Define Actions

1. Define Functions Select the option “Define Functions” (highlighted above) and provide the name.

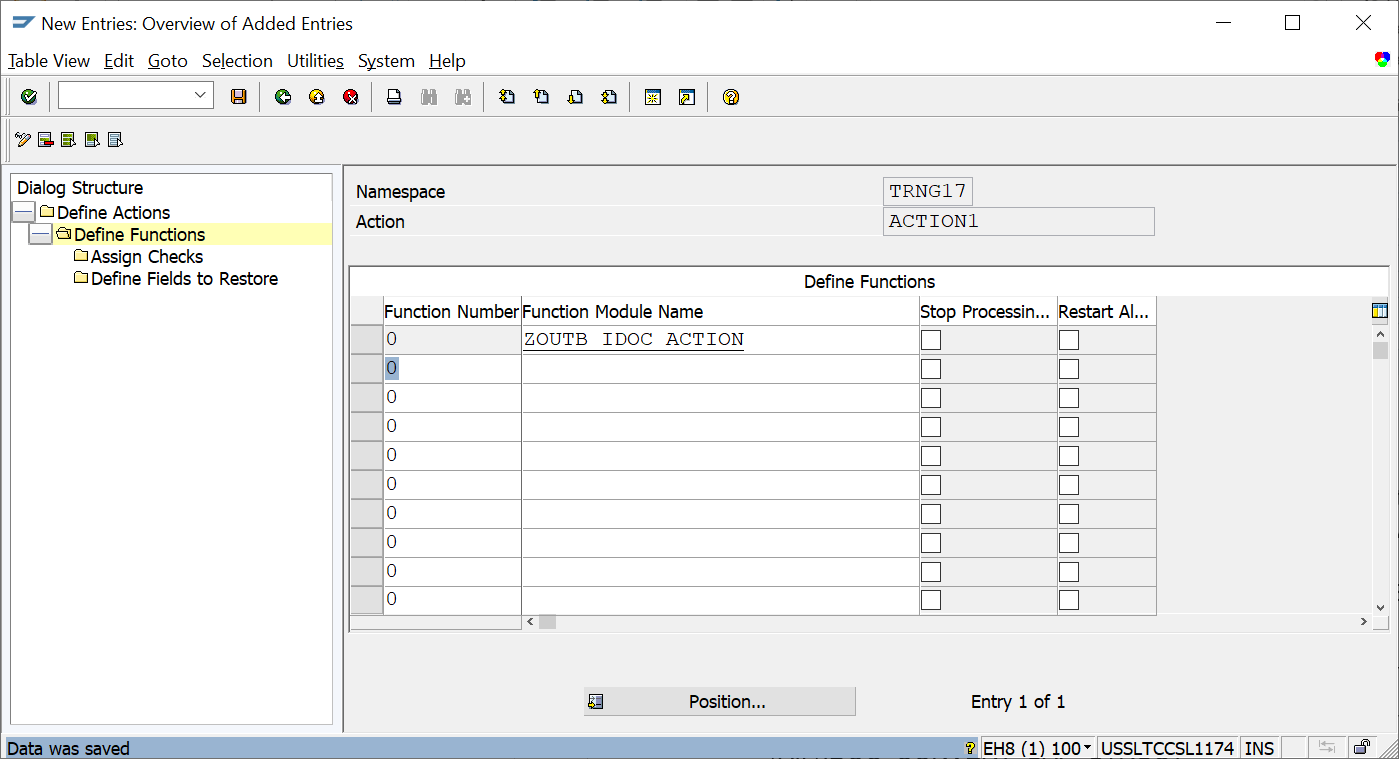


Figure 20: Define Functions

Double click on the function module name and it will prompt to create a new FM with that name. Select “Yes” and a new FM will be generated in AIF required template.

AIF Template FM for Action - /AIF/FILE\_TEMPL\_PROCESS

In outbound action FM, we need to call FM 'MASTER\_IDOC\_DISTRIBUTE' to process the second IDOC to actual receiver system.

For getting the IDOC data in structures, we can use FM - '/AIF/IDOC\_CONVERT\_SAP\_STRUCT'

1. Changeable Fields for Forward Navigation

Select the option ‘Define Interface-Specific Features’ in Error Handling for yiour respective interface, after maintaining an entry in Namespace-Specific Feature.

