

PUBLIC
2026-11-05

Product Assistance for Connector for SAP Multi-Bank Connectivity

© 2026 SAP SE or an SAP affiliate company. All rights reserved.

Content

1 About This Guide. 3

2 Connector for SAP Multi-Bank Connectivity. 4

2.1 Configuration. 6

Basic Settings. 6

Advanced Settings. 9

2.2 Basic Processes. 25

Outbound Message Processing. 25

Inbound Message Processing. 32

2.3 User Tasks. 40

Monitoring of Bank Messages. 40

Resending Outgoing Bank Messages. 42

Reprocessing Incoming Bank Messages. 42

Reprocessing of Single Processing Step. 42

Status Update After Manual Processing. 43

Requesting Bank Statements. 43

Creating Free Text Messages. 43

Manual Creation of Bank Messages. 44

2.4 User Interface. 45

Connector Monitor. 45

1 About This Guide

Purpose and Scope

This guide is only relevant for customers with older releases of SAP ECC or SAP SuccessFactors Employee Central Payroll (ECP) with installations of the connector for SAP Multi-Bank Connectivity (BSNAGT).

Customers with SAP S/4HANA as of version 1909 FP01, or SAP S/4HANA Cloud Private Edition should refer to the Connector for SAP Multi-Bank Connectivity Product Assistance.

Customers with SAP S/4HANA Cloud Public Edition should refer to the Connector for SAP Multi-Bank Connectivity Product Assistance.

2 Connector for SAP Multi-Bank Connectivity

Use

SAP Multi-Bank Connectivity is a connectivity solution based on a hosted, software as a service (SaaS) model, which enables banks and their corporate customers to exchange messages. To use this solution, the bank's corporate customers must use the connector for SAP Multi-Bank Connectivity.

A single instance of the connector can be configured to communicate with multiple banks.

Following a payment run, payment messages can be sent to the relevant bank automatically via SAP Multi-Bank Connectivity. In response, the connector receives bank messages, such as payment status reports and bank statements, through SAP Multi-Bank Connectivity, and then automatically processes them.

In addition to the payment process, the connector can send and receive messages from other business processes.

Constraints

- The functionality of the connector is closely integrated with the functionality of SAP Multi-Bank Connectivity. Certain features of the connector are dependent on the availability of the corresponding features in SAP Multi-Bank Connectivity. This means that some of the features delivered in the connector may not be available in the current version of SAP Multi-Bank Connectivity and vice versa. Therefore, some features of the connector for SAP Multi-Bank Connectivity may not be available in your current implementation.
- The connector doesn't modify the payload of the messages.

License

The connector for SAP Multi-Bank Connectivity doesn't require a separate license. However, you need a license for either SAP Multi-Bank Connectivity, SAP S/4HANA Cloud Private Edition, advanced payment management or SAP S/4HANA Cloud Public Edition, advanced payment management to be able to use it. For customers with older SAP ERP or SAP ECC releases, SAP Multi-Bank Connectivity must be licensed, as advanced payment management is not available for these environments.

Features

The connector has the following main features:

- **Outbound Interface for Messages**
  The connector sends outbound messages asynchronously, via a defined Web service interface.
- **Inbound Interface for Messages**
  The connector receives inbound messages asynchronously, via a defined Web service interface.
- **Outbound Payment Processing**
  The system can be configured to use the connector to send newly created payment messages automatically, following a payment run.
  For more information about the outbound payment process, see Outbound Message Processing [page 25].
- **Inbound Message Processing**
  For information about the processing of inbound messages (including payment messages and other types of messages), see Inbound Message Processing [page 32].
- **Integration Scenarios**
  In addition to payment messages, the connector can be used to send and receive messages from other business processes.
- **Security**
  The connector provides end-to-end security features using Secure Store and Forward (SSF) mechanisms. SSF ensures that only the intended recipient can access the content of a message. For selected message types, the connector provides functions for tamper protection to ensure end-to-end message consistency, even if format transformations in the SAP Multi-Bank Connectivity. For more information, see SSF Profile Settings [page 6].
- *Connector Monitor*
  The *Connector Monitor* is the central monitoring function for the connector for SAP Multi-Bank Connectivity. It displays a list of inbound and outbound messages based on your search criteria. From this screen, you can perform a range of functions such as viewing the content of messages, sending outbound messages, and changing the status of an inbound message. You can also test the connectivity to the SAP Multi-Bank Connectivity by creating a ping test. For more information, see Connector Monitor [page 45].
- **Pick-Up Files**
  The function for picking up files is used to create messages, based on files in a defined directory, and then send them via the connector. For more information, see File Pickup [page 28].
- **Send Messages**
  The function for sending messages is used to send newly created messages.
- **Pull Messages**
  The *Pull Messages* function is used to retrieve inbound messages synchronously. For more information, see Pull Messages [page 32].
- **Create MBC Messages**
  The *Create MBC Message* function is used to create MBC Messages manually. For more information, see Create Multi-Bank Connectivity Messages [page 30].
- **MBC Status Handling**
  SAP Multi-Bank Connectivity provides updates to, for example, payment messages that have been received by SAP Multi-Bank Connectivity or that have been forwarded to a bank using status messages. This allows a close tracking of whether outgoing (payment) messages have been sent to the bank by SAP Multi-Bank Connectivity. For more information, see MBC Status Messages [page 39].
- **Handling of Sensitive Messages**

Dedicated authorizations can be set up to control if users can display/download the payloads of sensitive messages or if users can see sensitive messages in the list of incoming/outgoing messages in *Connector Monitor*. For more information, see Handling of Sensitive Messages [page 23].

- **Compression**
  Connector for SAP Multi-Bank Connectivity allows compression of outgoing message payloads or decompression of incoming message payloads. For more information, see Handling of Compressed Messages [page 23].
- **Message Archiving**
  You can archive messages after a customized residence period. For more information, see Message Archiving [page 24].

2.1 Configuration

2.1.1 Basic Settings

Make the relevant settings.

The following basic settings are available:

- SSF Profile Settings [page 6]
- Number Range for Message ID [page 7]
- Variants for Processing Bank Statements [page 7]
- Ping Tests [page 8]

2.1.1.1 SSF Profile Settings

Use this Customizing activity to specify whether you want to sign and/or encrypt outbound messages, and whether you want to decrypt and verify inbound messages.

After setting up the relevant certificates and security configuration in transactions STRUST and SSFA, you need to define the SSF Profile Settings for the connector for SAP Multi-Bank Connectivity under *Multi-Bank Connectivity Connector* > *Maintain Secure Store and Forward (SSF) Profile Data*

In this Customizing activity, you need to specify the signatory/recipient name of the certificate to be used. You can define these settings depending on message type and sender/receiver ID. For security configuration, enter **BSNAGT** for *Application*.

On the right-hand side of the maintenance view, you can set the indicators for the specific security features that you want to apply to incoming and/or outgoing messages.

> ✤ Example
>
> In normal scenarios, it's sufficient to maintain one generic entry in this configuration view like this:
>
> - *Message Type*: **<initial>**

- *Sender/Receiver ID*: **<initial>**
- *Application*: **BSNAGT**
- *Signatory/Recipient Name*: **<the-relevant-name>**
- *Sign*: **true**
- *Encrypt*: **true**
- *Decrypt*: **true**
- *Verify*: **true**

2.1.1.2 Number Range for Message ID

Maintain the number range for the message ID to generate a unique message ID when the system is not supplied with a message ID during the creation of messages.

The message ID of outgoing MBC messages is created depending on the application that creates the MBC message. Besides that, the connector for SAP Multi-Bank Connectivity creates a message ID for outgoing messages if no ID is provided by the creating application. For this, the number range object `/BSNAGT/N2` is used.

Please use the Customizing activity under *Multi-Bank Connectivity Connector* > *Maintain Number Range for Message ID* to maintain the interval `N2` for this number range object.

2.1.1.3 Variants for Processing Bank Statements

Maintain selection variants for bank statements.

The connector for SAP Multi-Bank Connectivity supports direct processing of several bank statement formats. For the processing of bank statement messages, the payload of the incoming message is handed to transaction `FF.5`. Technically, a job is planned for the processing of bank statements. This job is planned using the relevant report that is also used by transaction `FF.5` during the manual upload of bank statements.

To use the direct processing of bank statements, you need to specify the `FF.5` variant that you want to use for incoming messages types. You can specify the variant under *Multi-Bank Connectivity Connector* > *Maintain Selection Variants for Bank Statements*.

You can specify the variant depending on sender ID and message type of the incoming statement messages. During runtime, the connector for SAP Multi-Bank Connectivity reads this configuration and calls transaction `FF.5` as a job using the assigned variant for processing the incoming bank statement message.

2.1.1.4 Ping Tests

Create ping test to test the connectivity between the connector for SAP Multi-Bank Connectivity and SAP Multi-Bank Connectivity.

