



Payment and Bank Communication Management with SAP®

- › Implement in-house banking, advanced payment management, cash management, and more with step-by-step instructions
- › Deploy bank connectivity methods and tools
- › Establish a payment factory in your SAP S/4HANA landscape

Adrian Matys
Jean-Michele Szczecina

Contents

Preface	13
1 Payments Within a Corporation	21
<hr/>	
1.1 Payments as a Business Imperative	22
1.2 Payments with SAP	26
1.2.1 Payments Management Within Financial Accounting	27
1.2.2 Bank Account Management	29
1.2.3 SAP Multi-Bank Connectivity and Bank Connectivity	34
1.2.4 SAP S/4HANA Finance for Advanced Payments Management	35
1.2.5 In-House Banking Versus SAP In-House Cash	42
1.2.6 SAP Bank Communication Management	50
1.2.7 SAP S/4HANA Finance for Cash Management	56
1.3 Payments Processing	59
1.3.1 Payments in the Name Of	59
1.3.2 Direct Debit	61
1.3.3 Intercompany Payments	63
1.3.4 Payments on Behalf Of	71
1.3.5 Collections/Receivables on Behalf Of	76
1.3.6 Cash Pooling	80
1.3.7 Treasury Payments	83
1.3.8 Preprocessing and Postprocessing	88
1.4 Summary	91
2 Bank Connectivity	93
<hr/>	
2.1 Connectivity Standards and Security in Payments	93
2.1.1 SWIFT	95
2.1.2 Host to Host	108
2.1.3 API Usage	108
2.1.4 EBICS	109
2.1.5 Bank Collection and Forwarding Service	120
2.2 Financial Standards	121
2.2.1 Payment Instruments	121

2.2.2	Payment Schemes	130
2.2.3	Payment Systems	137
2.2.4	Payment Formats	142
2.2.5	Bank Fee Analysis	148
2.3	Summary	149

3 Bank Account Management 151

3.1	Functions and Processes	151
3.1.1	Bank Keys	152
3.1.2	Creating Banks	153
3.1.3	Uploading, Migrating, and Replicating Bank Accounts	181
3.1.4	Bank Account Approvals	183
3.1.5	Manage Bank Account Hierarchies App	187
3.1.6	Manage Bank Account Reviews App	188
3.1.7	Closing Bank Accounts	189
3.1.8	Manage Bank Chains App	190
3.1.9	Foreign Bank Accounts App	191
3.1.10	Bank Relationship Overview App	192
3.1.11	Power of Attorney for Banking Transactions	193
3.1.12	Bank Fees	197
3.2	Configuration	201
3.2.1	Define Basic Settings	201
3.2.2	Enable Payment Approval	204
3.2.3	Number Ranges for Technical Accounts ID	205
3.2.4	Define Settings for Bank Account Master Data	206
3.2.5	Define Settings for Bank Account Contract Types	209
3.2.6	Define Field Status Groups	212
3.3	Summary	214

4 Advanced Payment Management and In-House Banking 215

4.1	Advanced Payment Management Functions and Processes	215
4.1.1	Key Components	216
4.1.2	Functions and Apps	230
4.1.3	Master Data	245

4.2	In-House Banking Functions and Processes	260
4.2.1	Prerequisites	261
4.2.2	Account Management Lifecycle	262
4.2.3	General Functions	264
4.2.4	Payments: In-House Operations	282
4.3	Configuration	295
4.3.1	Advanced Payment Management	295
4.3.2	In-House Banking	331
4.4	Integration of Intelligent Services	339
4.4.1	Function and Processes	340
4.4.2	Intelligent Service Integration	341
4.4.3	Configuration	343
4.5	Summary	347
5	SAP Multi-Bank Connectivity	349
5.1	Introduction to SAP Multi-Bank Connectivity	350
5.1.1	System Landscape and Components	351
5.1.2	Cloud Foundry Migration	354
5.2	Connectivity Options	356
5.2.1	EBICS	357
5.2.2	Host-to-Host	360
5.2.3	SWIFT Connectivity	365
5.2.4	Member Banks	366
5.3	Functions and Processes	366
5.3.1	Connector Monitor Transaction	367
5.3.2	Manage Bank Messages App	373
5.3.3	Pull Messages	377
5.3.4	Push Messages	378
5.3.5	Pickup Files	379
5.3.6	Handling of Sensitive Data	382
5.3.7	SWIFTRef Integration	383
5.4	Onboarding	385
5.5	Configuration	387
5.5.1	SAP S/4HANA Finance Configuration and Prerequisites	387
5.5.2	Basic Configuration	392
5.5.3	Inbound and Outbound Processing	402

5.5.4	Business Add-Ins	413
5.5.5	Authorizations	415
5.6	Summary	417
6	SAP S/4HANA Finance for Cash Management	419
6.1	Functions and Processes	419
6.1.1	Bank Account Balances	420
6.1.2	Cash Position	424
6.1.3	Cash Flow Analyzer	425
6.1.4	Short-Term Positioning	438
6.1.5	One Exposure from Operations	446
6.1.6	Manage Memo Records	452
6.1.7	Import Memo Records	454
6.1.8	Bank-to-Bank Transfers	456
6.1.9	Free-Form Payments	459
6.1.10	Cash Concentration	462
6.2	Configuration	467
6.2.1	Planning Levels	467
6.2.2	Planning Groups	470
6.2.3	Liquidity Items	471
6.2.4	Memo Records	472
6.2.5	Cash Pools	476
6.2.6	Define Source Application Accounting	477
6.2.7	Activate Source Application	478
6.3	Summary	479
7	SAP Bank Communication Management	481
7.1	Functions and Processes	482
7.1.1	Payment Approvals	484
7.1.2	Payment Run	487
7.1.3	Batching and Processing Payments	492
7.2	Configuration	498
7.2.1	Basic Configuration	498

7.2.2	Batching Rules Configuration	500
7.2.3	Workflow Activation and Configuration	502
7.2.4	Alerts	506
7.2.5	Useful BAdIs	509
7.3	Summary	509
8	Mapping Format Data	511
8.1	Transition from SWIFT MT to ISO 20022 MX	512
8.2	Functions and Configuration	513
8.2.1	Map Format Data Apps Overview	514
8.2.2	Map Format Data for Incoming Files from Banks	520
8.2.3	Map Format Data for Advanced Payment Management	536
8.3	Summary	538
9	Bank Statements	539
9.1	Bank Statement Processing	540
9.1.1	Overview	540
9.1.2	Bank Statement Reconciliation	543
9.1.3	Bank Statement Monitor	544
9.1.4	Manage Bank Statements	548
9.1.5	Upload Bank Statement	550
9.1.6	Japanese Bank Statements	557
9.1.7	Bank Statement Forwarding Using Advanced Payment Management	559
9.1.8	Intraday Bank Statements	559
9.1.9	Processing Rules	562
9.2	Bank Statement Configuration	565
9.2.1	New Bank Statement Import Program: FEB_FILE_HANDLING	565
9.2.2	Lockboxes	569
9.2.3	Bank Statement Posting	573
9.2.4	Search String Configuration	579
9.2.5	BAdIs	581
9.3	Summary	582

10 The Payment Factory 583

10.1 Designing a Payment Factory	583
10.2 Processes and Functions	588
10.2.1 In-House Banking Processes	590
10.2.2 Payment Factory Functions	595
10.2.3 Payment Orchestration	598
10.3 Quantitative and Qualitative Factors	600
10.4 The Reconciliation Factory	602
10.5 Summary	605

11 Outlook on Payments and Bank Communication with SAP 607

11.1 SAP Solutions	607
11.1.1 SAP Digital Payments Add-On	608
11.1.2 SAP Digital Currency Hub	613
11.2 SAP Outlook by Kolja Ewering	617
11.3 Summary	620

Appendices 621

A ISO 20022 Transformation	623
B Ongoing Regulations	639
C The Authors	645

Index	647
--------------	-----

Chapter 3

Bank Account Management

Bank accounts are the cornerstone of payment processing, and managing them effectively is crucial for smooth financial operations. A variety of activities, including managing bank relationships, revolve around maintaining and utilizing these accounts.

As described in Chapter 1, Bank Account Management (BAM) in SAP S/4HANA serves as the foundation for many other treasury functions. The bank account is the central master data element and the starting point for all treasury-related activities. Without a properly maintained and approved bank account, it is not possible to execute payments, upload or process electronic bank statements, or perform any meaningful cash or liquidity checks. In essence, the entire treasury process landscape depends on the availability and accuracy of bank account data managed within BAM.

In this chapter, we will explore the usage of BAM and its integration within the SAP S/4HANA system, focusing particularly on how SAP Fiori apps are utilized and their impact from a functional perspective. This chapter will highlight the key configuration steps necessary to enable core functionalities, offering insight into the most critical elements that support payment execution and help manage your bank account data.

3.1 Functions and Processes

In this section, we will explain how bank accounts are created and updated in the system, covering the main functionalities and how to configure subaccounts. Although SAP S/4HANA offers many additional features related to BAM, our focus will be on the core functionalities essential for setup and maintenance. This will provide a clear understanding of how to manage bank accounts effectively within the SAP S/4HANA environment.

In this section, we will explore how BAM serves as a central repository for managing all bank accounts in your SAP system and how you can use it as a single point of truth for all your banking relationships. You will gain a clear understanding of the concept of a bank key, its significance within SAP, and how it is essential for uniquely identifying each bank. We will walk you through the process of creating banks for your suppliers and customers, covering the key data and requirements you need to consider. Finally, we'll explain how to manage and establish all your bank relationships, including both

external and internal banks, to ensure streamlined and accurate handling of financial transactions within your SAP S/4HANA environment.



Bank Account Management Versus Electronic Bank Account Management

Electronic BAM (eBAM) represents the automation of key activities between banks and their corporate customers using software. These activities include opening bank accounts, maintaining them—such as changing account signatories or spending limits—closing them, and generating reports required by law or regulation. This automation typically involves generating and sending files to the bank to execute these activities. BAM is a core solution in SAP for managing bank accounts, but it can also be used as an eBAM solution when connected to a bank, providing an integrated platform for automating and managing these processes seamlessly.

In this chapter, we will briefly touch on the eBAM functionalities available within BAM. However, it is not a very popular feature to use. With the centralization of BAM and the implementation of the latest SAP modules, the goal is to reduce the number of bank accounts being used. When there are fewer accounts to manage, implementing an automatic exchange of messages for creating and closing bank accounts becomes less practical and cost-effective.

3.1.1 Bank Keys

In SAP S/4HANA, the bank key is a unique identifier used to represent a bank within the system. It stores essential details about a bank, such as its name, address, SWIFT/BIC code, and clearing information. The bank key is critical for payment processing as it ensures transactions are routed to the correct financial institution. In many countries, it corresponds to a standardized bank code—for example, the sort code in the UK or routing number in the US, or the Zengin code in Japan. The bank key is a critical identifier that represents a specific bank branch, not just the bank itself. This means that the same bank operating in different cities—or even different branches within the same city—can have separate bank keys.

The bank key is a required element when entering any bank account data in SAP S/4HANA. It must be provided not only for bank accounts created in BAM, but also for bank accounts assigned to business partners such as vendors and customers. Without a valid bank key, it is impossible to create or maintain any bank account in the system as it forms the foundation for identifying the banking institution associated with the account.



Creation of Master Data

Preparing master data in SAP S/4HANA, especially data related to bank accounts, can be a challenging and time-consuming task. One of the main difficulties is identifying and assigning the correct bank key for each bank account number. Although the bank key is

often embedded within the bank account number—particularly in countries where an IBAN is used—SAP still requires the bank key to be entered separately in the master data for vendors, customers, and internal accounts.

This requirement remains even though, from a payment execution perspective, the IBAN alone is sufficient in many countries. In SAP S/4HANA, however, the bank key must be explicitly maintained and linked to detailed bank master data, including the address. This means that before you can even load vendor or customer bank accounts, you must first prepare and upload all necessary bank keys, defining them in the system as banks with their corresponding information.

To facilitate this, companies often choose to import SWIFT files or purchase access to external bank master databases to prepopulate their systems. Alternatively, when the actual bank key is unknown, using the SWIFT code as the bank key is a common and accepted workaround, especially in global implementations. Nevertheless, the preparation of accurate and complete bank master data remains a key dependency and potential bottleneck in system setup.

3.1.2 Creating Banks

Banks can be created in SAP S/4HANA using several different methods, depending on the system version and preferred approach. Here are the main options:

- **The Manage Banks for Master Data app (F6437)**
Available in SAP S/4HANA 2023 and SAP S/4HANA Cloud Public Edition. This app is used to create simple banks used for master data.
- **The Manage Banks app (F1574)**
Available for lower versions of SAP S/4HANA (prior to SAP S/4HANA 2023). Offers similar functionality to the Manage Banks for Master Data app but provides more information.
- **Transaction F101**
This classic SAP GUI transaction is used primarily in SAP ERP systems or where SAP Fiori apps are not available. It is not recommended to use this transaction for creating bank keys in SAP S/4HANA.

To open a new bank account, you can choose between two primary approaches in SAP: creating a new bank account directly in the Manage Bank Accounts app or requesting a new bank account using the Submit Bank Account Applications feature. With the first option, you can implement an approval process to manage the creation of new bank accounts effectively, providing better control and governance. Alternatively, the second approach allows you to submit a bank account application for opening a new account using the Submit Bank Account Applications app. After the application is approved through the Approve Bank Account Applications app, an inactive bank account is created and can then be fully processed in the Manage Bank Accounts app.

Which method you choose depends on your organizational requirements, but using approval workflows is recommended to enhance visibility and oversight when opening new bank accounts. In this section, we will describe both approaches and explain how to manage your bank keys, as well as how to prepare for a bank migration.

Submit Bank Accounts

Starting with SAP S/4HANA 2023, there is an option to use the Submit Bank Account Applications app (F5861), which allows users to initiate a structured request for creating a new bank account before the account is created in the system. Historically, you would create a bank account in SAP only after you had contracted with the bank, as the system did not have built-in program processing for account creation. However, SAP has evolved, and it now offers options to automate and streamline this process through BAM. This is especially useful in organizations with multiple teams involved in the bank account lifecycle, such as treasury, accounting, compliance, or local finance departments. Using this app, users can fill in preliminary data and submit the application for review and approval. Once approved, the system can automatically generate an inactive bank account based on the submitted information. When opening a bank account and creating it in the system, there are several prerequisites and activities that need to be completed, including securing internal approvals and finalizing arrangements with the bank. The Submit Bank Account Applications app helps manage this process by guiding you through the necessary steps and facilitating the submission of bank approvals. Only after these approvals are in place can a bank account be officially created and configured in the system, ensuring compliance and proper setup before any financial transactions can take place.

To submit a new bank account request, you can use the Submit Bank Account Applications app. When you open the app, you'll see a list of all existing bank account requests. To create a new request, simply click **Create**. Figure 3.1 shows the main screen where you can submit bank applications. In the new application form, you need to provide an **Application Title** for the request and an **Application Reason** for opening the account. You will also need to specify key details such as the **Company Code**, the **Currency** of the new account, the **Bank Country/Region** where the account will be opened, and the **Contract Type**. There's also an option to upload attachments, which can be used to support the request; for example, you can upload a document explaining the business need for the new account. Once all the required data is entered, click **Create** to submit the request. In this example, the request is for opening a new INR (Indian Rupee) bank account in India.

Once you create the bank account application, it is initially saved as a draft. To initiate the approval process, you also need to submit the bank application for approval.

SAP

Bank Account Application

AM

Create new bank account

Application Information

Attachments

Application

Application Title: *
Created On/At: Submitted On/At:
Create new bank account04/19/2025, 20:19:29–

Created Bank Account

Company Code (Requested): Bank Country/Region (Requested):
1010 (Company Code 1010)IN (India)
Currency (Requested): Contract Type (Requested):
INRCurrent Account

Application Reason

Application Reason *:
Bank account to handle INR payments

Attachments

Attachments (0) Select Document Type to Upload: * FCLM_BAOR Upload

Create Discard Draft

Figure 3.1 Submit Bank Account Application

Approve Bank Account Applications

To approve bank accounts via the approval workflow, use the Approve Bank Account Applications app (F5859). When you open this app, you will see only those requests that are assigned to you for approval, meaning that the system filters the requests based on the workflow configuration and routing logic. You won't see all bank account applications, but only the ones you're authorized to approve in your current workflow step. The system is designed so that depending on the workload setup and user roles, you should not be able to approve the bank account application that you previously submitted. This ensures proper segregation of duties and prevents conflicts of interest. Instead, you should only be able to view bank account applications created by other users, ensuring that approval processes are handled by designated personnel to maintain security and compliance.

Once inside the app, click any visible request to open its details. As shown in Figure 3.2, you will have the following options:

- **Approve** the request to confirm the creation or update of the bank account.
- **Reject** the request to stop the process and mark it as rejected.
- **Edit** the request. This is often used when additional input is required before the account can be finalized. This is particularly useful if approvers need to add data such as the general ledger (G/L) account, validity dates, or other key attributes discussed earlier.