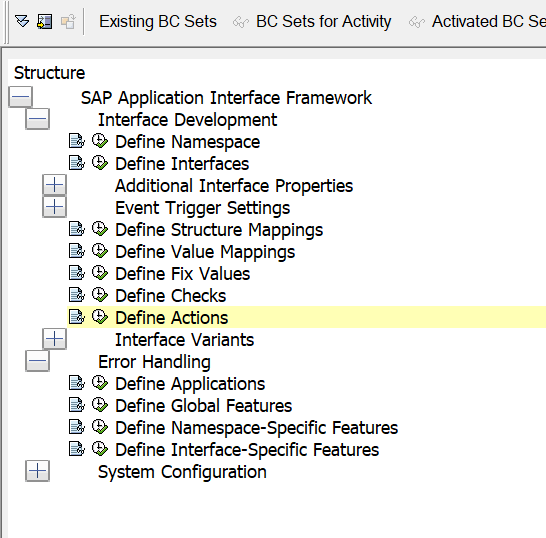


Figure 21: Interface Specific Features

Select option Define Changeable Fields for configuring forward navigation for error reprocessing,

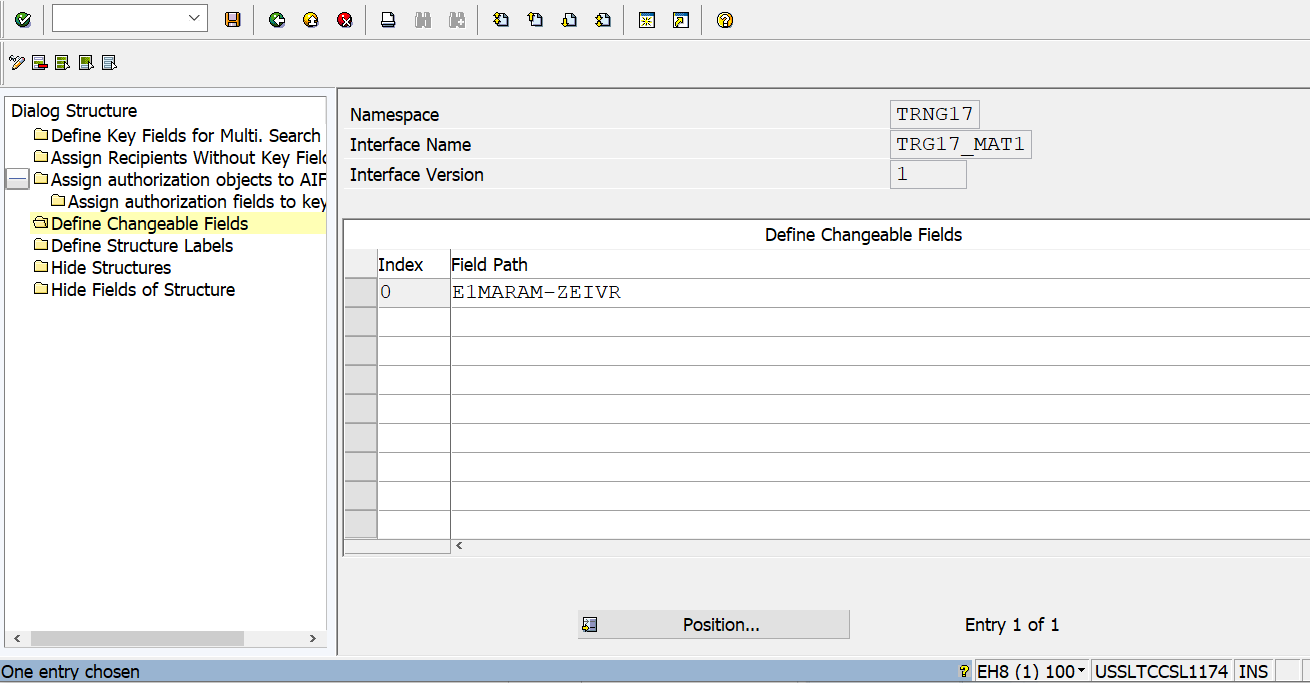


Figure 22: Changeable fields

**ECC Steps**

1. Partner Profile Configuration

Go to transaction code WE20 and in the receiver partner number, configure partner profile for both IDOCs

