

# Creating Interfaces for the SAP Application Interface Framework with Service Implementation Workbench



## Applies to:

SAP Application Interface Framework 2.0 with AIFX component installed

## Summary

This document describes how an interface for the SAP Application Interface Framework can be generated with the Service Implementation Workbench (SIW). An example for an interface creating flight bookings will be shown.

**Author:** Verena Wörner

**Company:** SAP AG

**Created on:** 27 November 2012

## Author Bio

Verena Wörner is a developer for SAP Custom Development. She is a member of the development team for the SAP Application Interface Framework

## Table of Contents

Table of Contents .....	2
Prerequisites .....	3
Create Interface with SIW .....	3
Complete Customizing in SAP Application Interface Framework.....	7
Simple 1: 1 Field Mappings.....	7
Complex Field Mappings.....	7
Value Mappings .....	8
Fix Values .....	8
Interface Specific Index Table and Key Fields.....	9
Test Flight Booking Interface .....	10
Related Content.....	11
Copyright.....	12

## Prerequisites

Before you can start to generate your interface with the Service Implementation Workbench (SIW), you have to create a service interface. Furthermore, you need to generate the proxy in your application system and activate it.

This document will use an example interface that will create flight bookings. The example will use a services interface that looks like the service interface displayed in Figure 1. A proxy class for the service interface should be generated and activated. Furthermore, you should check that the implementation of the class' method is empty and activated.

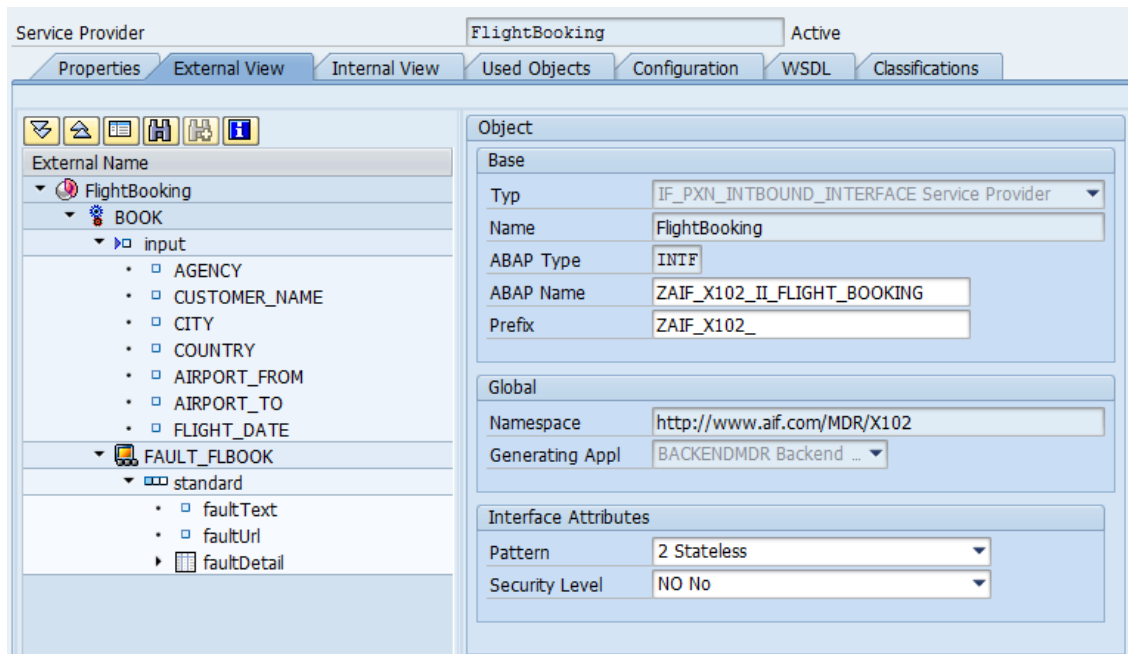


Figure 1: Service Interface for Example with SIW

## Create Interface with SIW

A template for the Service Implementation Workbench (SIW) is delivered with the SAP Application Interface Framework. Following objects can be generated with the SIW:

- Proxy class implementation
- AIF interface definition
- SAP data structure
- Structure mapping: A mapping between source root structure and destination root structure will already be created. Further mappings have to be created manually.
- Action and action function
- Action assignment in structure mapping
- Recipient
- A default interface specific selection screen which can be implemented in order to select key fields

In order to create an AIF interface with SIW, call transaction SIW.

