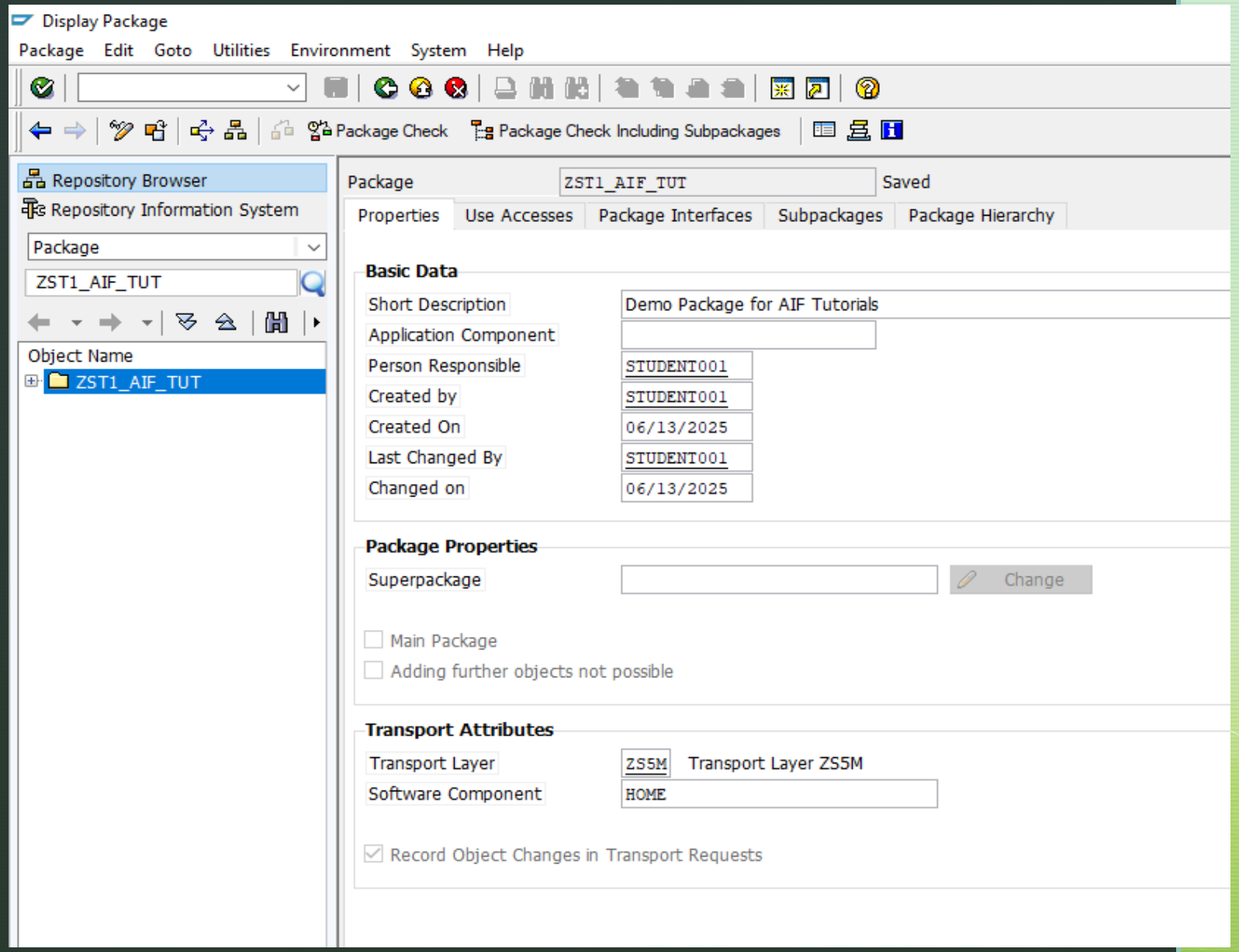
My training by tutorial
from developers.sap.com

https://developers.sap.com/tutorials/aif-proxy-monitoring-interface-create..html

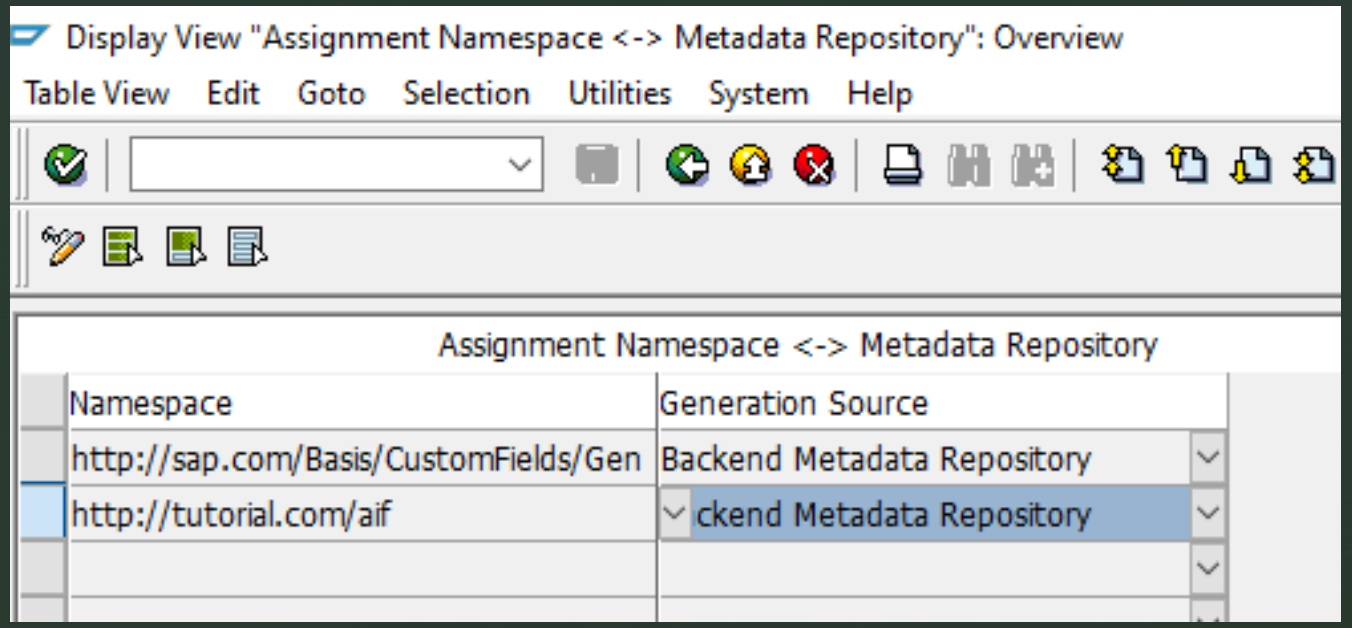
Create a Simple Proxy Interface

- Instructions and codes are taken from tutorial "Create a Simple Proxy Interface" (https://developers.sap.com/tutorials/).
  All screenshots made from my own development performed via this tutorial

Create a package



Next, you need to assign a new namespace to the Backend Metadata Repository.
Run transaction **Assignment Namespace Generating Application** (transaction code SPXNGENAPPL).
Switch to **Edit** mode, add a new entry, and enter or select the following details for your new namespace:
Save your changes.