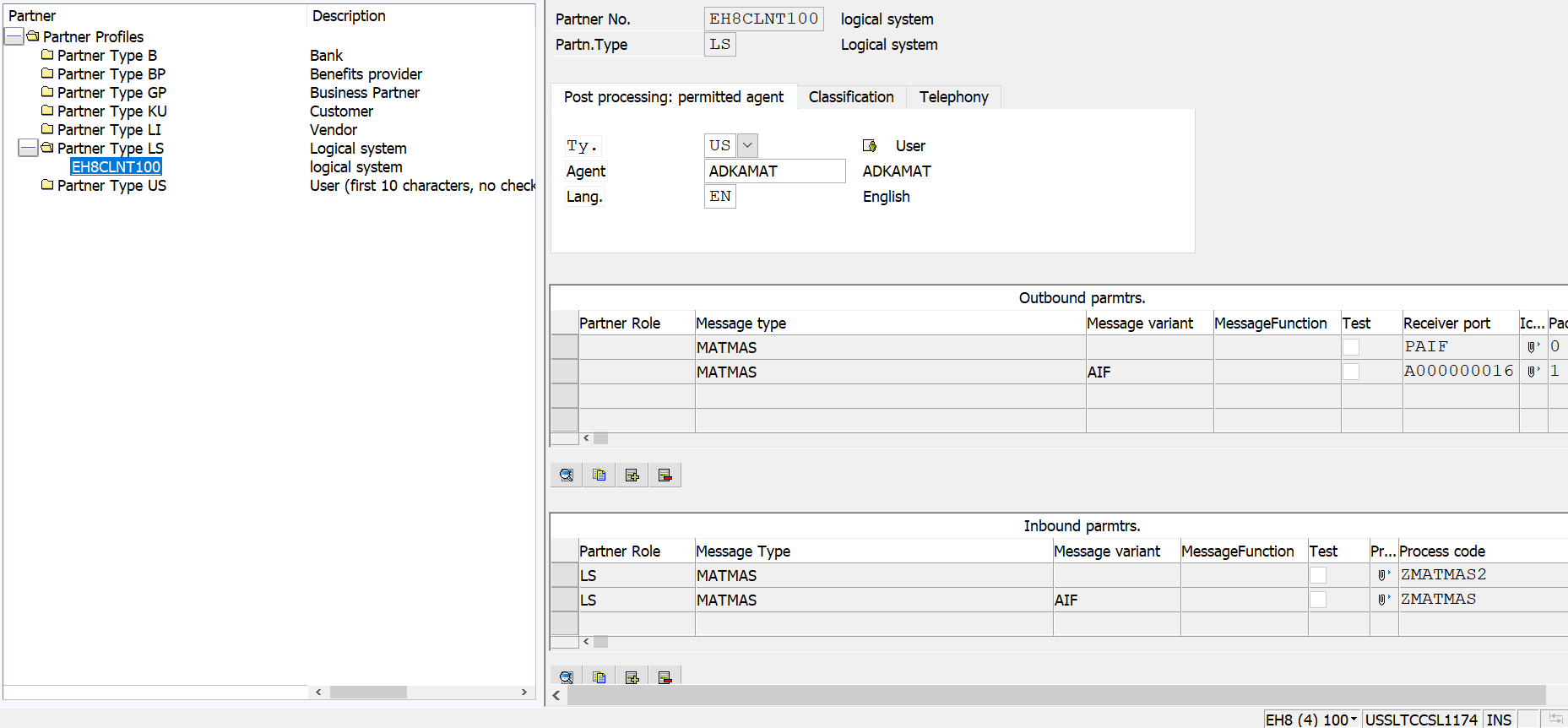
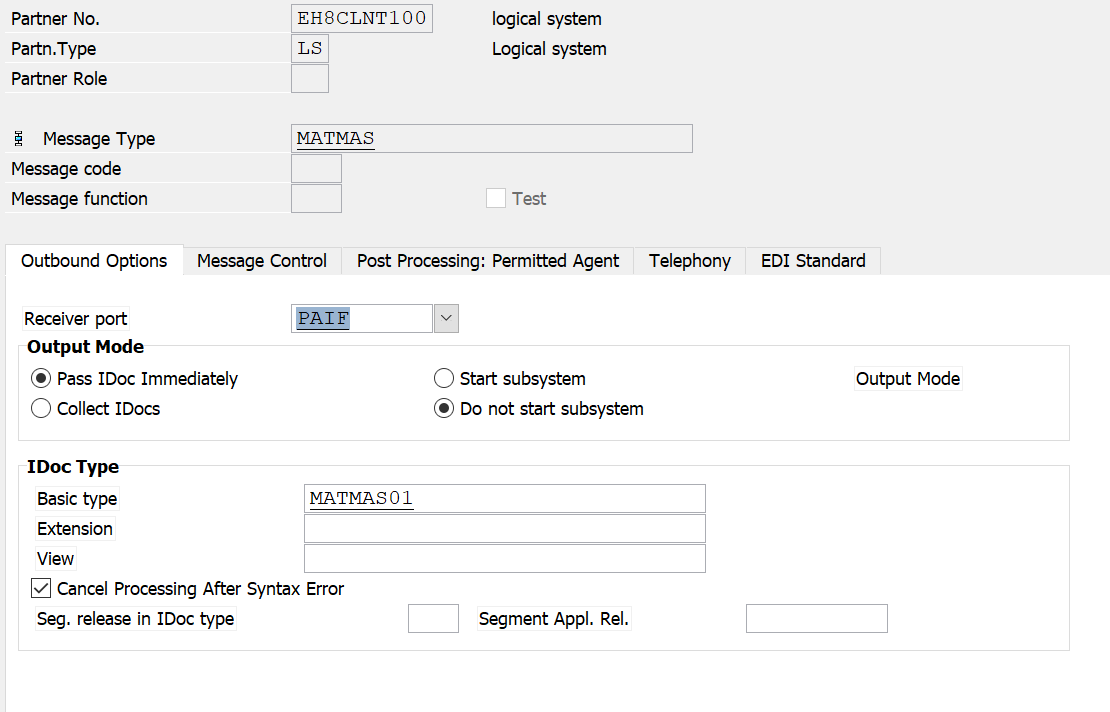