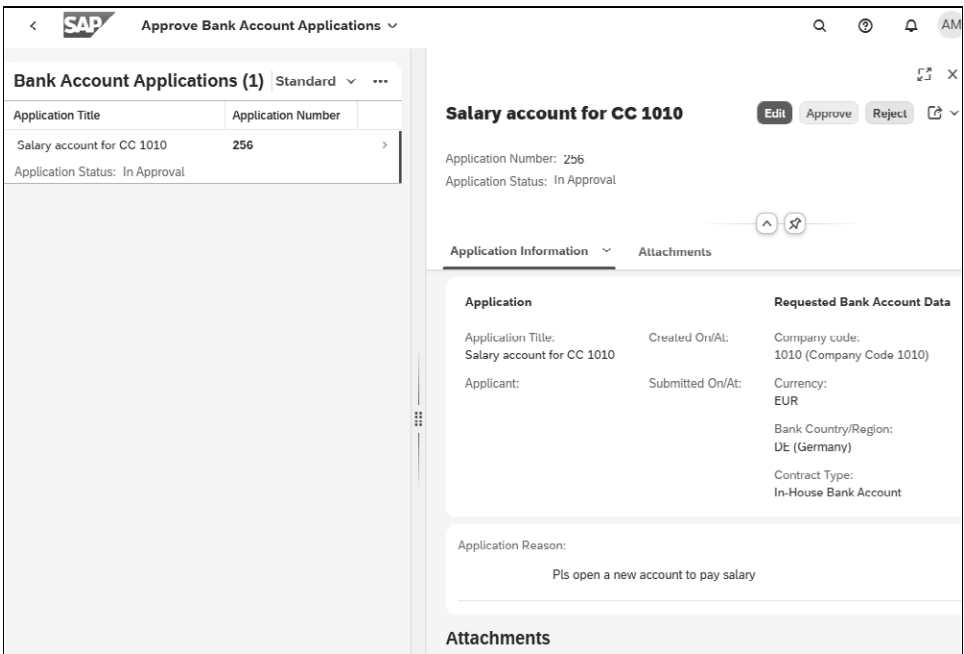


Figure 3.2 Approve Bank Account Application

This **Edit** option is quite common in more complex workflows, where the final approver is also responsible for enriching the bank account with technical or financial details before it's officially created in the system. Once the bank account application is approved, you can proceed with the creation of the new bank and bank account in the system. It's crucial to select the correct **Contract Type** during this process as it determines many further functionalities. As you'll see in Chapter 4, this step is particularly important because it helps differentiate between in-house bank accounts and external bank accounts, ensuring that the appropriate configurations and functionalities are applied to each type.

Once the application is approved, you can proceed with the creation of the bank accounts in the system. This step allows you to set up the necessary bank details and configure the account according to the specified contract type and other relevant parameters.

Create Bank

Let's now walkthrough how to actually create a bank. We'll examine three options for doing so:

- Manage Banks—Master Data app
- Manage Banks app
- Transaction FIO1

Manage Banks—Master Data App

Starting with SAP S/4HANA 2023, you should use the Manage Banks—Master Data app (F6437) to create master data for the banks your company, your business partners, your customers, and your suppliers use to conduct business. When you open this app (see Figure 3.3), you can either check the banks created for a certain country or click **Create** and create a new bank by entering the **Bank Country/Region** and the **Bank Key**.

The screenshot shows the SAP Manage Banks - Master Data app interface. At the top, there's a header with the SAP logo and the app name. Below the header, there's a search bar and several filter fields: Editing Status (set to All), Bank Country/Region, Bank Key, Bank Name, SWIFT/BIC, and Bank Number. A 'Go' button is next to the Bank Number field. Below the filters, there's a table with columns: Bank Country/Region, Bank Key (Bank Name), Bank Number, SWIFT/BIC, Street, City, and Bank Branch. The table is currently empty, and a message says 'To start, set the relevant filters and choose "Go".' A 'Create Bank' dialog box is open in the foreground, showing fields for 'Bank Country/Region' (set to CH) and 'Bank Key' (set to 00206). The dialog box has 'Create Bank' and 'Cancel' buttons.

Figure 3.3 Create Bank

In this example (see Figure 3.4), we will create the UBS bank located on the Paradeplatz in the center of Zurich. After you click **Create Bank** in Figure 3.3, you will need to fill in other information. When creating a bank in SAP S/4HANA, it is important to enter the **SWIFT/BIC** code and bank **Address**, both of which may be required for generating payment files and ensuring accurate payment processing. The SWIFT code ensures international compatibility, while the address (including international versions, e.g., to create addresses in different languages when different alphabets are used) can be used to fulfill country-specific formatting or compliance requirements. Once all the necessary details are entered, you can create the bank.

Upon creation, the system will generate the bank record, and you will be able to view its full details and history (see Figure 3.5), including any changes or updates made over time. This ensures transparency and traceability for audit or reconciliation purposes.

Once the bank is successfully created and saved, it becomes available for use in your counterparty master data, such as vendors, customers, or business partners. This allows you to link bank details to transactions like payments or collections. If you intend to create a house bank (a bank account owned by your company), you can do so using the Manage Banks—Cash Management app (F1574A). This app enables you to define house banks and assign the bank accounts necessary for treasury operations and payment processing; the functionality is similar to that of the classic Manage Banks app.

Figure 3.4 Create Bank Address

Figure 3.5 Change History Create Bank

An alternative option for creating banks in SAP S/4HANA is the Manage Banks app (F1574) (see Figure 3.6), which, although deprecated, is still available in versions after SAP S/4HANA 2023 (and you need to use it if you have a lower SAP S/4HANA version). This app not only allows you to create and maintain banks but also provides extended capabilities for managing house banks. It also enables you to enter more detailed information, such as contract types. Despite being phased out in favor of newer apps, it remains available and a useful tool.

The screenshot shows the SAP 'Manage Banks' app interface. At the top, there's a header with the SAP logo and 'Manage Banks' title. Below it, a 'Standard*' dropdown is visible. The main area contains several filter fields: 'Bank Country/Region:' (set to 'CH (Switzerland) x'), 'Bank Key:', 'City:', 'SWIFT/BIC:', 'Bank Number:', 'Linked House Banks:', 'Linked Bank Accounts:', and 'Mark as Deleted:'. A 'Bank Type:' dropdown is set to 'All'. A 'Go' button and 'Adapt Filters (2)' text are on the right. Below the filters is a table titled 'Banks (1) Standard'. The table has columns: 'Bank Key', 'Bank Country/Region', 'Bank Number', 'Bank Hierarchy Group', 'Business Partner', 'Number of Bank Accounts', and 'Number of House ...'. The first row shows '00206 (UBS)', 'CH (Switzerland)', '00206', and '0' in the last two columns.

Figure 3.6 Manage Banks

This app functions similarly to the one discussed previously for bank creation; however, as Figure 3.7 shows, when you click **Create**, you'll see more detailed fields to be filled in.

The screenshot shows the 'GENERAL DATA' form for bank '00230' in the SAP 'Manage Banks' app. The form is divided into three main sections: 'Control Data', 'Risk Information', and 'Address'. 'Control Data' includes fields for 'Bank Country/Region:*' (CH (Switzerland)), 'Bank Key:*' (00230), 'Bank Name:*' (UBS), 'SWIFT/BIC:' (UBSWCHZH80V), 'Bank Category:' (Standard Bank), 'Bank Group:', 'Bank Number:' (—), and 'Intraday Statements Rule (Bank Level):'. 'Risk Information' includes 'Risk Business Partner:' and 'Rating:'. 'Address' includes 'Region:', 'Street:' (BAHNHOFSTRASSE 45), 'City:' (Zürich), and 'Bank Branch:'. Each field has a copy icon next to it.

Figure 3.7 Manage Banks: General Data

In this app, you enter the same core information as in other bank creation tools—such as the **Address** and **SWIFT/BIC** code—but you also gain access to several additional, important fields, like the following:

■ **Bank Category**

This defines the type of bank being created. The default option is **Standard Bank**, but you can also select categories like **In-House Bank Account** for internal banking operations or **Technical Bank Account** for system-internal postings and testing. The selection here impacts how the bank accounts behave in later processes, which will be explained further in Section 3.2.

■ Intraday Bank Statement Rule

This setting determines whether intraday bank statements are expected from this bank, and how frequently. Its function and configuration will be detailed later in Chapter 9.

■ Risk Business Partner

This is used for integration with the Bank Risk app, allowing risk exposure to be tracked against specific banks.

■ Bank Rating

This field captures the rating of the bank, which is useful for treasury and risk management evaluations.

These additional options allow for more precise and business-aligned bank master data management.

Once you save the bank record, you will be able to provide more information, like **House Banks**, **Contact Info**, **Related Branches**, **Bank Service Mapping**, and **Netting Business Partner**, and you will be able to see the **Change History** information.

One of the most important options available in this app is the ability to create a house bank (see Figure 3.8). When creating a house bank, you must first assign it to a **Company Code** and define a **House Bank ID/code**. The house bank ID will be used later when you create bank accounts. After this, you can provide additional key information that enhances how the bank is used in various processes. This includes **Communication Data**, which specifies how the company will communicate with the bank and who the relevant contact person is. You can also maintain general data, as well as configuration for data medium exchange to enable electronic payment file processing.

Furthermore, you can define the account used for bank charges (**Charges Account** section) and execution (**Execution** section), and assign an **Instruction Key** that indicates, for example, whether the payer or payee will cover the transaction charges. You can also maintain the **Reporting Data** needed for central bank reports, ensuring compliance with regulatory requirements. There is also an option to make the house bank relevant for **EDI Partner Profiles** (relevant for SAP In-House Cash house banks; without this setting enabled, you won't be able to run SAP In-House Cash payments or post SAP In-House Cash statements).

In most cases, only the company code and house bank ID are commonly used and required when creating a house bank. These are the essential details needed to define and use a house bank in standard payment- and bank-related processes. Additional functionalities—such as communication data, instruction keys, reporting details, or EDI partner profiles—are optional and being slowly deprecated by SAP; they're typically used only for very specific business or regulatory requirements.

In the classic SAP In-House Cash solution, the bank accounts created in BAM and used for payments (such as those generating PAYEXT or DIRDEB IDocs) must be marked as EDI-relevant, and the associated payment methods must also be flagged as EDI-relevant. If these settings are not maintained, IDoc generation in SAP In-House Cash is

not possible. In contrast, with in-house banking, these EDI-relevant flags are no longer required, simplifying the configuration and execution of payments.

HOUSE BANK

House Bank Data

Company Code:*
1010 (Company Code 1010)

House Bank:*
UBS01

Control Data

Bank Country/Region:
CH (Switzerland)

Bank Key:
00230 (UBS)

Address

Bank Name:
UBS

Region:
-

Street:
BAHNHOFSTRASSE 45

Bank Branch:
-

Communications Data

Telephone:

Tax Number:

Contact Person:

General Data for Data Medium Exchange

Alternative Bank ID:

Bank No. Receiver:

Customer Number:

Company Number:

Charges Account

Bank Number:

Account Number:

Control Key:

ISO Currency Code:

Execution

Instruction Key:

Lead Days:

Reporting Data

Create Central Bank Report:

Area Number:

Forward Payment Data to the Central Bank:

EDI Partner Profiles

Partner Number:

Figure 3.8 Create House Bank

In addition, you can create contact information within the house bank setup to indicate who is responsible for managing the bank account on the bank's side, as well as who serves as the internal contact person from your organization. However, it's important to note that the contact person must first be created as a business partner in the system before they can be assigned to the house bank.

In many cases, bank address data is not straightforward; banks may have multiple addresses—such as branch-specific addresses, headquarters addresses, or addresses in different languages using special characters. SAP S/4HANA provides advanced functionalities to handle these scenarios, allowing users to define detailed and flexible address data. This is especially important because accurate and complete address information is often required to ensure smooth payment processing on the bank's side.

Such functionalities are available when creating a bank in SAP S/4HANA via the advanced address feature (as shown in Figure 3.9). To access this option, you first need to create a bank, then open it for editing and click the **Advanced Address** button. This option allows you to enter more detailed and structured address information beyond the standard fields. With advanced address, you can capture additional data such as the building name, floor number, district, PO box details, and more granular location specifics.

Figure 3.9 Advanced Address

This can be especially helpful in scenarios such as the following:

- **Payment file generation**

Some formats or local regulations may require extended address data to ensure the payment is accepted by the banking system.

- **Avoiding delays**

Providing a full and accurate address helps avoid rejections or delays due to incomplete information.

- **Compliance and reporting**

For certain jurisdictions, especially in regulated industries, full address details might be required for audit, compliance, or central bank reporting purposes.

- **Intercompany or in-house bank structures**

When managing a large number of banks and bank accounts, the advanced address structure helps to clearly differentiate and organize data.

Using advanced address information enhances data quality and reduces the likelihood of errors or missing fields during payment processing and reporting.

In the Manage Banks app (F1574), you also can view all the bank keys and banks that have been created in the system, along with their corresponding bank category. This information is particularly important when working with SAP In-House Cash, in-house banking for SAP S/4HANA Finance for advanced payment management, or any other technical accounts that will be used for specific purposes within your organization. The bank category helps determine how each bank account behaves in the system and ensures that the correct category is assigned, which is essential for proper configuration and usage.

Figure 3.10 shows how the Manage Banks app can be used for reporting functionalities. In this view, you also can see how many house banks and bank accounts have been created for the selected bank.

Bank Key	Bank Country/Region	Bank Number	Business Partner	Bank Category	Number of Bank Accounts	Number of House ...
90900000 (Commerzbank)	NL (Netherlands)			(Standard Bank)	1	1 >
00206 (UBS)	CH (Switzerland)	00206		(Standard Bank)	0	0 >
00230 (UBS)	CH (Switzerland)	00230		(Standard Bank)	0	0 >
00270 (UBS)	CH (Switzerland)	00270		(Standard Bank)	0	0 >
99 (In-House Bank Area B)	NL (Netherlands)	99		1 (In-House Cash Center)	0	0 >

Figure 3.10 Manage Banks

Transaction F101

Finally, as an alternative to using the SAP Fiori apps for bank creation, you can use the classic SAP GUI Transaction F101 (see Figure 3.11). This transaction allows you to manually create banks by entering the required information such as bank key, name, address, and SWIFT code. If you are still working in an SAP ERP system, Transaction F101 is the only option to create a bank, but the transaction is also still available in SAP S/4HANA.

Bank Ctry/Reg.: CH

Bank Key: 00270

Figure 3.11 Transaction F101: Create Bank

Transaction F101 is very straightforward to use. You simply need to enter the **Bank Ctry/Reg.** and the **Bank Key** to begin the bank creation process. Once these two fields are filled in, the system will allow you to proceed with entering the remaining bank master data, such as the **Bank Name**, **Address**, and **SWIFT/BIC** code, as shown in Figure 3.12.

The screenshot displays the 'Create Bank : Detail Screen' in SAP. The interface includes a header bar with the SAP logo and a title bar. Below the header, there are several input fields organized into sections. The 'Bank Country/Region' is set to 'CH' (Switzerland), and the 'Bank Key' is '00270'. The 'Address' section contains fields for '* Bank Name' (UBS), 'Bank Branch' (Wiedikon), 'Street/House Number' (BIRMENSCHWANDSTRASSE 123), 'Postal Code/City', 'Country/Region' (CH), and 'Region'. The 'Control data' section includes 'SWIFT/BIC' (UBSWCH33HAN), 'Bank Number' (00270), and 'Bank Group'.

Figure 3.12 Transaction FI01: Additional Bank Data

When using SAP S/4HANA version 2022 or higher, the system automatically creates all banks in the background as standard banks, meaning that the bank category (called the *contract type* in previous versions) is set to **Standard Bank**. This default behavior has important implications for future configuration activities. The bank category determines what kind of functionality and configuration options are available for that bank (such as the ability to switch off workflow approvals for technical or SAP In-House Cash accounts).



Change of Contract Type

It is not possible to change the bank category after the bank key has been created, so it is essential to choose the correct category from the beginning, especially when setting up house banks for SAP In-House Cash/in-house banking (see Chapter 4 to learn about the link between BAM and in-house banking accounts) or other technical purposes. This means that when you use SAP S/4HANA 2023 and up, you cannot use these functionalities and will need to use the newest apps. Otherwise, it won't be possible to determine the correct contract type.

Bank Migration

As you can see, when implementing a new SAP S/4HANA system, you may have to migrate hundreds of thousands or even millions of bank records, which is virtually

impossible to manage if you need to enter them one by one in the SAP Fiori app. However, there are several options available to streamline the migration process:

- **Using Transaction FIO1**

This transaction can be used for uploading bank master data in older systems or even in SAP S/4HANA. With SAP Legacy System Migration Workbench, you can map the fields from your source system to SAP S/4HANA and upload large sets of bank master data. However, it is recommended to use SAP Legacy System Migration Workbench only in SAP ERP systems as it may limit the ability to enter a lot of information, such as certain bank configurations or additional fields required in SAP S/4HANA. Using SAP Legacy System Migration Workbench in newer systems can restrict the full use of features and fields that are available in the SAP S/4HANA environment.

- **Using the SAP S/4HANA migration cockpit for bank master data**

In SAP S/4HANA systems, you can use the migration cockpit to load bank data into your new system in bulk, without having to manually create each bank key or bank record. You can prepare a data file and then use the standard migration objects provided by SAP.

- **Using a BAPI**

Alternatively, you can write a custom program using the `BAPI_BANK_CREATE` BAPI to automate the creation of banks and bank keys in the new system. This BAPI provides a standardized way to create banks and their details programmatically. This approach is highly beneficial when dealing with large volumes of data and complex migrations.

These options provide flexibility depending on your specific needs and the tools available in your SAP environment, ensuring that large-scale migrations can be performed efficiently.

Manage Bank Accounts

As mentioned previously, banks (and their corresponding bank keys) must be created in the SAP S/4HANA system for all relevant financial institutions, including those related to vendors, customers, and any other counterparties your organization interacts with.

The next step is to create house banks and corresponding bank account IDs for the banks your company uses for its own financial operations. These house banks and account IDs are critical because they will be used for executing payments, receiving funds, performing bank reconciliations, and supporting overall cash and liquidity management within the SAP S/4HANA system. Creation of, changes to, and closure of bank accounts is handled in the Manage Bank Accounts app (F1366A).

Figure 3.13 shows how you can display and manage your organization's own bank accounts, maintain bank relationships, and adjust their associated attributes and settings. This app serves as a central point for viewing and updating details such as account

numbers, currency, bank assignment, and usage information. To work efficiently within the app, you need to define selection criteria—such as **Company Code**, **House Bank**, or bank **Account Number** or, in the example shown, the **Bank Country/Region** information for where the bank accounts are located. This lets you filter the results and display the relevant bank accounts already created in the system.