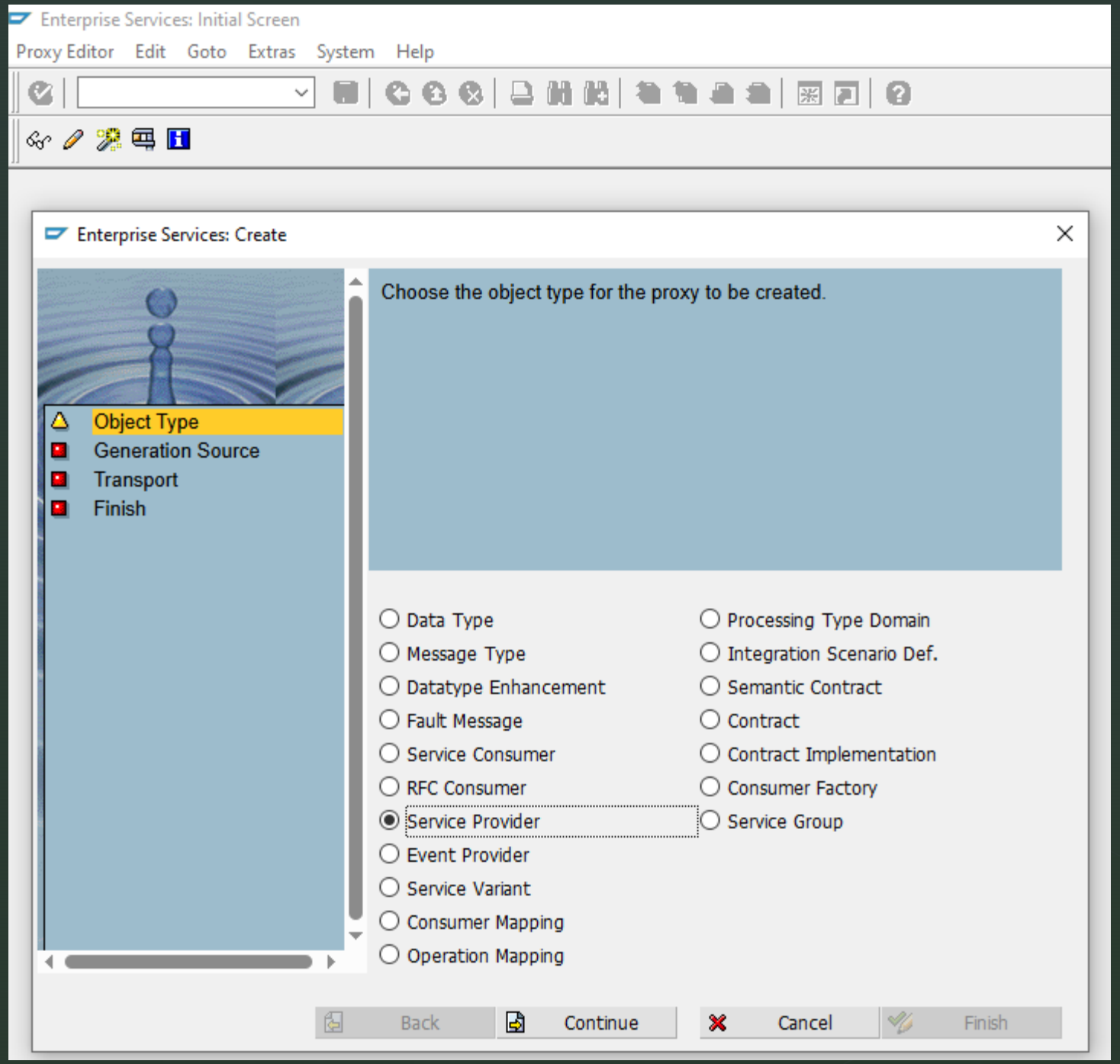


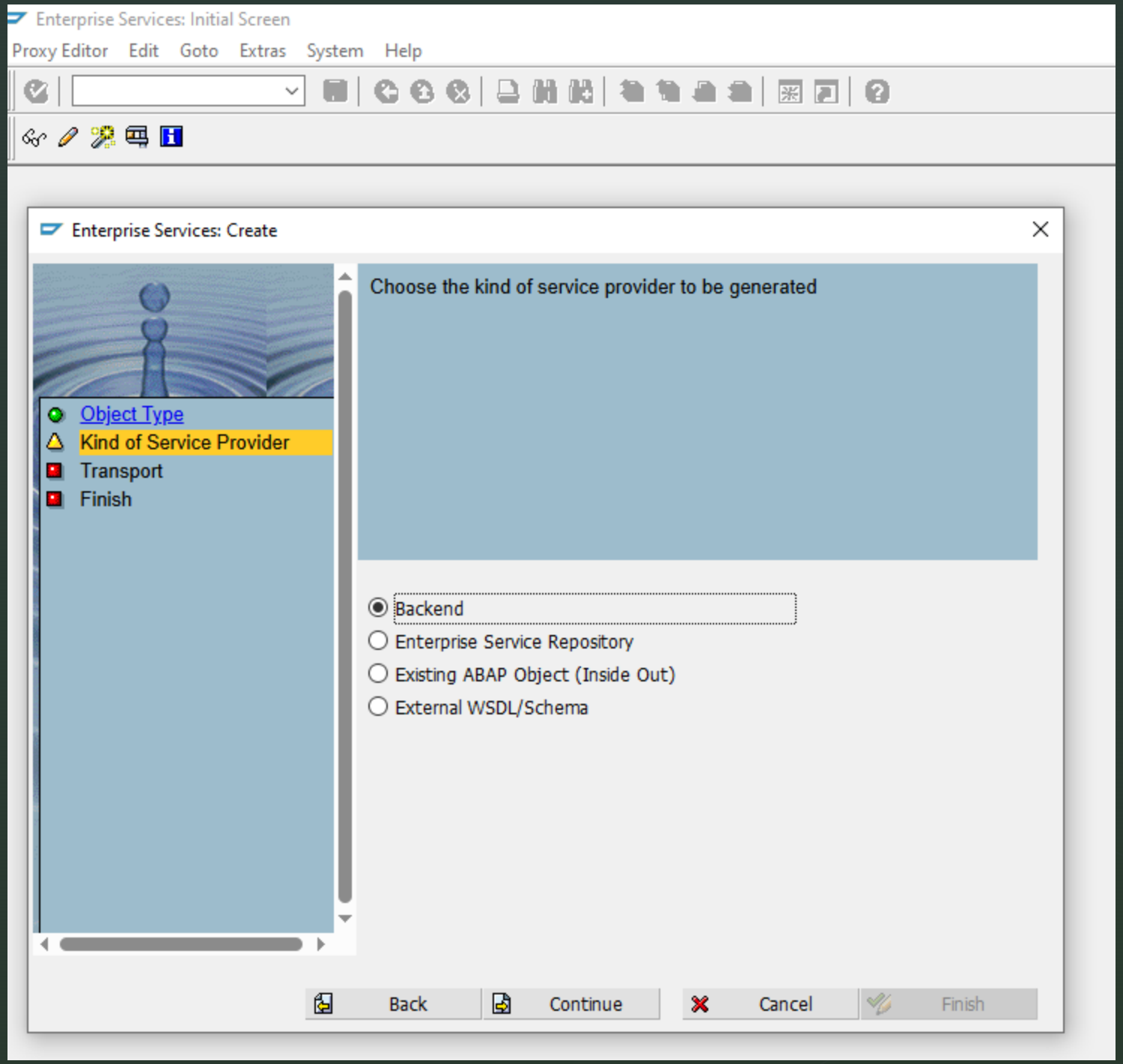
CREATE PROXY

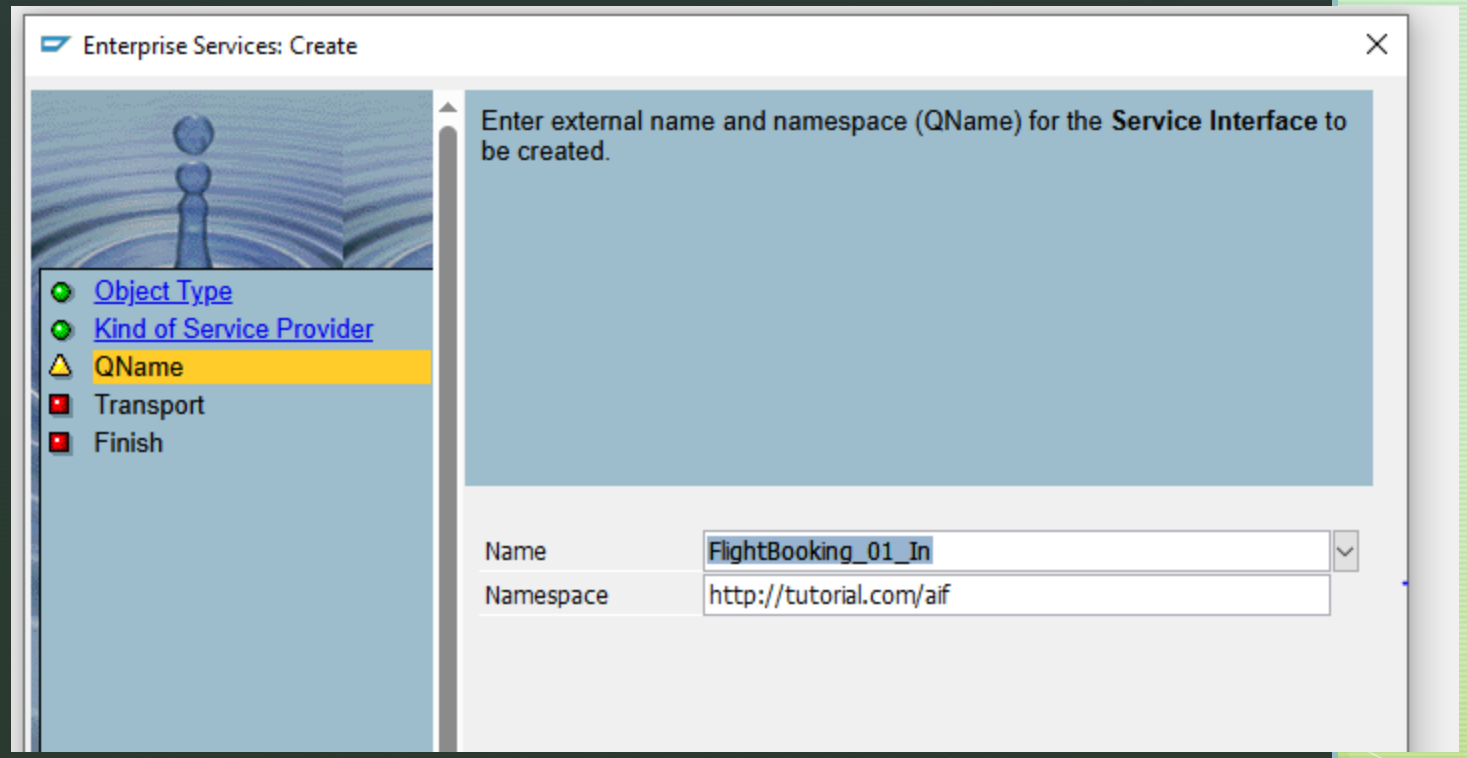
To create a new service interface, run the **proxy editor** (transaction code SPROXY_START).

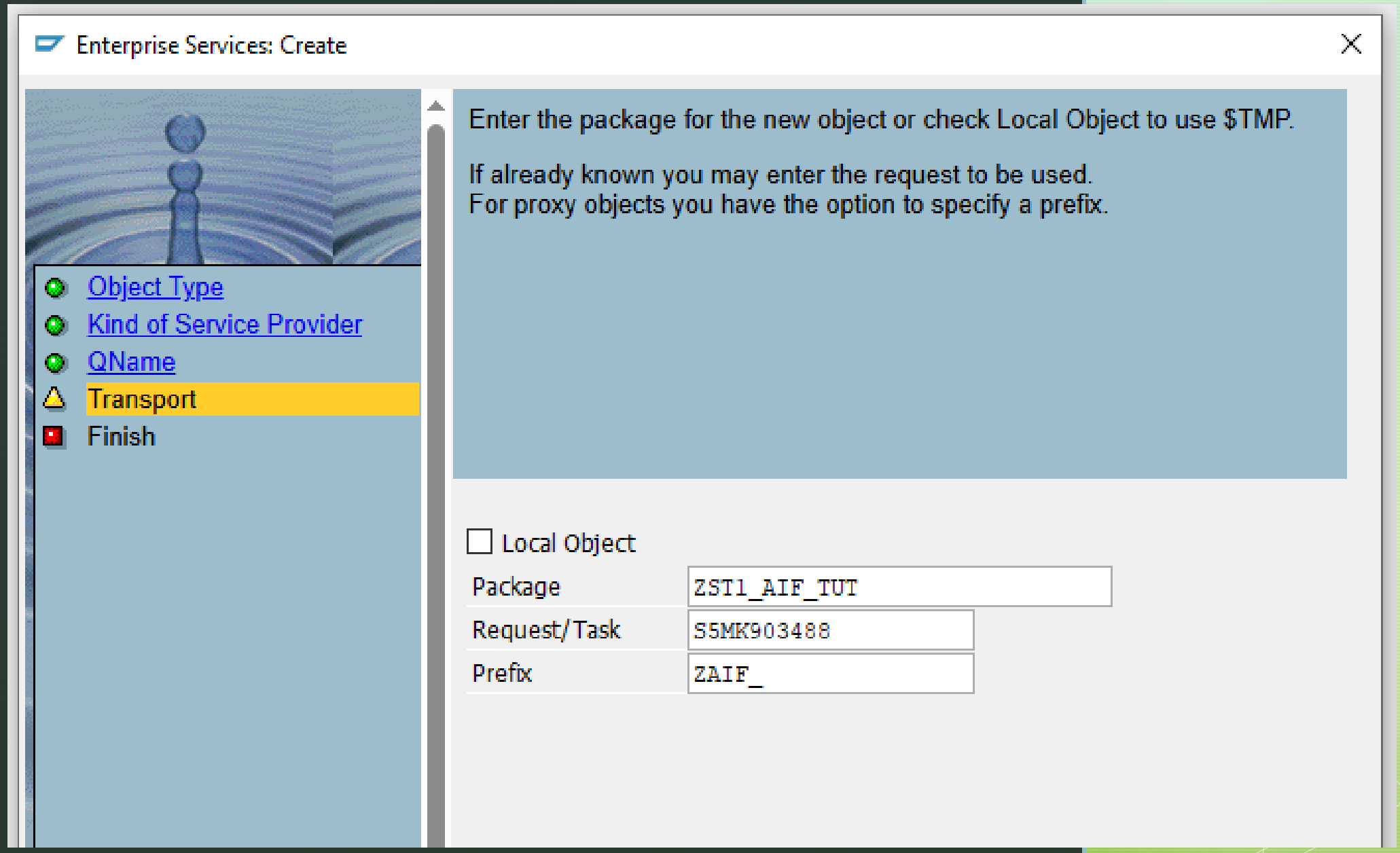
In **Enterprise Services: Initial Screen**, select **Create with Wizard** and carry out the following configuration steps:

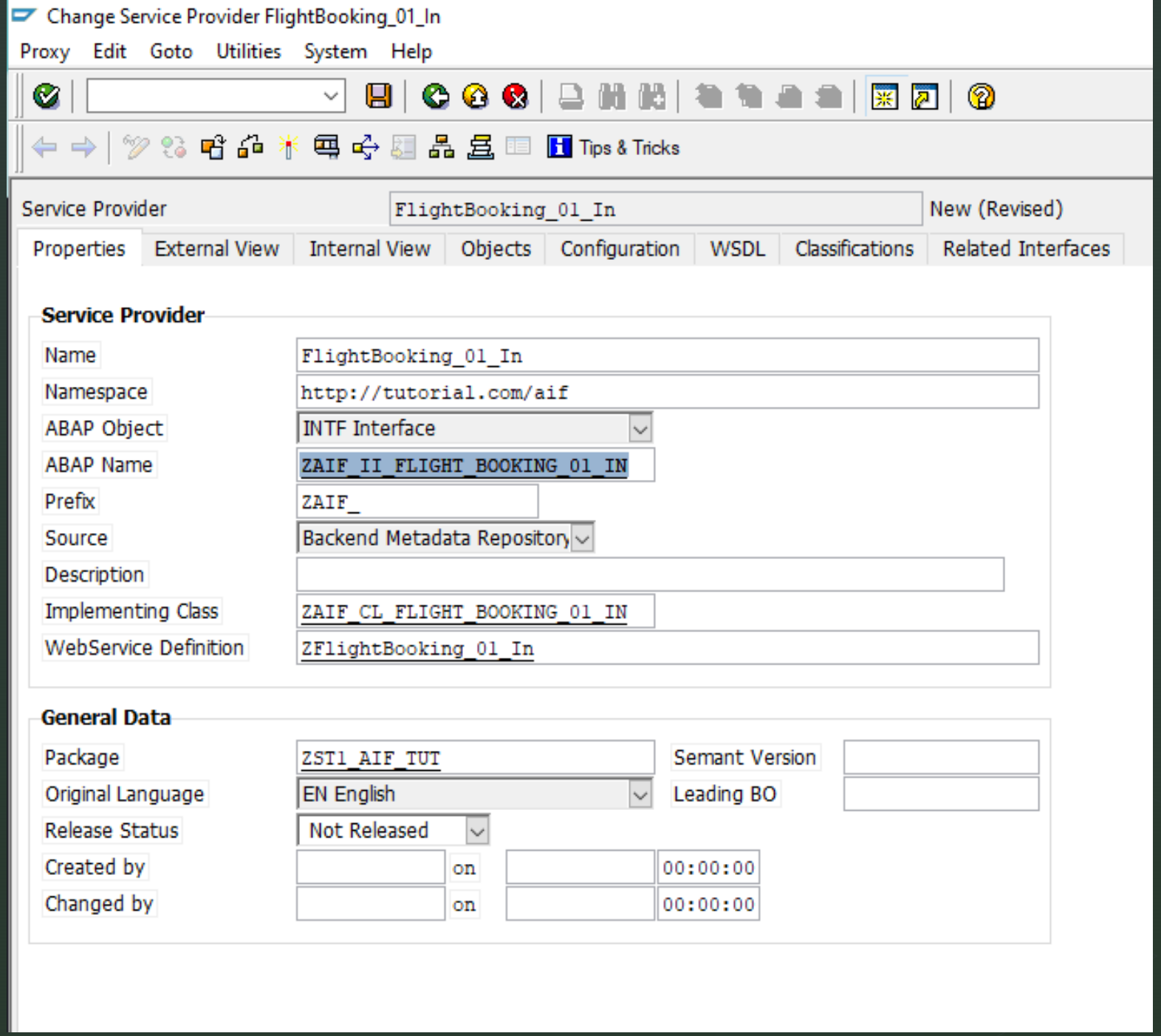
1. As **Object Type**, select **Service Provider**. Click **Continue**.
2. As **Kind of Service Provider**, select **Backend**. Click **Continue**.
3. As **Name**, enter and namespace **http://tutorial.com/aif**. Click **Continue**.
4. For the transport options, enter your package ZST1_AIF_TUT, select a workbench request, and enter the prefix **ZAIF_**. Select **Continue**.

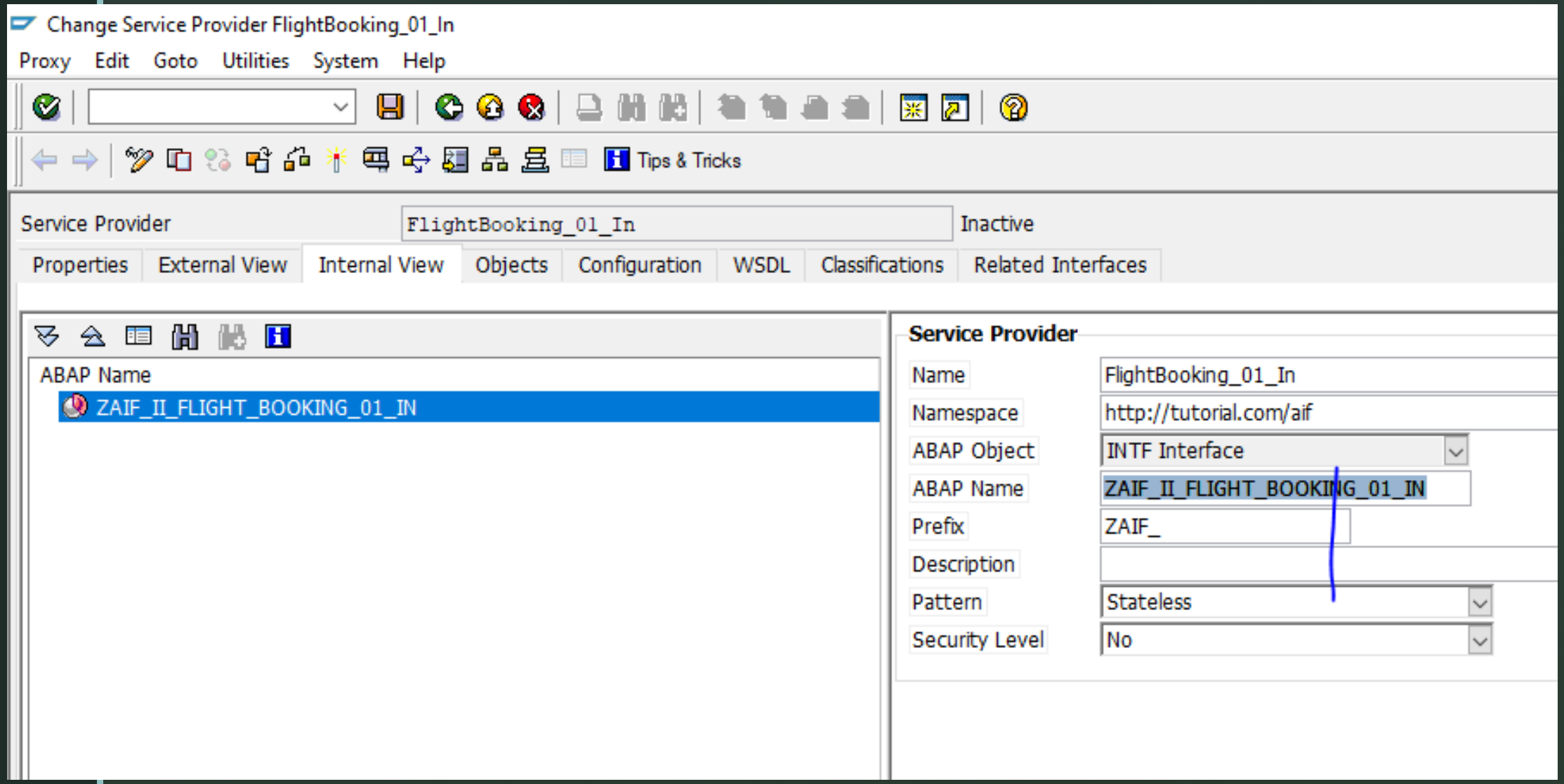

Enterprise Services: Initial Screen
Proxy Editor
Edit
Goto
Extras
System
Help
Enterprise Services: Create
Object Type
Generation Source
Transport
Finish
Choose the object type for the proxy to be created.
Data Type
Message Type
Datatype Enhancement
Fault Message
Service Consumer
RFC Consumer
Service Provider
Event Provider
Service Variant
Consumer Mapping
Operation Mapping
Processing Type Domain
Integration Scenario Def.
Semantic Contract
Contract
Contract Implementation
Consumer Factory
Service Group
Back
Continue
Cancel
Finish