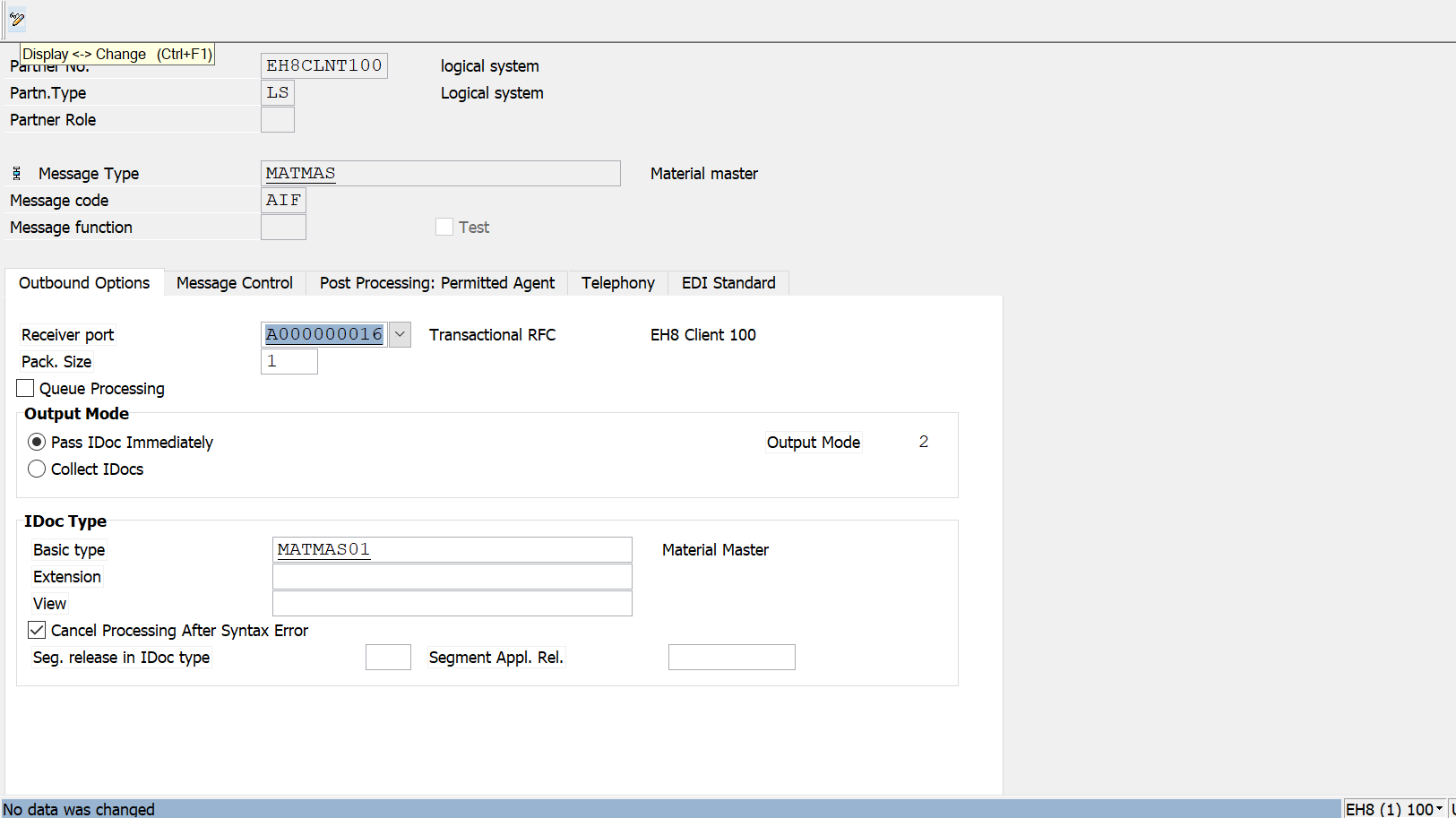