Context

A ping test is used to test the connectivity between the connector for SAP Multi-Bank Connectivity and SAP Multi-Bank Connectivity. When you run a ping test, the connector sends a test message to SAP Multi-Bank Connectivity. When SAP Multi-Bank Connectivity receives the ping test message, it generates a ping response message that will be retrieved during a subsequent message pull call from the connector. In this way, the ping test verifies that both the outbound route and the inbound route are functioning correctly.

In addition, outgoing ping test messages can be signed and encrypted and are then decrypted and verified when they're received in SAP Multi-Bank Connectivity. SAP Multi-Bank Connectivity also signs and encrypts the ping response that is then decrypted and verified when received by the connector for SAP Multi-Bank Connectivity. This means that, by using the ping test functionality, you can verify not only that the connection is working but also that the certificate for communication with SAP Multi-Bank Connectivity is working well.

Procedure via Connector Monitor

1. Log into your SAP S/4HANA, SAP S/4HANA Cloud Private Edition, SAP ECC, or SAP SuccessFactors Employee Central Payroll (ECP) system.
2. Choose *Multi-Bank Connectivity Connector* > *Connector Monitor*.
3. Enter any relevant parameters and choose *Execute*.
4. Choose *Connection* and *Create Ping Request*.
5. A ping request message is created and sent to SAP Multi-Bank Connectivity.
6. Initiate a pull to receive a ping response message in your ERP. If you've created an automatic job to pull, wait until the job has finished.

Procedure via /BSNAGT/PING

1. Log into your SAP S/4HANA, SAP S/4HANA Cloud Private Edition, SAP ECC, or SAP SuccessFactors Employee Central Payroll (ECP) system.
2. Run transaction code `/BSNAGT/PING`. The sender and receiver ID fields should be pre-populated with *CONNECTOR* and *MBC*.
3. Execute the ping test.

Result

Ping request messages and ping response messages are displayed in the *Connector Monitor*. If a ping response has an error status, you can view the error details in the *Connector Monitor* and also navigate directly to the application log to view more detailed information about the error.

Related Information

Connector Monitor [page 45]

2.1.2 Advanced Settings

Make the relevant settings.

The following advanced settings are available:

- Maintaining Sender/Receiver ID Mappings [page 9]
- Maintaining SWIFT Parameters [page 10]
- EBICS Order Type Determination (EBICS 2.5) [page 16]
- Downloading of Incoming Messages [page 19]
- Routing [page 21]
  - ERP to ERP Integration via RFC-API [page 21]
  - Integration with SAP Multi-Bank Connectivity via HTTP-API [page 22]
- Handling of Sensitive Messages [page 23]
- Handling of Compressed Messages [page 23]
- Approval User Determination [page 15]
- Mapping of External/Internal Message Types [page 24]
- Message Archiving [page 24]

2.1.2.1 Maintaining Sender/Receiver ID Mappings

Maintain the mapping of sender and receiver IDs.

In payment scenarios based on the *Payment Medium Workbench*, the sender ID and receiver ID of an MBC message are set up on house bank level. For scenarios where it isn't possible to set the sender ID and receiver ID information, the connector for SAP Multi-Bank Connectivity allows the IDs to be mapped to other values with the Customizing activity under *Multi-Bank Connectivity Connector* > *Maintain Custom Sender/Receiver IDs*.

You can set up the sender and receiver IDs depending on message type, receiver ID, and origin of an MBC message. During message creation using the *Payment Medium Workbench*, the connector for SAP Multi-Bank Connectivity reads this configuration and adjust the IDs accordingly.

2.1.2.2 Maintaining SWIFT Parameters

Maintain the relevant SWIFT parameters to send messages to banks using the SAP Multi-Bank Connectivity.

If you want to use SAP Multi-Bank Connectivity to send a message to a bank using the SWIFT network, the relevant SWIFT parameters need to be defined in the connector for SAP Multi-Bank Connectivity.

To do this, maintain the Customizing activity under *Multi-Bank Connectivity Connector* > *Maintain Swift Parameters*.

You can define SWIFT parameters depending on receiver ID and message type. The value help for the parameter name shows the available SWIFT parameters. Furthermore, you can define whether a parameter is used for productive or nonproductive scenarios, for example for tests, or in both types of scenario.

When a message is sent to MBC, the SWIFT parameters are created in the extended header section of the MBC message and in this way form part of the message that is being sent to MBC.

Procedure

1. Run transaction code **SPRO**.
2. In the *SAP Reference IMG*, go to *Multi-Bank Connectivity Connector* > *Outbound Processing* > *Maintain Swift Parameters for Outbound Messages*.
3. On the *Change View "Maintenance of Swift Parameters": Overview screen*, choose *New Entries*.
4. The parameter name entries in the following table are mandatory for SAP Multi-Bank Connectivity. Additional parameters can be added based on bank and/or corporate requirements, for example, logical system, FileInfo, and so on.

Receiver ID	Message Type	Parameter Name	System Role	Parameter Value
Specifies the ID of the recipient of the message, typically the bank identifier code (BIC)	Add the in-scope message type to be sent to Swift via SAP Multi-Bank Connectivity, for example, pain.001.001.03	IsNotificationRequested	All / Production / Non-Production, as required	Value to be set as *True* or *False*. If set to *False*, no delivery notification is returned from Swift for payment messages.

Receiver ID	Message Type	Parameter Name	System Role	Parameter Value
As above	As above	ReceiverBIC	As required	*FileAct (FA) (not relevant for Business Connect)*: 11-digit BIC of receiving bank for FileAct (FA) messages. **Example** ABCDDEX0XXX *FIN*: 12-digit BIC of receiving bank for FIN messages. **Example** ABCDDEX0AXXX **→ Remember** BICs must be entered with uppercase letters. Use the test BIC for your test environment, and the production BIC for your production environment.
As above	As above	RequestType	As required	Message type, as identified by Swift. **Example** fin.101, fin.300 or pain.001.001.03 **→ Remember** Message types are case-sensitive, and must be entered with all lowercase letters.

Receiver ID	Message Type	Parameter Name	System Role	Parameter Value
As above	As above	Responder	As required	**Only FileAct-relevant**: Receiver Bank Distinguished Name (DN), to be provided by the receiver bank. **Example** o=dcba-dexx,o=swift **→ Remember** Responder DNs must be entered with all lowercase letters.

Receiver ID	Message Type	Parameter Name	System Role	Parameter Value
As above	As above	SenderBIC	As required	*FileAct (FA) Payments (not relevant for Business Connect)*: 11-digit BIC of sender/ corporation for FA payments. **Example** DCBADEX0XXX *FIN Payments*: 12-digit BIC of sender/corporate for FIN payments. **Example** DCBADEX0AXXX **→ Remember** BICs must be entered with uppercase letters. Use the test BIC for your test environment, and the production BIC for your production environment.

Receiver ID	Message Type	Parameter Name	System Role	Parameter Value
As above	As above	SenderDN	As required	**Only FileAct-relevant**: Corporate Distinguished Name (DN), to be provided by Swift. **Example** o=abcd-dexx,o=swift **→ Remember** Sender DNs must be entered with all lowercase letters.
As above	As above	Service	As required	The Swift service the bank has established with Swift. The bank should inform the corporation of the appropriate service to be entered here. Common services include swift.corp.fa!p for an FA message, or swift.fin for a FIN message. **→ Remember** Services must be entered with lowercase letters.

5. Choose *Save*.

When a message is sent to SAP Multi-Bank Connectivity, the Swift parameters are created in the extended header section of the MBC message and in this way form part of the message that is being sent to MBC.

2.1.2.3 SwiftRef Directory

Learn about SwiftRef and SAP Multi-Bank Connectivity.

The SwiftRef directory is a collection of verified and secure financial reference data provided by Swift and collected from corporates, banks, and other financial institutions. SwiftRef takes away the effort of verifying

reference data, like bank account numbers, before starting a payment process since customers can rely on an up-to-date and consistent data pool. If you want to receive SwiftRef via SAP Multi-Bank Connectivity, a separate subscription is necessary that can be purchased via the SAP Store.

You can download the SwiftRef file from the connector for SAP Multi-Bank Connectivity. From there, you need to manually upload it into your ERP via transaction code `BIC2`.

Related Information

SwiftRef, the payments reference data solution

2.1.2.4 EBICS Connectivity

2.1.2.4.1 Approval User Determination

Maintain the settings for determination of approval users.

Whenever a payment, which has been approved in SAP Bank Communication Management, is sent out, the connector for SAP Multi-Bank Connectivity reads the approval users and sends up to two approval users to SAP Multi-Bank Connectivity along with the outgoing message.

The approval users are placed in the extended header parameters `APPROVAL_USER_1` and `APPROVAL_USER_2`.

By default the system determines the last two approval users and writes them into the extended header section of the MBC message.

If necessary, you can overrule this default behavior by setting up more detailed rules in the Customizing activity under *Multi-Bank Connectivity Connector* > *Maintain Approval User Settings*.

In this Customizing, you can define which approval users should be taken into account based on the *Sender ID*, *Receiver ID*, *Message Type* and *Origin* of the outgoing MBC message.