Account Description	Account Curr...	Account Number	Account Status	Account Type	Bank Country/Region	IBAN	Company Code
Dummy text	CHF	23050422318T	02 (Active)	01 (Current Account)	CH (Switzerland)	CH350023023050422318T	2210 (Company Code 2210)
Dummy text	CHF	25-9779-8	02 (Active)	01 (Current Account)	CH (Switzerland)	CH6309000000250097798	2210 (Company Code 2210)
Dummy text	EUR	22595291301G	02 (Active)	01 (Current Account)	CH (Switzerland)	CH510022522595291301G	2210 (Company Code 2210)
Dummy text	EUR	25-9034-2	02 (Active)	01 (Current Account)	CH (Switzerland)	CH0309000000250090342	2210 (Company Code 2210)
Bank 3 Giro	CHF	012345678009	02 (Active)	01 (Current Account)	CH (Switzerland)	CH5604839012345678009	2210 (Company Code 2210)
PayPal	CHF	999999911	02 (Active)	02 (Current Account (Internal))	CH (Switzerland)		2210 (Company Code 2210)

Figure 3.13 Manage Bank Accounts View

From the Manage Bank Accounts app, you have access to a wide range of functionalities beyond just displaying account details. You can import and export bank accounts, manage bank account hierarchies, create and attach documents, and initiate review and approval workflows. In addition, you can directly navigate to related SAP Fiori apps such as Manage Bank Statements, Cash Flow Analyzer, Manage Bank Account Balances, and Monitor Bank Account Balances. You also can create new bank accounts or delete existing ones directly from this interface. We will describe all these SAP Fiori apps and features in more detail in Chapter 6; for now, we will focus specifically on the creation of new bank accounts within this app.

Figure 3.14 shows what happens when you click the **Create** button. The system will start the creation of the new bank account and will ask you first to select the **Company Code** for which you will create the bank account and the **Account Type**. In this example, the account type is **01**, for a current bank account (representing an external bank account; the values in this dropdown list are from the configuration described in Section 3.2.4). The account type is created in the configuration and determines the fields available and the behavior of the account.

Create

Company Code: *

1010

Account Type: *

01

Continue Cancel

Figure 3.14 Create Bank Account

When creating a new bank account (see Figure 3.15), the first step is to fill in the **Header** data. This includes entering the **Account Description** and bank **Account Number**, and linking the account to the **Bank Country/Region** and **Bank Key** that you previously created. You'll also need to enter the **Bank Control Key** and account **Currency**. If the bank account is for a country that uses IBAN (i.e., an "IBANized" country), you can simply click **Generate IBAN**, and the system will automatically propose an IBAN based on the entered data. If the system is unable to generate the IBAN—for example, due to missing formatting rules—then you can always enter the IBAN manually.

The screenshot shows the SAP Bank Account Header Data entry screen. The title is "UBS Current Account USD" with the account number "439743019999". The "Header" tab is selected. The form contains the following fields and values:

- Account Description: UBS Current Account USD
- Bank Country/Region: CH (Switzerland)
- Currency: Swiss Franc (CHF)
- Account Number: 439743019999
- Bank Key: 00230 (UBS Bank)
- Account Status: 01 (Inactive)
- Bank Control Key: 35
- IBAN: CH3500230439743019999

There are two buttons at the bottom right: "Generate Account..." and "Generate IBAN".

Figure 3.15 Create Bank Account Header Data

Creation of the Bank Account

As previously mentioned, the contract type (bank category) cannot be selected or changed at this stage. It is automatically derived from the bank creation step completed earlier. Once a bank is created with a specific contract type, it is fixed and cannot be modified later, so it's important to ensure that the correct category is chosen during the initial bank setup.

Also, note that in SAP S/4HANA you can assign only one currency to each bank account. In practice, many banks offer multicurrency accounts, allowing multiple currencies to be handled under a single account number—but SAP S/4HANA does not support this functionality directly.

To manage multicurrency accounts in SAP S/4HANA, the common approach is to create one house bank and then multiple bank accounts under that house bank, each with the same account number but a different currency assigned. This allows for accurate processing and reconciliation (however, you will need to set up payments and configuration for each bank account separately) across various currencies while maintaining alignment with how SAP S/4HANA structures bank account data.

The next pieces of information to be checked or entered is under **General Data** (as shown in Figure 3.16). Here, the system automatically derives certain information from

the company code selected earlier. You can also specify the **Account Holder**, which is a free-text field that allows you to enter the name of the entity owning the account. This can be either an SAP-registered entity or any external party. You also can define the **Profit Center** and **Segment**, which are particularly useful for reporting and can also be automatically derived during bank statement reconciliation processes. You will also select the **Account Type**, which determines how the bank account behaves within the system (whether it's a current account, loan account, etc.). Another key field is **Opening Date**. This date is crucial because no transactions can be carried out before it (there would be a lot of errors in various places in SAP S/4HANA: payments, statements, treasury and risk management, and cash management). Once the account is created and approved, this opening date, along with other key data, usually cannot be changed, thus ensuring the consistency and auditability of bank master data.

General Data					
Entity Details		Account Details		Bank Details	
Company Code:	Controlling Area:	Account Type:	Technical ID:	Bank Number:	Street:
1010 (Company Code 1010)	A000 (Controlling Area A000)	01 (Current Account)	1000000253	–	Zurich
Company Country/Region:	Profit Center:	Opening Date:	Financial Object Number:	SWIFT/BIC:	City:
DE (Germany)		04/19/2025	–	UBSWCHZH80V	8001
Company Currency:	Segment:			Region:	Bank Branch:
European Euro (EUR)				–	Paradeplatz 6
Account Holder:					
Company Code 1010					

Figure 3.16 Create Bank Account General Data

The **Technical ID** is automatically generated by the system once the bank account is created and serves as a unique identifier for internal system processes. The **Financial Object Number** is particularly relevant for SAP Treasury and Risk Management; it's used to link the bank account to financial objects for risk analysis and reporting. If this functionality is configured in your system, the number will be derived automatically. All of the information under **Bank Details**—such as the bank name, address, SWIFT code, and bank key—are pulled directly from the bank record that was created in the previous step, ensuring consistency and avoiding redundant data entry.

Figure 3.17 shows the next tab in the bank account, **Bank Relationship**, which is very important and affects many functionalities in SAP S/4HANA.

What you select at this stage significantly impacts various functionalities later in the SAP S/4HANA system. Each field and value chosen here—**Account Type**, **Intraday Bank Statement**, **Balance Update Method**, and so on—can influence how the bank account behaves across different modules, from cash management and payment processing to reconciliation and reporting. In Chapter 6 and Chapter 9, we'll take a closer look at each of these settings and how they are derived, and we'll explain how they affect system behavior.

Bank Relationship

Bank Statement General

Bank Statement Group:

Tolerance Group for Reconciliation:

Bank Statement Forwarding:

No Forwarding

End-of-Day Bank Statement

Processing Status:

☒

Difference Status:

☒

Serial Number Status:

☒

Reconciliation Status:

☒

Interval:

1

Interval Unit:

1 (Calendar Days)

Intraday Bank Statement

Upload of Intraday Statements:

☐

Intraday Statements Rule (Account Level):

Intraday Statements Rule (Bank Level):

-

Planning Level (Memo Record):

Reconcile Intraday Memo Records:

☐

Posting Category (Intraday):

No Posting Processing

Balance Update Method

Enable Balance from Manual Entry:

No

Enable Balance from Spreadsheet Import:

No

Enable Balance from EPIC:

No

Contact Persons

Internal Bank Account Contact:

Internal Bank Account Supervisor:

Figure 3.17 Bank Account: Bank Relationship

In the next step, shown in Figure 3.18, you need to select the **Bank Statement Group**, which is defined in the system configuration. This grouping is used to organize bank statements, allowing them to be displayed and managed at an aggregated level. You must also specify the **Tolerance Group for Reconciliation**, which determines the acceptable tolerance levels when reconciling line items during the bank statement reconciliation process. This helps streamline reconciliation and ensures consistent handling of small discrepancies.

Bank Statement General

Bank Statement Group:

UBS (UBS Bank Statement)

Tolerance Group for Reconciliation:

Bank Statement Forwarding:

No Forwarding

Figure 3.18 Bank Account: Bank Statement General

In addition, there is an option for **Bank Statement Forwarding**, which defines whether the imported bank statement files—both end-of-day and intraday statements—should be automatically forwarded to subsequent systems (e.g., other SAP systems, legacy environments, or third-party applications). This functionality is particularly relevant in complex system landscapes and is governed by configuration settings.

The **End-of-Day Bank Statement** section (see Figure 3.19) is a critical component for leveraging the Bank Statement Monitor app, which is explained in detail in Chapter 9. In this section, you define how frequently you expect to receive the end-of-day bank statements by specifying the **Interval Unit** (e.g., daily, weekly), and this frequency is monitored against the selected **Factory Calendar**. This ensures that expected receipt dates align with working days and holidays based on the relevant calendar. This expectation forms the basis for monitoring whether statements are received on time and in accordance with your reconciliation processes.

End-of-Day Bank Statement

Processing Status:

☒

Difference Status:

☒

Serial Number Status:

☒

Reconciliation Status:

☒

Interval:

Interval Unit:

Factory Calendar ID:

Position:

Difference Amount:

Reconcile End-of-Day Memo Records:

☒

Posting Category (End of Day):

Transfer Mode:

Figure 3.19 Bank Account: End-of-Day Bank Statement

This setup enables the system to track several key statuses within the Bank Statement Monitor app, including the following:

■ **Processing Status**

Whether the bank statement has been imported and processed successfully.

■ **Difference Status**

Reflects any discrepancies between the G/L account balance and the balance reported in the bank statement.

■ **Serial Number Status**

Checks for gaps or duplication in the sequence of incoming statements.

■ **Reconciliation Status**

Indicates whether the items in the statement have been matched and cleared properly against the corresponding G/L entries.

■ **Reconciliation Differences**

These define the acceptable thresholds for automatic reconciliation of items.

■ **Reconciliation of End-of-Day Memo Records**

This option, used by certain apps, enables you to also reconcile memo records that appear at the end of the day with the forecasted flows. This is purely a treasury forecasting activity and is independent of the accounting reconciliation.

Moreover, you define the **Posting Category**, which indicates whether the statement should be posted automatically or kept for review. This influences the overall automation level of the bank statement processing.

Finally, the **Transfer Mode** setting determines how cash flows are updated in cash management. Both the posting category and the transfer mode come from the cash management setup and are not editable directly at the BAM level. These settings play a

crucial role in determining how data flows through SAP's treasury and cash management functions and directly impact downstream processes and visibility into cash positions.

Under the **Intraday Bank Statement** section, shown in Figure 3.20, you first indicate that intraday bank statements are expected and will be uploaded for the specific bank account by selecting the **Upload of Intraday Statements** checkbox.

The screenshot shows a configuration form titled "Intraday Bank Statement". It contains the following fields and options:

- Upload of Intraday Statements:** A checkbox that is checked.
- Intraday Statements Rule (Account Level):** A dropdown menu showing "UBS (Intraday Bank Statements ...)" with a selection icon.
- Intraday Statements Rule (Bank Level):** A dropdown menu showing a hyphen "-" with a selection icon.
- Planning Level (Memo Record):** A dropdown menu showing "ID (Intraday Bank Statement)" with a selection icon.
- Reconcile Intraday Memo Records:** A checkbox that is checked.
- Posting Category (Intraday):** A dropdown menu showing "No Posting Processing".

Figure 3.20 Bank Account Intraday Bank Statement

If enabled, several critical configurations follow:

■ Intraday Statement Rule

This defines how frequently you expect to receive intraday statements throughout the day (e.g., every hour, multiple times per day). It helps structure your expectations and set up monitoring properly in Bank Statement Monitor and related apps. This rule is created in Transaction FCLM_BRM_RULE or the Define Monitoring Rules – Intraday Statements app.

■ Planning Level

This determines the level at which line items from the intraday statements are posted into cash management. It affects how cash positions are visualized and analyzed across planning hierarchies.

■ Intraday Memo Records Reconciliation

Here you specify whether you will reconcile memo records from intraday statements using the dedicated Reconciliation of Memo Records app. This improves cash visibility and helps identify temporary entries or holds.

■ Posting Indicator

This setting specifies whether actual postings should be made in the system based on the content of intraday statements. This is especially relevant when real-time updates to cash positions or ledger postings are needed.

If you select and configure these entries correctly and proceed with creating them, you will enable the full functionality related to intraday bank statements. This includes the ability to receive and process intraday updates from your banks, as well as to use the Bank Statement Monitor—Intraday app (F3671) and to perform memo record reconciliation in dedicated SAP Fiori apps.

The balance update methods for bank accounts (see Figure 3.21) are predefined in the Define Bank Account Settings—Bank Statements app (F5488) and cannot be changed or selected directly within the Manage Bank Accounts app. These methods determine how bank account balances can be updated in the system and are typically assigned based on company code requirements.

Balance Update Method

Enable Balance from Manual Entry:

No

Enable Balance from Spreadsheet Import:

No

Enable Balance from EPIC:

No

Figure 3.21 Bank Account: Intraday Balance Update Methods

The available balance update methods include the following options:

- **Enable Balance from Manual Entry**
Allows users to manually input balances for bank accounts using the Manage Bank Account Balances app. This is useful for quick updates or ad hoc adjustments.
- **Enable Balance from Spreadsheet Import**
Enables users to upload a spreadsheet to update bank account balances in bulk via the same app. This method is efficient for handling multiple accounts at once.
- **Enable Balance from EPIC**
Allows automatic updating of bank account balances using an integration specific to the Chinese market—the Electronic Payment Integration for China (EPIC) functionality.

In the **Contact Person** section of the bank account (see Figure 3.22), you can define both internal and external contacts related to the selected account:

- For external contacts, you should assign the person responsible for the bank account on the bank's side. This person must be created in the system as a business partner.
- For internal bank account supervisors, you need to define relevant system users who will be responsible for managing the bank account internally. To do this, create business partners with the role category BUP003 (Employee), and then assign the system user IDs to these business partners. Some older SAP S/4HANA versions might require employees to be created with HR roles.

Contact Persons

Internal Bank Account Contact:
MA

Internal Bank Account Supervisor:
TES1

Figure 3.22 Bank Account: Contact Person

Now, scroll down to the **House Bank Account Connectivity** section, shown in Figure 3.23. Here you create a **House Bank** and **House Bank Account ID** and assign these to the selected bank account (by entering a value here in **House Bank Account**; or can you add this later by clicking the line for the correct account). A house bank in SAP represents your company's own bank within the system. It acts as an internal structure that organizes and manages your external bank accounts. The house bank is created at the bank level and is always linked to a specific bank key, which uniquely identifies a bank branch (via a routing number, sort code, Zengin code, etc.).

House Bank Account Connectivity (1) Standard

Manage House Banks Create Delete

ID Category	Remote System	Company Code	House Bank	House Bank A...	Additional ID	Valid From	Valid To
01 (Central System: House Bank Acco...		1010				04/19/2025	12/31/9999

Figure 3.23 Bank Account House Bank Account Connectivity

The house bank is not a bank account itself, but a logical container that groups one or more bank accounts (bank account IDs) for a particular company code. It helps with account management, payment processing, and reconciliation. You create the house bank in the Manage Banks—Cash Management app or the Manage Banks app during the earlier setup phase, when you also provide the company code and define the house bank ID. Once the house bank is created, you can proceed with assigning it to the relevant bank account ID, finalizing the structure that links your SAP S/4HANA bank account to an actual bank institution for operational use.

Figure 3.24 shows how to create a new bank account after clicking the **Create** button.

Once you click **Create**, you'll be prompted to select an option for **ID Category**, which defines how the bank account is managed:

- **00 Central System: G/L Account**

Used when you have only a G/L account centrally and the account is managed in the local system; this is not a popular option.

- **01 Central System: House Bank Account**

The bank account is fully managed in the current system. This is the most common setup when all postings and bank-related processes happen in the same SAP S/4HANA system and should be used for most scenarios.

■ **02 Remote System: House Bank Account**

The bank account is managed in a different system; you're only bringing the account details into this system for central visibility or consolidation purposes (e.g., for centralized cash management or reporting).

■ **03 Remote System: G/L Account**

You store the bank account centrally for reporting purposes, but the G/L account is created only in the local system, together with corresponding postings.

■ **04 Others**

Any other specific scenario not covered by the other options.

The screenshot shows the SAP 'New House Bank Account Connectivity' form. The form is titled 'New House Bank Account Connectivity' and has a 'Delete' button. It contains several sections: 'Connectivity Path' (selected), 'House Bank Account Data', 'Bank Details For Returned BoE Payable', and 'Bank Details For Returned Presentation'. The 'House Bank Account Data' section includes fields for 'ID Category' (01), 'Company Code' (1010), 'House Bank Account' (USD01), 'Valid From' (04/19/2025), 'Valid To' (12/31/9999), 'Risk Business Partner' (—), 'Netting Business Partner' (—), and 'Use in Cash Pooling' (checkbox). The 'Bank Details For Returned BoE Payable' section has a 'Returned BoE Payable' field. The 'Bank Details For Returned Presentation' section has three sub-sections: 'Details For BoE Presentation', 'Account for Debit Memos', and 'Account for Credit Memos'. An 'Apply' button is at the bottom right.

Figure 3.24 New House Bank Account Connectivity

House Bank is automatically derived based on the bank and bank key you selected earlier. You'll then need to do the following:

- Assign a **House Bank Account ID**, which uniquely identifies the specific bank account within the house bank.
- Define the validity period by entering the **Valid From** and **Valid To** dates.
- Select **Use in Cash Pooling** if you want this account used in cash pooling.
- Select the **G/L Account** that the bank account will post to. This G/L account will be used for all financial postings related to this specific bank account.

Other house bank account fields such as **Alternative Bank Account Number**, **DME Identification**, or **Reference Information** can be used to support specific bank statement processing scenarios. These fields become especially useful when you have one of the following situations:

- The bank sends additional identifiers in the bank statement (e.g., a different account number than the one maintained as the main account number).
- You need to use data medium exchange (DME) identification to differentiate between various payment files or bank accounts.
- The **Reference Info** can help with matching incoming bank statement lines to the correct internal bank account, particularly when banks include reference data that doesn't directly match your primary account settings.

House Bank and Bank Account ID

It's important to note that both the **House Bank Account ID** and the **Bank Account ID** can contain a maximum of five characters each. Because of this limitation, it's highly recommended to establish a consistent naming convention across the entire system to maintain clarity and avoid confusion.

This logic should be simple yet informative—ideally allowing users to identify key details about the account at a glance. A common and effective practice is to include abbreviations for the bank name and the currency. For example, an account at HSBC in USD could be named HSUSD, while one at Citi in EUR might be CTEUR.