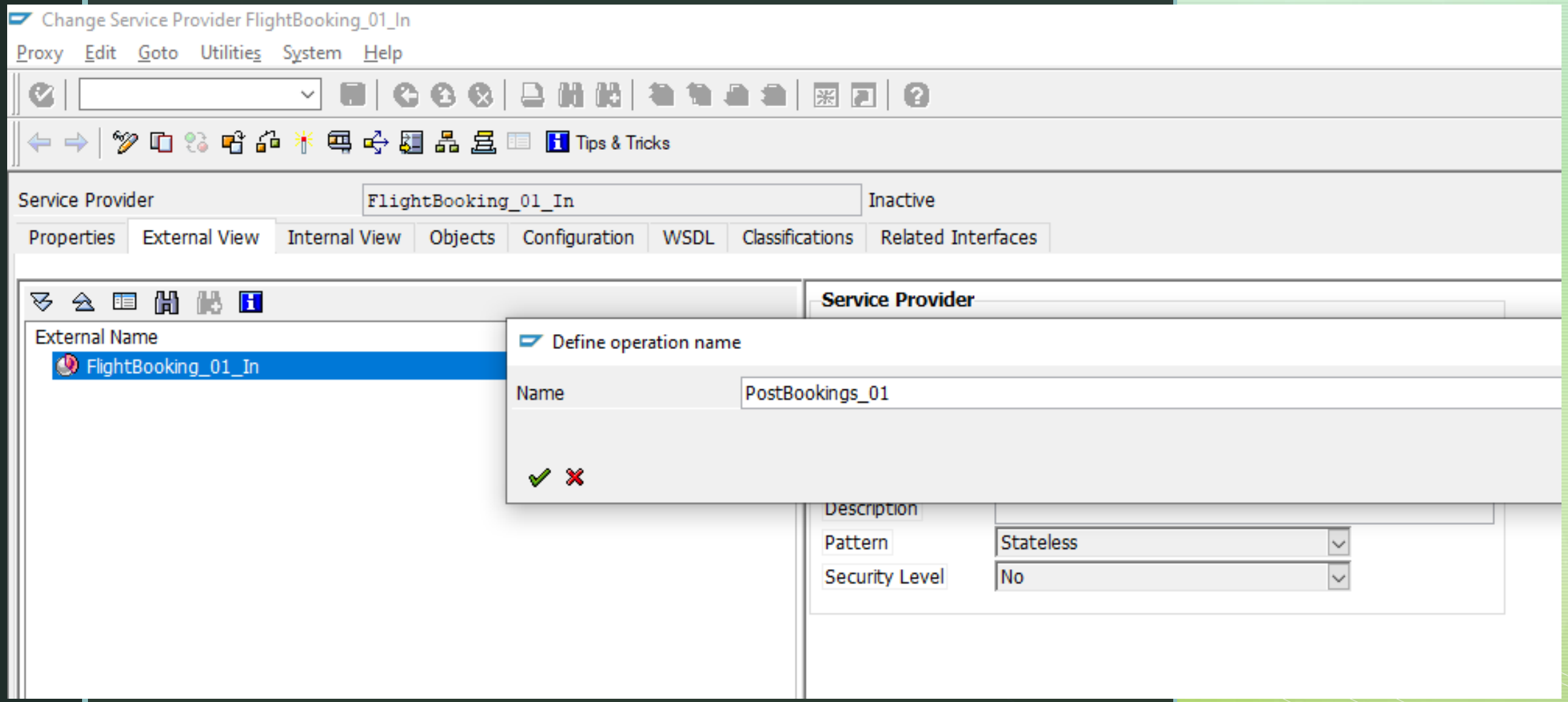

Enterprise Services: Initial Screen
Proxy Editor
Edit
Goto
Extras
System
Help
Enterprise Services: Create
Object Type
Kind of Service Provider
Transport
Finish
Choose the kind of service provider to be generated
Backend
Enterprise Service Repository
Existing ABAP Object (Inside Out)
External WSDL/Schema
Back
Continue
Cancel
Finish


Enterprise Services: Create
Object Type
Kind of Service Provider
QName
Transport
Finish
Enter external name and namespace (QName) for the Service Interface to be created.
Name
FlightBooking_01_In
Namespace
http://tutorial.com/aif


Enterprise Services: Create
Object Type
Kind of Service Provider
QName
Transport
Finish
Enter the package for the new object or check Local Object to use $TMP.
If already known you may enter the request to be used.
For proxy objects you have the option to specify a prefix.
Local Object
Package
ZST1_AIF_TUT
Request/Task
S5MK903488
Prefix
ZAIF_


Change Service Provider FlightBooking_01_In
Proxy Edit Goto Utilities System Help
Tips & Tricks
Service Provider
FlightBooking_01_In
New (Revised)
Properties
External View
Internal View
Objects
Configuration
WSDL
Classifications
Related Interfaces
Service Provider
Name
FlightBooking_01_In
Namespace
http://tutorial.com/aif
ABAP Object
INTF Interface
ABAP Name
ZAIF_II_FLIGHT_BOOKING_01_IN
Prefix
ZAIF_
Source
Backend Metadata Repository
Description
Implementing Class
ZAIF_CL_FLIGHT_BOOKING_01_IN
WebService Definition
ZFlightBooking_01_In
General Data
Package
ZST1_AIF_TUT
Semant Version
Original Language
EN English
Leading BO
Release Status
Not Released
Created by
on
00:00:00
Changed by
on
00:00:00


Change Service Provider FlightBooking_01_In
Proxy Edit Goto Utilities System Help
Tips & Tricks
Service Provider
FlightBooking_01_In
Inactive
Properties
External View
Internal View
Objects
Configuration
WSDL
Classifications
Related Interfaces
ABAP Name
ZAIF_II_FLIGHT_BOOKING_01_IN
Service Provider
Name
FlightBooking_01_In
Namespace
http://tutorial.com/aif
ABAP Object
INTF Interface
ABAP Name
ZAIF_II_FLIGHT_BOOKING_01_IN
Prefix
ZAIF_
Description
Pattern
Stateless
Security Level
No

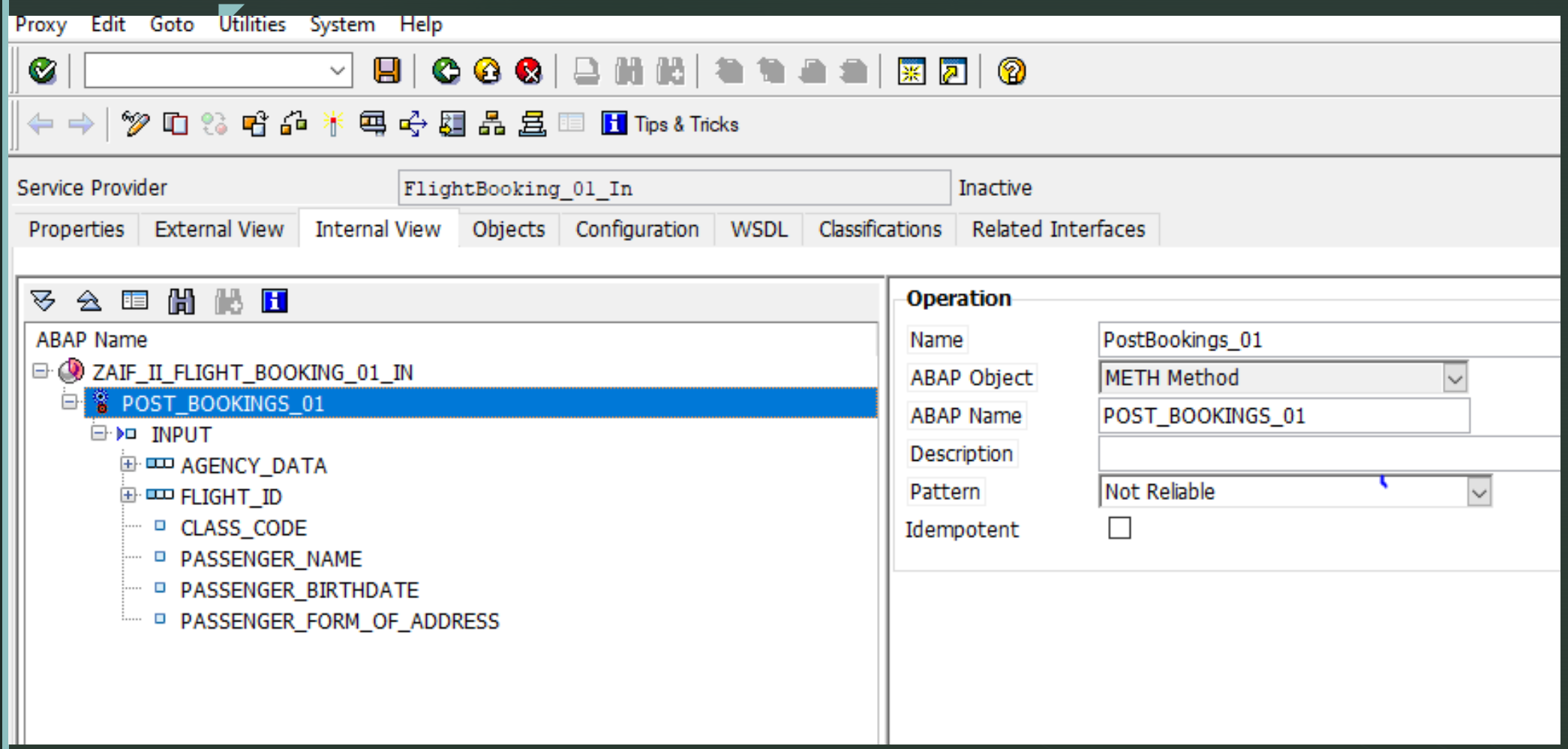

Change Service Provider FlightBooking_01_In
Proxy Edit Goto Utilities System Help
Tips & Tricks
Service Provider
FlightBooking_01_In
Inactive
Properties
External View
Internal View
Objects
Configuration
WSDL
Classifications
Related Interfaces
External Name
FlightBooking_01_In
Service Provider
Define operation name
Name
PostBookings_01
Description
Pattern
Stateless
Security Level
No

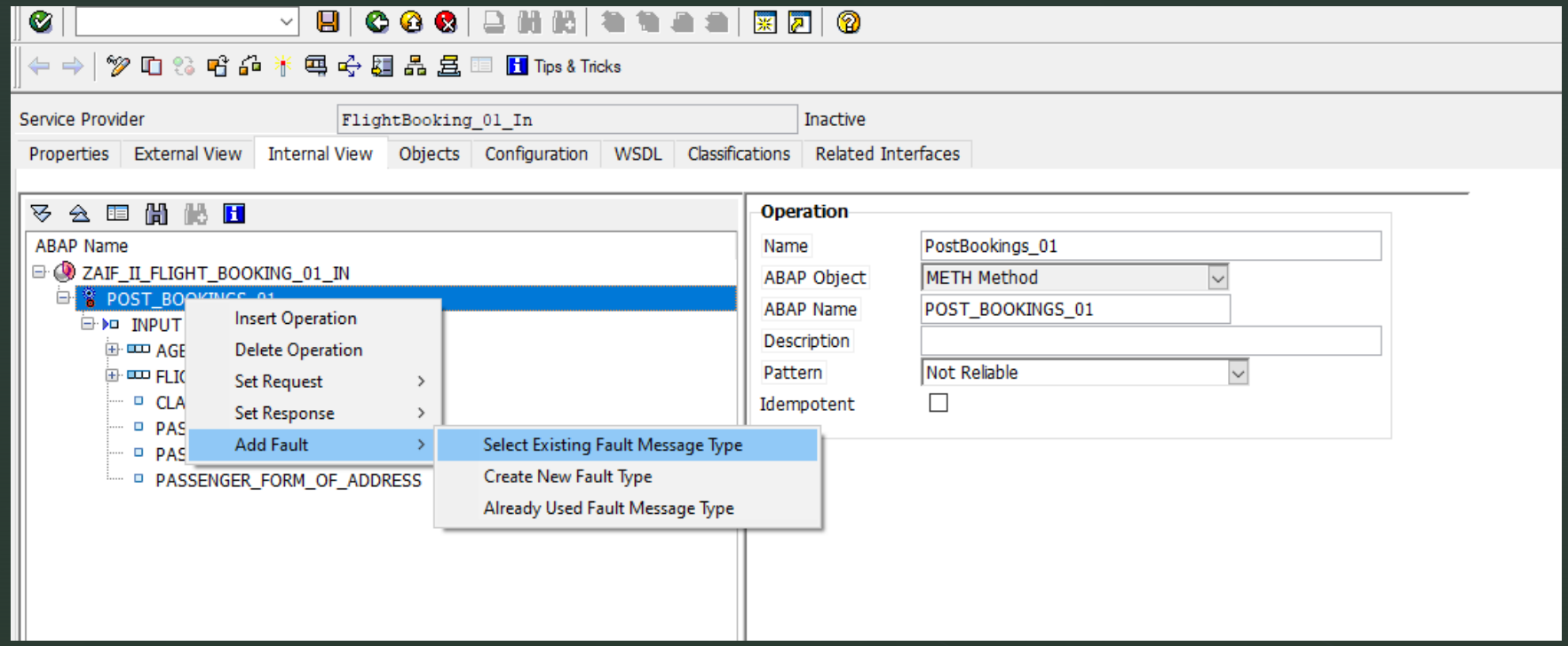
DEFINE PROXY STRUCTURES

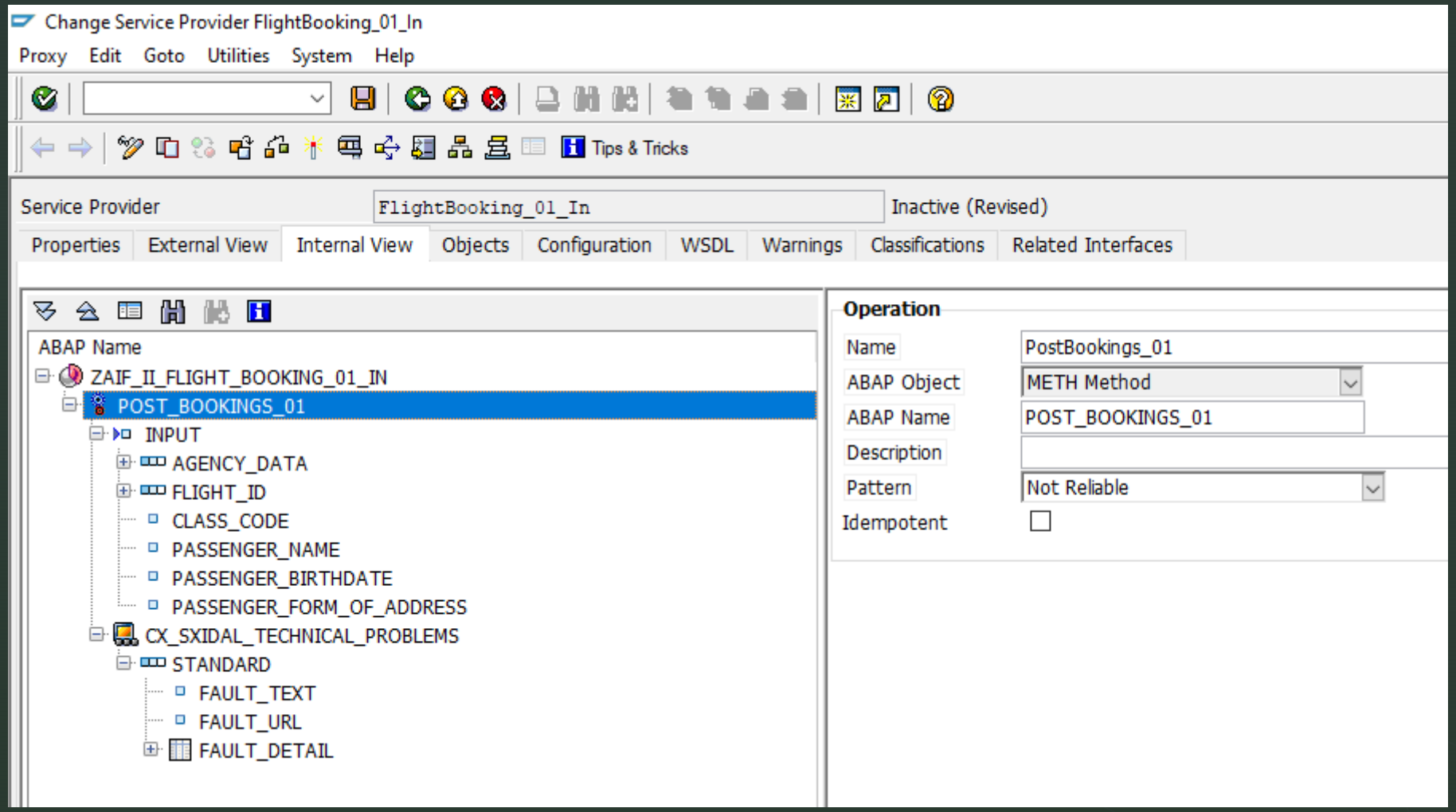
Next, you need to add an operation and a fault message type in the proxy editor.