Figure 23: Partner Profile

 Figure 24: Partner Profile for 1st IDOC



**Figure 25: Partner Profile for 2nd IDOC**

# Error Handling

For error handling go to t-code /n/AIF/ERR

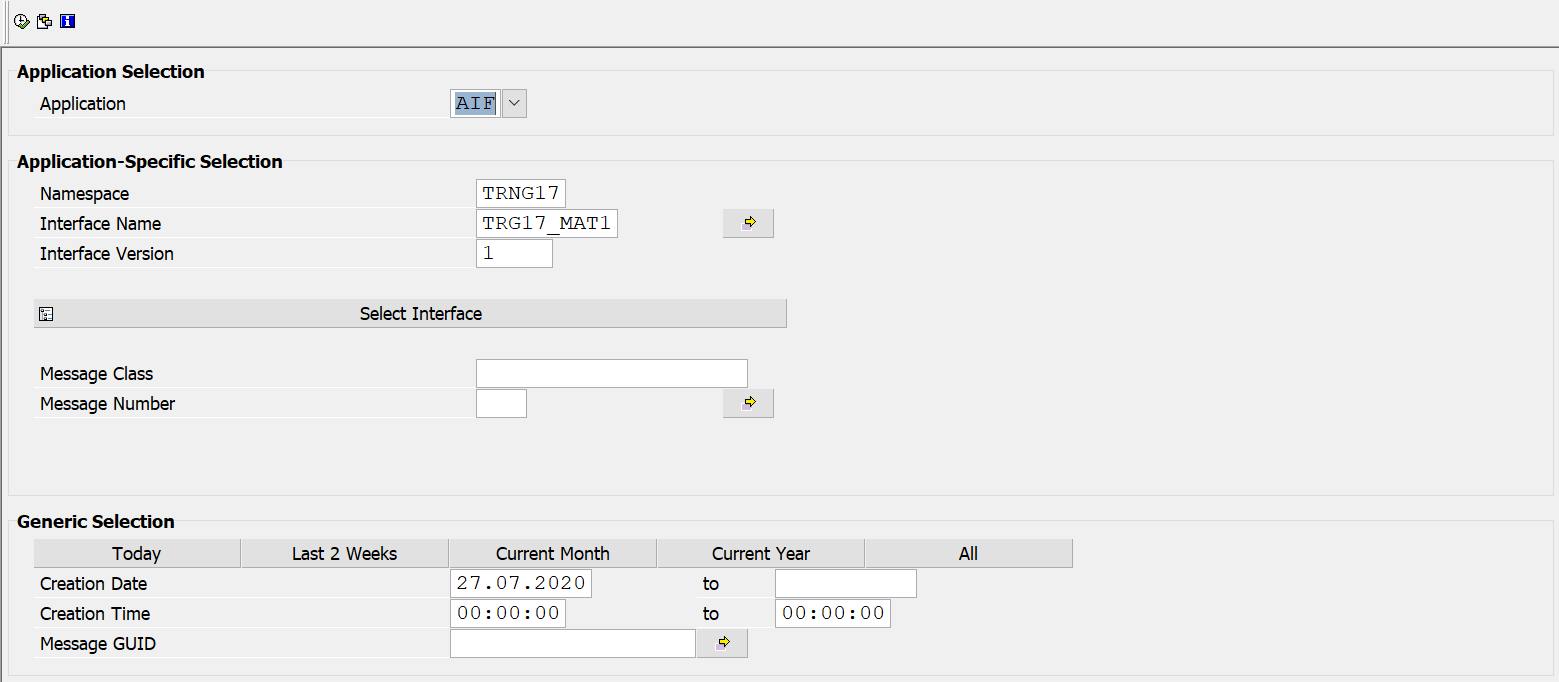


Figure 26: AIF Error Handling Screen

On the selection screen, enter your Interface, date range and message statuses and click execute.