Defining and adhering to a system-wide standard ensures better governance and simplifies ongoing bank account maintenance, reporting, and integration activities across different areas of SAP S/4HANA and company codes.



G/L Accounts Used for Bank Accounts

In the classic approach, typically, each bank account is associated with its own G/L account, and at least one bank clearing account is created for each bank. However, with the new approach introduced by SAP S/4HANA, there is now the concept of a single G/L account for all bank accounts within a particular company code. This allows for greater simplicity in the chart of accounts by reducing the number of G/L accounts needed.

Before moving forward, it is important to verify if your system supports this functionality and whether it will enable you to perform bank reconciliations with a single G/L account. If this feature is available and you can reconcile your bank accounts using just one G/L, it is highly recommended as it simplifies the overall financial structure and reduces the complexity of managing multiple G/L accounts for different bank accounts.



When you further scroll down your screen in the Manage Bank Accounts app, you will see the **Payment Approvers** tab (see Figure 3.25), where you can assign payment approvers and define which groups or individuals are authorized to approve payments based on specific amounts. If the approval process is not dependent on the amount, you can select the **Unlimited Amount** option. Here, you will also specify the validity period for the payment approver and define which person or group is responsible for each step in the approval process.









Payment Approvers (2) Standard ⁺													Active Payment Approvers (2)		All Payment Approvers (3)		Show Approval Steps		Delete	📄	🔍	⚙️	👤
	Payment Approver Group	Payment Approver	Unlimited A...	Maximum Amount for Payment		Maximum Amount for Batch		Valid From		Valid To		Appr Grp Description	Currency										
<input type="radio"/>			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	e.g. 12/3...		e.g. 12/3...												
<input type="radio"/>	G002	 S4H_CASH	<input type="checkbox"/>	<input type="checkbox"/>	100,000.00	USD	1,000,000.00	USD	10/20/2017	12/31/9999	Two Steps Approval - 2nd Step		USD										
<input type="radio"/>	G001	 S4H_FIN	<input type="checkbox"/>	<input type="checkbox"/>	100,000.00	USD	1,000,000.00	USD	10/20/2017	12/31/9999	Two Steps Approval - 1st Step		USD										

Figure 3.25 Bank Account Payment Approver

By clicking **Show Approval Steps**, you will be able to view the configured approval flow, including who is responsible for which step of the payment approval process and the corresponding approval patterns. This allows for clear visibility into and control over the payment approval workflow, ensuring that the right people are involved at the right stages of the process.

Note that from the BAM perspective, the payment approval rules can only be created based on amounts. In addition, you need to maintain the individuals and groups responsible for approvals for each bank account. If your approval process requires more complex rules or logic beyond simple amount-based conditions, you would either need to implement custom development or utilize SAP Bank Communication Management or advanced payment management for more advanced payment approval workflows and processing. These solutions provide greater flexibility and customization for handling payment approvals and related processes.



Maintain Payment Approvals for Multiple Bank Accounts App

There is also an alternative SAP Fiori app called Maintain Payment Approvals for Multiple Bank Accounts (F1372), shown in Figure 3.26, which allows you to create and maintain payment approval rules for more than one bank account at the same time.

Once you open the app, you can select the bank accounts you want to modify. Then, click **Add Payment Approver** to assign a new approver. You will be able to select the **Approver** and **Approval Group**, define the **Amounts** and **Validity**, and assign the **Payment Approval Group**. Alternatively, you can select multiple bank accounts and use the **Change Payment Approver** option to update the approvers for the selected accounts. You also can update the overdraft amount limits, or remove approvers entirely by clicking **Revoke Authorization**.

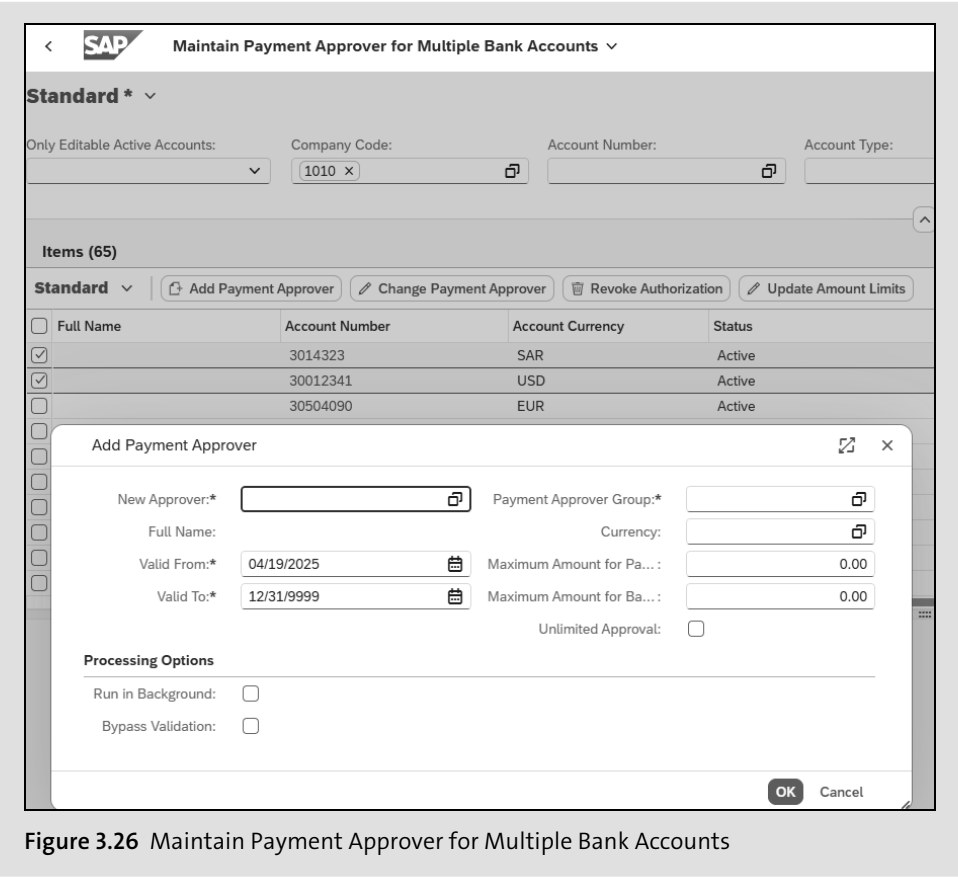


Figure 3.26 Maintain Payment Approver for Multiple Bank Accounts

The next step in maintaining the bank account, shown in Figure 3.27, is defining the overdraft limits. Here, you can specify the overdraft limit for a particular bank account, including the validity period, the overdraft amount, and the currency. These overdraft limits are checked and utilized in various applications within cash management to ensure proper monitoring and management of cash flow.

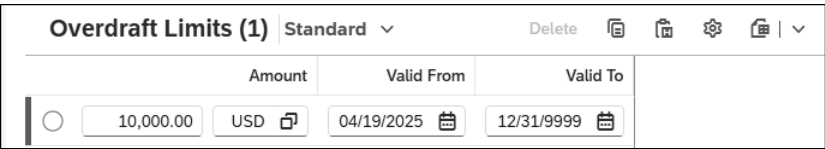


Figure 3.27 Bank Account Overdraft Limits

Overdraft limits are available and used in a few SAP S/4HANA solutions, primarily to monitor and control cash availability. However, the overall functionality around overdraft limits is still limited, and they are not yet widely integrated into broader payment or liquidity management processes. Their use remains basic, mainly serving as a reference point rather than a dynamic control mechanism.

In the next step (see Figure 3.28), you can create a description for the bank account in multiple languages. This feature is particularly useful for multinational organizations that need to manage bank accounts in different countries and regions. By maintaining descriptions in local languages as well as centrally in English or any other standard language, you can ensure consistency and clarity across all locations while also accommodating local language preferences for users in different countries.

Figure 3.28 Bank Account: Multilanguage Descriptions

This helps with efficient communication and streamlines the management of bank accounts globally. This functionality allows you to conduct thorough research and meet international reporting requirements across various countries, especially when it comes to payment processing and variant generation. It also ensures that descriptions are clear and understandable for all stakeholders involved, supporting consistent and compliant communication across different regions and systems.

The next step involves entering **Cash Pool** data for the bank account (see Figure 3.29). This step is important for determining whether the account serves as a header account or a subaccount within the cash pool structure. If the value is populated under the **As Header Account** area, it means that the account is a header account for the selected cash pool. In this case, you also need to specify a payment method that will be used for sweeps in case of any transfers made from this header account. If the value is populated under the **As Subaccount** area, it means that the account is designated as a subaccount for the selected cash pool. For subaccounts, you also need to specify the payment method used for the sweeps from this account, the target balance—which indicates the expected balance on this account after the sweep is completed—and the maximum and minimum transfer amounts to be executed.

Figure 3.29 Cash Pool Data

Cash pooling, also known as *cash concentration* in previous versions of SAP S/4HANA, allows companies to manage liquidity more efficiently by consolidating balances from

multiple accounts into one central account. This functionality is highly relevant for payment processing and cash management. However, note that in SAP S/4HANA 2023, new dedicated functionalities have been introduced for this specific solution. Cash pools are no longer managed within a single master data object but are now handled centrally through the Manage Cash Pools (Version 2) app (F3266A), provided as part of SAP S/4HANA Finance for cash management. This means that if you are using at least SAP S/4HANA 2023, you maintain participating accounts in the Manage Cash Pools apps. If you have an older version, you need to specify and assign them individually in each bank account.

Also note that SAP plans to remove this functionality from the Manage Bank Accounts app in the future and move it fully to dedicated apps. It is already deprecated as of SAP S/4HANA Cloud Public Edition 2025, but it can still be used for now.

The bank correspondence functionality (see Figure 3.30), available just below the **Cash Pool** tab in Manage Bank Accounts app, adds features that enable automatic correspondence with the bank, making it easier to manage communications electronically. One key feature is the ability to create e-banking functionality, allowing your company to automate the exchange of electronic messages and documents with the bank. We will cover the e-banking functionality in Chapter 4—specifically, during the process of creating an in-house bank account from a BAM account.

Bank Correspondence										
<div> <div>▼</div> <div>Output Details</div> </div>							<div> <div>Send Output</div> <div>Duplicate</div> <div>Retry</div> <div>Set to Completed</div> <div>⚙️</div> </div>			
<input type="checkbox"/>	Item ID	Log	Status	Dispatch Time	Output Type	Role	Recipient	Channel	Form Language	Form Template
No data available										

Figure 3.30 Bank Correspondence

This capability enhances your bank communication by enabling you to automatically generate and send correspondence, reducing the manual effort involved in creating and managing letters. For instance, automatic notifications can be sent to the bank when key account changes occur, such as account openings or updates to payment approvers. E-banking functionality also can integrate with digital banking services to facilitate smoother, faster transactions and communication between your company and the bank. However, the main idea behind using centralized payment systems is to reduce the number of banks. Using and paying for e-banking services, which require technical support, can be too costly to justify reducing the number of banks. Therefore, it is generally recommended to manage this manually.

Once you have created a bank account, and if you have configured the SAP Document Management service in your SAP S/4HANA system, you can easily attach relevant documents to a bank account. To do this, go to edit mode, click **Manage Attachments** (see

Figure 3.31), and the system will open the Manage Attachments app (F4812; see Figure 3.32), in which you can create and attach documents related to the bank account. This app is used for all attachments in the system, so you need to select **Bank Attachments** from the list.

USBK3
1234567899

Copy View History Manage Attachments

Bank Country/Region: US (USA) Currency: United States Dollar (USD) In Workflow Process: Yes (X)
Bank Key: 88888777 (Bank 77) Account Status: 02 (Active)
Bank Control Key: — IBAN: —
Contract Type: Current Account

Figure 3.31 Manage Attachments

This functionality allows you to store and organize all related documents, such as contracts, agreements, or any other files, in a centralized location, making it easier to access and manage them when needed.

SAP Document

Edit

Create Document Add URL

Document	Document Type	Document Part	Document Version	Authorization Gro...	File	Description	Created At
No data available							

Upload Files

Document Type:* BAM

Authorization Group:

Storage category:* DMS_C1_ST

Description: Bank Account Data

File Path: Bank Details.pdf

Upload Cancel

Figure 3.32 Add Attachment to Bank Account

Once the documents are attached, you will be able to see all the relevant files linked to a particular bank account (see Figure 3.33). The Manage Attachments app (F4812) allows you to store and organize all information coming from the banks, such as agreements, bank statements, contracts, and correspondence. By centralizing these documents, you ensure easy access and efficient management, making it simpler to track and reference important information associated with each bank account.

Once you have populated all the required information for the bank account, you can save it. At this stage, the account will be saved as an inactive account. The next steps in

the process will depend on the configuration settings, but typically, bank accounts should go through an approval process. The approval process, including who approves the account and the approval steps, will be determined by the configuration setup. This ensures that all necessary checks and validations are performed before the bank account is fully activated and ready for use. We will discuss approvals in Section 3.1.4, but first we'll talk about options for mass import/migration of bank accounts.



Document	Document Type	Document Part	Document Version	Authorization Gro...	File	Description	Created At	Created By	Last Changed At	Changed By
BAM					Bank Details.pdf	Bank Account Data	04/19/2025 17:53:...	Adrian Matys	04/19/2025 17:53:...	Adrian Matys

Figure 3.33 Manage Bank Account Attachments

3.1.3 Uploading, Migrating, and Replicating Bank Accounts

When you have multiple ERP systems across your organization, it's highly beneficial to establish a central bank account repository to manage all bank accounts in one place and replicate them into connected systems. This approach minimizes manual work, reduces inconsistencies, and eliminates redundant updates across environments.

BAM is an excellent tool to serve as this central repository. It allows for centralized creation, maintenance, and governance of bank account data, while also supporting automatic replication to other ERP systems via integration. This ensures that all systems are always aligned with up-to-date and accurate bank account information, leading to operational efficiency, data integrity, and compliance benefits.

Replication options for bank accounts are as follows:

- XML-based replication (import/export via SAP Fiori app)
- IDoc replication
- Web services/APIs

Currently, SAP is also developing some additional features for bank account replication in SAP S/4HANA Cloud Public Edition.

Import and Export Bank Accounts App

Creating all bank accounts one by one in the Manage Bank Accounts app can be extremely time-consuming, especially during system go lives, large-scale migrations, or major system updates. To streamline this process, SAP provides the Import and Export Bank Accounts app, which is a much more efficient alternative. You can also open this app from the Manage Bank Accounts app by clicking **Import and Export Bank Accounts**.

In this application (see Figure 3.34), you can first download the predefined Excel template that includes all the necessary fields expected by the system by clicking **Download XML Spreadsheet Template**. Once you've populated the template with the required

bank account data, you can upload it directly into the system, and SAP S/4HANA will automatically create the bank accounts based on the information provided in the file.

Figure 3.34 Import and Export Bank Accounts

The Import and Export Bank Accounts app processes files in XML format. This standardized format ensures compatibility and allows for structured data exchange between systems.

You can download the current bank account data from the system in XML format, make the necessary changes or additions, and then upload it back—either to the same system or to another system, such as quality or development. This makes it particularly useful for keeping environments synchronized or regularly updated.

In addition, during project testing phases or cutover activities, you can prepare an XML file with all the finalized bank account information and upload it into the production system as part of the data migration process. This helps avoid manual data entry, ensures consistency, and speeds up the transition to the new environment.

IDoc Bank Accounts Replication

SAP has provided standard message types to facilitate the replication of house banks, house bank accounts, and bank accounts using the data replication framework (DRF). This framework is especially useful when setting up a centralized BAM approach or during system landscape transitions, such as implementing a new SAP S/4HANA instance. We only want to briefly mention this functionality here; there are many other sources for detailed information on how it works technically and how it integrates into various SAP processes. For example, you can check out SAP Note 3524248.

Transaction DRFOUT triggers the replication; here you can select the appropriate replication model and filter criteria to define which bank data should be sent to which system. Transaction DRFOUT is a standard SAP transaction used for triggering the replication of master data, including bank accounts, business partners, materials, and more. Its functionality is highly dependent on the DRF configuration. Within Transaction DRFOUT, you define replication models, filter criteria, and target systems, which control how and where the data is sent.

When it comes to bank account replication, Transaction DRFOUT supports sending data via IDoc message types like BAMMAST (for bank accounts) and HBHBMAST (for house banks and house bank accounts). The flexibility of this transaction makes it a powerful tool not only for bank-related data but also for a wide range of master data objects, if the appropriate replication models and message types are configured. Proper setup in Transaction DRFIMG, along with maintaining partner profiles in Transaction WE20, is essential for enabling and controlling these replications

Web Services/APIs

Replication of bank accounts using web services or APIs with custom code is the most flexible and robust option for integrating systems, but it does not use a standard tool. It allows you to write custom logic using OData services or SOAP/REST APIs, enabling real-time replication across multiple systems. This approach is particularly well-suited for organizations adopting microservices architecture or requiring on-demand updates as it offers scalability and adaptability for complex enterprise environments.

3.1.4 Bank Account Approvals

Once a bank account is created in SAP, there are three main options for how it can be approved, depending on the configuration settings:

- The first option is *no approval*, in which the account becomes active immediately after creation without requiring any further validation.
- The second option is *dual control*, in which one user creates the bank account and another user must approve it using the Approve Bank Account Changes—Two-Person Verification app (F6264). This provides a simple layer of control and can be performed by any authorized user.
- The third and most flexible option is the use of a *workflow*, which allows for building more complex approval processes. In a workflow setup, predefined users must approve the account based on specific rules, and in some steps, they may even be required to input additional information. For example, the first person may create the bank account, and another designated user can later assign the house bank account ID or other attributes. This level of flexibility in workflows ensures proper governance and accountability, tailored to the organization's internal control requirements.

Two-person verification is shown in Figure 3.35. When you open the Approve Bank Account Changes—Two-Person Verification app, you will see a list of bank accounts requiring two-step verification. The process is straightforward: Simply click **Request** to view the details of the account, including any attached documents. From there, all you need to do is click **Approve** or **Reject**.