After clicking *Create Project* a popup is displayed. You have to enter a *Project ID* and a description for you project. Furthermore, you have to enter a *Configuration ID*. The configuration contains the construction plan and defines which objects should be generated. In order to create an AIF interface insert *Configuration ID* /AIFX/DEFAULT\_WO\_ESR.

On the following *Create Project* screen you have to specify the service interface. Therefore, two options exist. The first option is to specify it by service interface name. In this case you have to enter the *Software Component*, the *Version*, the *Namespace*, and the *Service Interface*. The second option is to specify the service interface by proxy name. In this case you have to maintain *Namespace*, *Direction* and *Proxy Name*. Furthermore, you have to specify the service operation.

On the last subscreen *Specify Application Interface Framework Attributes* you have to maintain the attributes needed for the generation for the SAP Application Interface Framework specific objects. Following fields have to be filled.

- *Prefix*: Used for all repository objects generated with the SIW
- *Package*: The package storing generated objects. Alternatively, you can set the Local Object indicator.
- *Function group*: Contains the generated action function module
- *Namespace / interface name / interface version*: The AIF interface keys. The namespace is not generated by SIW, but must be defined in advance.
- *Interface description*
- *Action*: The AIF action used by the AIF interface. The generated action function module is assigned as a function of this action.
- *Action description*

Figure 2 shows how the screen for the flight booking example could look like.

Specify Application Interface Framework Attributes	
Prefix	ZX102
Package	ZAIF_TEST_X102 <input type="checkbox"/> Local Object
Function group	ZAIF_X102_TEST
Namespace	X102_6
Interface Name	FLBOOKING
Interface Version	1
Description	Example to create flight bookings
Action	FLBOOKING_CREATE
Action Description	Create flight booking

**Figure 2: SIW Attributes for Flight Booking Example**

On the *Project Implementation Screen* all objects that will be generated will be displayed. In the first section *Interface Definition and Structure Mapping* the objects related to the AIF interface will be displayed. Some of the values can still be changed. Following AIF attributes are displayed:

- **PREFIX**: Used for all repository objects generated with the SIW
- **API\_NAME\_FOR\_STRUCTURE**: You have to enter the name of a BAPI or any other function module. This BAPI/function module will be called in the action function module. Furthermore, the SAP structure will be generated depending on the BAPIs/function modules parameters. **Note**: The function module's parameters should not have the type of a generated proxy structure (or contain a generated proxy structure anywhere as component type), since it will not be possible to activate the SAP structure in this case.
- **SAP\_STRUCTURE**: Defines the Name of the SAP data structure that will be generated.
- **STRUCTURE\_DESCRIPTION**: will be used as short description for the SAP data structure that will be generated

- NS: Namespace where the AIF interface will be created.
- IFNAME: Name of the AIF interface that will be created
- IFVERSION: Version of the AIF interface that will be created
- INTERFACE\_DESCRIPTION: Description of the AIF interface
- IFACTION: Name of the Action that will be created. The action will be assigned in structure mapping of the interface that will be created.
- ACTDESC: Description of the action that will be created
- ACT\_FUNCTION: Name of the action function module that will be generated. Within this function module the BAPI specified in API\_NAME\_FOR\_STRUCTURE will be called. Furthermore, the action function will be assigned to the action specified in IFACTION.
- PACKAGE\_NAME: Defines the package where the repository objects will be stored.
- RECIPIENT\_NS: Namespace the generated recipient is assigned to
- RECIPIENT\_ALERT: Name of the recipient that will be created. The recipient will be assigned to the interface. After assigning users to the recipient, the users will be able to see the messages in Interface Monitor and receive the alert messages in their in inbox.
- SCREEN\_NUMBER: Defines the number of the selection screen that can be used if you want to have an interface specific selection screen.
- REPORT\_NAME: Defines the name of the report where the interface specific selection screen can be implemented.
- ECH: Defines if the AIF interface should also support ECH functionality.

The second section *Proxy and Service Repository Data* displays the data related to the selected service interface.

After you have entered the BAPI into *API\_NAME\_FOR\_STRUCTURE* field and checked the suggested names of all other objects, you can click *Generate*. In order to be able to generate the interface for flight bookings you have to insert *BAPI\_FLBOOKING\_CREATEFROMDATA* into *API\_NAME\_FOR\_STRUCTURE*.