The definition takes place by specifying a predefined *Sequence Number* and by mapping it to the relevant *Parameter*.

The following predefined sequence numbers are available:

- First Approval User
- Second Approval User
- Last but two Approval User
- Last but one Approval User
- Last Approval User

When the determination of approval users is configured in this Customizing activity, you can leave each of the key fields blank which then works as a wildcard.

> **Note**
>
> Please make sure to define only one relevant entry for each parameter to avoid that unexpected approval users are determined during output processing.

2.1.2.4.2 EBICS Order Type Determination (EBICS 2.5)

When SAP Multi-Bank Connectivity connects to banks via EBICS, an EBICS order type needs to be provided.

You can configure the rules for determining EBICS order types in the Customizing activity under *Multi-Bank Connectivity Connector* > *Maintain EBICS Order Type Determination*.

During the output processing (sending) of messages, this Customizing setting is checked and a configured EBICS order type is added as an extended header parameter *EBICS Order Type* to the outgoing message.

As this configuration is optional, no error will be raised even if no EBICS order type is determined for the outgoing message.

For setting up EBICS order type determination, you can use several attributes in which are available during the payment media creation via PMW or DMEE. Some attributes can be wildcarded by leaving them blank.

Attributes and Possible Wildcards

Attributes	Possible Wildcard
CoCd (Paying company code)	Yes
House Bk(Short key for a house bank)	Yes
Acct ID (ID for account details)	Yes
PM (Payment method)	Yes
Urgt (Checkbox for urgent payment)	No
IK (Instruction key for data medium exchange)	Yes
Code Word for Instructions 1	No
Code Word for Instructions 2	No
Code Word for Instructions 3	No
Code Word for Instructions 4	No
Direct Debit Type (Local instrument code)	No
SB_IND (Single/bulk indicator)	Yes

> **Note**
>
> For troubleshooting purposes, these fields are also logged in the MBC message. For releases other than SAP S/4HANA Cloud Public Edition, please have a look at these SAP notes:
>
> - 3007500 (Maintenance View for EBICSOrderType Determination)
> - 3032850 (Adjustment of parameter name for EBICS.OrderType)
> - 3001090 (Configuration for EBICSOrderType determination)

- 2991619 (New Extended Header Parameter for EBICSOrderType)
- 3091493 (EBICS Order Type Value Helps)
- 3007569 (Determination of EBICS Order Type in PMW/DMEE-based Payment Runs)
- 3276937 (File Pickup - EBICS Order Type parameter missing)

2.1.2.4.3 EBICS BTF Settings (EBICS 3.0)

The Business Transaction Format (BTF) concept has been introduced in EBICS 3.0 to replace order types. Learn how to configure BTF settings in EBICS.

Prerequisites

- Before configuring the BTFs, make sure to always have the latest support package installed (at least SP08). If you have reasons not to install the latest support package, please have a look at the SAP Notes below in the Related Information section that apply to specific versions. It is recommended to install the notes in the order that they are displayed.
- BTF parameters are determined by the bank and might differ per bank, country/region, and other factors. SAP Multi-Bank Connectivity does **not** determine any BTF parameters. Please confirm them beforehand with your banks or financial service providers.

Configurations

When using EBICS 3.0, you need to configure the relevant BTFs in connector for SAP Multi-Bank Connectivity. This configuration is covered by several configuration views.

Defining and Maintaining BTF Types and Parameters

1. Go to *IMG* > *Multi-Bank Connectivity Connector* > *EBICS BTF* > *Maintain EBICS BTF Types* to define the relevant BTFs. Provide a name and a description for the defined BTFs.
2. Go to *IMG* > *Multi-Bank Connectivity Connector* > *EBICS BTF* > *Maintain EBICS BTF Parameter assignments* to assign the relevant parameters to each of the BTF that you've defined in step 1. Please align the necessary parameters with the relevant bank and configure them appropriately.
3. Go to *IMG* > *Multi-Bank Connectivity Connector* > *EBICS BTF* > *Maintain EBICS BTF Type Determination Rules* to define the rules for determination of BTFs. During payment media creation, the system will check for the fields and assign the relevant BTF to the respective MBC Message. When defining rules for BTF determination, the following fields will be treated as wildcards if left blank:
   - Company Code
   - House Bank ID
   - Account ID
   - Payment Method

All other fields won't be treated as wildcards as the initial value might have an actual meaning.

Examples of Mandatory BTF Parameters

There are many different BTF parameters but not all of them are used by every customer. The BTF parameters in the table are mandatory:

BTF Parameters	Value (Example)
EBICS.BTF.AdminOrderType	BTU
EBICS.BTF.MsgName	pain.001
EBICS.BTF.ServiceName	SCT
EBICS.BTF.SignatureFlag	TRUE

Deleting BTF Types

Before deleting BTF types, please make sure to delete the corresponding configuration entries which make use of the to-be-deleted BTF type. BTF types are used in the following configuration options:

- *IMG* > *Multi-Bank Connectivity Connector* > *Outbound Processing* > *EBICS BTF* > *Maintain EBICS BTF Parameter assignments*
- *IMG* > *Multi-Bank Connectivity Connector* > *Outbound Processing* > *EBICS BTF* > *Maintain EBICS BTF Type Determination Rules*
- *IMG* > *Financial Supply Chain Management* > *Advanced Payment Management* > *External Interfaces* > *File Handler* > *Multi-Bank Connectivity Connector Integration* > *Maintain EBICS Order Type Determination*

Related Information

SAP Note 3351194
SAP Note 3351495
SAP Note 3351627
SAP Note 3351946
SAP Note 3352016
SAP Note 3653169

2.1.2.4.4 EBICS User Mapping

In scenarios in which SAP Multi-Bank Connectivity connects to banks using EBICS, EBICS users are required for sending messages to the banks. Two EBICS users are usually required in such a scenario.

You can configure the mapping of approval users to EBICS users in the Customizing activity under *Multi-Bank Connectivity Connector* > *Map Approval Users to EBICS Users*.

If you map approval users to EBICS users, the EBICS users that are defined are used as the values for the extended header parameters `EBICS.USER_1` and `EBICS.USER_2` in the MBC message.

> **Note**
>
> For more information about the handling of approval users, see Approval User Determination [page 15].

This configuration is optional. No error is raised if no EBICS users are assigned during the outbound processing. In this case, no EBICS user parameters are created in the extended header section of the MBC message.

When you set up mapping rules for EBICS users, you must specify the receiver ID that matches the bank that you want to send the messages to.

You can optionally also specify a sender ID if you want different rules for different sender IDs.

2.1.2.4.5 Static EBICS Users

In scenarios in which SAP Multi-Bank Connectivity connects to banks using EBICS, EBICS users are required for sending messages to the banks. Two EBICS users are usually required in such a scenario.

You can define static EBICS users that can be used by SAP Multi-Bank Connectivity to send messages through EBICS in the Customizing activity under *Multi-Bank Connectivity Connector* > *Maintain Static EBICS Users*.

Static EBICS users are used in scenarios where there are no approval users, but you must define EBICS users to be able to send messages via EBICS, for example, if the messages are created by the file pickup function.

To handle such cases, you can define static EBICS users with the following message header fields:

- *Sender ID*
- *Receiver ID*
- *Message Type*
- *Origin*

For a simpler configuration, you can also leave one or more of these fields empty. In this case, an empty field is treated as a wildcard (*).

For each static EBICS user entry, assign the extended header parameter `EBICS.USER_1` or `EBICS.USER_2` and the relevant EBICS user known to the bank.

2.1.2.5 Downloading of Incoming Messages

Automatic download of incoming messages can be set up by defining processing steps. This allows for detailed control over which messages are downloaded and where they are stored on the server.

It is possible to set up an automatic download of incoming messages. This means that the *processing* of an incoming message is actually to download it to a defined directory on or accessible by the application server.

In addition to specifying the logical file name, you can also define which fields should be mapped to the logical file name during the determination of the physical file name. You can map specific fields of the MBC message

header to the parameters that can be used during the definition of the pattern that tells how the physical file name should be structured.

Procedure

1. Define an inbound processing step in the Customizing activity under *Multi-Bank Connectivity Connector* > *Maintain Inbound Processing Steps*
2. To define the automatic download of incoming messages, assign a processing step from the processing application DOWNLOAD - Download to Application Server. You can define such a download step as an additional step to other steps or as the only step.

   > **Note**
   >
   > Creation of some default processing steps will be switched off by defining additional processing steps. For this reason, please make sure to define all the relevant steps for the relevant messages when defining processing steps.