This app does not support workflows; instead, it follows a simple dual control mechanism. Any user with the appropriate role can approve any bank account, regardless of the specific content. However, the only restriction is that you cannot approve changes, openings, or closures for accounts that you yourself have requested. This ensures a basic level of control and separation of duties in the approval process.

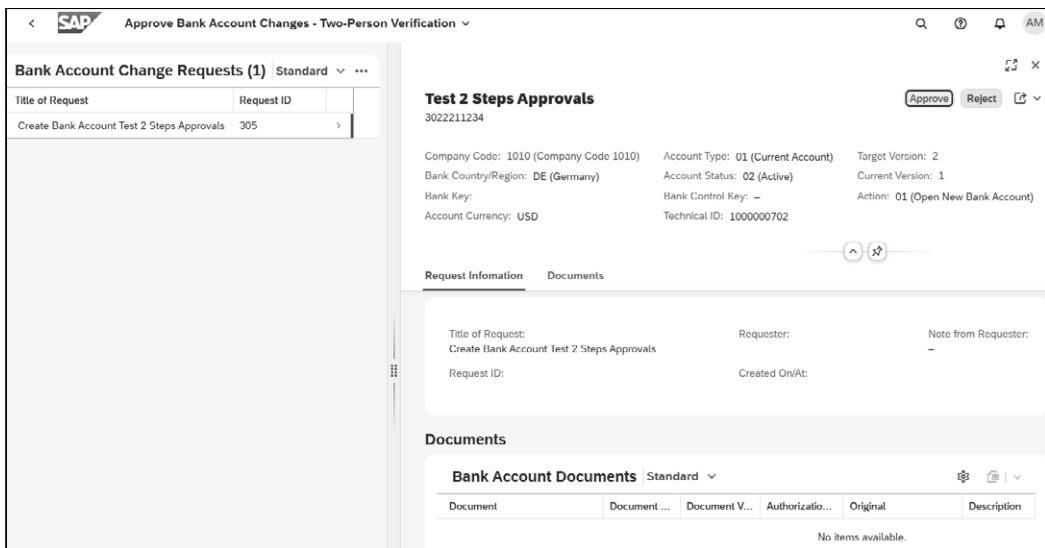


Figure 3.35 Approve Bank Account Changes—Two-Person Verification

The Bank Account Change Requests app (F6265) is designed to facilitate the management and monitoring of bank account change requests within the two-person verification process. This app (see Figure 3.36) allows users to view all bank account change requests that require approval, providing a centralized overview of their statuses. Users can filter and check the details of each request, including whether they have been approved, are pending approval, have been rejected, or have been withdrawn. This functionality ensures transparency and control over bank account modifications, aligning with organizational compliance requirements.

The status of all submitted bank account applications (no matter the type of approval selected) can be tracked in this app. It provides a centralized overview of all requests created for opening, changing, or closing bank accounts. You can view detailed information such as the status (e.g., **Draft**, **Submitted**, **In Approval**, **Approved**, **Rejected**, or **Withdrawn**), the applicant, the submission date, and other key details. It's especially

useful for monitoring the progress of applications in organizations with multiple teams involved in BAM, ensuring transparency and timely follow-up.

Request ID	Request Status	Created On/At	Account Number	Account Curr...	Bank Country/Region	Company Code	Technical ID	Contract Type	Account Type	Bank Contr...
324	Withdrawn	05/30/2025, 16:19:48	65656565	EUR	DE (Germany)	1012 (Beispiel AG)	1000000680	Current Account	08 (Salary Account)	
305	Approved	03/24/2025, 08:52:16	3022211234	USD	DE (Germany)	1010 (Company Code 1010)	1000000702	Current Account	01 (Current Account)	
304	Approved	03/18/2025, 15:50:28	98989899	EUR	DE (Germany)	1012 (Beispiel AG)	1000000714	Current Account	09 (Checking Account)	
303	Approved	03/18/2025, 11:49:02	98989899	EUR	DE (Germany)	1012 (Beispiel AG)	1000000714	Current Account	09 (Checking Account)	

Figure 3.36 Bank Account Change Request

If you are approving bank accounts using a workflow (instead of the dual control approach), you will use the Approve Bank Account Applications app (F5859).

This app allows you to manage the approval process for bank account creation or changes in a more controlled and flexible way. It is especially useful in organizations that require more structured approval flows, possibly involving multiple steps or designated approvers. Key features of this app include the following:

- **Approve or reject bank account applications**

You can review the application details and either approve or reject the request directly within the app.

- **Automatic creation of inactive bank account**

Once approved, the system automatically creates an inactive bank account based on the details provided in the application.

- **Email notifications**

The system automatically sends an email to the applicant to inform them of the approval or rejection status.

- **Withdraw application**

You can confirm the withdrawal of a bank account application. If the application is withdrawn, the process of creating the bank account is stopped. If an inactive bank account was already created before the withdrawal, it will be automatically marked as **Withdrawn** in the system.

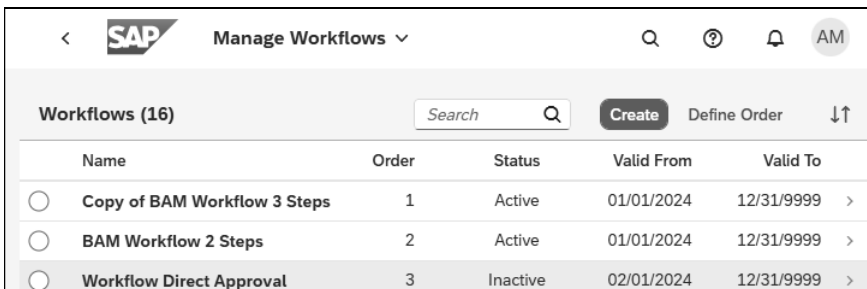
This app supports structured and auditable workflows, making it a strong alternative to the simpler two-step, dual control method.

Withdrawal of a request can be done directly from the Manage Bank Accounts app, regardless of whether you're using dual control or a workflow-based approval. To withdraw a request, follow these steps:

1. Open the Manage Bank Accounts app.
2. Find and select the bank account for which you’ve submitted a request (this could be a creation, change, or closure).
3. If the request is still pending approval, you’ll see an option to withdraw the request. Simply click **Withdraw**, and the system will cancel the approval process.

Once a bank account is approved, its status changes to **Active** and it becomes available for use. At this point, the house bank and bank account ID are also activated. It’s important to note that once a bank account reaches the **Active** status, it can no longer be deleted. Only **Inactive** accounts can be deleted. If an account has already been activated, the only option is to close it. This ensures that all activated accounts remain in the BAM system for audit, monitoring, and reporting purposes.

BAM workflows can be maintained using the Manage Workflows—For Bank Accounts app (F2796). This app (see Figure 3.37) enables business process specialists to define and manage approval workflows tailored to bank account processes within a line of business or across the entire organization. With this app, you can review predefined workflows or create new ones by copying existing templates. You can customize the workflow logic based on your company’s requirements by specifying preconditions, step sequences, and recipients. It’s also possible to configure automatic approvals for certain steps by setting specific criteria such as the change request creator, account type, or company code. The app also allows you to activate, deactivate, and delete workflows and to define the order in which workflows are evaluated, ensuring that the correct workflow is triggered for various bank account scenarios.



The screenshot shows the SAP Manage Workflows app interface. At the top, there's a header bar with the SAP logo, the title 'Manage Workflows', and several icons (search, help, notifications, and a user profile icon labeled 'AM'). Below the header, there's a section titled 'Workflows (16)' with a search bar and buttons for 'Create', 'Define Order', and a sort icon. The main content is a table with five columns: Name, Order, Status, Valid From, and Valid To. The table lists three workflows: 'Copy of BAM Workflow 3 Steps' (Order 1, Active), 'BAM Workflow 2 Steps' (Order 2, Active), and 'Workflow Direct Approval' (Order 3, Inactive). Each row has a radio button in the first column and a chevron icon in the last column.

Name	Order	Status	Valid From	Valid To
<input type="radio"/> Copy of BAM Workflow 3 Steps	1	Active	01/01/2024	12/31/9999 >
<input type="radio"/> BAM Workflow 2 Steps	2	Active	01/01/2024	12/31/9999 >
<input type="radio"/> Workflow Direct Approval	3	Inactive	02/01/2024	12/31/9999 >

Figure 3.37 Manage Workflow for BAM Applications

When creating a workflow in the Manage Workflows—For Bank Accounts app, you have a wide range of possibilities to tailor the process to your organization’s needs. You can define the triggering events that initiate the workflow, such as creation, modification, or closure of a bank account. You can also specify rules and conditions, such as which company codes, account types, or change initiators the workflow should apply to. Within the workflow, you can build multistep approval processes, assigning different recipients for each step and even enabling or skipping steps based on certain criteria. In addition, you can decide which fields should trigger the workflow when changed and

whether a step requires user input or just approval. This flexibility allows you to create complex and dynamic workflows that reflect real organizational policies and approval hierarchies.

3.1.5 Manage Bank Account Hierarchies App

Once your bank accounts are created and approved, it's a common practice to group them into bank account hierarchies for better organization, reporting, and operational management. These hierarchies help structure accounts based on various criteria, such as cash pooling structures, geographical locations, currencies, or other custom-defined reporting needs. This is managed through the Manage Bank Account Hierarchies app (F4973). Within this app (see Figure 3.38), you can define and maintain flexible hierarchical structures by assigning bank accounts to different nodes, allowing you to visualize and analyze your account structure more effectively. These hierarchies can also be leveraged in reporting tools, cash concentration processes, and cash management analysis across your organization. When you open the Manage Bank Account Hierarchies app, the system will display the existing bank account hierarchies. To create a new hierarchy, simply click the **Create** button.

The screenshot displays the SAP 'Manage Bank Account Hierarchies' app interface. A 'New Hierarchy' dialog box is open, allowing users to create a new hierarchy. The dialog includes the following fields and options:

- Type:** A dropdown menu currently set to 'Bank Account Group'.
- Hierarchy ID:** A text field containing '0001'.
- Hierarchy Description:** A text field containing 'Cash Pool Switzerland'.
- Valid From:** A date field set to '04/13/2025'.
- Valid To:** A date field set to '12/30/9999'.
- Note:** An empty text area for additional information.
- Buttons:** 'Create' and 'Cancel' buttons at the bottom of the dialog.

The background interface shows a 'Hierarchy List (0)' table with columns for 'Valid From/To' and 'Status'. It also includes a search bar, a 'Valid From/To' filter with a date example 'e.g. 12/22/2025-12/31/2025', and buttons for 'Go', 'Adapt Filters', 'Import Hierarchy', and 'Create'.

Figure 3.38 Manage Bank Account Hierarchies

You will then need to select the **Type** for the hierarchy, enter the **Hierarchy ID**, provide a **Hierarchy Description**, specify the validity period (**Valid From** and **Valid To**), and click **Create**. The app also allows you to import hierarchies from an Excel file, making it easier to set up multiple hierarchies at once, especially for large-scale environments or migrations. Then you can simply start populating your hierarchies by creating headers and assigning bank accounts, as shown in Figure 3.39.

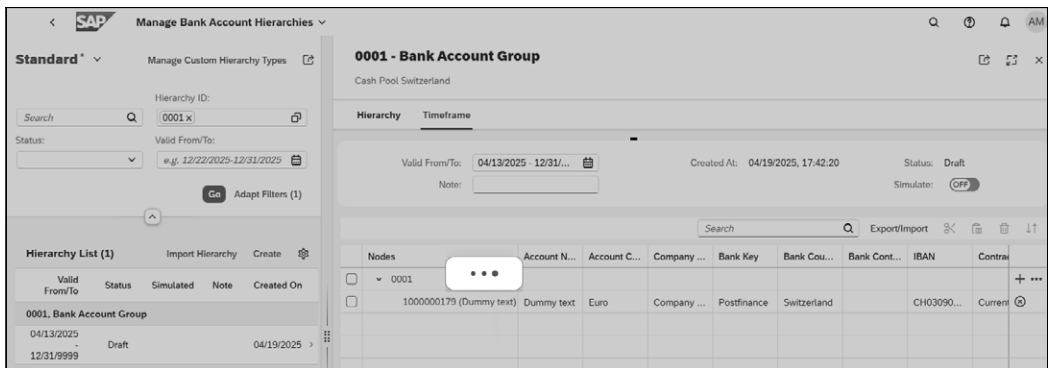


Figure 3.39 Create Bank Account Hierarchy

You can update existing bank account hierarchies by selecting the desired hierarchy, opening a new view, and clicking **Edit**. Once you're in edit mode, you can click the **+** icon to add new nodes or bank accounts to the hierarchy. A node represents a grouping level; for example, you can have a **Europe** hierarchy containing nodes that represent countries in Europe. The app also provides an export/import feature, which allows you to download and upload hierarchy files. This is especially useful when you need to transfer hierarchies between systems, such as during user acceptance testing (UAT): You can create a hierarchy, download it, and upload it to a new system. After you make any changes, be sure to save your work. Note that once a hierarchy has been updated, it must be activated by another user to ensure proper approval.

In addition to bank account hierarchies, there are multiple apps available for creating similar hierarchies in SAP S/4HANA. In Chapter 6, we will explore liquidity item hierarchies, which are created in the same way as bank account hierarchies. These concepts also tie into broader accounting structures such as profit centers, G/L accounts, and other types of hierarchies. The approach to managing and organizing these hierarchies remains consistent across various functionalities, ensuring streamlined and unified management across different accounting and financial processes.

Hierarchies created in this app are used in a couple of reports we'll describe in Chapter 6—mainly, the Cash Flow Analyzer app.

3.1.6 Manage Bank Account Reviews App

The Manage Bank Account Reviews app (F7510) enables you to initiate, manage, and monitor periodic reviews of bank account master data to ensure its accuracy and completeness. This app allows you to organize reviews by reference date and divide the process into specific review areas, each tailored to different responsible reviewers. For each area, the system takes a snapshot of the bank account master data at the time of review initiation. This snapshot serves as the basis for the review and is preserved throughout the process. Once a review is started, a workflow is triggered to guide the assigned

reviewers, who complete their tasks via the My Inbox—All Items app. You can continuously track the progress of each review area in the Manage Bank Account Reviews app, ensuring the process stays on schedule and facilitating issue resolution if needed. Once all checks are finalized, you can mark the review as **Completed**. This app is a key component for enforcing governance and control over bank account data in a structured and auditable manner.

Prior to SAP S/4HANA 2024, initiating the bank account revision process required starting directly from the Manage Bank Accounts app or from the individual bank account itself. Users would then switch to a dedicated bank account revision app to validate and manage the revisions they initiated. However, SAP S/4HANA is now shifting toward a more centralized and structured approach with the Manage Bank Account Reviews app, which streamlines the process and offers better oversight and control. It is considered best practice to perform regular reviews of bank accounts to ensure that all accounts are still required and properly maintained. This is particularly important for large organizations, where it's common to find unused or outdated bank accounts that have not been formally closed. A structured revision process helps identify such cases, improves centralization of BAM, and supports clean and efficient financial structures across the organization.

3.1.7 Closing Bank Accounts

The closure of a bank account in SAP S/4HANA is performed through the Manage Bank Accounts app. To initiate the process, open the relevant bank account and click **Mark for Closing**. The system will open the popup shown in Figure 3.40, in which you need to select a date for the **Planned Closing Date** and add a **Note**.

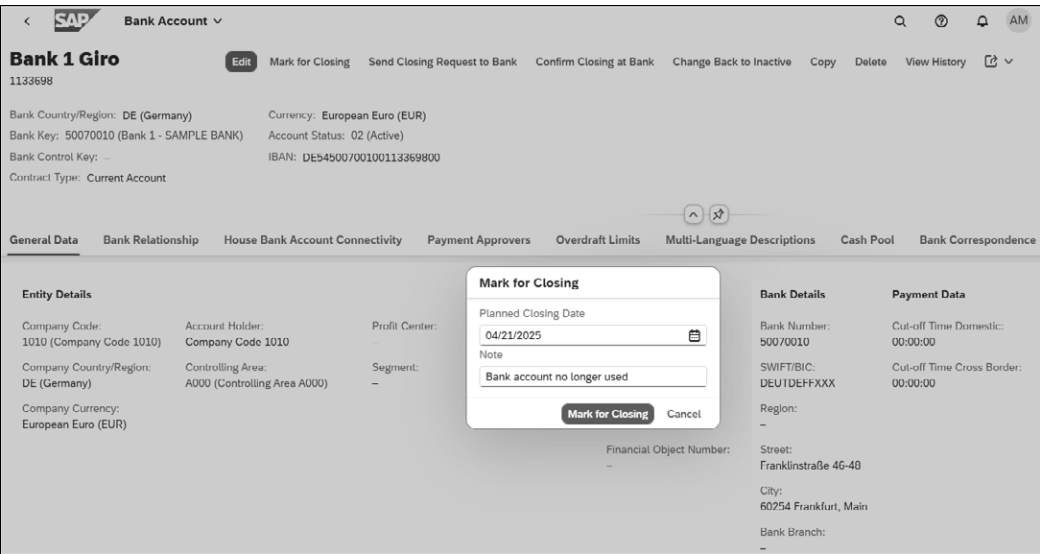


Figure 3.40 Close Bank Account

Once this action is taken, the system—depending on your configuration—will trigger a workflow requiring approval of the closure request. After approval, there may be additional steps in the process, such as sending a closure request to the bank, confirming the closure at the bank, and finalizing the account closure in the system. These steps vary based on how the workflow is set up, which will be covered in more detail later in Section 3.2. It’s crucial to note that once a bank account has been activated, it cannot be deleted from the system. It can only be closed, and this closure will always follow a structured approval and workflow process to ensure compliance, traceability, and proper recordkeeping for audit purposes.

3.1.8 Manage Bank Chains App

Bank chains in SAP S/4HANA are used for managing and processing intermediary bank accounts, which are often required in international or complex payment scenarios. When a direct connection to the beneficiary bank is not possible or practical, intermediary banks are involved in routing the payment through one or more additional financial institutions. These intermediary relationships are maintained as bank chains in SAP S/4HANA using the Manage Bank Chains app (F4004). In this app, you can define the sequence of intermediary banks that should be used for specific transactions, ensuring accurate and efficient payment processing. This setup is particularly important in global organizations in which multiple banking relationships exist and payments need to be routed through a network of banks.