1.Switch to the **Internal View** tab. Right-click your service provider and select **Add Operation**. Enter the operation name PostBookings_01. With the new operation selected, switch the **Pattern** of the operation to **Not Reliable** to simplify testing.

1.Right-click the operation and select **Set Request > Select Existing Message Type** from the context menu.
2.In the upcoming **Restrict Value Range** dialog, remove all filters.
3.Enter the message type SXIDAL_FBO_REQUEST_MT in the **ABAP Name** search filter,
4. and the namespace http://sap.com/xi/XI/Demo/Airline in the **Namespace** search filter,
5.then press **Enter**. In the search result, select the found entry, and select **Copy**.
2.Right-click the operation and select **Add Fault > Select Existing Fault Message Type**.
3.Similar to the message type search, remove all filters. Then search for the fault message
4.type CX_SXIDAL_TECHNICAL_PROBLEMS.
3.**Save** and activate the proxy.

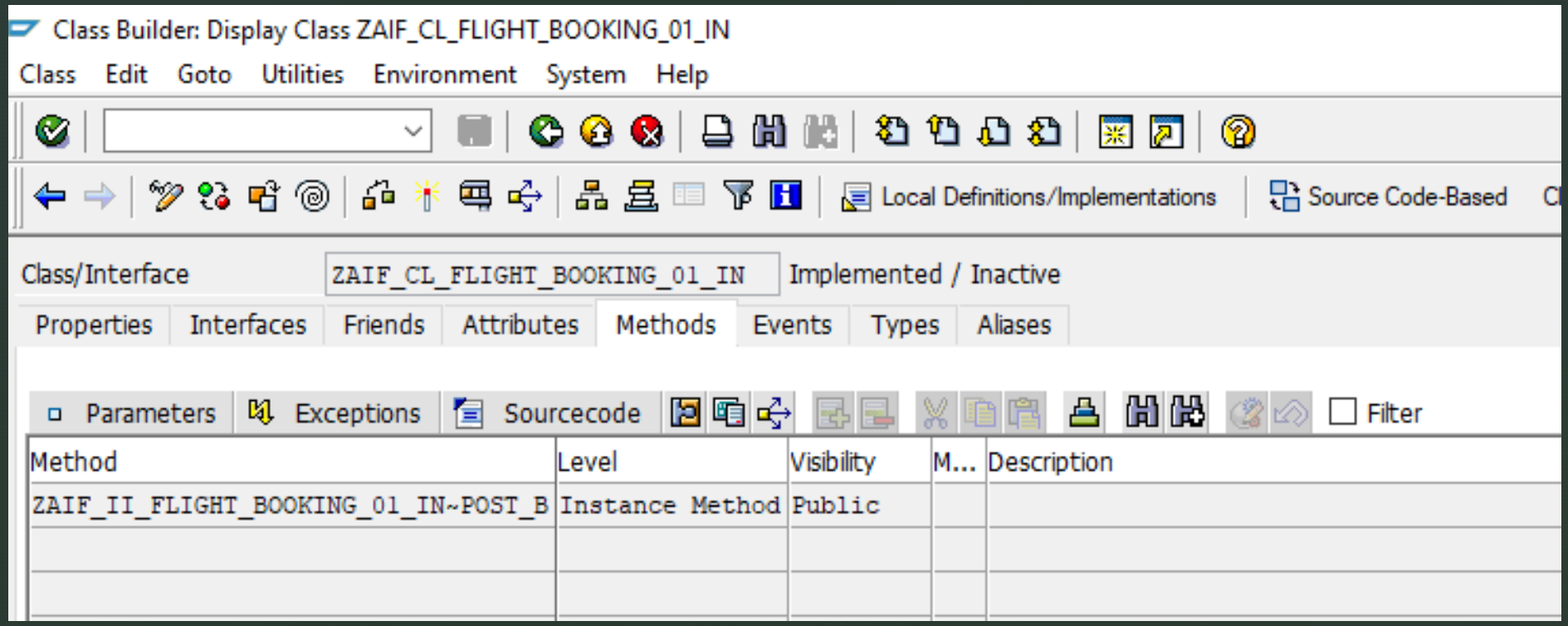

Proxy Edit Goto Utilities System Help
Tips & Tricks
Service Provider
FlightBooking_01_In
Inactive
Properties
External View
Internal View
Objects
Configuration
WSDL
Classifications
Related Interfaces
ABAP Name
ZAIF_II_FLIGHT_BOOKING_01_IN
POST_BOOKINGS_01
INPUT
AGENCY_DATA
FLIGHT_ID
CLASS_CODE
PASSENGER_NAME
PASSENGER_BIRTHDATE
PASSENGER_FORM_OF_ADDRESS
Operation
Name
PostBookings_01
ABAP Object
METH Method
ABAP Name
POST_BOOKINGS_01
Description
Pattern
Not Reliable
Idempotent


Tips & Tricks
Service Provider
FlightBooking_01_In
Inactive
Properties
External View
Internal View
Objects
Configuration
WSDL
Classifications
Related Interfaces
ABAP Name
ZAIF_II_FLIGHT_BOOKING_01_IN
POST_BOOKINGS_01
INPUT
AGE
FLI
CLA
PAS
PAS
PASSENGER_FORM_OF_ADDRESS
Insert Operation
Delete Operation
Set Request
Set Response
Add Fault
Select Existing Fault Message Type
Create New Fault Type
Already Used Fault Message Type
Operation
Name
PostBookings_01
ABAP Object
METH Method
ABAP Name
POST_BOOKINGS_01
Description
Pattern
Not Reliable
Idempotent


Change Service Provider FlightBooking_01_In
Proxy Edit Goto Utilities System Help
Tips & Tricks
Service Provider
FlightBooking_01_In
Inactive (Revised)
Properties External View Internal View Objects Configuration WSDL Warnings Classifications Related Interfaces
ABAP Name
ZAIF_II_FLIGHT_BOOKING_01_IN
POST_BOOKINGS_01
INPUT
AGENCY_DATA
FLIGHT_ID
CLASS_CODE
PASSENGER_NAME
PASSENGER_BIRTHDATE
PASSENGER_FORM_OF_ADDRESS
CX_SXIDAL_TECHNICAL_PROBLEMS
STANDARD
FAULT_TEXT
FAULT_URL
FAULT_DETAIL
Operation
Name
PostBookings_01
ABAP Object
METH Method
ABAP Name
POST_BOOKINGS_01
Description
Pattern
Not Reliable
Idempotent

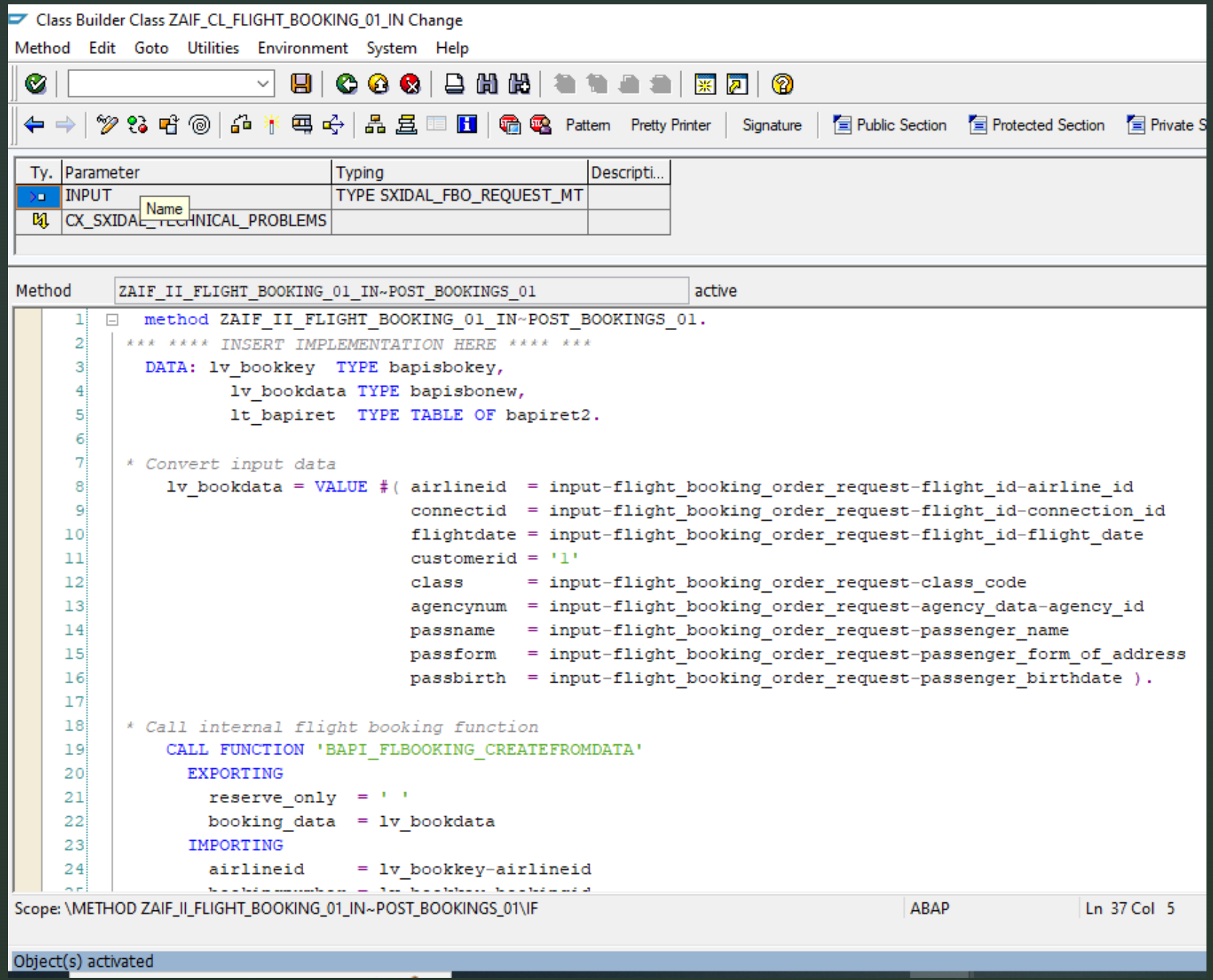
Implement proxy class method

Finally, to book the flights in your test scenario,
the proxy class method needs to be implemented.
Switch to the **Properties** tab.
Double-click the implementing class
ZAIF_CL_FLIGHT_BOOKING_01_IN
and then double-click the method
ZAIF_II_FLIGHT_BOOKING_01_IN~POST_BOOKINGS_01.
Maintain the implementation of the method by copying and pasting the following:


Class Builder: Display Class ZAIF_CL_FLIGHT_BOOKING_01_IN
Class Edit Goto Utilities Environment System Help
Local Definitions/Implementations
Source Code-Based
Class/Interface
ZAIF_CL_FLIGHT_BOOKING_01_IN
Implemented / Inactive
Properties Interfaces Friends Attributes Methods Events Types Aliases
Parameters
Exceptions
Sourcecode
Filter
Method
Level
Visibility
M...
Description
ZAIF_II_FLIGHT_BOOKING_01_IN~POST_B
Instance Method
Public

```
 method ZAIF_II_FLIGHT_BOOKING_01_IN~POST_BOOKINGS_01.

DATA: lv_bookkey  TYPE bapisbokey,
lv_bookdata TYPE bapisbonew,
lt_bapiret  TYPE TABLE OF bapiret2.

* Convert input data
lv_bookdata = VALUE #( airlineid  = input-flight_booking_order_request-flight_id-
airline_id
                       connectid  = input-flight_booking_order_request-flight_id-connection_id
                       flightdate = input-flight_booking_order_request-flight_id-flight_date
                       customerid = '1'
                       class      = input-flight_booking_order_request-class_code
                       agencynum  = input-flight_booking_order_request-agency_data-
agency_id
                       passname   = input-flight_booking_order_request-passenger_name
                       passform   = input-flight_booking_order_request-
passenger_form_of_address
                       passbirth  = input-flight_booking_order_request-passenger_birthdate ).
```

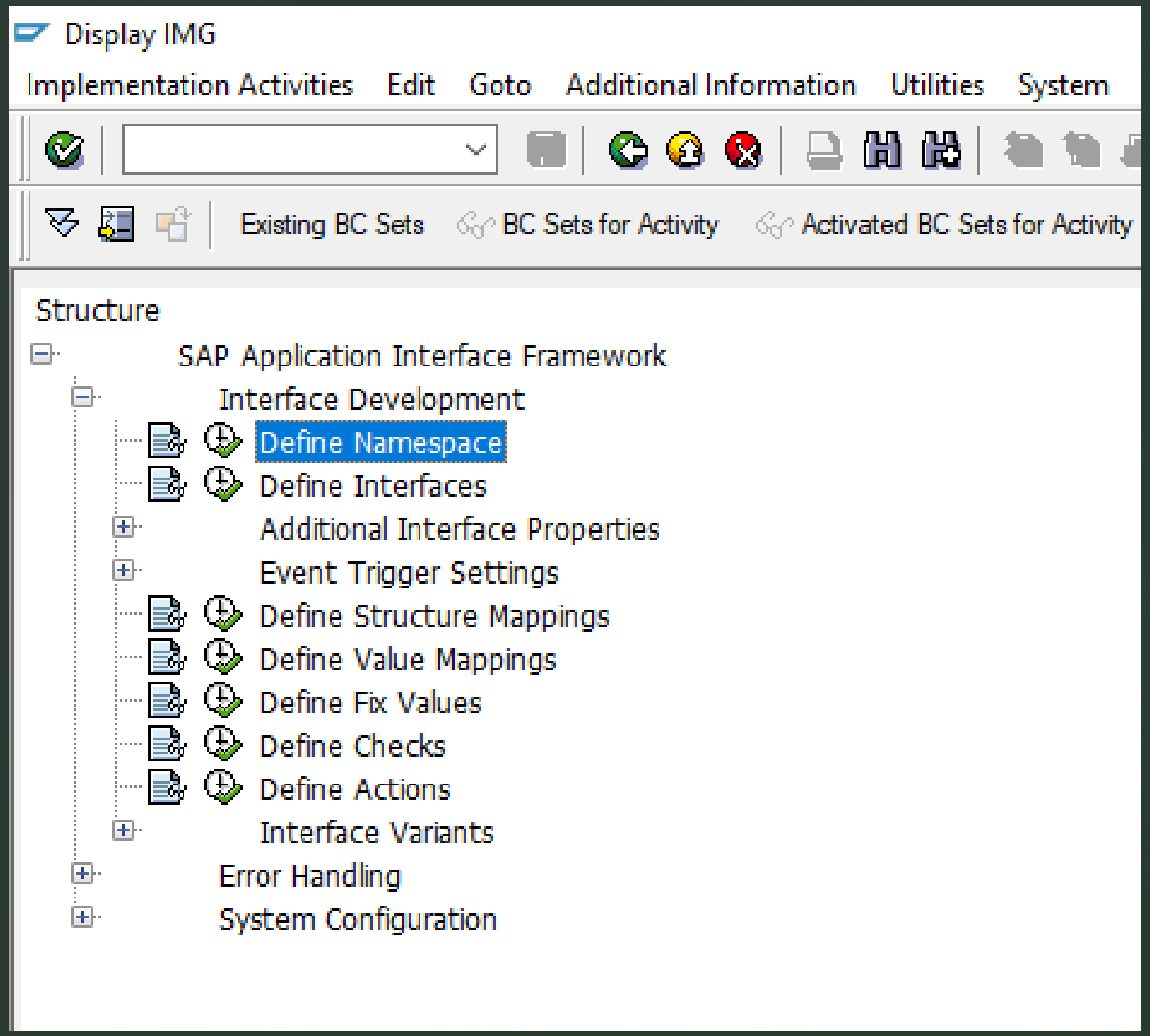
```
* Call internal flight booking function
    CALL FUNCTION 'BAPI_FLBOOKING_CREATEFROMDATA'
      EXPORTING
        reserve_only  = ' '
        booking_data  = lv_bookdata
      IMPORTING
        airlineid     = lv_bookkey-airlineid
        bookingnumber = lv_bookkey-bookingid
      TABLES
        return        = lt_bapiret.

* error case
    IF line_exists( lt_bapiret[ type = 'E' id = 'BAPI' number ='001' ] ).

      CALL METHOD cl_proxy_fault=>raise
        EXPORTING
          exception_class_name = 'CX_SXIDAL_TECHNICAL_PROBLEMS'
          bapireturn_tab       = lt_bapiret.

    ENDIF.
ENDMETHOD.
```