3. After assigning the `DOWNLOAD` processing step, define a logical path name that can be used for downloading via transaction codes `FILE`.
   1. Start transaction *FILE*.
   2. To enter a new path name, select an existing path name, or choose *New Entries*. This must be as descriptive as possible.
   3. In the *Logical File Path Definition* dialog structure, double-click on the *Assignment of Physical Path and Logical Paths* substructure. Double-click to select a syntax group if there are several. If there's only one, it's automatically selected. In a heterogeneous system landscape (such as UNIX and Windows NT servers) all system-specific syntax groups must be maintained. Ensure that the definitions for the various syntax groups point to the same directory.
   4. Assign a physical path name to the logical path name.
4. Save your entry.
5. Define a logical file name. Logical file names are mostly created cross-client-specific via transaction `FILE`. Optionally, you can do client-specific settings with transaction `SF01`. A client-specific definition always overrides a cross-client definition. Therefore, ensure that in every client any unnecessary client-specific definitions are deleted. By default, the system uses the logical file name /BSNAGT/DOWNLOAD_FILE.
   1. Start transaction *FILE*.
   2. To enter a new file name, select an existing file name, or choose *New Entries*. This must be as descriptive as possible.
   3. Double-click on the *Logical File Name Definition, Cross-Client* dialog structure.
   4. Double-click on the relevant logical file name.
   5. Maintain the *Physical File* name field. This file name may contain placeholders that are substituted at runtime. For SAP Multi-Bank Connectivity, some message header attributes are assigned to the following three parameters that may be selected as a placeholder:
      - Parameter 1 <PARAM_1> = Message type
      - Parameter 2 <PARAM_2> = File name
      - Parameter 3 <PARAM_3> = Message id

      If you need more flexibility, please see step 6.
   6. In the *Logical Path* field, you assign the logical path name defined previously to the current logical file name. You can assign a logical path name to multiple logical file names.

6. **Optional:** If a more flexible approach is required for downloading, you can also define additional settings in the Customizing activity under *Multi-Bank Connectivity Connector* > *Maintain Download Settings*. In this Customizing activity you can define what logical file should be used for downloading based on *Message Type*, *Sender ID*, *Receiver ID* and *Origin*. This Customizing activity enables you to define in a more detailed way which messages should be downloaded into which server directories, and which message header attributes should be mapped to the generic parameters of the logical file name.
7. Save your entry.

2.1.2.6 Routing

The connector for SAP Multi-Bank Connectivity provides routing capabilities to enable scenarios where MBC messages are exchanged between different instances of connector for SAP Multi-Bank Connectivity (in different ERP systems).

An example for such a scenario might be the integration of an HR system which is not connected to SAP Multi-Bank Connectivity directly, but routes its messages via another ERP system or connector for SAP Multi-Bank Connectivity which is connected to SAP Multi-Bank Connectivity.

You can define routing conditions in the sending system in the Customizing activity under *Multi-Bank Connectivity Connector* > *Maintain Routing Settings*.

In this Customizing activity you configure the rules to determine a *Routing Category* which tells the technical channel that an outgoing message should take.

The following routing category options are available:

- HTTP-API
  Messages are sent via an HTTP API.
- RFC-API
  Messages will be sent via RFC API. For more information, see ERP to ERP Integration via RFC-API [page 21].
- Web Service Runtime
  The outgoing message will be sent via Web service runtime. Depending on the configuration of the Web service runtime, XI protocol or Web service runtime can be used.

2.1.2.6.1 ERP to ERP Integration via RFC-API

ERP-to-ERP integration via RFC-API enables message transfer between connectors by configuring routing and RFC destination settings in SAP Multi-Bank Connectivity.

For sending messages from one connector to another one, you can use RFC API once you have configured the *Routing Category* with the RFC API in the Customizing activity under *Multi-Bank Connectivity Connector* > *Maintain Routing Settings*.

You need to specify the technical details for usage of the RFC API in the Customizing activity under *Multi-Bank Connectivity Connector* > *Maintain Settings for RFC API*.

In this Customizing activity you configure the determination of the *RFC Destination* that should be used for calling the RFC API. You can define the RFC destination based on *Receiver ID* and *Origin* of the outgoing MBC message.

When setting up an RFC destination that should be used with the RFC API, the recommended connection type is 3 (RFC connection to ABAP system).

2.1.2.6.2 Integration with SAP Multi-Bank Connectivity via HTTP-API

For communication between SAP Multi-Bank Connectivity and connector for SAP Multi-Bank Connectivity, an HTTP based API allows a direct/synchronous communication without using Web service runtime or XI protocol. To send messages via HTTP-API, you need to set up the following logical ports using transaction **SM59**.

> ⓘ Note
>
> Please align with the SAP Multi-Bank Connectivity onboarding/support team to clarify the recommended way of communication between SAP Multi-Bank Connectivity and connector for SAP Multi-Bank Connectivity. The communication via HTTP-API might not be available in all scenarios.

If required, you can name the RFC destinations as needed and mention them in the Customizing for HTTP-API. If no special requirements exist to the naming of RFC destinations, however, some default names are assumed by the connector for SAP Multi-Bank Connectivity when communicating via HTTP-API. The relevant RFC destinations and their default names are listed in the following table:

RFC Destinations and Default Names

Purpose	Default Name	Connection Type
Send Request	MBC_SEND_REQ_<SYSID><CLIENT>	G - HTTP Connection to External Server
Send Ack/Response	MBC_SEND_ACK_<SYSID><CLIENT>	G - HTTP Connection to External Server
Pull Request	MBC_PULL_REQ_<SYSID><CLIENT>	G - HTTP Connection to External Server
Pull Ack/Response	MBC_PULL_ACK_<SYSID><CLIENT>	G - HTTP Connection to External Server

To send messages via HTTP-API, set up one or multiple routing rules which use the routing category **HTTP API** in the Customizing activity under *Multi-Bank Connectivity Connector* > *Maintain Routing Settings*.

To specify the technical settings like RFC destinations for HTTP-API, maintain the settings in the Customizing activity under *Multi-Bank Connectivity Connector* > *Maintain Settings for HTTP API*.

Please only mention the send-related RFC destinations in this Customizing activity. Pull-related RFC destinations are maintained when setting up the pull type.

> ⓘ Note
>
> These settings only need to be made if you don't use the default names for RFC destinations as recommended above.

For setting up RFC destinations for receiving messages, you need to set up a pull type in the Customizing activity under *Multi-Bank Connectivity Connector* > *Maintain Pull Types*.

To use HTTP-API for pulling, you can assign the pull category **`HTTP API`**. If you don't use the default names for RFC destinations as recommended above, you also need to specify the two pull-related RFC destinations in the respective fields.

To define pulling via HTTP-API as the standard way of communicating with SAP Multi-Bank Connectivity, you can define an initial pull type entry.

2.1.2.7 Handling of Sensitive Messages

Define which incoming or outgoing messages are treated as sensitive messages.

You can define which incoming or outgoing messages are treated as sensitive. For sensitive messages, additional authorizations are required to display and/or download the message payload or to control whether a user can see the message in *Connector Monitor*.

Use the Customizing activity under *Multi-Bank Connectivity Connector* > *Maintain Sensitive Data Classification* to define whether messages are classified as sensitive depending on the sender/receiver ID, message type, and origin of the MBC message.

When receiving an incoming message, the connector for SAP Multi-Bank Connectivity checks this configuration and classifies the message accordingly.

MBC messages can also be classified by providing the extended header parameter `CONNECTOR.SensitivePayload` with the value **`true`**. Incoming messages that have this parameter set are classified directly as sensitive regardless of what is defined in the Customizing activity.

The parameter `CONNECTOR.SensitivePayload` is also added to outgoing sensitive messages during sending.

2.1.2.8 Handling of Compressed Messages

Maintain the settings for compressing and decompressing message payloads.

The Connector for SAP Multi-Bank Connectivity enables message payloads to be compressed and decompressed. Use the Customizing activity under *Multi-Bank Connectivity Connector* > *Maintain Compression Settings* to define the compression category and the decompression category depending on sender and receiver ID and message type:

- The compression category defines whether an outgoing message is expected to be compressed during sending.
- The decompression category defines whether the payload of an incoming message is expected to be received in compressed form and needs to be decompressed before processing.

2.1.2.9 Mapping of External/Internal Message Types

Map external message types to internal message types.

When SAP Multi-Bank Connectivity or a bank can't provide message types exactly in the way the connector for SAP Multi-Bank Connectivity expects them, message types can be mapped. Use the Customizing activity *Multi-Bank Connectivity Connector* > *Maintain external-to-internal MessageType Mapping* to define which external message types are mapped to which internal message types.

2.1.2.10 Message Archiving

Archive your message in the *Connector Monitor*.

Use

Periodically, you can archive selected messages so that they're no longer displayed in the *Connector Monitor*.

Integration

The archiving object `/BSNAGT/AO` is used to archive message data from table `/BSNAGT/FILE_INF` and message status information from the tables `CRM_JSTO` and `CRM_JEST`.

Features

The system enables you to configure a minimum residence time for each message type. The residence time is the minimum amount of time that messages are retained before becoming eligible for archiving.

You maintain the residence time in the Customizing activity under *Multi-Bank Connectivity Connector* > *Maintain Residence Time for Message Archiving*.