Once you open the Manage Bank Chains app and click **Create**, you’ll arrive at the screen shown in Figure 3.41, where you can create a new bank chain and intermediary bank.

Standard

Search

Business Partner

Bank Key of Recipient Bank

Country/Region Key of Sender Bank

Currency

Country/Region Key of Recipient Bank

Account Number at Recipient Bank

Bank Key of Sender Bank

Payment Method Supplement

Go

Adapt Filters

Bank Chains

Standard

Create

Delete

Account Number at Recipient Bank

To start, set the relevant filters and choose "Go".

4074262436

ZA

Bank Data

Items

Change Log

Partner Bank Data

Business Partner: 64100010

Country/Region Key of Recipient Bank: ZA (South Africa)

Bank Key of Recipient Bank: 632005 (ABSA 1 - SAMPLE BANK)

Account Number at Recipient Bank: 4074262436

IBAN: -

House Bank Data

Country/Region Key of Sender Bank: South Africa (ZA)

Bank Key of Sender Bank: ZAEPI (042010)

Currency: South African Rand (ZAR)

Payment Method Supplement

Intermediary Banks (2)

Standard

Add

Remove

Order

Type of Bank in a Bank Ch...

Country/Region Key of Co...

Bank Key of a Correspond...

Account Number at Corres...

IBAN

1

2 (Intermediary bank)

ZA

091001

432143219

2

3 (Recipient's corre...

ZA

632005

78643521

Change Log

Change Documents

Standard

Create

Discard Draft

Figure 3.41 Create New Bank Chain

After you click **Create**, select the business partner for whom you want to execute the payment. Then choose the account from which the payment will be made, the currency in which you would like to execute the payment, and the payment method to supplement it. Then click **Add**, select the intermediary bank, and specify its country, bank key, and bank account. Next, select the recipient's bank account (the ultimate beneficiary account), including its country, bank key, and bank account.

In the example shown in Figure 3.42, we processed payments through a bank chain by utilizing an intermediary bank. Specifically, we initiated payments from a bank account located in South Africa, denominated in South African Rand, and routed the funds through an intermediary bank account—also based in South Africa—before they reached the final beneficiary's local bank account.

Account Number at Recipient Bank	Country/Region Key of Recipient Bank	Bank Key of Recipient Bank	IBAN	Currency	Bank Key of Sender Bank	Country/Region Key of Sender Bank
4074262436	ZA (South Africa)	G32005 (ABSA 1 - SAMPLE BANK)		ZAR	042010	ZA

Figure 3.42 Manage Bank Chain

3.1.9 Foreign Bank Accounts App

In many countries, there is a regulatory requirement to register foreign bank accounts held by entities located within that country, along with their respective balances. To support compliance with these regulations, SAP S/4HANA offers the Foreign Bank Accounts app (F1575). This app enables you to identify and manage foreign bank accounts owned by your company and to track employees who hold the power of attorney over these accounts. The collected information can be used to file mandatory reports—such as the Report of Foreign Bank and Financial Accounts (FBAR) required for US persons—or for internal analytical purposes, depending on your country or region's legal requirements.

When you open the app (see Figure 3.43), you'll notice that the system has already predefined certain countries, but you can easily add more as needed; these are the countries where the entity is located. Simply select the timeline for which you want to review the accounts (in the **Time Range** field), choose the foreign accounts created for the relevant entities by selecting values for **Home Country/Region**, and click **Go** to view the results.

<

SAP

Foreign Bank Account Report ▾

🔍

?

🔔

AM

Standard ▾

Home Country/Region:*

Company Code:

Time Range:*

Consider Account Value:

[-US x]

[-MP x]

[-AS x]

[-GU x]

[-PR x]

[-VI x]

🔒

🔒

Last Year (01/01/2024 – 12/31/2024)

🗓️

No

▾

Go

Adapt Filters (3)

⬆️

⬆️

Employee

Company

Accounts (4)

Standard ▾

⚙️

🔒

▾

Bank Account Description	Account Number	Bank Account ...	Financial Institution	Country/Region (Bank)
Company Code: 9080 (US_Chicago), Country/Region (Company): DE (Germany)				
Current Account	1000000008	EUR	Deutsche Bank	DE (Germany)
Company Code: 9070 (US_Houston), Country/Region (Company): DE (Germany)				
Current Account	1000000007	EUR	Deutsche Bank	DE (Germany)
Company Code: 9060 (US_Detroit), Country/Region (Company): DE (Germany)				
Current Account	1000000006	EUR	Deutsche Bank	DE (Germany)
Company Code: 1710 (Company Code 1710), Country/Region (Company): US (USA)				
Affiliate Hong Kong USD Account	3001234567	USD	JPMORGAN CHASE BANK, N.A., HONG KONG BRANCH	HK (Hong Kong)

Figure 3.43 Foreign Bank Account Report

As a result, the system will display all foreign bank accounts associated with entities in the selected countries that are valid within the chosen time period. By switching to the **Employee** tab, you can also view all employees who hold power of attorney and are authorized to receive payments in the selected bank accounts.

This report is designed to provide much of the information needed for various regulatory reporting requirements. Some countries, such as France, may also require that you report the balances of these foreign accounts. In such cases, you'll need to refer to other reports as well, which we'll explore later in this section. You can then combine the relevant information and manually prepare the necessary regulatory submissions. Although the app supports multiple use cases, its primary focus is to fulfill the requirements of US authorities.

3.1.10 Bank Relationship Overview App

The Bank Relationship Overview app (F3775) provides a wide range of valuable information about your organization's banking landscape. It offers insights into both incoming and outgoing payments—showing their volumes and amounts—based on selected bank accounts. You can also access detailed bank profiles, including how many banks are associated with specific bank keys, the number of bank accounts, and how many company codes are using them. In addition, the app displays a breakdown of bank fees for selected accounts, account statuses (**Active**, **Closed**), workflow statuses (such as open requests), and revision history. In essence, this app consolidates all key data previously available across various apps and bank statements, offering a comprehensive view of your entire bank relationship setup within the system.

principals, authorized representatives, associated bank accounts, payment approval rules, and permitted activities assigned to each representative.

You can create a power of attorney record using the Manage Powers of Attorney for Banking Transactions app (F5742). Once you open this app, you will see the screen shown in Figure 3.45.

TEST01
Power of Attorney for payments approvals

Retrieve Bank Accounts

Header Principals Bank Accounts Authorized Representatives Activities Predecessor

Power of Attorney ID: * TEST01
Authorization Group: * SALL (All Company Codes)
Status: Created
Bank Account Assignment Type: * Via Selections, Existing and Futur...
POA Description: Power of Attorney for payments appro...
Valid From: * 04/01/2025
Valid To: 06/30/2026
Payment Approval Rule:

Principals

Assigned Company Codes (1) Standard Search Add Multiple Company Codes Add Remove

Company Code	Company Name
1010	Company Code 1010

Bank Accounts

Bank Account Selections (1) Search Remove

Selection Group	Selection Field	Select Option	Value
	Bank Country/Region	equal to	DE

Figure 3.45 Create Power of Attorney

When creating a power of attorney record, you first need to assign a **Power of Attorney ID** and provide a **POA Description**. In this example, we are creating a power of attorney specifically for the approval of payments. Here, select the relevant **Authorization Group** and define the validity period by specifying the start and end dates. Next, you must indicate whether the power of attorney applies to a specific **Payment Approval Rule** or if it should cover all existing bank accounts in the system, as well as any bank accounts that will be created in the future.

To select the *principals*—the companies that grant the power of attorney to one or more individuals (authorized representatives)—you need to choose the appropriate **Company Codes** in the system; these represent the principals of the power of attorney holders. After selecting the relevant company code, you then define the bank accounts to which the power of attorney will apply. In this example, we are selecting bank accounts created within a specific country, but you can also search by bank keys, specific bank

accounts, currencies, technical IDs, or any other relevant criteria. Once you have made your selections under **Bank Account Selections**, click the **Retrieve Bank Accounts** button. The system will then automatically retrieve all bank accounts that meet the specified criteria, as shown in Figure 3.46.

Assigned Bank Accounts (5) Standard ▼							<input type="text" value="Search"/>					
Technical ID	Bank Key	Account Number	Bank Country/Region	Currency	Bank Control Key	Contract Type	Account Type	Bank Account Assigned At				
1000000000	50070010	1133698	DE (Germany)	EUR			01 (Current Account)					
1000000004	99999911	99999911	DE (Germany)	EUR			02 (Current Account (Internal))					
1000000003	88888886	1133888	DE (Germany)	EUR			01 (Current Account)					
1000000002	50070010	40414243	DE (Germany)	USD			01 (Current Account)					
1000000001	82080000	2580061	DE (Germany)	EUR			01 (Current Account)					

Figure 3.46 Power of Attorney: Assigned Bank Accounts




Next, you need to select *authorized representatives*; these are individuals (typically employees) to whom your company has granted the power of attorney.

Authorized Representative

For your authorized representatives, you must create a business partner and assign the business partner role **TR1000 Authorized Representative** (see Figure 3.47). Only business partners with this role can be assigned as authorized representatives in a power of attorney record.

Figure 3.47 Authorized Representative

After selecting the representatives, you must define *activities* (see Figure 3.48), which are the banking transactions for which authorization has been granted. The system offers a set of predefined activities, but you can also define custom activities as needed. Once you have selected all the required information, click **Create**. After the power of attorney record has been created, you must submit it for approval.

Authorized Representatives (1) Standard * ▾			Search 			▾
Business Partner	Business Partner Name	Approver Group				
1000090	Power of Attorney Representative Power of Attorney Representative					




Activities (1) Standard ▾			Search 			▾
Activity ID	Activity Description					
SPAY	Approve Payments					

Figure 3.48 Create Power of Attorney Representatives and Activities

When a power of attorney record is submitted for approval using the Manage Powers of Attorney for Banking Transactions app, and an approval workflow is enabled, a workflow task is automatically created in the My Inbox—All Items app for all users in the responsible team who have the authorization to approve powers of attorney. To ensure compliance with the dual control principle, the workflow only assigns approval tasks to users other than the one who submitted the power of attorney record (the initiator of the workflow).

If the workflow is configured with email notifications, the approvers also receive an email informing them that the approval process has started, including a direct link to the My Inbox—All Items app.

After the powers of attorney have been granted and are active, administrative tasks must be carried out to implement them—for example, by setting up changes and communication to go through a relevant bank branch (see Figure 3.49) or setting up authorizations for online banking managed by the bank. The Implement Powers of Attorney for Banking Transactions app (F6374) supports you in managing this process, helping to ensure that the granted powers are properly communicated and activated with the banks. For each power of attorney record, you must create exactly one power of attorney implementation.

Once your implementation is complete, you can use the Implement Powers of Attorney app to manage several key tasks. You can create outgoing correspondence objects to communicate with banks, as well as register incoming correspondence objects. If you do not have an automated connection or if eBAM is not implemented through SAP Multi-Bank Connectivity, then you can manually update the status of outgoing correspondence objects. The app also allows you to replace an existing power of attorney record or revoke one when necessary, helping you maintain accurate and up-to-date records of your banking authorizations.

The power of attorney and automatic exchange of information features are not very widely used. In most cases, only power of attorney documents are maintained to clearly identify who is authorized to use the accounts and who has access to them in the online

banking platform. This is primarily done for audit and reporting purposes, ensuring that there is a clear record of account access and usage rights.

Power of Attorney Implementation

Edit

Revoke

Header

Power of Attorney

Channels

Implementation ID:

Implementation Description:
Power of Attorney Implementation

Implementation Status:
Active

Power of Attorney

Assigned Power of Attorney (1) | Standard ▾

Power of Attorney ID

POA Description

Channels

Assigned Channels (1) | Standard ▾

Channel ID

Channel Description

SBRA

Bank Branch

>

Figure 3.49 Implement Power of Attorney

3.1.12 Bank Fees

Bank fees are an essential aspect of managing bank accounts, and they are typically incurred for each transaction. Maintaining bank accounts is associated with various costs, including account maintenance fees, transaction fees, and other charges depending on the services provided by the bank. These fees are closely tied to the number and type of transactions processed, as well as the specific bank account terms. Identifying and tracking the exact cost of these bank relationships can be quite complex, as fees can vary based on account activity, transaction volume, and other factors. Often, the detailed breakdown of these costs is not immediately clear, making it challenging to have full visibility into how much is being paid for the services provided by the bank.

Banks usually send this information about bank fees as part of the camt.86 file, which provides detailed fee breakdowns. SAP S/4HANA offers tools designed to help with the reconciliation of these bank fees, making it easier to match the bank's statements with your internal records and ensure accurate accounting. However, the process still requires careful review and reconciliation to ensure all costs are accounted for correctly.

The camt files you receive from your bank can be uploaded into SAP S/4HANA for further analysis using the Import Bank Services Billing Files app (F3002). This app allows you to import and process the bank fee data from the camt files, enabling you to perform detailed analysis, reconcile fees, and ensure that the bank charges align with your internal records.

When you open the Import Bank Services Billing Files app (F3002), you will see a single option to import your file. Simply select the file you wish to upload and import it into the system (see Figure 3.50) via the **Select Source File** field. It is important to ensure that the bank account reflected in the file already exists in the system. If the bank account is not created in the system, the file import will fail. However, if the bank account is set up correctly, the file will be uploaded successfully without any issues.

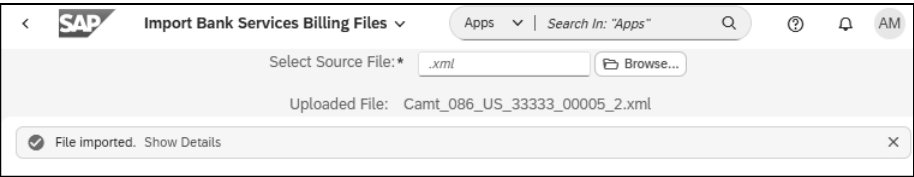


Figure 3.50 Import Bank Services Billing Files



File Version

Note that banks may send various versions of the camt86.xml files. Older versions of SAP S/4HANA prior to version 2023 might not be able to process newer versions of these XML files. In such cases, you may need to build a transformation to convert the file into a compatible format, or you can check SAP Notes (SAP's online service system) for any necessary updates. SAP typically releases updates through SAP Notes that can be uploaded to your system, allowing newer versions of the files to be processed correctly.

The Manage Bank Fee Conditions app (F3185) allows you to create, edit, and delete bank fee conditions. The condition defines how bank fees should be charged for specific bank services, as shown in Figure 3.51. The system uses these conditions to validate the imported bank fee data, checking for any errors or improper charges. The main features of this app are as follows:

- **Create, edit, and delete bank fee conditions**
You can define conditions for how fees should be applied to various banking services.
- **Assign conditions to bank services**
Once conditions are defined, you can assign them to relevant bank services and use them to validate imported bank fee data in the Monitor Bank Fees app.

To create a new bank fee condition, you first need to enter the **Condition ID** and specify the **Valid From** date. Then, provide a **Condition Description**, select the **Pricing Type**, and choose the **Condition Currency**. Once these fields are completed, click **Create**. Now enter

the conditions that are applied by the bank, which should come from your contract. These include thresholds, unit price, annual rates (as a percentage), base charge, minimum charge, and maximum charge. After filling in all the required information, click **Create** again to finalize the setup.

THRESHOLD

Condition Data Condition Items

Condition ID: * THRESHOLD Condition Description: Condition Currency: * United States Dollar (USD)

Valid From: * 04/02/2025 Pricing Type: * 06 (Custom Pricing)

Condition Items

Condition Item (2) Standard * Create Delete Copy Paste Settings Refresh Filter

	Base Charge	Minimum Charge	Maximum Charge	Threshold Volume	Annual Rate (%)	Unit Price	Currency
<input type="checkbox"/>	0.00 USD	15.00 USD	0.00 USD	2000000	0.3000000	0.00000000	USD
<input type="checkbox"/>	0.00 USD	5.00 USD	0.00 USD	1000000	0.2500000	0.00000000	USD

Figure 3.51 Manage Bank Fee Conditions: Create

In this example, we are creating conditions and charges that the bank can apply based on certain thresholds. For instance, if there is an excessive balance in your bank account, then the bank may charge fees. These conditions allow you to define how such charges are applied depending on the account balance or other predefined criteria.

Once you've created the bank fee conditions, you can analyze the fees uploaded from the statements using the Monitor Bank Fees app (see Figure 3.52).

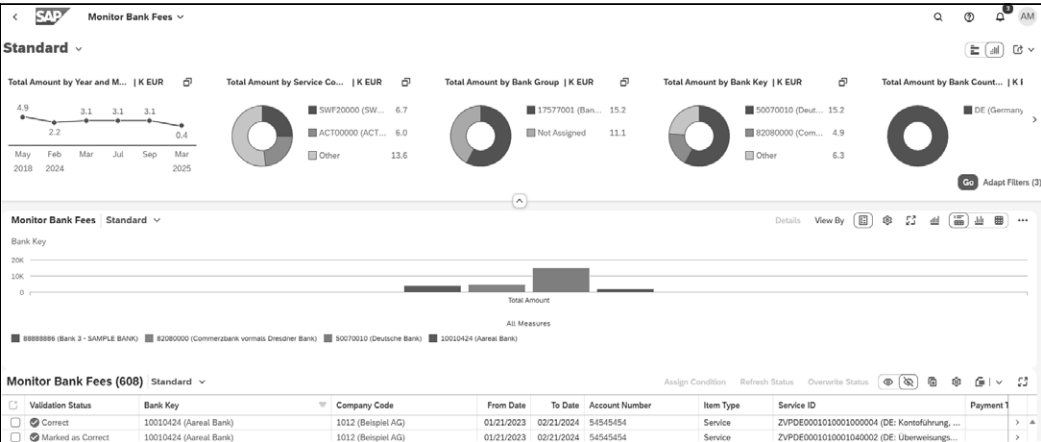


Figure 3.52 Monitor Bank Fees

This app allows you to view a detailed breakdown of your fees per bank account, including the online items that come from the billing services file. It provides a comprehensive overview, enabling you to ensure that the fees charged align with the conditions you've set and helping you to identify any discrepancies or unexpected charges.