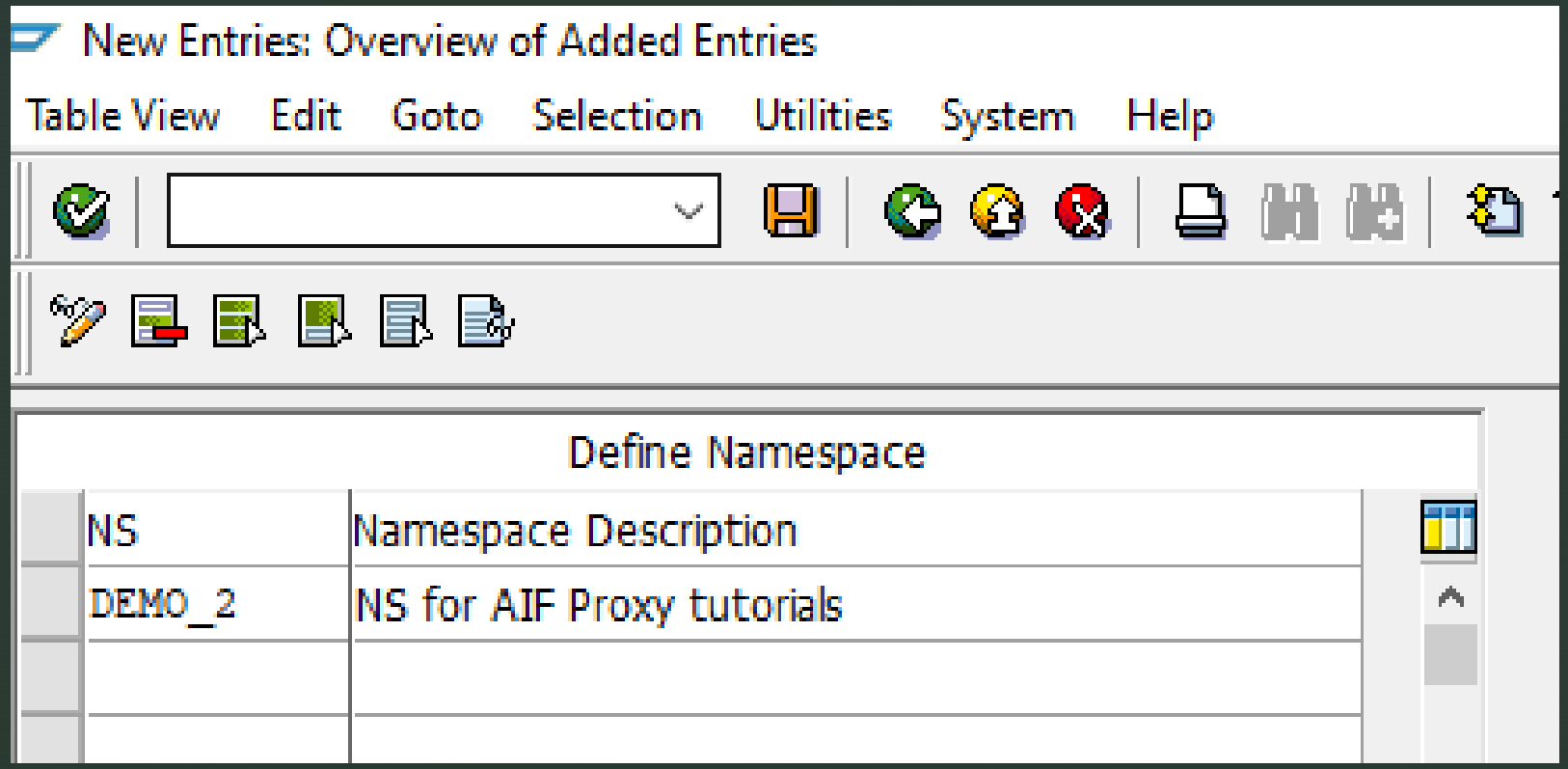

Class Builder Class ZAIF_CL_FLIGHT_BOOKING_01_IN Change
Method Edit Goto Utilities Environment System Help
Pattern Pretty Printer Signature Public Section Protected Section Private S
Ty. Parameter Typing Descripti...
INPUT TYPE SXIDAL_FBO_REQUEST_MT
Name
CX_SXIDAL_TECHNICAL_PROBLEMS
Method ZAIF_II_FLIGHT_BOOKING_01_IN~POST_BOOKINGS_01 active
method ZAIF_II_FLIGHT_BOOKING_01_IN~POST_BOOKINGS_01.
*** **** INSERT IMPLEMENTATION HERE **** ***
DATA: lv_bookkey TYPE bapisbokey,
lv_bookdata TYPE bapisbonew,
lt_bapiret TYPE TABLE OF bapiret2.
* Convert input data
lv_bookdata = VALUE #(airlineid = input-flight_booking_order_request-flight_id-airline_id
connectid = input-flight_booking_order_request-flight_id-connection_id
flightdate = input-flight_booking_order_request-flight_id-flight_date
customerid = '1'
class = input-flight_booking_order_request-class_code
agencynum = input-flight_booking_order_request-agency_data-agency_id
passname = input-flight_booking_order_request-passenger_name
passform = input-flight_booking_order_request-passenger_form_of_address
passbirth = input-flight_booking_order_request-passenger_birthdate).
* Call internal flight booking function
CALL FUNCTION 'BAPI_FLBOOKING_CREATEFROMDATA'
EXPORTING
reserve_only = ' '
booking_data = lv_bookdata
IMPORTING
airlineid = lv_bookkey-airlineid
Scope: \METHOD ZAIF_II_FLIGHT_BOOKING_01_IN~POST_BOOKINGS_01\IF
ABAP
Ln 37 Col 5
Object(s) activated

CREATE NAMESPACE

As interfaces in SAP Application Interface Framework are grouped using namespaces, you must create a namespace.
Go to **Customizing** for SAP Application Interface Framework (transaction code /n/AIF/CUST)
and navigate to **Interface Development** > **Define Namespace**.
Select **New Entries** and enter the following name and description for your new namespace:

Namespace	Namespace Description
DEMO_2	NS for AIF Proxy tutorials


Display IMG
Implementation Activities
Edit
Goto
Additional Information
Utilities
System
Existing BC Sets
BC Sets for Activity
Activated BC Sets for Activity
Structure
SAP Application Interface Framework
Interface Development
Define Namespace
Define Interfaces
Additional Interface Properties
Event Trigger Settings
Define Structure Mappings
Define Value Mappings
Define Fix Values
Define Checks
Define Actions
Interface Variants
Error Handling
System Configuration


New Entries: Overview of Added Entries
Table View
Edit
Goto
Selection
Utilities
System
Help
Define Namespace
NS
Namespace Description
DEMO_2
NS for AIF Proxy tutorials

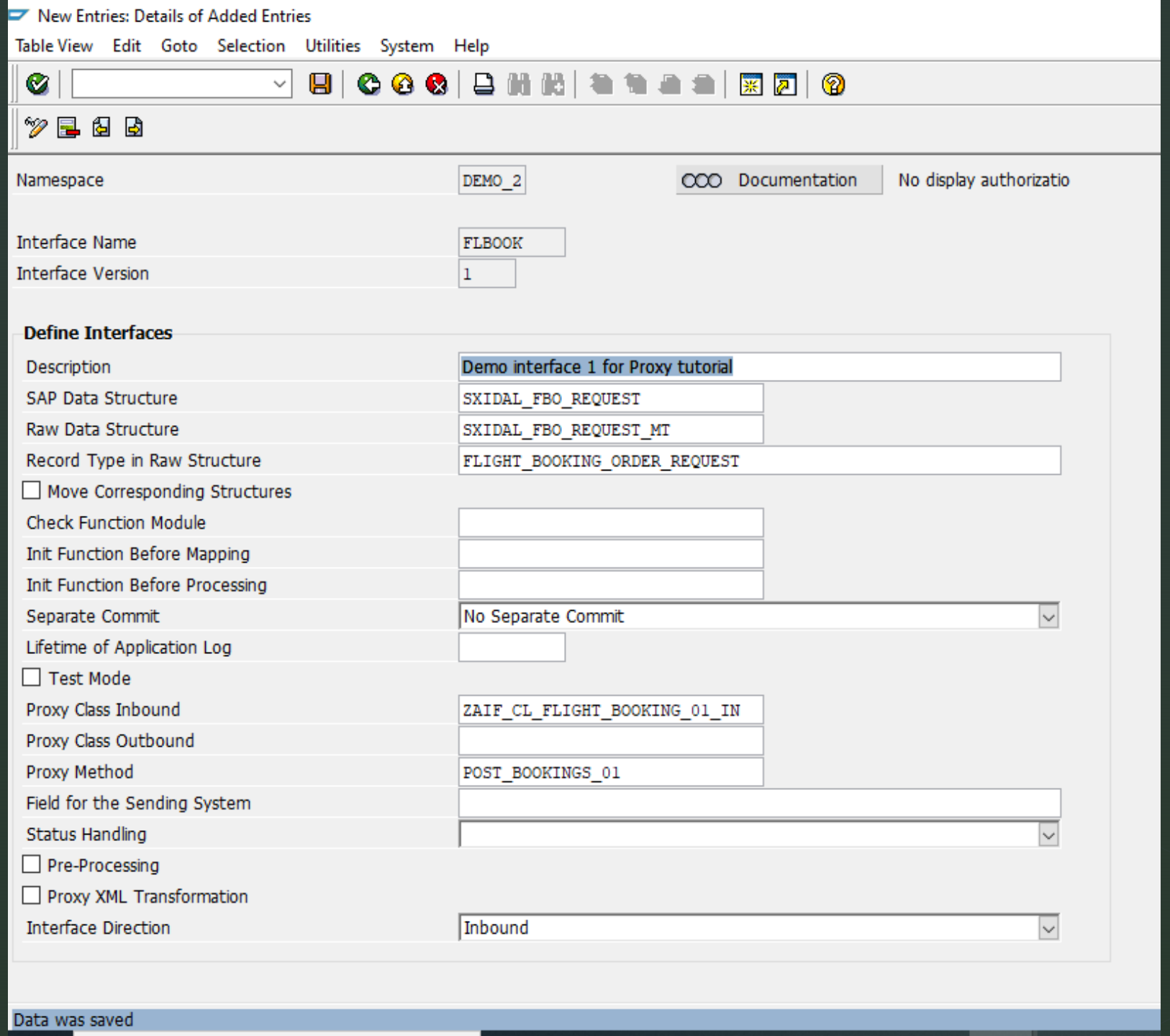
CREATE INTERFACE

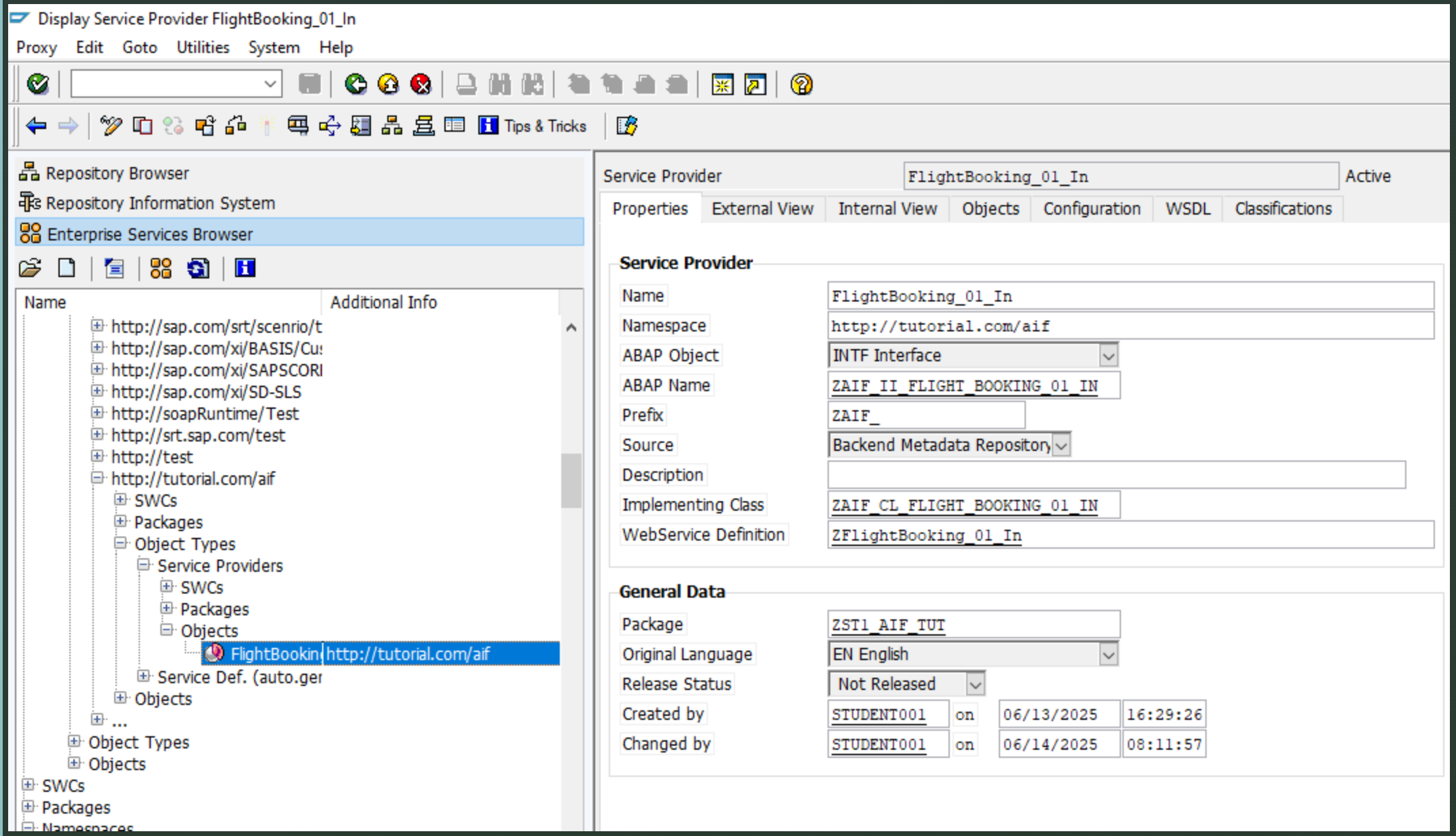
While still in **Customizing** (transaction code /n/AIF/CUST), navigate to **Interface Development** > **Define Interfaces**.
In the upcoming dialog, enter your previously created namespace **DEMO_2**
and press **Enter**.
Select **New Entries** and enter the following parameters based on your proxy class and implementation.
You can double-check this information in transaction code SPROXY

> Be aware that entering the **Proxy Class Inbound** automatically fills in **Raw Data Structure**, **Record Type in Raw Structure**, and **Proxy Method**.