After the generation you can check the generated objects on the *Objects* tab of the *Change Project Implementation* screen. If you click on an object name, you will be forwarded to the generated repository object. For example, if you click on the implementation of the proxy method the call to the AIF should be implemented (see Figure 3).

```

METHOD ZAIF_X102_II_FLIGHT_BOOKING~BOOK.
 * This is generated via SIW on 20120817 / 114406 by ██████████
 * Other manually entered code will be removed during re-generation.
 /aif/cl_enabler_proxy=>process_message(
     is_input = input
     iv_exception_classname = 'ZAIF_X102_CX_FAULT_FLBOOK' ).
ENDMETHOD.
"if siw configuration setup~setup construction plan

```

**Figure 3: Implementation of a Proxy Method after Generation with SIW**

If you click on the object name of MODULE\_CREATE the implementation of the interface-specific selection screen is displayed. The program contains two code slots. Code implemented within this code slots will not be overwritten after a re-generation of the project with SIW.

In order to select for interface specific key fields an interface specific index table is needed. In order to create an index table you can copy table /AIF/STD\_IDX\_TBL and add your key fields (for more information refer to the [Cookbook](#)). For example, an index table for the flight booking example could look as follows:

Component	Key	Data element	Group
MANDT	X	MANDT	
MSGGUID	X	GUID_32	
.INCLUDE		/AIF/IFKEYS	AIFKEYS
.INCLUDE		/AIF/ADMIN	ADMIN
PID		SXMSPID	
AIRPORT_FROM		S_AIRPORT	
AIRPORT_TO		S_AIRPORT	

After you have created an interface specific index table you can continue to implement the generated program containing the selection screen. As mentioned before the program contains two code slots. Within the first code slot you can specify your single index table, multi index table and, document index table. Within the second code slot you have to define the parameters and select-option that will be displayed on the interface specific-selection screen. Below the implementation of the program for the flight boeing example is displayed.

```
PROGRAM zx102_rep_x102_6_flbooking.

TABLES: /aif/std_idx_tbl.
* code slot 1: additional tables
TABLES: zaif_flbook_idx.
* end code slot 1

SELECTION-SCREEN BEGIN OF SCREEN 0001 AS SUBSCREEN.
* code slot 2: key fields
SELECT-OPTIONS: s_from FOR zaif_flbook_idx-airport_from,
                s_to   FOR zaif_flbook_idx-airport_to.
* end code slot 2

SELECTION-SCREEN END OF SCREEN 0001.

AT SELECTION-SCREEN OUTPUT.
/aif/cl_global_tools=>get_value_from_mem( ).
```

## Complete Customizing in SAP Application Interface Framework

Now the necessary mappings have to be created. Therefore, go to Customizing for the SAP Application Interface Framework (transaction /AIF/CUST). In Customizing activity *Define Structure Mappings* the mapping from the source root structure to the destination root structure has already been created. Furthermore, the action was already assigned. However, you might have to define further structure mappings. Additionally, field mappings have to be created. If you want to use checks, fix values and value mappings you have to create them as well.

In the following sections the customizing like field mapping, value mappings and interface key fields needed for the flight booking example are described. You can find more information on how to set up an AIF interface in the [cookbook](#).

### Simple 1: 1 Field Mappings

Field in Destination Structure	Fieldname 1
BOOKING_DATA-AGENCYNUM	AGENCY
BOOKING_DATA-FLIGHTDATE	FLIGHT_DATE

### Complex Field Mappings

Field	Value
Field in Destination Structure	BOOKING_DATA-AIRLINEID
Fieldname 1	AIRPORT_FROM
Fieldname 2	AIRPORT_TO
Fieldname 3	FLIGHT_DATE
Namespace	X102_6
Value Mapping	FLIGHT_AIRLINE

Field	Value
Field in Destination Structure	BOOKING_DATA-CONNECTID
Fieldname 1	AIRPORT_FROM
Fieldname 2	AIRPORT_TO
Fieldname 3	FLIGHT_DATE
Fieldname 4	@BOOKING_DATA-AIRLINEID
Namespace	X102_6
Value Mapping	FLIGHT_CONNECTION

Field	Value
Field in Destination Structure	BOOKING_DATA-CUSTOMERID
Fieldname 1	CUSTOMER_NAME
Namespace	X102_6
Value Mapping	CUSTOMERID_BY_NAME

**Note:** To create the value mappings insert the value mapping's name and press enter. A popup will ask you if you want to create the value mapping. Afterwards you can double click the value mapping and maintain the value mapping.