Activities

You perform the following activities:

- **Select Messages to Archive**
  The system provides a report that enables you to select a range of messages and then set the status of the messages to indicate that they're ready for archiving.

The report has two modes: test and production. You must run the report in production mode to set the relevant message status for archiving. The archiving status is maintained in the ARC_STATUS field in table /BSNAGT/FILE_INF.

To select messages for archiving, choose *Multi-Bank Connectivity Connector* > *Message Archiving* > *Select Messages to Archive*.

- **Archive Selected Messages**
  This report copies all the messages that are ready for archiving to an archive, and then sets the status of the messages to archived.
  To copy messages to the archive, choose *Multi-Bank Connectivity Connector* > *Message Archiving* > *Archive Selected Messages*.
- **Delete Archived Messages**
  This report deletes all the messages that have been archived.
  To delete archived messages, choose *Multi-Bank Connectivity Connector* > *Message Archiving* > *Delete Archived Messages*.
- **Reload Archived Messages**
  If archived data is required in the system again, this report can be used to reload it from the archive.
  To reload messages from the archive, choose *Multi-Bank Connectivity Connector* > *Message Archiving* > *Reload Archived Messages*.

2.2 Basic Processes

2.2.1 Outbound Message Processing

Use

The connector for SAP Multi-Bank Connectivity sends outbound messages via an API.

SAP S/4HANA provides the option of using either *Payment Medium Workbench* or classic payment medium programs to generate payment media. If you use the *Payment Medium Workbench*, then you can use the connector for SAP Multi-Bank Connectivity to send payment messages automatically following a payment run, as described in this process. If you use classic payment medium programs, you need to run the *File Pickup* function to generate the messages.

Prerequisites

- You've maintained the payment medium formats and assigned the payment medium type **`SAP Multi-Bank Connectivity Connector`** to the relevant formats.
- You've maintained your Secure Store and Forward (SSF) product and SSF format using transaction SSFA.
- You've maintained your security certificates in trust manager using transaction STRUST.
- You've setup connectivity with SAP Multi-Bank Connectivity or the relevant receiver system.

- You've maintained SWIFT parameters in case you want to leverage SWIFT connectivity. For more information, see the chapter on *Maintaining SWIFT Parameters*

Payment Run Process

1. You perform a payment run and choose the relevant payment method for SAP Multi-Bank Connectivity.
2. If you're using SAP Bank Communication Management and the batch requires approval, the system triggers the payment approval process as follows:
   1. For each step in the approval process, the approver enters the required data and accepts the batch.
   2. The final approver triggers the creation of the payment medium for the batch.
3. The system creates the payment formats.

   > **Note**
   >
   > If using *Payment Medium Workbench*, the payment media creation can directly create an outgoing MBC message and no longer needs to create a physical file.

4. The system applies the security settings you've defined in Customizing.
   Optionally, the messages can be signed, encrypted, or signed and encrypted.
5. The system calls an API to send the messages to SAP Multi-Bank Connectivity.

   > **Note**
   >
   > Depending on your configuration settings, the system attaches the relevant parameters, for example for SWIFT, to the outgoing message during sending.

The *Connector Monitor* displays a list of all outbound messages and their corresponding statuses. For more information, see the chapter on the *Connector Monitor*.

Related Information

File Pickup [page 28]
Maintaining SWIFT Parameters [page 10]
Connector Monitor [page 45]

2.2.1.1 Send Messages

Use

Some messages, such as payment messages, are generated and sent automatically. However, the connector for SAP Multi-Bank Connectivity can also be used by other business processes, and in some cases, the messages are created but not sent. The *Send Messages* function is specifically provided to send messages that

have been generated by the connector, but not sent automatically. For example, these messages could include messages generated by the *Pick-up Files* function.

The *Send Messages* function can also be used to re-send messages that have been previously sent. This could be required, for example, in a disaster recovery situation.

Features

Before running the function, you can optionally enter selection criteria such as the *Message Type*, *Receiver ID*, and *Sender ID*.

If you select the *Include Sent Msg.* checkbox, the connector includes the previously sent messages according to the entered selection criteria. For messages that have previously been sent, the system creates a copy of the original message and sends this copy to SAP Multi-Bank Connectivity. The *Message ID* of the original message is saved as the *Related Message ID* in the new copy of the message.

Activities

To send messages, on the *SAP Easy Access* screen, choose *SAP Menu* > *Connector for Multi-Bank Connectivity* > *Send Messages*.

Additionally to viewing the status of outbound messages, you can also send messages via the *Connector Monitor*. Choose *SAP Menu* > *Connector for Multi-Bank Connectivity* > *Connector Monitor*. Adapt the search filters if necessary and start the search. Mark the message you would like to send, elect the chevron icon next to *Send* and select *Send*.

Related Information

Connector Monitor [page 45]

2.2.1.2 Resend Messages

Send messages again that were previously sent successfully.

Use

Usually, once a message is sent, it is not possible to send it again to avoid duplicate payments being initiated with banks. But there might still be cases where it is necessary to send a message again which was already sent before. Duplicate messages are normally caught in a checking process within SAP Multi-Bank Connectivity. The

Send Again function attaches specific parameters to the MBC message that allow it to go around this checking process. This activity requires a separate authorization which usually not every user needs to have.

> **Note**
>
> You can resend messages only up until 14 days after they have been created.

Activities

To resend messages, on the *SAP Easy Access* screen, choose *SAP Menu* > *Connector for Multi-Bank Connectivity* > *Connector Monitor*. Adapt the search filters if necessary and start the search. Mark the message you would like to resend, select the chevron icon next to *Send* and select *Send again*.

2.2.1.3 File Pickup

Upload files and send them using SAP Multi-Bank Connectivity.

Use

You can upload files of a defined directory on the application server (visible via `AL11`), and send them using SAP Multi-Bank Connectivity. Use transaction `/BSNAGT/PICK_FILES` to upload the files. Please note that only clear text files are accepted. Encrypted files or other storage options like SFTP or WebDAV are not supported by the *File Pickup* function.

There are two possible scenarios for using the file pickup.

Scenario 1: Header Data Determination Using File Names

In this scenario, the file names are expected to contain all the information that is required in the header of the MBC messages. The file name is expected to be structured as a comma-separated list with the following information:

- `MessageType`
- `SenderId`
- `ReceiverId`
- `NumberOfRecords`
- `FileName`
- `MessageId (optional)`

When picking up a file, the connector for SAP Multi-Bank Connectivity parses the file name and propagates the values into the corresponding fields of the message header.

Scenario 2: Fix Value Header Fields

In this scenario, you can define fixed values for message type, sender ID, and receiver ID. In this case, the file names are not required to be structured to hold header information.

When picking up the files, the connector for SAP Multi-Bank Connectivity populates the fields *Message Type* and *Sender/Receiver ID* using the fixed values. The *File Name* field is populated with the name of the file that is picked up without making any adjustments to it. The message ID is created by the connector for SAP Multi-Bank Connectivity using a number range object.

Setting Up File Pickup

To use the file pickup, you need to define a logical file name, for example, by using the transactions `FILE` or `SF01`, that specifies the directory from which files are read. In the file pickup report, this is called the source directory.

Optionally, you can set up a target directory using a logical file name. If a target directory is set up, the connector for SAP Multi-Bank Connectivity moves the files that have been picked up to the target directory. This enables the source directory to remain in a clean state. Setting up a target directory is highly recommended as it helps reduce the number of files in the source directory, which improves performance when the connector for SAP Multi-Bank Connectivity scans the source directory for new files.

To set up the source and target directories, you can use transactions `FILE` or `SF01`. The standard delivery includes some examples for logical files and logical directories that show you how to set up the source/target directory and file:

- **`/BSNAGT/PICKUP_SRC_DIR`**
- **`/BSNAGT/PICKUP_TRG_DIR`**

It isn't required that you use these logical files or directories directly, but you can create your own logical files and directories as necessary.

If necessary, the file pickup can also be planned as a regular job by planning a job for report `/BSNAGT/REP_PICKUP_FILES`.

Procedure

1. Define a logical path name via transaction codes `FILE`. Examples are the aforementioned **`/BSNAGT/PICKUP_SRC_DIR`** or **`/BSNAGT/PICKUP_TRG_DIR`**.
    1. Start transaction *FILE*.
    2. To enter a new path name, select an existing path name, or choose *New Entries*. This must be as descriptive as possible.
    3. In the *Logical File Path Definition* dialog structure, double-click on the *Assignment of Physical Path and Logical Paths* substructure. Double-click to select a syntax group if there are several. If there's only one, it's automatically selected. In a heterogeneous system landscape (such as UNIX and Windows NT servers) all system-specific syntax groups must be maintained. Ensure that the definitions for the various syntax groups point to the same directory.
    4. Assign a physical path name to the logical path name.
2. Save your entry.