Field name	Description	VALUE
Interface Name	Name of the interface to be created, for example (an abbreviation of) the basic type	`FLBOOK`
Interface version	Version number of the interface	`1`
Description	Description of the interface	`Demo interface 1 for Proxy tutorial`
SAP Data structure	Input substructure of the proxy class	`SXIDAL_FBO_REQUEST`
Raw Data structure	Input structure of the proxy class	`SXIDAL_FBO_REQUEST_MT`
Record Type in Raw Structure	Main component of the raw data structure	`FLIGHT_BOOKING_ORDER_REQUEST`
Proxy Class Inbound	Name of the proxy class	`ZAIF_CL_FLIGHT_BOOKING_01_IN`
Proxy Method	Method name of the generated proxy class	`POST_BOOKINGS_01`
Interface Direction	Indicates the direction of the interface	`Inbound`

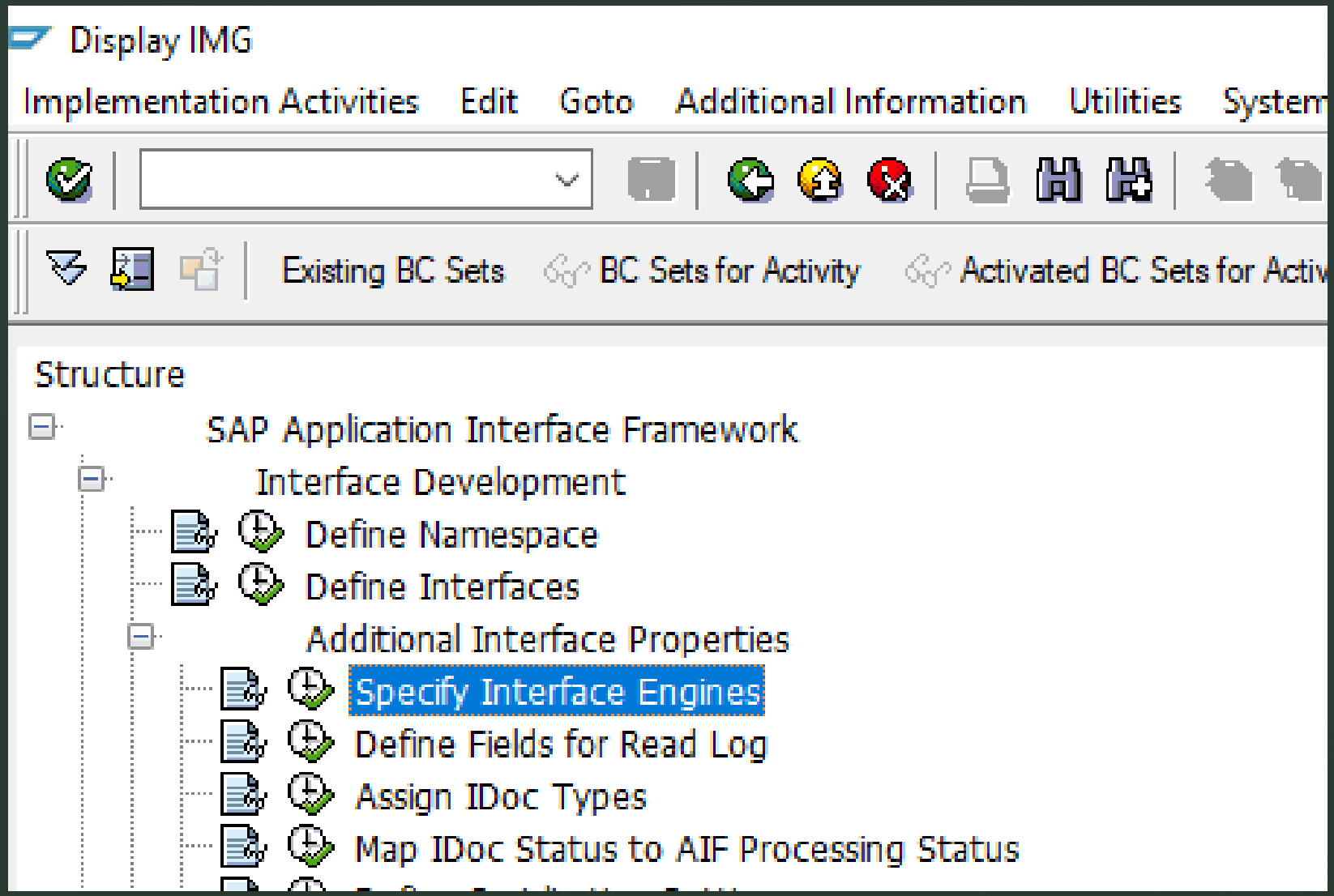
Save your changes.

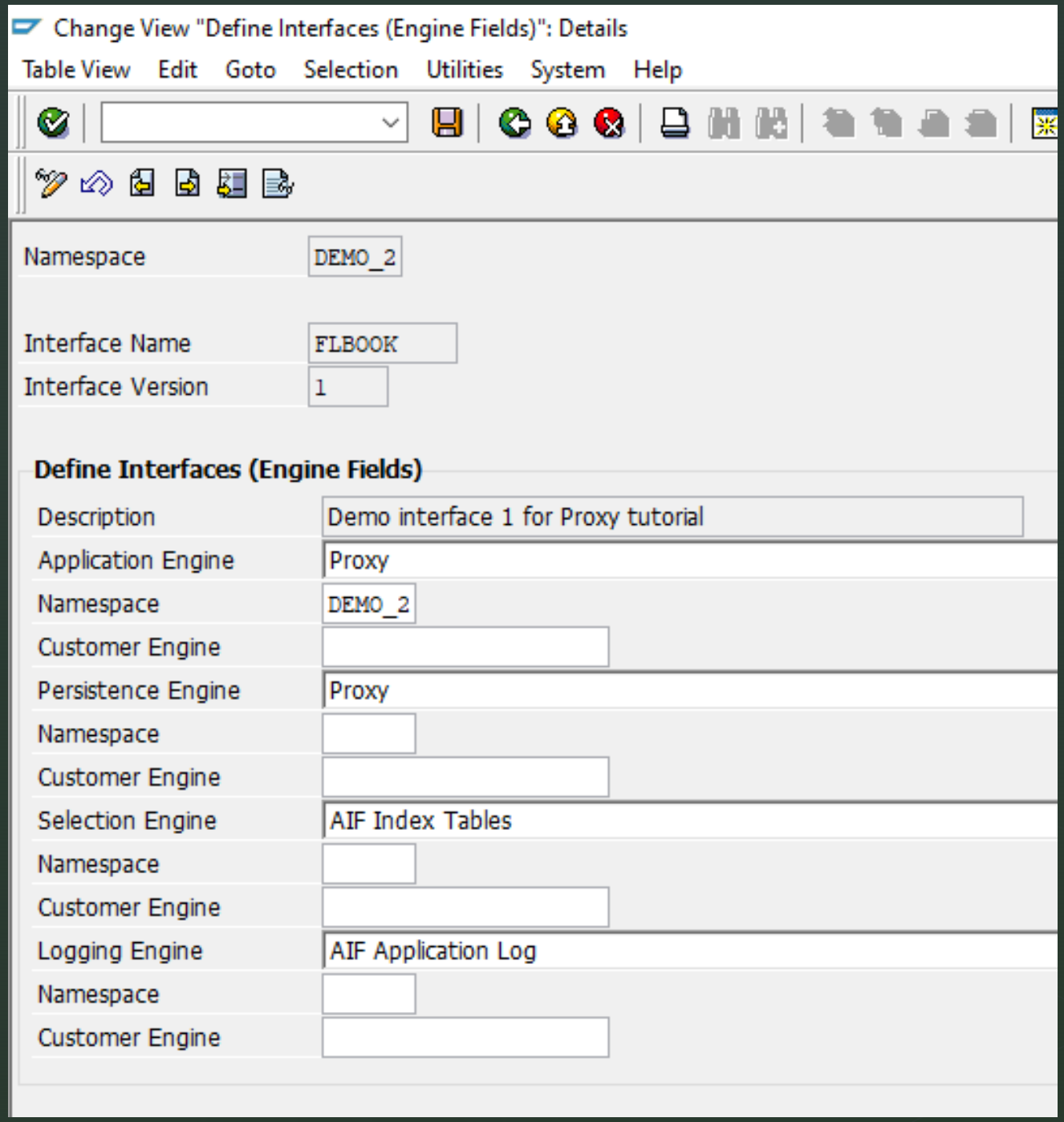

New Entries: Details of Added Entries
Table View Edit Goto Selection Utilities System Help
Namespace DEMO_2
Documentation
No display authorizatio
Interface Name FLBOOK
Interface Version 1
Define Interfaces
Description Demo interface 1 for Proxy tutorial
SAP Data Structure SXIDAL_FBO_REQUEST
Raw Data Structure SXIDAL_FBO_REQUEST_MT
Record Type in Raw Structure FLIGHT_BOOKING_ORDER_REQUEST
Move Corresponding Structures
Check Function Module
Init Function Before Mapping
Init Function Before Processing
Separate Commit No Separate Commit
Lifetime of Application Log
Test Mode
Proxy Class Inbound ZAIF_CL_FLIGHT_BOOKING_01_IN
Proxy Class Outbound
Proxy Method POST_BOOKINGS_01
Field for the Sending System
Status Handling
Pre-Processing
Proxy XML Transformation
Interface Direction Inbound
Data was saved


Display Service Provider FlightBooking_01_In
Proxy Edit Goto Utilities System Help
Tips & Tricks
Repository Browser
Repository Information System
Enterprise Services Browser
Name
Additional Info
http://sap.com/srt/scenrio/t
http://sap.com/xi/BASIS/Cu
http://sap.com/xi/SAPSCORI
http://sap.com/xi/SD-SLS
http://soapRuntime/Test
http://srt.sap.com/test
http://test
http://tutorial.com/aif
SWCs
Packages
Object Types
Service Providers
SWCs
Packages
Objects
FlightBookin
http://tutorial.com/aif
Service Def. (auto.ger
Objects
...
Object Types
Objects
SWCs
Packages
Service Provider
FlightBooking_01_In
Active
Properties
External View
Internal View
Objects
Configuration
WSDL
Classifications
Service Provider
Name
FlightBooking_01_In
Namespace
http://tutorial.com/aif
ABAP Object
INTF Interface
ABAP Name
ZAIF_II_FLIGHT_BOOKING_01_IN
Prefix
ZAIF_
Source
Backend Metadata Repository
Description
Implementing Class
ZAIF_CL_FLIGHT_BOOKING_01_IN
WebService Definition
ZFlightBooking_01_In
General Data
Package
ZST1_AIF_TUT
Original Language
EN English
Release Status
Not Released
Created by
STUDENT001
on
06/13/2025
16:29:26
Changed by
STUDENT001
on
06/14/2025
08:11:57

SPECIFY INTEERFACE ENGINES

Next, you have to select the engines that should be used to handle
the messages that are processed.
If you create a new interface, by default, SAP Application Interface Framework
handles the messages as proxy messages,
so you can keep the default settings.
To double-check the settings, go to **Customizing** for SAP Application Interface
Framework (transaction code /AIF/CUST)
and navigate to **Interface Development** > **Additional Interface Properties** > **Specify Interface Engines**.
In the upcoming dialog, enter your beforehand created namespace **DEMO_2**, and press **Enter**.
Check that the following engines are preselected:

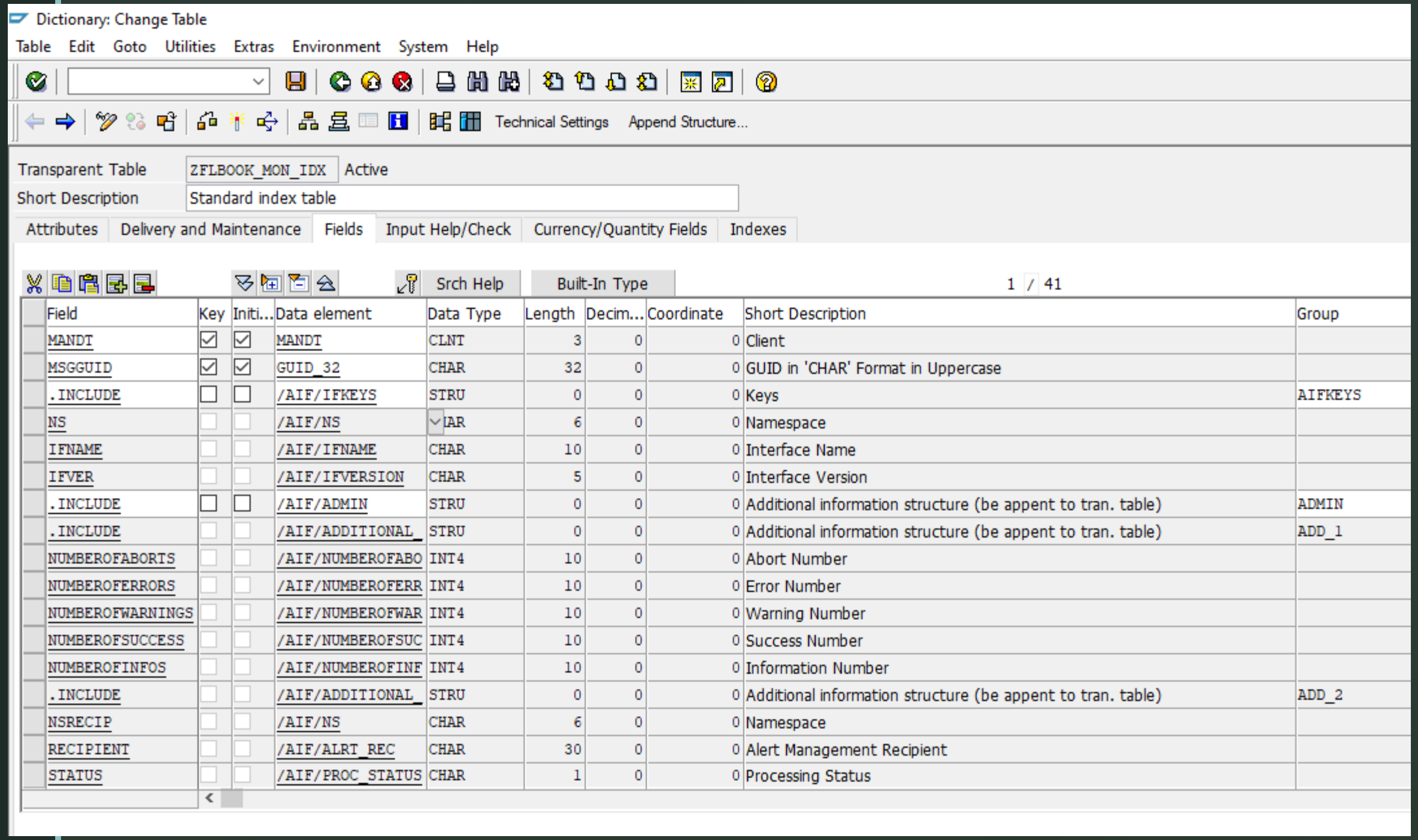

Display IMG
Implementation Activities
Edit
Goto
Additional Information
Utilities
Existing BC Sets
BC Sets for Activity
Activated BC Sets for Activ
Structure
SAP Application Interface Framework
Interface Development
Define Namespace
Define Interfaces
Additional Interface Properties
Specify Interface Engines
Define Fields for Read Log
Assign IDoc Types
Map IDoc Status to AIF Processing Status