The app helps you analyze how much you've paid per selected bank account. You can adjust the selection criteria to focus on specific bank accounts; under **Monitor Bank Fees**, you'll see all the line items from the bank services file. The app shows whether these line items have been assigned to any condition type. If a line item hasn't been automatically assigned a condition, you can manually assign it. Simply click the line item, and the system will open a new view in which you can assign the appropriate condition type to that line item.

From the line item in the **Bank Billing Services** file, select a line item and click the **Assign Condition** button, then select **Condition** and click **Assign** (see Figure 3.53) or click **New Condition** (system will move you to the Manage Bank Fee Conditions app). Once the condition is assigned correctly, it will be considered in the analytics and statistics for your bank fees, allowing for more accurate tracking and reporting. This ensures that your bank fees are properly categorized and integrated into future analyses.

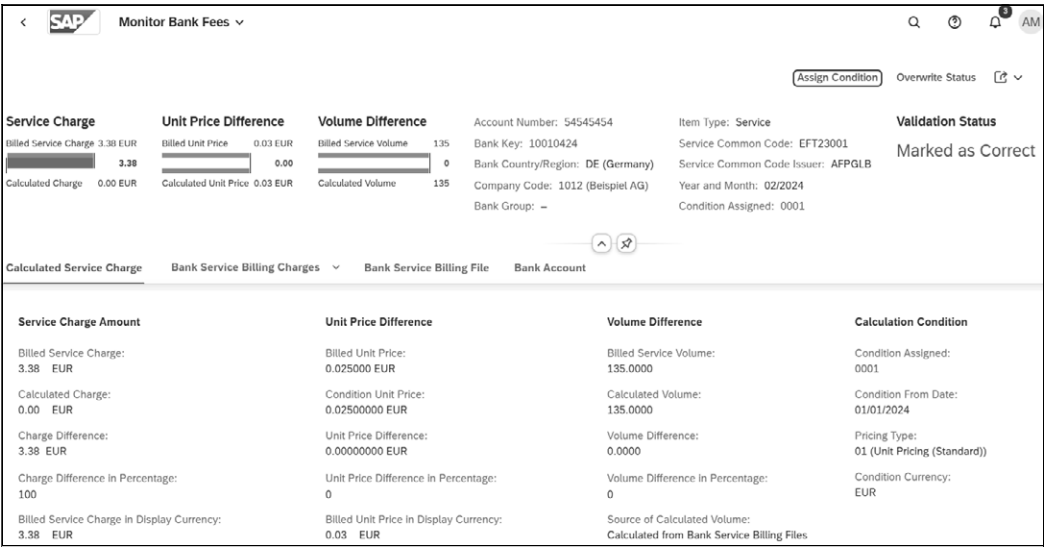


Figure 3.53 Assign Conditions to Bank Fee

Once all the relevant conditions have been assigned, you can begin analyzing and monitoring bank fees. The system will automatically track these fees and update them across various reporting sections, ensuring consistent and accurate visibility into your financial data.

3.2 Configuration

As you've learned throughout this book, in cash and bank activities, SAP S/4HANA is increasingly shifting functionalities to SAP Fiori apps, allowing business users to manage many tasks directly through the applications without needing technical support. This move supports a more user-friendly and accessible approach to daily operations. However, there are still certain activities that require backend configuration, particularly those related to defining the behavior of accounts or objects used in cash management. These configurations ensure that the system processes and interprets financial data correctly in line with organizational requirements.

In this section, we'll first examine two configuration activities that are also necessary for SAP S/4HANA Finance for cash management (Chapter 6) and SAP Bank Communication Management (Chapter 7): defining basic settings and enabling payment approvals. From there, we'll get into BAM-specific configurations for number ranges, bank account master data, bank account contract types, and field status groups.

3.2.1 Define Basic Settings

Defining basic settings is one of the most critical configuration activities as it determines how cash and bank relationship management and SAP Bank Communication Management will function within the SAP S/4HANA system. Follow menu path **SAP Financial Supply Chain Management • Cash and Liquidity Management • General Settings • Define Basic Settings** to reach the screen shown in Figure 3.54. This activity lays the foundation for the behavior of cash and liquidity management by establishing essential parameters such as the **Cash Scope** of functionality (**Basic Scope** or **Full Scope**, depending on your license and the associated cost).

You will also find change request grouping, bank account control modes, snapshot settings, and the availability of cash request features. These settings shape how the system processes cash flows, manages liquidity data, and interacts with other areas of SAP S/4HANA, like accounting and BAM. This activity also includes key technical setup areas such as defining the source application for accounting data, runtime parameters for the flow builder, and default liquidity items for liquidity analysis. Choosing the right configuration here is vital as it directly influences how cash operations are structured and monitored across the organization. Let's now describe each function a bit more:

■ Enable snapshot

The enable snapshot feature in SAP S/4HANA Finance for cash management allows you to activate the snapshot function within the full scope of cash management. When the **Enable Snapshot** field is enabled, cash flows are segmented by the time they are generated in One Exposure from Operations. This allows you to take snapshots of cash flows at different times and compare various versions of cash management data, such as cash positions or forecasts for a particular day. For example, you

can use snapshots to trace cash position reports for the previous month and assess how they have changed over time. It's important to note that the snapshot functionality is not supported in the basic scope. Even in the full scope, this feature is not enabled by default; it must be activated manually. Note that in this book, we are not covering snapshots in cash management in detail.

Figure 3.54 Define Basic Settings

■ Cash trade request

A cash trade request in cash management is a feature that allows you to create, track, and manage short-term investments and borrowing activities within your organization's cash position management processes. It can be triggered directly from the cash management module and can also be used to send trade requests to external trading platforms to create deals. This capability helps treasurers to manage liquidity more effectively, ensuring that surplus funds are invested efficiently or short-term borrowing needs are met in a timely manner. It's worth noting that cash trade requests fall within the broader topic of treasury and risk management in SAP, which we do not cover in detail in this book.

■ New reconciliation model

The **Enable New Reconciliation Model** option activates the new cash flow reconciliation model within cash management, providing enhanced capabilities for reconciling cash flows. This new model offers more functionalities and can be used to

reconcile cash flows against a broader set of data compared to the old model. If you have this functionality available in your system, you should use the new reconciliation model to take advantage of its advanced features. When the new reconciliation model is enabled, you should use the Reconcile Cash Flows—Intraday app for your daily cash flow reconciliation tasks. If the new reconciliation model is not enabled, you will need to use the Reconcile Cash Flows—Intraday Memo Records app instead. This flexibility allows you to tailor your reconciliation processes based on your specific requirements and system configuration.

■ **Additional bank account derivation model**

The **Disable Additional Bank Account Derivation Model** option in cash management allows you to control how the system fills in missing bank account information during accounting flows. By default, the system applies an additional bank account derivation logic when key details such as the bank account, house bank, or house bank account information is absent. This additional logic is executed before saving the data. However, if you activate the **Disable Additional Bank Account Derivation Model** option, the system suspends this additional derivation process, allowing you to manage and control bank account assignments manually if needed. It's important to note that in cash management, having all flows assigned to the correct bank accounts is critical for maintaining an accurate cash flow forecast. Therefore, if you choose to disable this option, you must implement an alternative method to assign flows to bank accounts to ensure accuracy. For more information on these alternatives, see Chapter 6.

■ **Grouping options for mass change requests**

The grouping options for mass change requests in BAM allows you to control how change requests for updating payment approvers in multiple bank accounts are grouped together for workflow processing. By default, when you initiate a mass change to update payment approvers across several accounts, a change request is generated for each bank account, and these requests are grouped into a single mass change request. However, you can customize how these requests are grouped: into a single mass change request, or into multiple mass change requests by company code, account type, or a combination of both. This flexibility ensures that workflows align with your organization's structure and responsibilities. It's important to note that this setting is only relevant when workflows are enabled for BAM and is not applicable to basic cash management. If you choose **Group into a Single Change Request**, all change requests are combined into one, requiring approvers to be authorized for all company codes and account types. **Group by Company Code** creates separate mass change requests for each company code, so approvers need responsibility for specific company codes. **Group by Account Type** creates separate requests by account type, so approvers must be responsible for specific account types. Lastly, **Group by Company Code and Account Type** generates highly specific mass change requests for each company code and account type pair, giving you the most granular control over approval workflows.

■ Bank account control mode

Via the bank account control mode, you can select how you would like to approve bank changes within BAM. By choosing the appropriate grouping option, you determine how change requests are organized and routed to approvers based on your company's structure and workflow needs, as described in Section 3.1.4.

■ Agent rules for workflow

An agent in BAM is a user who can receive workflow tasks (work items) in their inbox and perform necessary actions on them. In the **Agent Rules for Workflow** section, you can define whether the workflow should consider teams and functions only, or also include responsibility rules. Teams and functions group users by role or organizational structure, whereas responsibility rules allow for more granular assignments based on criteria such as company code, account type, or other attributes. Typically, you would select both teams and functions and activate responsibility rules to ensure that the right users—whether organized by teams or by functions or assigned via responsibility rules—receive and process workflow tasks efficiently.

■ Bank account contract types

The **Enable Bank Account Contract Types** setting in BAM allows you to activate the use of contract types for bank accounts within your system. By enabling this option, you can classify and categorize bank accounts based on different contract types, such as checking, savings, or term deposit accounts, depending on how they are managed within your organization. This feature helps ensure that bank accounts are appropriately categorized and managed according to your internal standards and contract-specific requirements. The use of bank account contract types and when you select them is described in Section 3.1.2.

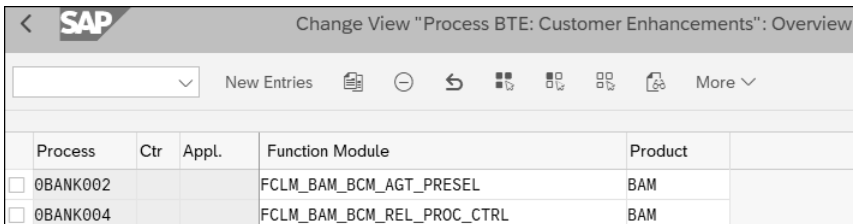
3.2.2 Enable Payment Approval

In the configuration activity for enabling payment approvals, you will define which payment approval mechanism will be used: payment approvers maintained in the BAM master data, or via the SAP Bank Communication Management configuration, which is described in detail in Chapter 7. This decision is crucial, as it determines where and how approval rules are managed and executed. As described earlier, using BAM allows for a simpler, payment-based approval setup, while SAP Bank Communication Management provides more flexibility and control, enabling complex approval workflows based on various parameters such as amount, payment method, or company code.

You can find this configuration under **SAP Financial Supply Chain Management • Cash and Liquidity Management • Bank Account Management • Enable Payment Approval**.

To enable the payment approval process in BAM, add the following two entries to table TPS34 (however, we recommended you maintain them via the menu path) as shown in Figure 3.55:

- **Process “OBANK002”: Function Module “FCLM_BAM_BCM_AGT_PRESEL”: Product “BAM”**
- **Process “OBANK004”: Function Module “FCLM_BAM_BCM_REL_PROC_CTRL”: Product “BAM”**



	Process	Ctr	Appl.	Function Module	Product
<input type="checkbox"/>	OBANK002			FCLM_BAM_BCM_AGT_PRESEL	BAM
<input type="checkbox"/>	OBANK004			FCLM_BAM_BCM_REL_PROC_CTRL	BAM

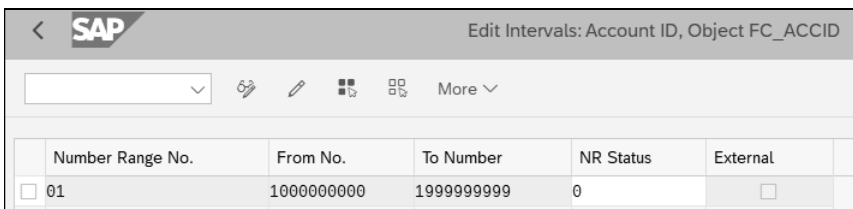
Figure 3.55 Enable Payment Approver

If you want to continue working with the SAP Bank Communication Management setup, you do not need to do anything here; leave the blank values, and the system will pick up rules and workflow assignments from the SAP Bank Communication Management customizing, as described in Chapter 7.

3.2.3 Number Ranges for Technical Accounts ID

The first thing you always need to set up when working with BAM in SAP S/4HANA is the number range. SAP S/4HANA uses various number ranges for different purposes, and several of them are specifically related to BAM. Among these, one of the most important is the number range for technical IDs (see Figure 3.56), which represents the internal numbering of bank accounts in the system. As shown in the earlier screenshots, these technical IDs play a crucial role in identifying and managing bank accounts. Before you can use BAM effectively, it's essential to first create and assign the appropriate number range for these technical IDs.

You can find the setup for the number ranges under **SAP Financial Supply Chain Management • Bank Account Management • Basic Settings • Define Number Ranges for Bank Account Technical IDs**.



	Number Range No.	From No.	To Number	NR Status	External
<input type="checkbox"/>	01	1000000000	1999999999	0	<input type="checkbox"/>

Figure 3.56 Number Range for Bank Account Technical ID

Once you open this customizing activity, you can click **Edit** and provide the desired number range.

Here, you need to specify the range used for BAM accounts. By default, this range can typically be set from 1 to 9999999999, providing a wide scope for assigning internal technical IDs to bank accounts. However, for specific needs, you can use any other number range; these numbers are used only for internal purposes. The example shown here also presents a different number range. This number range ensures that each bank account receives a unique internal identifier, which is crucial for tracking and managing accounts effectively within SAP S/4HANA. Make sure this range is defined and assigned before creating any bank accounts in BAM.

3.2.4 Define Settings for Bank Account Master Data

In this customizing activity, you are specifying the most important settings for BAM, including the account types and how they are going to behave within the system. This activity can be found under **SAP Financial Supply Chain Management • Bank Account Management • Basic Settings • Define Settings for Bank Account Master Data**.

First you need to specify a bank account **Type** (see Figure 3.57) used for BAM and an **Account Type Description**; an underlying **Contract Type—Current Account** represents external bank accounts, whereas a **Payment Service Provider Account** can be used for technical accounts provided by third-party payment providers such as PayPal. **Technical Account** is for any other technical or interim accounts you may want to use. **In-House Bank Account** is used for accounts required in in-house bank processing, and **Other** can cover any remaining accounts. It's good practice to use different contract types for various accounts to differentiate them clearly and give yourself the flexibility to use different functionalities.

Type	Account Type Description	Contract Type
<input type="checkbox"/> 01	Current Account	Current Account
<input type="checkbox"/> 02	Current Account (Internal)	Current Account
<input type="checkbox"/> 03	Deposit Account	Current Account
<input type="checkbox"/> 04	Loan Account	Current Account
<input type="checkbox"/> 05	Investment Account	Current Account
<input type="checkbox"/> 06	Tax Account	Current Account
<input type="checkbox"/> 07	Margin Account	Current Account
<input type="checkbox"/> 08	Salary Account	Current Account
<input type="checkbox"/> 09	Checking Account	Current Account
<input type="checkbox"/> 10	Lockbox Account	Current Account
<input type="checkbox"/> 88	In-House Bank Account Type	In-House Bank Account

Figure 3.57 Bank Account Types

Next, click **Sensitive Fields for Activated Banks**. Here you need to specify which fields are going to trigger approval workflows and which can be changed without triggering a workflow (see Figure 3.58). To specify which fields are sensitive fields and workflow fields in BAM, simply click **New Entries**, then select the object that represents the tab where the field is located in the bank account (for example, **General Data** or **Payment Approver**), and choose the field name you want to include. Once you've made your selections, click **Save** to confirm your entries. This setup ensures that only the designated fields are monitored for sensitive changes or included in workflow triggers. Fields specified here will trigger the approval workflow.

Change View "Sensitive Fields for Activated Bank Accounts": Overview

🔍
New Entries
📄
⌂
🔄
📊
📈
📉
📋
More ▾

Dialog Structure

- Account Type Definition
- Sensitive Fields for Activated Bank Accounts**
- Sensitive Fields for Bank Accounts
- Define Selection Variants for Sensitive Fields
- Define Import Methods for Bank Statements
- Define Bank Statement Groups
- Define Payment Approver Groups
- Define Approval Patterns
- Assign Approval Patterns
- Define Automatic Approval/Rejection

Sensitive Fields for Activated Bank Accounts

Object	Field name	Field Description	Selection Variant ID	Inactive
<input type="checkbox"/>	Bank Account ▾ ACC_NUM	Bank Account Number		<input type="checkbox"/>
<input type="checkbox"/>	Bank Account ▾ ACC_NUM	Bank Account Number	S001	<input type="checkbox"/>
<input type="checkbox"/>	Bank Account ▾ CASH_POOL_ID	Cash Pool ID		<input type="checkbox"/>
<input type="checkbox"/>	Bank Account ▾ CASH_POOL_ID_PARTICIPANT	Cash Pool for Subaccount		<input type="checkbox"/>
<input type="checkbox"/>	Bank Account ▾ CASH_POOL_TARGET_AMT_MAX	Target Balance for Subaccount		<input type="checkbox"/>
<input type="checkbox"/>	Bank Account ▾ CASH_POOL_TARGET_AMT_MIN	Minimum Target Amount for Subaccount		<input type="checkbox"/>
<input type="checkbox"/>	Bank Account ▾ CASH_POOL_TRANSFER_AMT_MIN	Minimum Transfer Amount for Subaccount		<input type="checkbox"/>
<input type="checkbox"/>	Bank Account ▾ EXT_RULE_ID	Monitoring Rule for Intraday Statements		<input type="checkbox"/>
<input type="checkbox"/>	Bank Account ▾ IBAN	IBAN (International Bank Account Number)		<input type="checkbox"/>
<input type="checkbox"/>	Bank Account ▾ PAYMENT_METHOD_L	Payment Method for Header Account		<input type="checkbox"/>
<input type="checkbox"/>	Bank Account ▾ PAYMENT_METHOD_P	Payment Method for Subaccount		<input type="checkbox"/>
<input type="checkbox"/>	Bank Account ▾ RECONCILE_INTRADAY	Reconcile Intraday Memo Records		<input type="checkbox"/>
<input type="checkbox"/>	Bank Account ▾ TOLERANCE_GROUP	Tolerance Group Key		<input type="checkbox"/>
<input type="checkbox"/>	Bank Account ▾ UPLOAD_BS_FLAG	Upload of Intraday Statements		<input type="checkbox"/>

Figure 3.58 Sensitive Fields for Approval Workflow

In the next step, click **Define Import Methods for Bank Statements**, where you need to specify the import methods for the selected bank statements (see Figure 3.59).

