

**SAP** PRESS

**SAP**  
for Utilities

# SAP

# Financial Services

# Network Connector

**POWERED BY SAP HANA**

**SAP** S/4 HANA

**A Business and Technical  
Roadmap to Deploying SAP**

Copyright(c) 2018 by Nixon Vunganai.

All rights reserved.

Neither this document nor any part of it may be copied or reproduced in any form or by any means or translated into another language, without the prior consent of Nixon Vunganai. The information contained in this document is subject to change without notice.

WHATSAPP +255738656506

---

	<b>0</b>
<b>INTRODUCTION</b>	<b>3</b>
<b>FINANCIAL SERVICES NETWORK CONNECTOR</b>	<b>5</b>
<b>MAINTAIN NUMBER RANGE FOR MESSAGE ID</b>	<b>5</b>
<b>MAINTAIN SSF APPLICATION PARAMETERS</b>	<b>6</b>
<b>MAINTAIN SECURE STORE AND FORWARD (SSF) PROFILE DATA</b>	<b>6</b>
<b>MAINTAIN SELECTION VARIANTS FOR BANK STATEMENTS</b>	<b>7</b>
<b>MAINTAIN RESIDENCE TIME FOR MESSAGE ARCHIVING</b>	<b>7</b>
<b>MAINTAIN CUSTOM SENDER/RECEIVER IDS</b>	<b>8</b>
<b>MAINTAIN ALTERNATIVE LOGICAL PORTS FOR OUTBOUND MESSAGES</b>	<b>8</b>
<b>BUSINESS ADD-INS</b>	<b>9</b>
SETTINGS FOR MESSAGE PROCESSING	9
BAdI: Inbound Message Processing	9
BAdI: Outbound Message Processing	10
SETTINGS FOR DETERMINATION OF LOGICAL PORT	12
Notes on Implementation	12
BAdI: Determination of Logical Port for Outbound Interface	12
BAdI: Determination of Logical Port for Synchronous Interface	13

# INTRODUCTION

Welcome to the fascinating world of SAP. This book helps you crack the tricks of mastering SAP HANA Customization

WHATSAPP [+255738656506](https://wa.me/91255738656506)



## Financial Services Network Connector

### Maintain Number Range for Message ID

#### Use

In this Customizing activity, you maintain the number range for Message IDs. The system uses this number range to generate a unique message ID when a message ID is not provided to the system during message creation.

#### Activities

To maintain the number range, choose the change *Intervals* pushbutton.

## Maintain SSF Application Parameters

### Use

In this Customizing activity, you maintain the parameters for the secure store and forward (SSF) applications used by the connector for the SAP Financial Services Network.

### Activities

For its security features, the connector makes use of two SSF applications. Depending on whether you plan to use payload security and/or tamper protection, you need to maintain the parameters for the applications BSNAGT and/or FSNCFP.

Application BSNAGT is used by the connector's payload security. This means that these parameters define how the message content of an FSNMessage is encrypted/signed and decrypted/verified.

Application FSNCFP is used by the connector's tamper protection functions. This means that these parameters define how, for example, the payment transaction fingerprints are signed.

## Maintain Secure Store and Forward (SSF) Profile Data

### Use

In this Customizing activity, you maintain the secure store and forward (SSF) profile data in the view /BSNAGT/V\_SSF.

### Requirements

You have configured the applications in Customizing under *Financial Services Network Connector -> Maintain SSF Application Parameters* (transaction SSFA).

You need to maintain the security parameters for the following applications, depending on whether you are planning to use payload security features and/or tamper protection security features:

- BSNAGT (payload security)
- FSNCFP (tamper protection)

### Activities

You specify whether you want to sign and/or encrypt outbound messages, and whether you want to decrypt and verify inbound messages. You configure these settings for a *Message Type*, a *SenderID/ReceiverID*, and an *Application*.

*Message Type* and *SenderID/ReceiverID* are optional. If you do not specify a *Message Type*, the settings you enter apply to all message types. Therefore, you can configure default settings for all messages, and then override those settings for a specific message type.

Similarly, if you do not enter a *SenderID/ReceiverID*, the settings you enter apply to all senders/receivers. As for message types, you can configure default settings for all senders/receivers, and then override those settings for a specific sender or receiver.

If you select one or more of the security options, you must also enter the *Signatory/Recipient Name*. If no security options are selected, it is not necessary to enter the *Signatory/Recipient Name*.

To use tamper protection functions, you maintain an entry with application FSNCFP. When you maintain such an entry, the tamper protection functions are switched on for the corresponding combinations of *Message Types* and *Receiver IDs*. Note that tamper protection functions only exist for selected *Message Types* (for example, pain.001.001.03). When sending a message with such a *Message Type* and *Receiver ID*, the system creates one or more fingerprints and includes them into the extended header of the outgoing FSN Message.

## Maintain Selection Variants for Bank Statements

### Use

In this Customizing activity, you maintain the *Variant* for a *SenderID/ReceiverID* and *Message Type* in the view BSNAGT/V\_VARI. The system processes incoming messages according to the message type, which is specified in the Web service interface. For MT940 (bank statement), MX camt.053 and MT942 (account report), or MX camt.052, the report RFEBKA00 (transaction FF.5) is executed with the relevant variant, as configured in this Customizing activity.