**Value Mappings****Value Mapping: FLIGHT\_AIRLINE**

Field	Value
Table Name	SFLIGHTS
Field Name	CARRID
Where Condition for Select Statement	AIRPFROM = '\$1' AND AIRPTO = '\$2' AND FLDATE = '\$3'
Do Not Buffer SELECT Statement	X

**Value Mapping: FLIGHT\_CONNECTION**

Field	Value
Table Name	SFLIGHTS
Field Name	CONNID
Where Condition for Select Statement	AIRPFROM = '\$1' AND AIRPTO = '\$2' AND FLDATE = '\$3' AND CARRID = '\$4'
Do Not Buffer SELECT Statement	X

**Value Mapping: CUSTOMER\_ID\_BY\_NAME**

Field	Value
Table Name	SCUSTOM
Field Name	ID
Where Condition for Select Statement	NAME = '\$1' AND CUSTTYPE = 'P'
Do Not Buffer SELECT Statement	X

**Fix Values**

For the class a fix value will be used. Enter Field Name BOOKING\_DATA-CLASS and enter value Y.



## Interface Specific Index Table and Key Fields

After the structure mapping is completed, you can define the interface specific key fields. Therefore, the interface's index table needs to be maintained. In Customizing for the SAP Application Interface Framework select *Error Handling* → *Namespace-Specific Features*. Maintain your index table in *Message Index Table* field. Furthermore, you have to maintain the program and screen number that contains your interface-specific selection screen in *Program Name* and *Screen number* fields.

Furthermore, you have to maintain the key fields in *Error Handling* → *Interface-Specific Features*. For the flight booking example following key fields can be maintained:

### Key Field AIRPORT\_FROM

Field	Value
Field Sequence Number	10
Key Field Name	AIRPORT_FROM
Data element	S_AIRPORT
Name Select-Options/Parameter	S_FROM
Field Is Select-Option	X
Field Name	AIRPORT_FROM
Hide Tree Node in View 1 Tree	X

### Key Field AIRPORT\_TO

Field	Value
Field Sequence Number	20
Key Field Name	AIRPORT_TO
Data element	S_AIRPORT
Name Select-Options/Parameter	S_TO
Field Is Select-Option	X
Field Name	AIRPORT_TO
Hide Tree Node in View 1 Tree	X

Optionally, you can assign the recipient created by the SIW to the users responsible for your interface. This will enable the users to see the interface in *Interface Monitor* and to receive email notifications in case of errors.

Furthermore, you should check which engines are assigned to your interface. Which engines will be used for the interface depends on your scenario. In most cases you can probably use the default engines when you created your interface with SIW. You can maintain an interface's engines in Customizing for the SAP Application Interface Framework in *Interface Development* → *Additional Interface Settings* → *Specify Interface Engines*. In order to be able to test the flight booking example interface *Application Engine XML* and *Persistence Engine XML* should be selected.

## Test Flight Booking Interface

In order to test the flight booking example interface you can use transaction /AIF/PERS\_TEST. Fill the selection screen according to your interface:

- Source Structure: Enter the name of the raw data structure of your interface
- Recordtype Name: Enter the record type of your interface

For all other fields you can use the default values.

Assuming that your user is assigned to the created recipient you should see the messages in *Interface Monitor (/AIF/IFMON)*. Otherwise, you can start *Monitoring and Error Handling* directly (transaction /AIF/ERR). Select your interface. If you have maintained an interface-specific selections screen, this screen should be displayed. You can use the selection screen to refine the selection. Furthermore, you should select all statuses. Depending on your selection, the processed messages should be displayed.

**Note:** If you do not have any valid flights you can generate data for the flight model by executing transaction BC\_DATA\_GEN.

## **Related Content**

[SAP Application Interface Framework – Cookbook](#)

[Service Implementation Workbench Support for AIF](#)

[Service Implementation Workbench](#)

## Copyright

© Copyright 2012 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Oracle Corporation.

JavaScript is a registered trademark of Oracle Corporation, used under license for technology invented and implemented by Netscape.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects S.A. in the United States and in other countries. Business Objects is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.