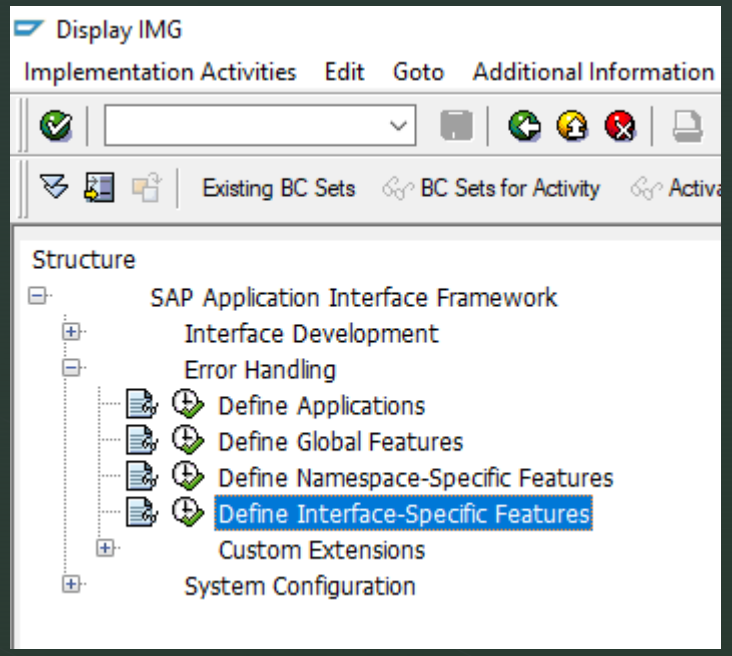

Change View "Define Interfaces (Engine Fields)": Details
Table View Edit Goto Selection Utilities System Help
Namespace DEMO_2
Interface Name FLBOOK
Interface Version 1
Define Interfaces (Engine Fields)
Description Demo interface 1 for Proxy tutorial
Application Engine Proxy
Namespace DEMO_2
Customer Engine
Persistence Engine Proxy
Namespace
Customer Engine
Selection Engine AIF Index Tables
Namespace
Customer Engine
Logging Engine AIF Application Log
Namespace
Customer Engine

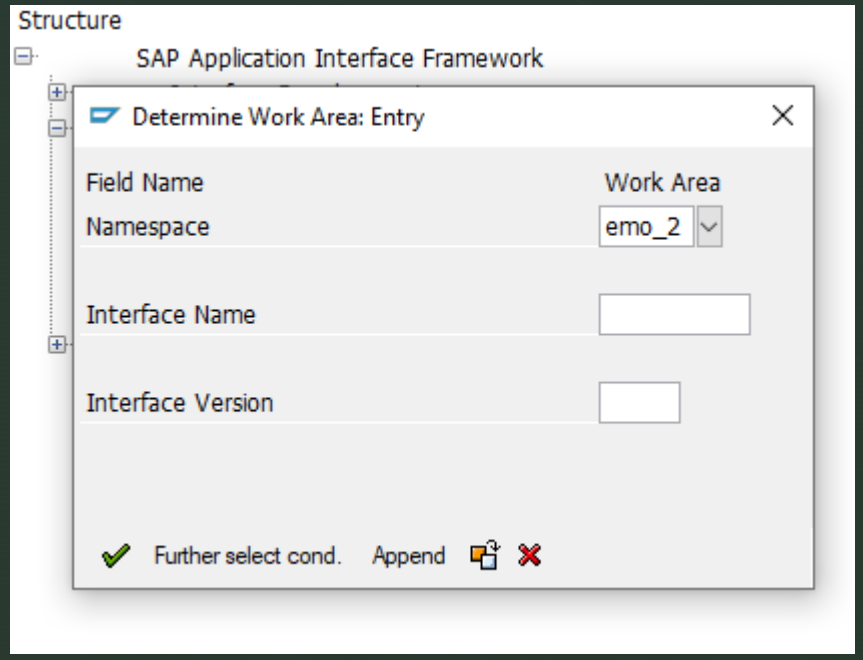
Create interface-specific single index table

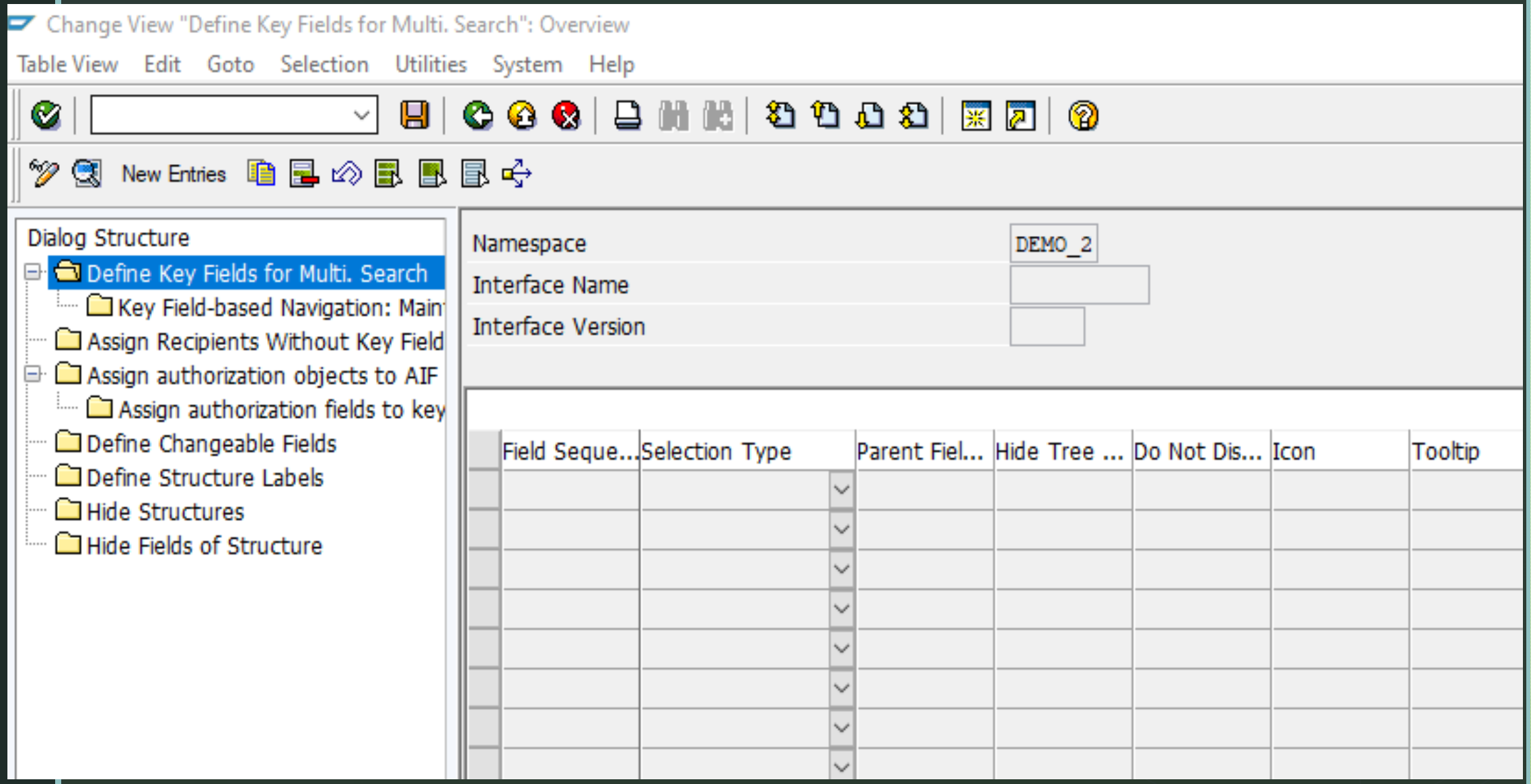
It’s recommended to implement an interface-specific single index table to ensure full flexibility, especially if you expect a high load of messages or if you plan to define key fields for your interface (now or later).

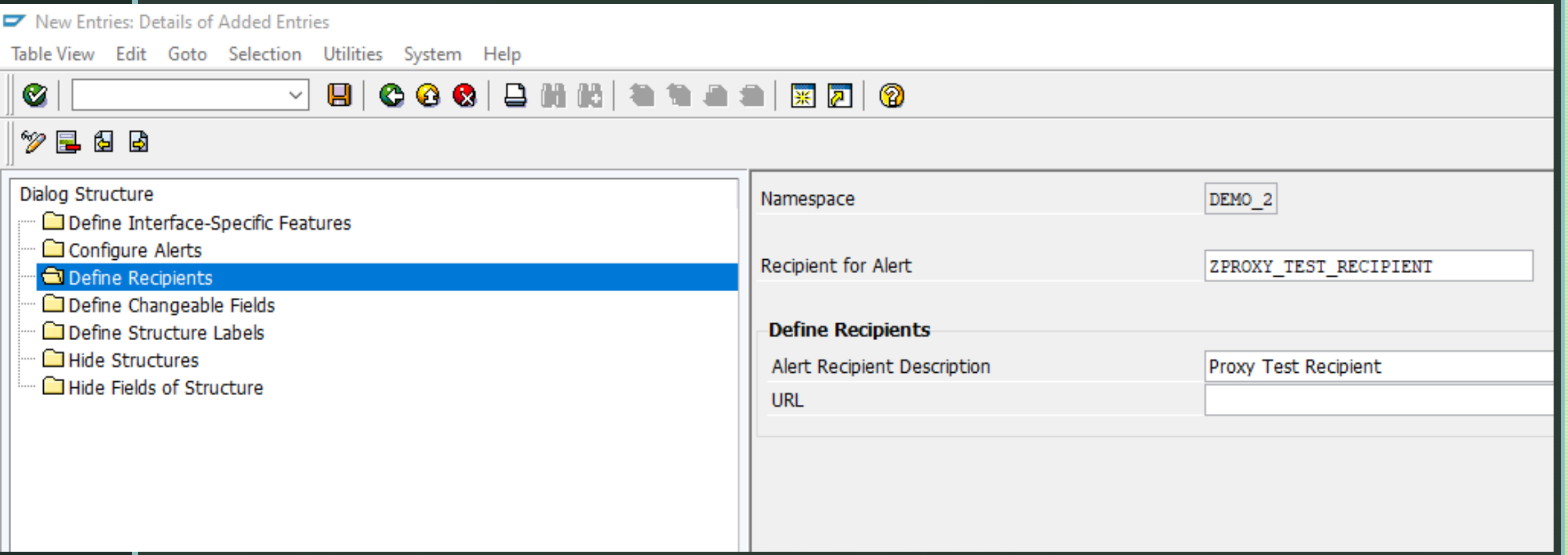
1.Create a table via transaction SE11. You can use table /AIF/STD_IDX_TBL as a template by entering **/AIF/STD_IDX_TBL** in the field **Database table**, right-clicking it and selecting **Copy…**. Enter the name **ZFLBOOK_MON_IDX** for the new table and select **Continue**. When prompted, enter package **ZDEMO**, which you created earlier.
2.After creating the single index table, activate it by selecting **Display** and then **Activate**.


Dictionary: Change Table
Table Edit Goto Utilities Extras Environment System Help
Technical Settings
Append Structure...
Transparent Table ZFLBOOK_MON_IDX Active
Short Description Standard index table
Attributes Delivery and Maintenance Fields Input Help/Check Currency/Quantity Fields Indexes
Srch Help
Built-In Type
1 / 41
Field Key Initi... Data element Data Type Length Decim... Coordinate Short Description Group
MANDT MANDT CLNT 3 0 0 Client
MSGGUID GUID_32 CHAR 32 0 0 GUID in 'CHAR' Format in Uppercase
.INCLUDE /AIF/IFKEYS STRU 0 0 0 Keys AIFKEYS
NS /AIF/NS CHAR 6 0 0 Namespace
IFNAME /AIF/IFNAME CHAR 10 0 0 Interface Name
IFVER /AIF/IFVERSION CHAR 5 0 0 Interface Version
.INCLUDE /AIF/ADMIN STRU 0 0 0 Additional information structure (be appent to tran. table) ADMIN
.INCLUDE /AIF/ADDITIONAL_ STRU 0 0 0 Additional information structure (be appent to tran. table) ADD_1
NUMBEROFABORTS /AIF/NUMBEROFABO INT4 10 0 0 Abort Number
NUMBEROFERRORS /AIF/NUMBEROFERR INT4 10 0 0 Error Number
NUMBEROFWARNINGS /AIF/NUMBEROFWAR INT4 10 0 0 Warning Number
NUMBEROFSUCCESS /AIF/NUMBEROFSUC INT4 10 0 0 Success Number
NUMBEROFINFOS /AIF/NUMBEROFINF INT4 10 0 0 Information Number
.INCLUDE /AIF/ADDITIONAL_ STRU 0 0 0 Additional information structure (be appent to tran. table) ADD_2
NSRECIP /AIF/NS CHAR 6 0 0 Namespace
RECIPIENT /AIF/ALRT_REC CHAR 30 0 0 Alert Management Recipient
STATUS /AIF/PROC_STATUS CHAR 1 0 0 Processing Status


Display IMG
Implementation Activities
Edit
Goto
Additional Information
Existing BC Sets
BC Sets for Activity
Structure
SAP Application Interface Framework
Interface Development
Error Handling
Define Applications
Define Global Features
Define Namespace-Specific Features
Define Interface-Specific Features
Custom Extensions
System Configuration

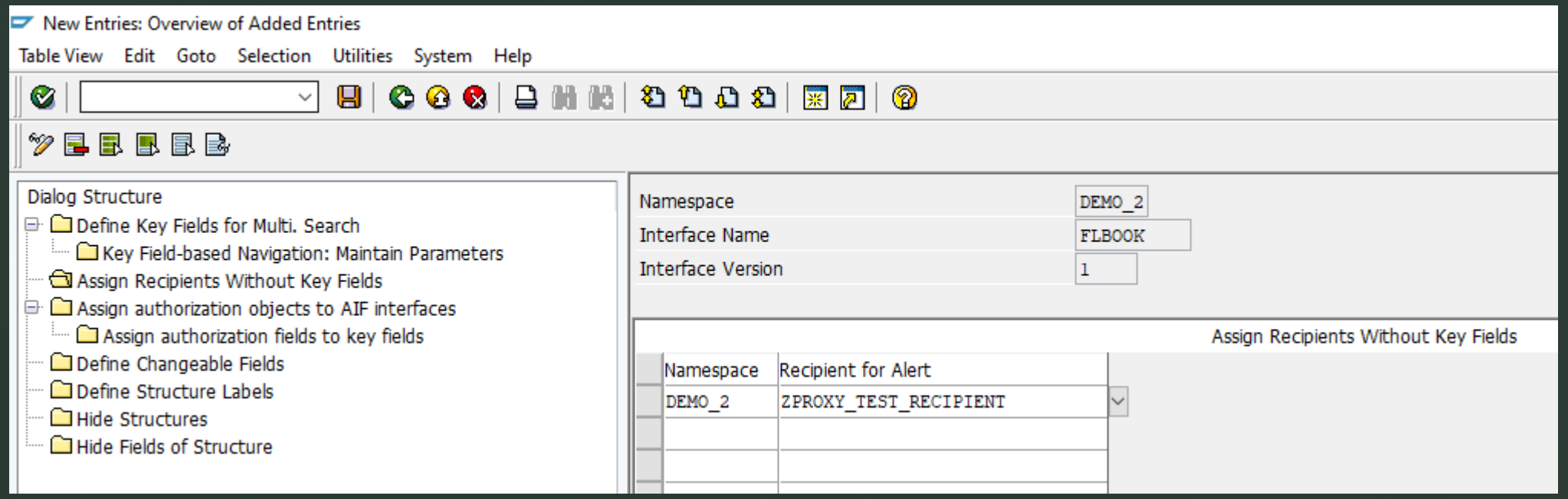

Structure
SAP Application Interface Framework
Determine Work Area: Entry
Field Name
Work Area
Namespace
emo_2
Interface Name
Interface Version
Further select cond.
Append