### Requirements

You have saved the variant in transaction FF.5.

## Maintain Residence Time for Message Archiving

### Use

In this Customizing activity, you maintain the residence time for each message type in the view /BSNAGT/V\_ARC\_CU. The residence time defines the length of time messages are stored before becoming eligible for archiving. The fields *Message Type*, *Residence*, and *Unit* are all required.



## Requirements

You have defined the message types in the table /BSNAGT/FILE\_INF.

## Maintain Custom Sender/Receiver IDs

### Use

In this Customizing activity you can define a custom *Sender ID* for a *Message Type* and/or *Receiver ID*. For example, you could define different *Sender IDs* for communicating with different banks.

These custom *Sender IDs* are used by the connector for the SAP Financial Services Network whenever a message is created. For payment messages created via Payment Medium Workbench, the system uses the *Customer Number* maintained at House Bank level as the *Sender ID* (the corresponding custom *Sender ID* entries are ignored). However, the system uses the custom *Sender ID* maintained in this Customizing activity as a fallback, and it is used in the event that the *Customer Number* has not been maintained at House Bank level.

You can leave the *Message Type* and/or *Receiver ID* blank. The system treats such entries as relevant for all *Message Types* and/or *Receiver IDs*. In the event of multiple valid entries, the system uses the most specific one.

When you are planning to manually create FSN Messages by uploading files to the connector from your local computer (the *Pick-Up Files* report), you might consider maintaining the exact *Sender IDs* and *Receiver IDs*, because the value helps from the *Pick-Up Files* report also use these Customizing settings.

## Maintain Alternative Logical Ports for Outbound Messages

### Use

In this Customizing activity, you can optionally define the alternative logical port to be used for outbound messages. You use this Customizing activity in cases where the default logical port, which is defined in transaction SOAMANAGER, should not be used. You enter the logical port for each *Receiver ID*.

When sending messages, the system checks this table. If an entry is found for the current *Receiver ID*, then the system uses the corresponding logical port to send the message. Otherwise, the default port defined in transaction SOAMANAGER is used.

## Business Add-Ins

### Settings for Message Processing

#### BAdI: Inbound Message Processing

##### Use

This Business Add-In (BAdI) can be used to enhance the capabilities of the connector for the SAP Financial Services Network. You can use this BAdI to enhance the supported message types or enhance how message types are handled.

A BAdI implementation can be assigned for each message type. The implementing class contains several methods that are called by the connector during the inbound handling for displaying the content of a message, or for specific requirements that relate to how file names are parsed by the pick-up report.

The standard implementations are inherited from a default implementation, which is always called when a specific implementation for the given message type is not found.

Implementations that are added for specific message types should inherit from the default implementation too, and you should only redefine the methods that are required to work differently to the methods in the default implementation.

This BAdI has the following methods:

- EXECUTE
- MAP\_DATA
- SET\_DATA
- GET\_DATA
- UPDATE\_STATUS
- DISPLAY\_MESSAGE\_CONTENT
- NAVI\_TO\_RELATBUSOBJ
- FIND\_PARENT\_MESSAGE
- REPROCESS
- DETERMINE\_HEADER\_FIELDS

##### Standard settings

For more information about the standard settings (filters, single or multiple uses), see the *Enhancement Spot Element Definitions* tab in the BAdI Builder (transaction SE18).

##### BAdI Implementations

- /BSNAGT/BADI\_MESSAGE\_DEFAULT  
Default BAdI Implementation for inbound messages.
- /BSNAGT/BADI\_IDOC\_PAYMENT  
BAdI Implementation for IDOC payments.
- /BSNAGT/BADI\_MESSAGE\_BULK  
BAdI Implementation for bulk messages.
- /BSNAGT/BADI\_MESSAGE\_PING\_TEST  
BAdI Implementation for ping test messages.
- /BSNAGT/BADI\_MESSAGE\_STATUS  
BAdI Implementation for status messages.
- /BSNAGT/BADI\_MESSAGE\_STM  
BAdI Implementation for statement messages.
- /BSNAGT/EHI\_FREE\_TEXT  
BAdI Implementation for free text messages.
- /BSNAGT/EHI\_PAYMENT  
BAdI Implementation for payment messages.

**See also**

For more information about implementing BAdIs as part of the enhancement framework, see SAP Library for SAP NetWeaver Platform on SAP Help Portal at [http://help.sap.com/nw\\_platform](http://help.sap.com/nw_platform). Choose a release and then *Application Help*. In SAP Library, choose *SAP NetWeaver Library: Function-Oriented View -> Application Server -> Application Server ABAP -> Application Development on AS ABAP -> ABAP Customer Development -> Enhancement Framework*.

## **BAdI: Outbound Message Processing**

### **Use**

This Business Add-In (BAdI) can be used to enhance the capabilities of the connector for the SAP Financial Services Network. You can use this BAdI to enhance the supported message types or enhance how outgoing message types are handled.