It will display all the messages based on the selection criteria.

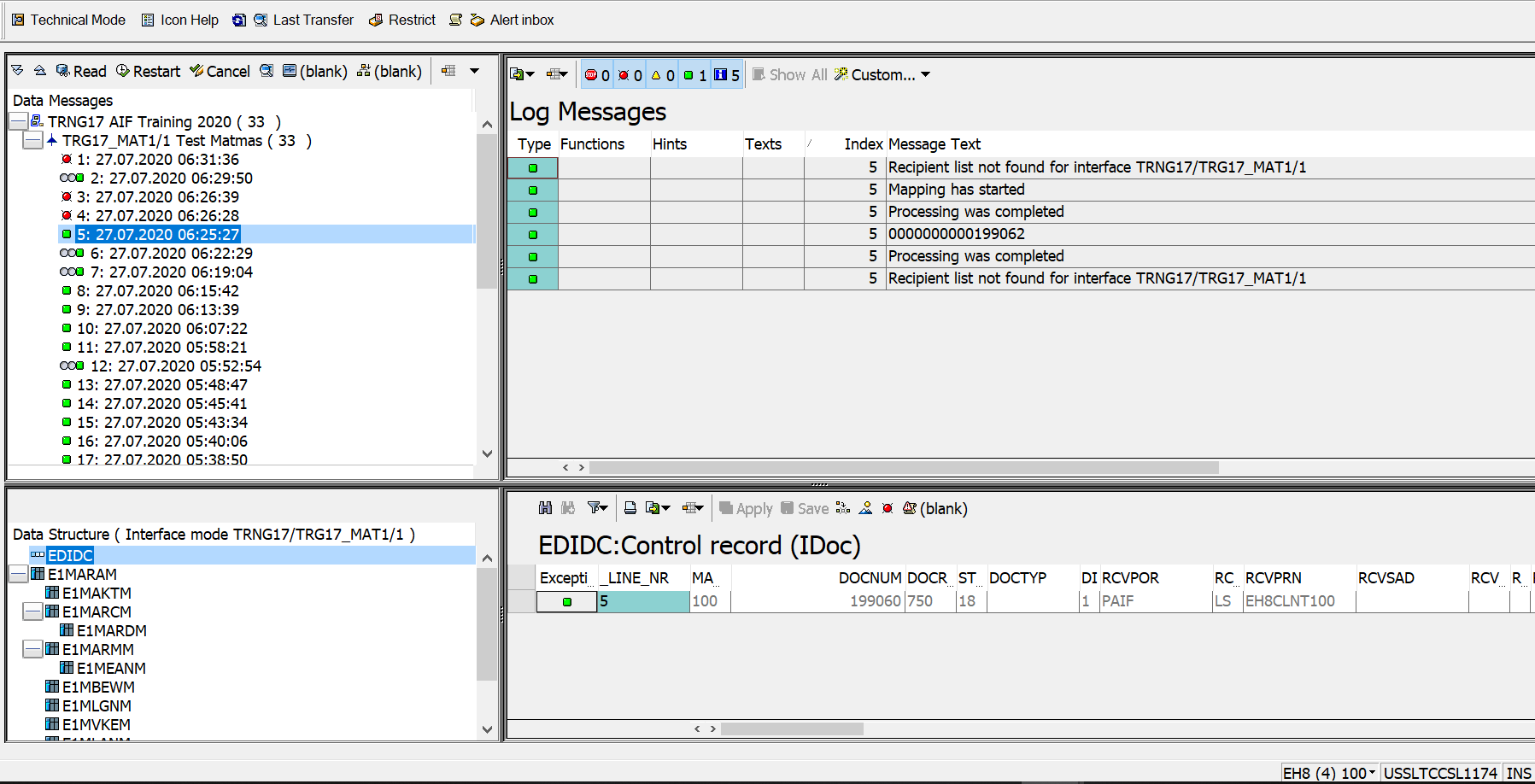


Figure 27: Error Handling Screen

# Conclusion

SAP Application Interface Framework is a powerful, user-friendly tool for implementing Business validations and data messaging before the message is processed in SAP Backend system. By configuring the interfaces in SAP Application Interface Framework, Technical Framework gets differentiated with the Business/Functional Framework. It enriches the business users to perform business validation and monitoring of interfaces as well as take corrective actions when an error occurs.