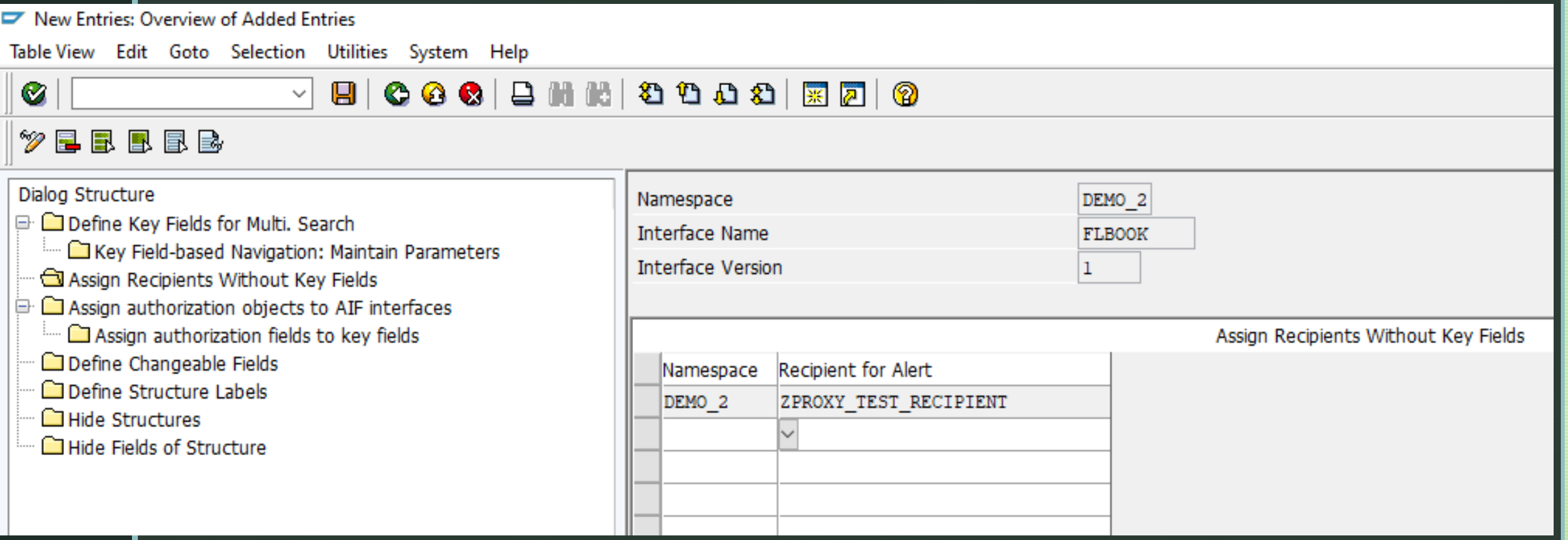

Change View "Define Key Fields for Multi. Search": Overview
Table View
Edit
Goto
Selection
Utilities
System
Help
New Entries
Dialog Structure
Define Key Fields for Multi. Search
Key Field-based Navigation: Main
Assign Recipients Without Key Field
Assign authorization objects to AIF
Assign authorization fields to key
Define Changeable Fields
Define Structure Labels
Hide Structures
Hide Fields of Structure
Namespace
DEMO_2
Interface Name
Interface Version
Field Seque...
Selection Type
Parent Fiel...
Hide Tree ...
Do Not Dis...
Icon
Tooltip


New Entries: Details of Added Entries
Table View
Edit
Goto
Selection
Utilities
System
Help
Dialog Structure
Define Interface-Specific Features
Configure Alerts
Define Recipients
Define Changeable Fields
Define Structure Labels
Hide Structures
Hide Fields of Structure
Namespace
DEMO_2
Recipient for Alert
ZPROXY_TEST_RECIPIENT
Define Recipients
Alert Recipient Description
Proxy Test Recipient
URL

ASSIGN RECIPIENT TO INTERFACE

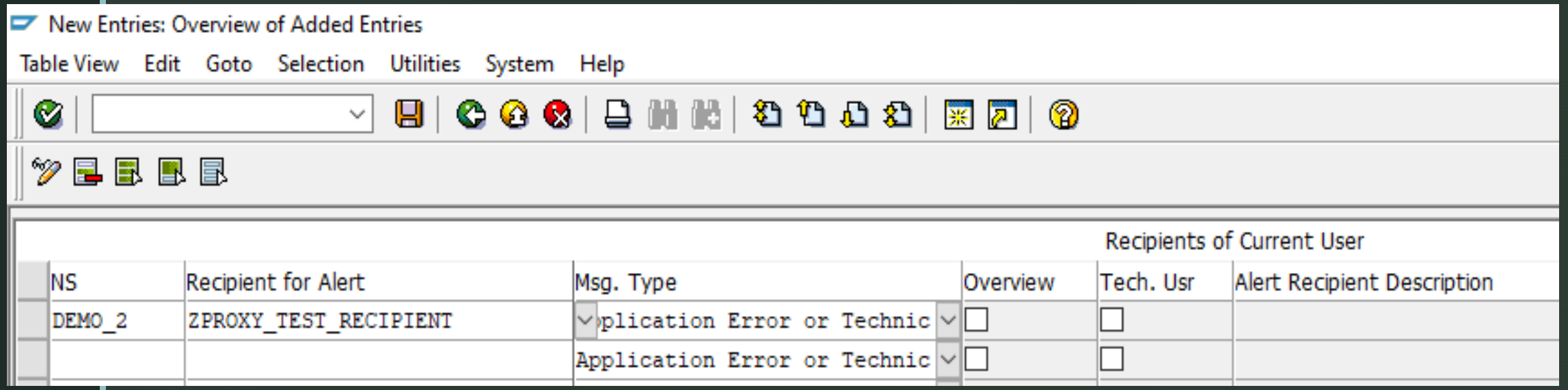
To be able to see any data in the Interface Monitor or the Message Dashboard, a recipient must be assigned to the interface you want to monitor.
Go to **Customizing** (transaction code /AIF/CUST) and navigate to **SAP Application Interface Framework** > **Error Handling** > **Define Interface-Specific Features**. Enter or select your namespace **DEMO_2**, as well as your interface name **FLBOOK** and interface version **1**. Select **Continue**.
In the menu on the left, double-click **Assign Recipients Without Key Fields** and create a new entry. Enter or select the namespace and the recipient you created before.


New Entries: Overview of Added Entries
Table View
Edit
Goto
Selection
Utilities
System
Help
Dialog Structure
Define Key Fields for Multi. Search
Key Field-based Navigation: Maintain Parameters
Assign Recipients Without Key Fields
Assign authorization objects to AIF interfaces
Assign authorization fields to key fields
Define Changeable Fields
Define Structure Labels
Hide Structures
Hide Fields of Structure
Namespace
DEMO_2
Interface Name
FLBOOK
Interface Version
1
Assign Recipients Without Key Fields
Namespace
Recipient for Alert
DEMO_2
ZPROXY_TEST_RECIPIENT


New Entries: Overview of Added Entries
Table View
Edit
Goto
Selection
Utilities
System
Help
Dialog Structure
Define Key Fields for Multi. Search
Key Field-based Navigation: Maintain Parameters
Assign Recipients Without Key Fields
Assign authorization objects to AIF interfaces
Assign authorization fields to key fields
Define Changeable Fields
Define Structure Labels
Hide Structures
Hide Fields of Structure
Namespace
DEMO_2
Interface Name
FLBOOK
Interface Version
1
Assign Recipients Without Key Fields
Namespace
Recipient for Alert
DEMO_2
ZPROXY_TEST_RECIPIENT

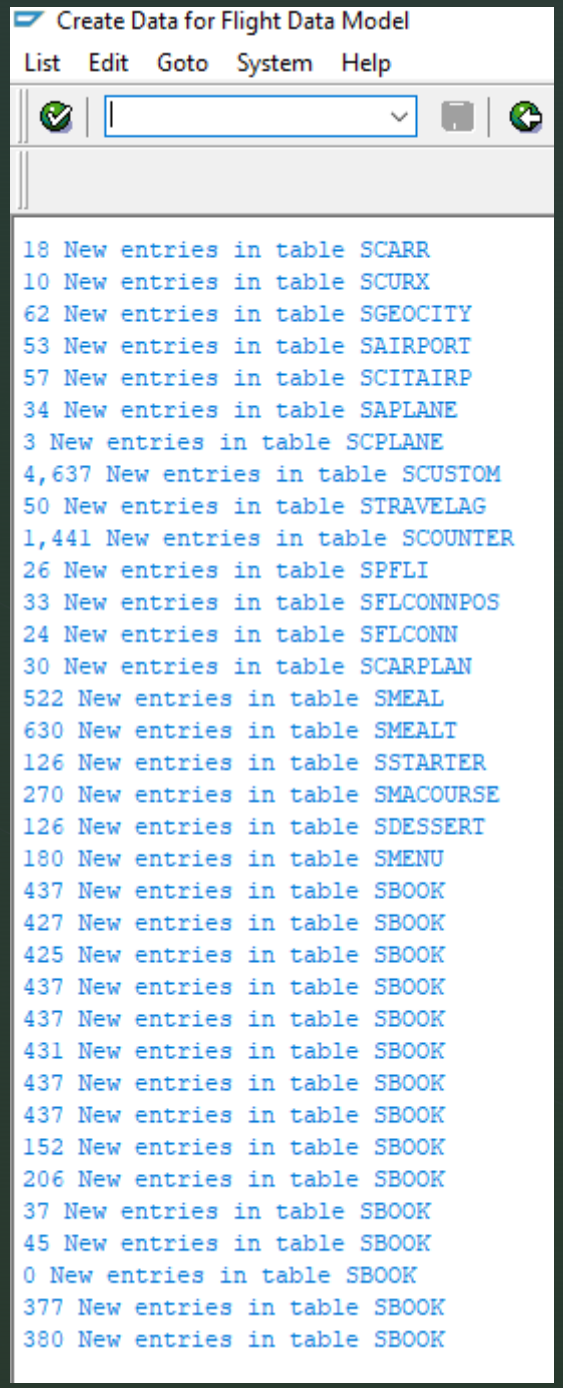
ASSIGN USERS TO RECIPIENT

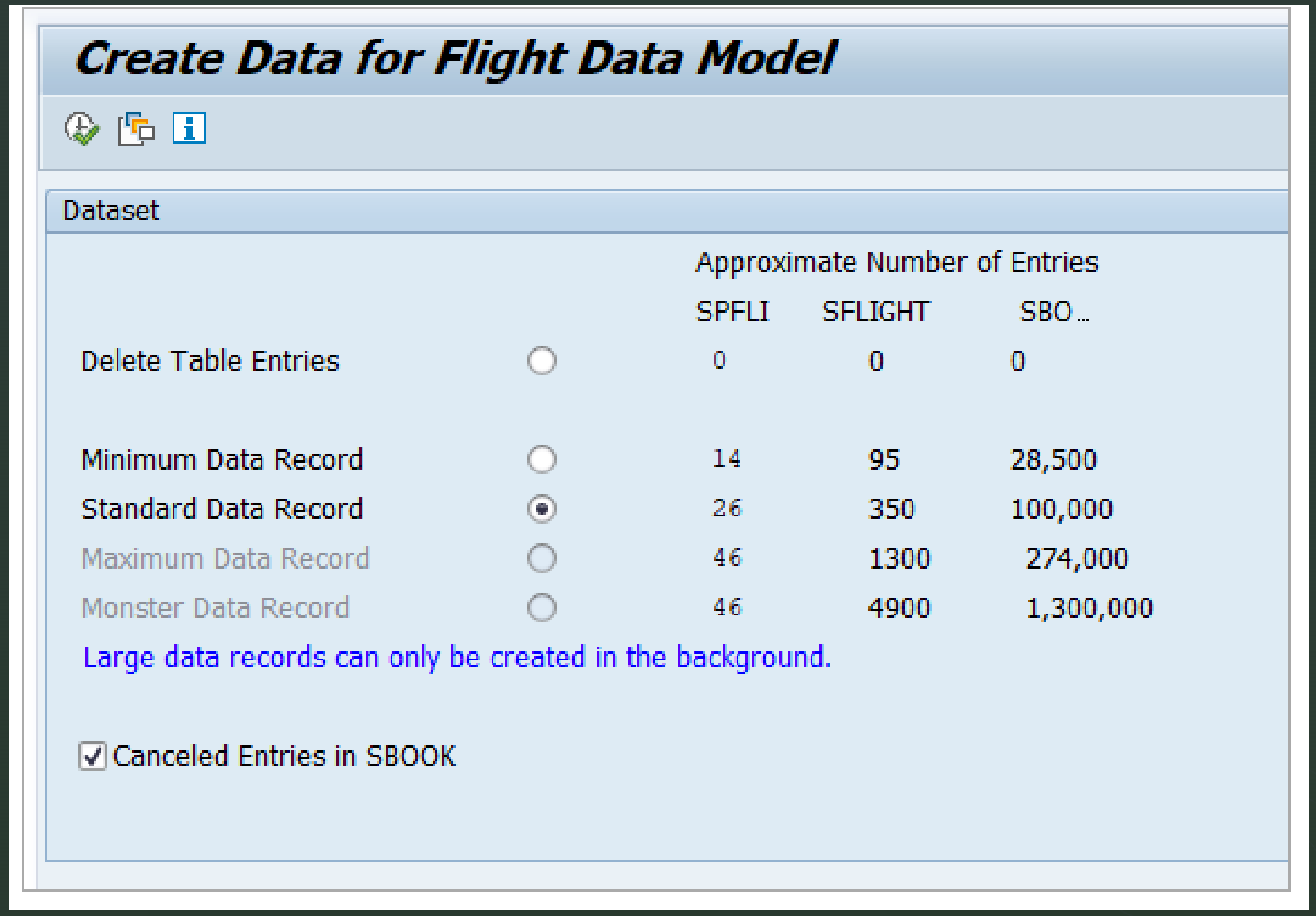
Now the users in charge of monitoring the proxy must be assigned to the recipient.
Run transaction /AIF/MYRECIPIENTS and create a new entry. Select the namespace **DEMO_2** and recipient ZPROXY_TEST_RECIPIENT you created in the steps before. Check the boxes for **Overview** and **Technical User**.


New Entries: Overview of Added Entries
Table View
Edit
Goto
Selection
Utilities
System
Help
Recipients of Current User
NS
Recipient for Alert
Msg. Type
Overview
Tech. Usr
Alert Recipient Description
DEMO_2
ZPROXY_TEST_RECIPIENT
plication Error or Technic
Application Error or Technic

CREATE TEST DATA

Before you can create flight bookings, you need to generate test data. To do so, run transaction BC_DATA_GEN, select the standard data record, and execute the report.


Create Data for Flight Data Model
List Edit Goto System Help
18 New entries in table SCARR
10 New entries in table SCURX
62 New entries in table SGEOCITY
53 New entries in table SAIRPORT
57 New entries in table SCITAIRP
34 New entries in table SAPLANE
3 New entries in table SCPLANE
4,637 New entries in table SCUSTOM
50 New entries in table STRAVELAG
1,441 New entries in table SCOUNTER
26 New entries in table SPFLI
33 New entries in table SFLCONNPOS
24 New entries in table SFLCONN
30 New entries in table SCARPLAN
522 New entries in table SMEAL
630 New entries in table SMEALT
126 New entries in table SSTARTER
270 New entries in table SMACOURSE
126 New entries in table SDESSERT
180 New entries in table SMENU
437 New entries in table SBOOK
427 New entries in table SBOOK
425 New entries in table SBOOK
437 New entries in table SBOOK
437 New entries in table SBOOK
431 New entries in table SBOOK
437 New entries in table SBOOK
437 New entries in table SBOOK
152 New entries in table SBOOK
206 New entries in table SBOOK
37 New entries in table SBOOK
45 New entries in table SBOOK
0 New entries in table SBOOK
377 New entries in table SBOOK
380 New entries in table SBOOK


Create Data for Flight Data Model
Dataset
Approximate Number of Entries
SPFLI
SFLIGHT
SBO…
Delete Table Entries
0
0
0
Minimum Data Record
14
95
28,500
Standard Data Record
26
350
100,000
Maximum Data Record
46
1300
274,000
Monster Data Record
46
4900
1,300,000
Large data records can only be created in the background.
Canceled Entries in SBOOK

TO BE CONTINUED