A BAdI implementation can be assigned for each message type. The implementing class contains several methods that are called by the connector during the outbound handling for applying outbound security, preparation of extended header, or processing steps that should be executed after the outgoing message is sent.

The standard implementations are inherited from a default implementation, which is always called when a specific implementation for the given message type is not found.

Implementations that are added for specific message types should inherit from the default implementation too, and you should only redefine the methods that are required to work differently to the methods in the default implementation.

This BAdI has the following methods:

- APPLY\_SECURITY\_OUT  
Used to apply outbound security.
- PREP\_EXT\_HEADER\_BEFORE\_SEC  
Used to prepare extended header before security.
- PREP\_EXT\_HEADER\_AFTER\_SEC  
Used to prepare extended header after security.
- BASE64\_ENCODE  
Used to encode message content in Base 64.
- PROC\_AFTER\_SEND  
Used to perform processing after sending.
- DETERMINE\_TRANSPORT\_LAYER\_CATEGORY  
Used to determine transport layer category.

#### **Standard settings**

For more information about the standard settings (filters, single or multiple uses), see the *Enh. Spot Element Definitions* tab in the BAdI Builder (transaction SE18).

#### **BAdI Implementations**

- /BSNAGT/BADI\_MESSAGE\_OUT\_DEF  
Default BAdI Implementation for outbound messages.
- /BSNAGT/BADI\_OUT\_PAIN00100102  
Outbound handler for PAIN.001.001.02 messages.
- /BSNAGT/BADI\_OUT\_PAIN0010103  
Outbound handler for PAIN.001.01.03 messages.

#### **See also**

For more information about implementing BAdIs as part of the enhancement framework, see SAP Library for SAP NetWeaver Platform on SAP Help Portal at [http://help.sap.com/nw\\_platform](http://help.sap.com/nw_platform). Choose a release and then *Application Help*. In SAP Library, choose *SAP NetWeaver Library: Function-*

*Oriented View -> Application Server -> Application Server ABAP -> Application Development on AS ABAP -> ABAP Customer Development -> Enhancement Framework.*

## **Settings for Determination of Logical Port**

### **Notes on Implementation**

#### **Use**

All Business Add-Ins (BAdIs) grouped under this structure node belong to the enhancement spot /BSNAGT/ES\_MESSAGE. You can use the BAdIs of this enhancement spot to make specific settings for the logical ports used for the outbound and synchronous interfaces.

#### **See also**

The following BAdIs are included under this enhancement spot:

- BAdI: Determination of Logical Port for Outbound Interface
- BAdI: Determination of Logical Port for Synchronous Interface

## **BAdI: Determination of Logical Port for Outbound Interface**

#### **Use**

This Business Add-In (BAdI) is used in the connector component for *SAP Financial Services Network*. You can use this BAdI to override the logical port prior to sending an outbound message.

Implementation of this BAdI is optional. If there is no active implementation, the system uses the default logical port (defined in transaction SOAMANAGER) or the alternative port specified in Customizing for *Financial Services Network Connector* under *Maintain Alternative Logical Ports for Outbound Messages*.

This BAdI is available in the Enhancement Spot /BSNAGT/ES\_MESSAGE and it has the method GET\_LOGICAL\_PORT\_NAME.

#### **Standard settings**

For more information about the standard settings (filters, single or multiple uses), see the *Enhancement Spot Element Definitions* tab in the BAdI Builder (transaction SE18).

### See also

For information about implementing BAdIs as part of the enhancement framework, see SAP Library for SAP NetWeaver Platform on SAP Help Portal at [http://help.sap.com/nw\\_platform](http://help.sap.com/nw_platform).

Choose a release and then *Application Help*. In SAP Library, choose *SAP NetWeaver Library: Function-Oriented View -> Application Server -> Application Server ABAP -> Application Development on AS ABAP -> ABAP Customer Development -> Enhancement Framework*.

## BAdI: Determination of Logical Port for Synchronous Interface

### Use

This Business Add-In (BAdI) is used in the connector component for the *SAP Financial Services Network*. You can use this BAdI to override the logical port used for synchronous communication with the SAP Financial Services Network.

Implementation of this BAdI is optional. If there is no active implementation, the system uses the default logical port (defined in transaction SOAMANAGER).

This BAdI is available in the Enhancement Spot /BSNAGT/ES\_MESSAGE and it has the method GET\_LOGICAL\_PORT\_NAME.

### Standard settings

For more information about the standard settings (filters, single or multiple uses), see the *Enhancement Spot Element Definitions* tab in the BAdI Builder (transaction SE18).

### See also

For information about implementing BAdIs as part of the enhancement framework, see SAP Library for SAP NetWeaver Platform on SAP Help Portal at [http://help.sap.com/nw\\_platform](http://help.sap.com/nw_platform).

Choose a release and then *Application Help*. In SAP Library, choose *SAP NetWeaver Library: Function-Oriented View -> Application Server -> Application Server ABAP -> Application Development on AS ABAP -> ABAP Customer Development -> Enhancement Framework*.