3. Define a logical file name. Examples are the aforementioned **/BSNAGT/PICKUP_SRC_DIR** or **/BSNAGT/PICKUP_TRG_DIR**. Logical file names are mostly created cross-client-specific via transaction FILE. Optionally, you can do client-specific settings with transaction SF01. A client-specific definition always overrides a cross-client definition. Therefore, ensure that in every client any unnecessary client-specific definitions are deleted.
    1. Start transaction *FILE*.
    2. To enter a new file name, select an existing path name, or choose *New Entries*. This must be as descriptive as possible.
    3. Double-click on the *Logical File Name Definition, Cross-Client* dialog structure.
    4. Double-click on the relevant logical file name.

    > ⓘ Note
    >
    > When setting up logical file names, please leave the *Physical File* field empty. Despite the name, the logical file name should point to a directory, not to a file, and with content in this field, the process doesn't work anymore.

    5. In the *Logical Path* field, you assign the logical path name defined previously to the current logical file name. You can assign a logical path name to multiple logical file names.
4. Save your entry.

> ⚠ Caution
>
> Avoid running concurrent operations that can interfere with each other:
>
> - Don't write files into the pickup directory while simultaneously picking up files from there. Doing so can result in empty or partially picked-up files.
> - Ensure that only one pickup job runs per directory at a time. The pickup function's duplicate check does not support parallel pickups.
> - Keep pickup and pull jobs sequential. If a pickup job sends a payment and a pull job starts before the pickup job has finished persisting that outgoing message, the pull response may return an MBC_STATUS for a message that cannot yet be found in the system, causing the MBC_STATUS to fail in processing.

2.2.1.4 Create Multi-Bank Connectivity Messages

Upload local files to create Multi-Bank Connectivity messages.

You can create an outgoing Multi-Bank Connectivity message by uploading a local file from your computer using transaction /BSNAGT/UPLOAD_FILE. You find this transaction under *Connector for Multi-Bank Connectivity* > *Create Multi-Bank Connectivity Message*.

When creating a message manually, you need to provide the message type, the sender/receiver ID, and the path/filename of the local file that is to be sent.

In the case of text-based files, you also need to specify the code page of the file.

Finally, you can use the parameter *Send* to define whether the message is sent immediately. If this parameter isn't set, the new message is created only and remains in the created status. Sending can then be triggered manually with the *Connector Monitor*.

2.2.1.5 API

Integrate to third-party systems with an API.

For a closer integration with third-party systems, you can leverage the API of the connector for SAP Multi-Bank Connectivity. You can use the function module `/BSNAGT/BAPI_CREATE_FILE API` to create MBC messages and optionally trigger sending after creation.

2.2.1.6 Payments in Contract Accounting

You can automatically communicate outgoing payments that you generate in Contract Accounting (FI-CA) to your bank via the connector for SAP Multi-Bank Connectivity. For more information, see the Contract Accounting documentation.

Related Information

Contract Accounting Integration with SAP Multi-Bank Connectivity (SAP S/4HANA as of version 1809
Contract Accounting Integration with SAP Multi-Bank Connectivity (SAP S/4HANA Cloud Public Edition)

2.2.1.7 Treasury Correspondence

Purpose

This integration scenario is a part of the *Exchange Treasury Correspondence via SWIFT* process that allows you to send and receive treasury correspondence for your FX and money market transactions. The integration is enabled using SAP Multi-Bank Connectivity for communication with the SWIFT infrastructure.

Related Information

Treasury Correspondence for S/4HANA Cloud
Treasury Correspondence for S/4HANA

2.2.2 Inbound Message Processing

Use

The connector for SAP Multi-Bank Connectivity receives inbound messages via a Web service interface.

The connector decrypts the content of the incoming message and verifies the signature. Then, the message and the metadata are added to the database table for inbound messages. At this point, the message details can be viewed in the *Connector Monitor*.

Based on the message type, the system either triggers automatic processing or sets the status of the message to indicate that manual processing is required.

Process

1. Depending on your specific implementation scenario, messages can be received synchronously with a pull mechanism or asynchronously with a push mechanism.
   For more information, see the chapter *Pull Messages*.
2. When a message is received, the system decrypts it and verifies the signature depending on the security settings.
3. The system saves the message and its metadata to a database table.
4. The system uses the message type, which is specified in the Web service message, to determine how to process the message. Processing steps are determined via the message type. Each message can have one or more message types that can be defaulted by the system or can be defined in the Customizing activity *Maintain Inbound Processing Steps*.
   For information about which message types are supported and about how the corresponding messages are processed, see the chapter on *Supported Message Types*.

Related Information

Pull Messages [page 32]
Automatically Processed Message Types [page 33]
Connector Monitor [page 45]

2.2.2.1 Pull Messages

Retrieve inbound messages using pull messages.

The pull function can be used to retrieve inbound messages from SAP Multi-Bank Connectivity synchronously. By using the pull function, you avoid having to open up a corporate network for inbound connections. When the pull function is triggered, a synchronous outbound connection to SAP Multi-Bank Connectivity is established to request new inbound messages. If new messages exist, SAP Multi-Bank Connectivity provides them in the

response of the synchronous pull. After that, the connector sends receipt messages for each inbound message that was received from SAP Multi-Bank Connectivity. This whole process is then repeated until SAP Multi-Bank Connectivity doesn't send any more messages.

Technically, the response from SAP Multi-Bank Connectivity is a bulk message that can contain several MBC messages. Normally, only the MBC messages are persisted in the connector for SAP Multi-Bank Connectivity, but not the bulk message itself. By setting the parameter *Save Bulk Messages* to **`true`**, you can control if the bulk message is persisted as well. This can be useful in development or test environments.

Procedure via Connector Monitor

1. Log into your SAP S/4HANA, SAP S/4HANA Cloud Private Edition, SAP ECC, or SAP SuccessFactors Employee Central Payroll (ECP) system.
2. Choose *Multi-Bank Connectivity Connector* > *Connector Monitor*.
3. Enter any relevant parameters and choose *Execute*.
4. Choose *Pull* and *Schedule*.

Procedure via /BSNAGT/MSG_PULL or Pull Messages

Via /BSNAGT/MSG_PULL

1. Log into your SAP S/4HANA, SAP S/4HANA Cloud Private Edition, SAP ECC, or SAP SuccessFactors Employee Central Payroll (ECP) system.
2. Run transaction code `/BSNAGT/MSG_PULL`.
3. Select the pull type you previously defined and execute the pull.

Via Pull Messages

1. Log into your SAP S/4HANA, SAP S/4HANA Cloud Private Edition, SAP ECC, or SAP SuccessFactors Employee Central Payroll (ECP) system.
2. Choose *Multi-Bank Connectivity Connector* > *Pull Messages*.
3. Select the pull type you previously defined and execute the pull.

2.2.2.2 Automatically Processed Message Types

The connector for SAP Multi-Bank Connectivity supports the processing of any sent or received message type. Some of the need to be manually processed while the majority can be processed automatically.

Automatic Processing of Bank Statements

The connector can trigger processing of incoming bank statement messages for several known message types.

Automatic Processing of pain.002 Status Messages

Processing of incoming pain.002 status messages can be triggered by the connector. For this, the pain.002 XML payload is handed to the Bank Communication Management component for processing.

If Bank Communication Management isn't available, incoming pain.002 status messages are set to status *For Manual Processing*.

Automatic Processing of camt.086 Bank Fee Statement Messages

Processing of incoming camt.086 bank fee messages can be triggered automatically by connector for SAP Multi-Bank Connectivity.

Recognized Message Types

The following message types are processed **automatically** by connector for SAP Multi-Bank Connectivity and subsequently handed over to the ERP.

Message Type	Description
PAIN.002.xxx.xx	Messages of this type are handed over to the Bank Communication Management component for further processing, which entails finding and updating the corresponding payment batch.
PAIN.001.xxx.xx	Customer Credit Transfer Initiation messages are usually outbound, but are processed also as inbound messages.
PAIN.008.xxx.xx	Customer Direct Debit Initiation messages are usually outbound, but are processed also as inbound messages.
CAMT.052.xxx.xx CAMT.053.xxx.xx CAMT.054.xxx.xx MT940 MT942	Bank statements and payment status reports are processed automatically. The system processes bank statements by running the *Import Electronic Bank Statement* process (transaction FF.5).
FIN.940 FIN.942	Swift Business Connect message formats (equivalent to MT940 and MT942) are processed automatically with SAP 3846899.

Message Type	Description
BAI2_LOCKBOX BAI2_STATEMENT BAI_LOCKBOX BAI_STATEMENT	Lockbox files (`BAI` and `BAI2`) are imported with the standard import for such messages.
CAMT.086.xxx.xx	Bank fee messages can be received and are automatically handed over to bank fee management for further processing.
MBC_STATUS	Internal status messages for SAP Multi-Bank Connectivity and its connector to update, for example, outgoing payment messages once they've been received by SAP Multi-Bank Connectivity and once they've been forwarded to the bank by SAP Multi-Bank Connectivity.
MT300 MT320 MT535	This integration scenario is a part of the *Exchange Treasury Correspondence via SWIFT* process, which allows you to send and receive treasury correspondence using MT300 and MT320 SWIFT messages for your FX and money market transactions, and to receive MT535 messages for external securities account statements from your depository banks. Relevant incoming correspondence objects are generated automatically. For more information, see the chapter on *Treasury Correspondence*.
CFONB_STATEMENT CSB43_STATEMENT ITAU_STATEMENT NO_AUTOGIRO_BBS NO_OCR NZ_BNZ	For these message types, bank statement processing will be triggered.
DELIVERY_REPORT and TRANSMISSION_REPORT	Implement SAP Note 3841981. Otherwise set the inbound process step in the configuration to *processed*.

Related Information

Treasury Correspondence [page 31]

2.2.2.3 Handling of Unknown Message Types

Manually process unknow message types in *Connector Monitor*.

If messages are received with message types that aren't known to the connector, these messages acquire the status *Manual Processing Required*. A user can then check the payload of those messages in the *Connector Monitor* and perform the relevant manual steps.

Once manual processing is finished, the user can set the message to status *Processed* using the *Completed* button in the *Connector Monitor*.

Related Information

Connector Monitor [page 45]

2.2.2.4 Processing Steps

Define how incoming messages are processed.

Use Customizing activity *Multi-Bank Connectivity Connector* > *Maintain Inbound Processing Steps* to specify processing steps to override the standard processing for known message types as well as define steps for message types that aren't known to the standard connector.

Inbound processing steps can be defined depending on sender/receiver ID and message type.

If an incoming message should be processed by more than one application, you can define several steps for the relevant combinations of sender/receiver ID and message type.

The connector assigns default processing steps to known message types. You can overrule the default processing steps for message types known to the connector or define new processing steps for unknown message types.

Use the Customizing activity *Multi-Bank Connectivity Connector* > *Maintain Inbound Processing Steps* to define processing steps depending on message type, sender ID, and receiver ID.

When you define processing steps with this configuration table, the connector no longer creates any default steps during runtime. So, if you still want to perform the default step in addition to manually defined steps, please be sure to define all relevant steps via this configuration table.

To deactivate processing of a specific step, you can define the step in this configuration table and remove the *active* indicator. Alternatively, you can also assign a generic processing step like *Manual Processing* to set incoming messages to status *Manual Processing Required*.

2.2.2.5 Standard Inbound Handlers

The standard delivery for the connector contains several inbound handlers that can be used for automatic processing of messages from SAP Multi-Bank Connectivity. Inbound handlers are represented by processing

application identifiers within the connector. The following list explains the standard inbound handlers and the respective processing application identifiers.

FI Bank Statements (MT940, MT942)

`FIBNKSTMT1`

This inbound handler can be used to process text-based bank statement messages that aren't in XML format. Payloads are handed over for upload to bank statement processing.

For the processing of bank statements, please make sure you've configured a variant in transaction `FF.5`. Also be sure to assign the relevant variant to the relevant message type in Customizing activity *Maintain Selection Variants for Bank Statements*.

XML or Bank-Specific Formats (e.g camt.053)

`FIBNKSTMX1`

This inbound handler can be used to process xml-based bank statement messages or bank-specific formats that can be processed in transaction `FF.5`. Payloads are handed over for upload to bank statement processing.

For the processing of bank statements, please make sure you've configured a variant in transaction `FF.5`. Also be sure to assign the relevant variant to the relevant message type in Customizing activity *Maintain Selection Variants for Bank Statements*.

BCM Status (pain.002)

`FIBCMSTAT1`

This inbound handler can be used for the automatic processing of pain.002 status messages. Incoming messages are handed over to status processing and can in this way update the status of the relevant payment batches.

BAI Lockbox

`FIBAILOCKB`

This inbound handler can be used for the automatic processing of incoming lockbox messages. Incoming messages are handed over to lockbox processing.

For the processing of lockbox files, please make sure you've configured a variant for report `RFEKA700`. Also be sure to assign the relevant variant to the relevant message type in Customizing activity *Maintain Selection Variants for Bank Statements*.

BAI Statement

`FIBAISTMT1`

This inbound handler can be used for the automatic processing of incoming BAI/BAI2 statement messages. Incoming messages are handed over to the BAI/BAI2 processing component.

For the processing of such statements, please make sure you've configured a variant in transaction `FF.5`. Also be sure to assign the relevant variant to the relevant message type in Customizing activity *Maintain Selection Variants for Bank Statements*.

CFONB Statement

`FICFONBSTM`

This inbound handler can be used for the automatic processing of incoming CFONB bank statements. Incoming messages are handed over to the component for bank statement processing.

For the processing of such statements, please make sure you've configured a variant in transaction `FF.5`. Also be sure to assign the relevant variant to the relevant message type in Customizing activity *Maintain Selection Variants for Bank Statements*.

Bank Fee Analysis

`FICAMT086`

This inbound handler can be used for the automatic processing of incoming camt.086 messages.

Incoming messages are handed over to the component for bank fee analysis for further processing.

Free Text Message

`FREETEXT`

This inbound handler can be used to handle incoming free text messages. Messages acquire the status *Manual Processing Required* and can be checked by users afterwards in the SAP Fiori app *Manage Bank Messages* or the *Connector Monitor*.

Manual Processing

`MANUAL`

This inbound handler can be used to set incoming messages directly to status *Manual Processing Required*.

This can be used in scenarios where a user needs to perform manual activities after receiving a corresponding message.

Set to status *Processed* directly

`PROCESSED`

This inbound handler can be used to set an incoming message directly to status *Processed* without any other activities being triggered.

This can be used in scenarios where information messages are received by the bank that don't require any other actions to be triggered.

CSB43 Statement

`FICSB43STM`

This inbound handler can be used for the automatic processing of incoming CSB43 bank statements. Incoming messages are handed over to the component for bank statement processing.

For the processing of such statements, please make sure you've configured a variant in transaction `FF.5`. Also be sure to assign the relevant variant to the relevant message type in Customizing activity *Maintain Selection Variants for Bank Statements*.

ITAU Statement (Brazil)

`FIITAUSTM`

This inbound handler can be used for the automatic processing of incoming ITAU (Brazil) bank statements. Incoming messages are handed over to the component for bank statement processing.

For the processing of such statements, please make sure you've configured a variant in transaction `FF.5`. Also be sure to assign the relevant variant to the relevant message type in Customizing activity *Maintain Selection Variants for Bank Statements*.

Forward Message

`FORWARD`

This inbound handler can be used to forward an incoming message out of the connector and based on your routing settings. See Routing [page 21] for more information.

Download

`DOWNLOAD`

This inbound handler can be used for the automatic download of incoming messages to an application server directory. For more information about message download, see Downloading of Incoming Messages [page 19].

MBC Status Message

`MSG_STATUS`

This inbound handler is used for the automatic processing of incoming status messages from SAP Multi-Bank Connectivity. These status messages are exchanged between SAP Multi-Bank Connectivity and the connector to update the status of outgoing payment messages once they've been received by SAP Multi-Bank Connectivity or have been forwarded to the bank by SAP Multi-Bank Connectivity.

Whenever a payment message is sent to SAP Multi-Bank Connectivity, a status message is sent back by SAP Multi-Bank Connectivity when it has received the payment message. In the connector, this status message changes the status of the outgoing payment message from *Sent to MBC* to *Received by MBC*. Once SAP Multi-Bank Connectivity has forwarded the payment message to the bank, another status message is sent back to the connector. This status message updates the status of the outgoing payment message to *Forwarded by MBC*.

Ping Test

`PING_TEST`

This inbound handler is used for the automatic processing of incoming ping response messages that are sent by SAP Multi-Bank Connectivity when triggered via a ping request message by the connector.

2.2.2.6 MBC Status Messages

Receive MBC status messages by SAP Multi-Bank Connectivity whenever a payment message is sent to and has been received by SAP Multi-Bank Connectivity.

There are five statuses a message in SAP Multi-Bank Connectivity can have:

- *Sent to MBC*: status that the outgoing payment message receives
- *Received by MBC*: status updated from *Sent to MBC* to *Received by MBC* in the connector for SAP Multi-Bank Connectivity
- *Forwarded by MBC*: status that the message receives once SAP Multi-Bank Connectivity has forwarded the payment message to the bank

- *Failed*: status if something fails during the processing of the message
- *Info*: status of a message that has reference to a payment message and holds additional information

2.2.2.7 Duplicate Check for Incoming Messages

Messages received on the ERP system via the pull process are checked for duplicates.

To check if an incoming message is a duplicate, the system compares the following fields:

- Message ID
- Sender ID
- File Name

If another message is found with the exact same header information in these fields, the new message is flagged as a duplicate. During the pull job, only the duplicate flag is set and retained with the message.

When messages are processed, the duplicate flag is checked and the message with the flag will be ignored for the processing.

Only if the processing mode is set to process duplicate messages, the processing of such messages will be triggered. This can be done, for example, by manually triggering the processing from Connector Monitor [page 45].

Even if messages are ignored during the processing, decryption of these messages will be triggered anyway to make sure that the payload is enabled to be displayed or downloaded, if necessary.

2.3 User Tasks

2.3.1 Monitoring of Bank Messages

Monitor incoming and outgoing messages.

You can use the *Connector Monitor* to monitor incoming and outgoing messages, like status, log or related messages, from SAP Multi-Bank Connectivity.

You can set selection criteria and filters to find the relevant messages. You can monitor the status of incoming and outgoing messages and display the payload of messages as well as the application log for each message.

If errors occur while messages are being sent or received, you can trigger the reprocessing or sending of a message and you can also change the status of messages manually to *Processed* or *Obsolete*.

The following table describes all statuses of messages.

Status	Description	Further Information
IBC80	Ping Request Created	Ping got created but didn't get through to the SAP Multi-Bank Connectivity tenant
IBC81	Waiting For Ping Response	Ping was successfully sent but the tenant hasn't responded yet.
IBC82	Ping Response Received	Ping response successfully received
IBC83	Ping Successfully Finished	Ping test successful, meaning the connection between the connector for SAP Multi-Bank Connectivity and the tenant is fine.
IBC84	Obsolete Message	Message isn't used anymore.
IBC90	Message Created	Once the message is created but not yet sent, this is status is assigned.
IBC91	Message sent to SAP Multi-Bank Connectivity	Created SAP Multi-Bank Connectivity message is successfully sent to the tenant.
IBC92	Message received from SAP Multi-Bank Connectivity	The ERP successfully received an incoming message from the SAP Multi-Bank Connectivity tenant, but it's not yet processed.
IBC93	Message Processed	Incoming message is processed successfully.
IBC94	Error	Error in processing (check the log for details).
IBC95	Manual Processing Required	Incoming message couldn't be automatically processed, and a manual processing is necessary.
IBC96	Message Received by SAP Multi-Bank Connectivity	The SAP Multi-Bank Connectivity tenant successfully received a message from the ERP.
IBC97	Message forwarded by SAP Multi-Bank Connectivity	The message is forwarded by SAP Multi-Bank Connectivity to the receiving party (bank or financial service institution).
IBC98	Processing started	The job to process the statement files is not yet finished.
IBC99	Partially Processed	One or several steps in the inbound processing steps are not completed (check log for details).

Related Information

Connector Monitor [page 45]

2.3.2 Resending Outgoing Bank Messages

Check and resend messages if an error occurred.

If an error occurs while a message is being sent, you can check the message log in the *Connector Monitor*. You can navigate to the relevant message and trigger resending.

In exceptional cases, you can also trigger the sending of messages that have already been sent. For use of this *Send Again* function, special authorizations are required, and the message may not be older than a few days.

Related Information

Connector Monitor [page 45]

2.3.3 Reprocessing Incoming Bank Messages

Trigger reprocessing of incoming bank messages after an error occurred.

If errors occurred during input processing, for example, because an issue with the certificate prevented a bank message from being decrypted, you can trigger reprocessing of the whole message from within the *Connector Monitor*. If a message has already been processed partially, you can trigger reprocessing for just the steps yet to be processed.

Related Information

Connector Monitor [page 45]

2.3.4 Reprocessing of Single Processing Step

Display processing steps and trigger the processing of single steps

If issues occurred during processing of a single step of an incoming message, you can display the processing steps of that message in the *Connector Monitor*. You can trigger processing of single steps from within this dialog.

It's also possible to trigger processing of already processed steps in this dialog. This might be necessary when a message payload has been successfully handed over to the processing application but couldn't be processed there.

Related Information

Processing Steps [page 36]
Connector Monitor [page 45]

2.3.5 Status Update After Manual Processing

Display or download status messages.

Incoming messages that have unknown message types are normally set to status *Manual Processing Required*. In the *Connector Monitor*, you can display or download the payload of such messages to perform necessary manual processing steps. Once the manual processing is done, you can change the status of the respective message by clicking the *Completed* button.

Related Information

Connector Monitor [page 45]

2.3.6 Requesting Bank Statements

If your bank offers this functionality, you can use the *Connector Monitor* to initiate a request of bank statements. To create such a request, you need to specify whether you want to request an end-of-day statement or an intraday statement. You can also specify whether a copy of a statement is requested.

Once a statement request reaches SAP Multi-Bank Connectivity, the relevant statements are requested from the banks. If this is not possible, the connector is informed via an MBC status message.

Statements that are provided as an answer to a request are processed as usual in the connector for SAP Multi-Bank Connectivity. The only exception is that, if a copy of a bank statement was requested, the corresponding bank statement is set to status *Manual Processing Required*.

2.3.7 Creating Free Text Messages

Context

You can create and send free text messages in the *Connector Monitor*. The recipient of the message can be either a bank or *SAP Multi-Bank Connectivity*.

Procedure

1. Open the *Connector Monitor* (transaction code **`/BSNAGT/MONITOR`**).
2. Specify the selection criteria as required and run the *Connector Monitor*.
3. On the *Outgoing* tab, choose the *Create Free Text Message* button.
4. Modify the receiver ID as needed in the *Free Text Message* dialog box.
5. Enter your free text and save your entries.
6. Select the message and choose *Send* to send the message to SAP Multi-Bank Connectivity.

Related Information

Connector Monitor [page 45]

2.3.8 Manual Creation of Bank Messages

Create bank messages manually by uploading local files.

You can create MBC messages by uploading files from your local computer. The uploaded file is copied into the *MessageContent* tag of the MBC message.

When creating messages manually, you need to provide the message type, sender ID, and receiver ID. You also need to specify the local file name. For text-based message types, such as XML formats, for example, `pain.001.001.03`, you need to specify the code page of the uploaded file. If necessary, the system does a conversion so that the message content can be stored in UTF-8 encoded format. For non-text-based message types or unknown message types, the system doesn't do a code page conversion but treats the file as binary. To have the newly created message sent immediately, select the *Send* checkbox.

To create a message manually, choose *Multi-Bank Connectivity Connector* > *Create MBC Message*.

2.4 User Interface

2.4.1 Connector Monitor

Use

The *Connector Monitor* is the central monitoring function for the connector for SAP Multi-Bank Connectivity. You use the *Connector Monitor* to view detailed information about all inbound and outbound messages, and you can also navigate to other functions that enable you to process the messages further.

Features

The *Connector Monitor* has the following features:

- **Message Selection**
  When you run the *Connector Monitor*, the system displays a selection screen where you can choose the messages to be displayed based on various criteria, such as the *Message Type*, the *Receiver ID*, and the *Sender ID*.
- **Detailed Message Information**
  The *Connector Monitor* has two tabs: one for inbound messages and one for outbound messages. Each tab has an SAP List Viewer (ALV) that displays detailed information about the messages.
- *Message Processing*
  The *Connector Monitor* provides the following options for further processing the messages:
  - View the sending or receiving details of a message
  - View the XML content of the message
  - Change the status of an inbound message
  - Send outbound messages
  - Display the related messages
  - Navigate to the application log (for messages with an error status)
  - Navigate to a related business function, such as the *Bank Communication Monitor*
  - Create a text message
- Ping Test
  You can navigate to the *Ping Test* screen. For more information, see Ping Tests [page 8].
- Pull Messages
  You can navigate to the *Pull Messages* screen. For more information, see Pull Messages [page 32].

Activities

To access the *Connector Monitor*, log into your ERP system, choose *Connector for Multi-Bank Connectivity* > *Connector Monitor*.

In the *Connector Monitor*, you can select a message and then perform further processing using the options summarized below.

Procedure

To access the individual functions shown in the table, log into your ERP system, choose *Connector for Multi-Bank Connectivity* > *Connector Monitor*.

Functions in Connector Monitor

Function	More information
Display the sending or receiving details for the selected message	Displays the details of the selected message in the XML monitor.
Display the XML content of a message	Displays the XML content of the selected message in a new dialog. You also have the option to save the message to a local file.
Change the status of an inbound message	Applies to messages that are processed manually.
Send (or resend) outbound messages	Sends the selected outbound messages. These messages can include messages that haven't been sent yet (for example, messages created by the *Pick-up Files* function) and messages being resent due to previous errors.
Display the related messages for a selected message	Opens the *Related Messages* screen.
Navigate to the application log	Opens the application log (only for messages with an error status).
Navigate to the relevant system function for the message	For example, for payment-related messages, the *Bank Communication Monitor* is opened (if active).
Create a free text message	Opens the *Free Text Message* dialog.
Reprocess an inbound message	Applies to inbound messages with an error status.
Display list of processing steps	Opens the *Steps* dialog that shows the processing steps and their status.
Display extended header	Displays the extended header parameters of the selected message.

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.
About the icons:

- Links with the icon  : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering an SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.
The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

www.sap.com/contactsap

© 2026 SAP SE or an SAP affiliate company. 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 SE or an SAP affiliate company. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company 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.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

Please see https://www.sap.com/about/legal/trademark.html for additional trademark information and notices.