▼

More ▼

Dialog Structure

Account Type Definition

Sensitive Fields for Activated Bank

Sensitive Fields for Bank Accounts

Define Selection Variants for Sensitive Fields

Define Import Methods for Bank Statements

Define Bank Statement Groups

Define Payment Approver Groups

Define Approval Patterns

Assign Approval Patterns

Define Automatic Approval/Rejection

Define Import Methods for Bank Statements

Import Method ID	Account Description
<input type="checkbox"/> ZEBS	Automatic Import of the Electronic Bank Statement
<input type="checkbox"/> ZMBS	Manual Bank Statement
<input type="checkbox"/> ZNO_BS	No Bank Statement
<input type="checkbox"/>	

Figure 3.59 Import Methods for Bank Statements

This is a setting you define within the BAM account, and it serves to inform you and others about how a particular bank statement is being uploaded into the system—for example, via manual upload, file interface, or an automated connection with the bank. (There are no predefined options in the system; these are text fields, so you can enter any values here.) Although this setting does not have a significant impact on subsequent processing or system behavior, it plays an informative role, helping to describe the source and method of statement delivery.

In the next step, click **Define Payment Approver Groups** (see Figure 3.60). Here you can specify any values; enter the approver group (**Appr. Grp.**) and its **Description**. You will then select these approver groups in the Manage Bank Accounts app—specifically, on the **Payment Approver** tab (provided that you have already enabled payment approver functionality in BAM in the earlier configuration steps).

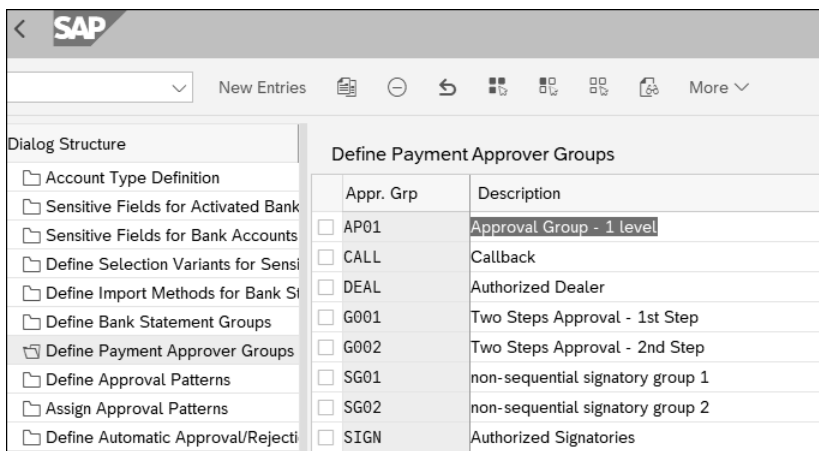


Figure 3.60 Define Payment Approvers

In the next configuration activity, **Define Approval Patterns**, you need to define the approval pattern (see Figure 3.61), which determines how payments are approved within the system. Click **New Entries**, then simply specify the name for the pattern and select how many sequences are required. SAP provides up to four levels of payment approvals, supporting the eight-eyes principle. You then assign each approval level to an approver group and specify the currency, as well as the minimum amount for both payments and batches that will trigger the workflow. This involves specifying the sequence and structure of approvals, such as the number of approval levels and the roles responsible for each step. Once the approval pattern is defined, it must be assigned to the relevant company codes and bank codes to ensure that the correct approval process is applied during payment processing for those entities.

Pattern	Non-Seq.	Appr. Seq.	Appr. Grp	CrCY	Min. Amount for Payment	Min. Amount for Batch
<input type="checkbox"/> AP01	<input type="checkbox"/>	First Step	✓ AP01	EUR	1.00	10,000.00
<input type="checkbox"/> P001	<input type="checkbox"/>	First Step	✓ G001	EUR		
<input type="checkbox"/> P001	<input type="checkbox"/>	Second Step	✓ G002	EUR		
<input type="checkbox"/> S001	<input checked="" type="checkbox"/>		✓ SG01			
<input type="checkbox"/> S001	<input checked="" type="checkbox"/>		✓ SG02			
<input type="checkbox"/> T001	<input type="checkbox"/>	First Step	✓ G001	EUR		
<input type="checkbox"/> T001	<input type="checkbox"/>	Second Step	✓ G002	EUR		

Figure 3.61 Define Approval Patterns

3.2.5 Define Settings for Bank Account Contract Types

As described earlier, this configuration activity is used to set up additional settings related to the behavior and capabilities of bank accounts—specifically, when bank account contract types are enabled. Here you define how bank accounts behave throughout their lifecycle—for example, which statuses are relevant, how accounts are activated in each status, and what repair actions can be taken. You can also control the visibility of specific tabs in the bank account master data, depending on the type of bank account. These settings can be tailored based on contract type, account type, and company code, allowing for a more flexible and structured approach to managing bank account data.

Note that this activity is only applicable if you have already enabled bank account contract types in your configuration. The prerequisites include activating the contract type feature in the **Define Basic Settings** activity (see Section 3.2.1) and defining the contract types themselves in the **Define Settings for Bank Account Master Data** activity (see Section 3.2.4).

You can find this configuration activity under **SAP Financial Supply Chain Management • Bank Account Management • Basic Settings • Define Settings for Bank Account Contract Types**.

First, click **Define Repair Function for Bank Accounts**. In this activity (see Figure 3.62), you define repair actions for bank accounts to manage corrections during their lifecycle. Two options are available:

- **Return to Preceding Status**

Allows for reverting a bank account to its previous status; enabled by default.

- **Restore Predecessor Version**

Restores the account to an earlier version; useful for both status and data changes. A version is created when the account is created, its status changes, or sensitive data is updated.

Both options have an impact on the next configuration steps, in which you define how account statuses should change depending on user actions.

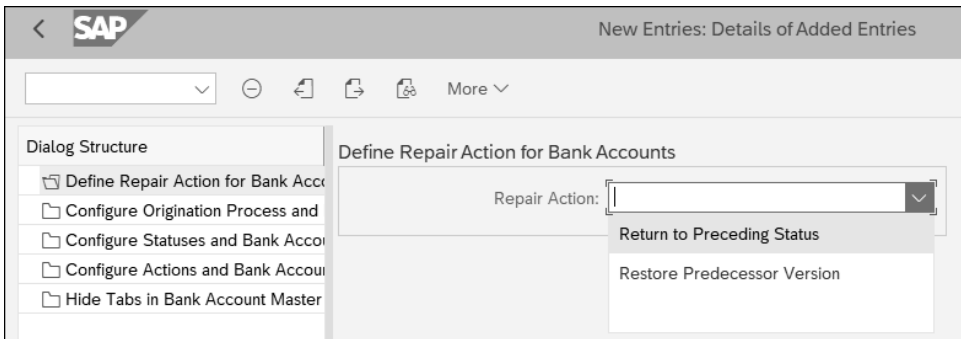


Figure 3.62 Repair Action for Bank Accounts

Next, click **Configure Origination Process and Manual Creation** (see Figure 3.63). When you click **New Entries** in this configuration step, you can select a contract type and decide whether to enable the originating status, which requires you to submit a bank account for creation. You can also choose to disable manual creation for accounts that are generated automatically—such as those created through the eBAM interface or the default BAM—thus restricting manual entries and ensuring consistency.

When the bank account contract type is enabled but the bank account origination process is not activated in the **Define Settings for Bank Account Contract Types** configuration, the system uses the default opening process, which includes only two statuses: **Inactive** and **Active**. If you enable the bank account origination process, additional statuses become available—**In Opening Process**, **Opened at Bank**, and **Active**. This allows for the use of eBAM correspondence and the possibility to design more advanced workflow scenarios. When manual creation is disabled, new bank accounts can only be created through the bank account application process. This means that a formal application must be submitted using the Submit Bank Account Applications app to request the creation of a new account.

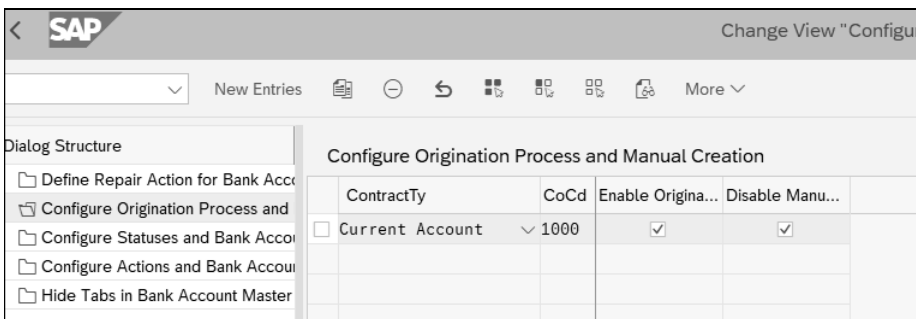


Figure 3.63 Configure Origination Process and Manual Creation


In the next step, click **Configure Statuses and Bank Account Control Modes**. In this activity, you configure the statuses and control modes for the bank account lifecycle. SAP S/4HANA provides predefined mandatory and optional statuses. Mandatory statuses are always enabled, whereas optional ones are enabled by default, unless you create an entry and deselect the **Enable Status** option.

The statuses include **Inactive**, **Active**, **Closed**, **Application Approved**, and others depending on whether the bank account origination process is enabled. For each status, you define the relevant bank account contract type, account type, and company code. Two control modes are defined for each status:

- **Control Mode to Next Status**
Controls progression to the next status.
- **Control Mode to Preceding Status**
Controls regression to the previous status.

Once a bank account reaches the **Pre-Opening Preparation** status, it cannot be reverted to **Application Approved**.

In our example (see Figure 3.64), you can, for example, set up **Direct Change** for **In-House Bank Accounts**, meaning that no workflow will be triggered for changing statuses in in-house bank accounts. In this way, you can differentiate the setup for the external and internal bank accounts.



Contract Type	Account Type	Company Code	Status	Enable Status	Control Mode to Next Status	Control Mode to Preceding Status
jk Ac...	88	1010	02 Active	<input checked="" type="checkbox"/>	Direct Change	Direct Change
In-House Bank Ac...	88	1010	15 Closed	<input checked="" type="checkbox"/>		
In-House Bank Ac...	88	1010	09 Marked for Closing	<input checked="" type="checkbox"/>	Direct Change	Direct Change

Figure 3.64 Configure Statuses and Bank Account Control Modes

In the next activity (see Figure 3.65), click **Configure Action and Bank Account Control Modes**. Here you can configure the bank account control modes for actions performed on bank account master data, such as modifying or deleting a bank account. In this configuration step, select the contract type and the company code, then choose the action—whether it applies to changing the status or deleting the account. You also specify if it is a direct change, requires two-person verification, or should go through a workflow. Finally, you set the control mode for restoring the account, which applies only to tracing and managing attending bank accounts—for example, transitioning an account from inactive to active or closing a bank and its associated accounts completely. This setup is mainly required when you’re using eBAM functionality as normally you only need the **Active**, **Inactive**, and **Closed** bank account statuses. SAP provides these additional statuses for more complex scenarios if needed.

If the **Restore Predecessor Version** repair action is enabled, you will also need to specify the **Control Mode for Restore**. This setting determines how the bank account is restored to its previous version. For our example, we want to have autoupdating turned on for in-house bank accounts.

Dialog Structure	Configure Actions and Bank Account Control Modes				
Define Repair Action for Bank Accounts	Contract Type	Account Type	Com... Action	Control Mode	Control Mode for Restore
Configure Origination Process and	<input type="checkbox"/> In-House Bank Ac... 88		Change	Direct Change	Direct Change
Configure Statuses and Bank Account	Current Account				
Configure Actions and Bank Account	Current Account				
Hide Tabs in Bank Account Master					

Figure 3.65 Configure Actions and Bank Account Control Modes

In the **Hide Tabs in Bank Account Master Data** activity, you define which tabs are visible in the bank account master data. For example, the **Cash Pools** tab may not be needed for technical bank accounts, so you can deselect this option for bank accounts with the technical bank contract type.

You can specify the bank account **Contract Type**, **Account Type**, and **Company Code** for each entry. The combinations you can use, in order of priority, are as follows:

1. Contract type + account type + company code
2. Contract type + account type
3. Contract type

Keep in mind that the tab visibility settings will override the field status group settings from the **Manage Field Status Groups** activity. In our example (see Figure 3.66), we have hidden the tabs that are not needed for in-house bank accounts.

Dialog Structure	Hide Tabs in Bank Account Master Data									
Define Repair Action for Bank Accounts	Contract Type	Account Type	Company Code	Bank Relationship	House Bank Account Conn...	Payment Approvers	Overdraft Limits	Multi-Language Descriptions	Cash Pool	Bank ...
Configure Origination Process and	<input type="checkbox"/> In-House Bank Account 88		1010	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Configure Statuses and Bank Account	Current Account			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Configure Actions and Bank Account	Current Account			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Hide Tabs in Bank Account Master				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 3.66 Hide Tabs in Bank Account Master Data

3.2.6 Define Field Status Groups

This customizing activity (see Figure 3.67) allows you to define field status groups and specific rules that control how fields appear and behave in the Manage Bank Accounts app. You can set fields to be mandatory, hidden, editable, or read-only on the user interface. The process involves creating field status groups, selecting variants, and assigning them to specific activities and account types. In this activity, you first need to specify the **Define Field Status Group** and then select **Define UI Field Status**. The system will then display a long list of all the fields available in BAM; currently, there are 235 fields. If you have any custom fields, you also need to assign them here. For each field that is

relevant to you, you can choose whether the field status should be read-only, hidden, editable, or mandatory. Note that in most cases, you don't need to change anything, because SAP already provides a wide range of field status groups that you can use. You should only create or modify entries here for specific cases that truly require it.

In most cases, the default settings are sufficient, but you may adjust them based on your needs—such as by making the **IBAN** field mandatory for certain account types or hiding fields that are not relevant (e.g., **Intraday**).

You can find this configuration activity under **SAP Financial Supply Chain Management • Bank Account Management • Manage Field Status Groups**.

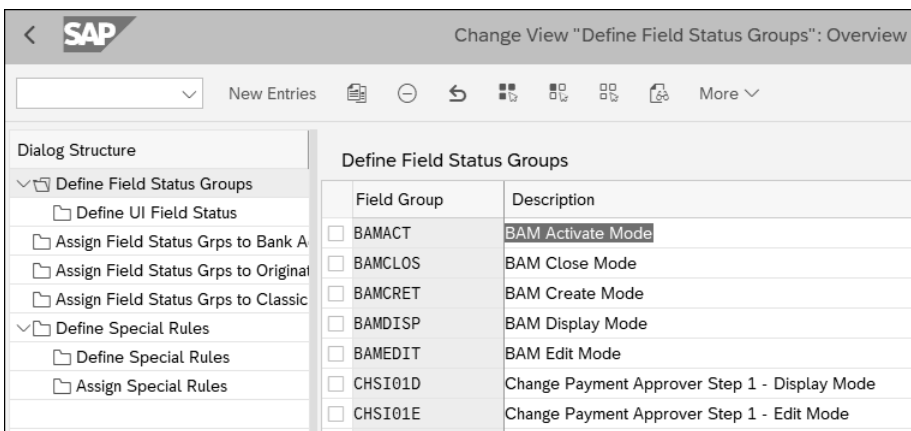


Figure 3.67 Define Field Status Groups

If you create new field status groups, you need to assign them to bank account master data (see Figure 3.68). However, SAP has predefined these values, so changes are only required if you have specific needs. If you need to change anything, click **New Entries**, select the **Contract Type** and **Scenario**, and then assign your newly created field status group.

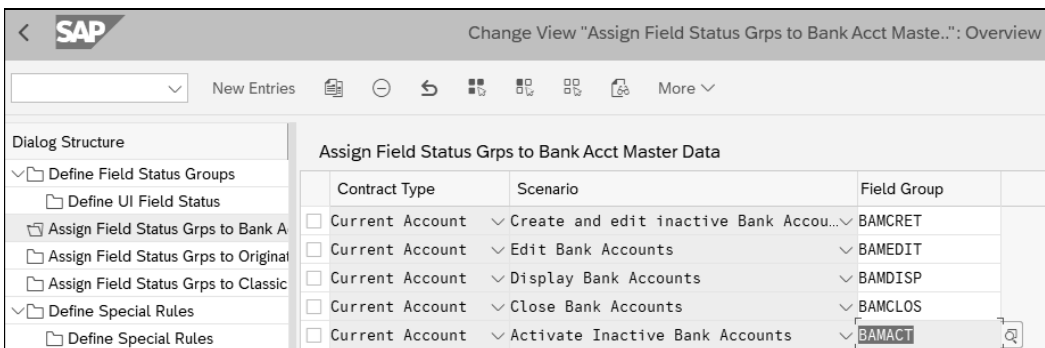


Figure 3.68 Assign Field Status to Bank Account

3.3 Summary

In this chapter, we explored the core functionalities of BAM within SAP S/4HANA, focusing on its essential role in maintaining a centralized and controlled structure for managing bank accounts. We walked through the prerequisites and configuration steps needed to set up BAM and highlighted the key apps and tools provided by SAP to simplify and standardize your banking processes. From creating and managing bank accounts to understanding their integration into payment workflows, we laid the foundation for more advanced treasury operations. We also clarified that in-house bank accounts are created and managed within BAM, further reinforcing its central role in your treasury setup.

With BAM accounts now in place, you're ready to move on to the next stage—implementing more advanced payment scenarios using advanced payment management and in-house banking. These solutions allow you to model complex payment workflows across your internal and external landscape, enabling greater control and efficiency. From there, you can progress into cash management, where you'll manage balances, assign cash flows, and forecast liquidity across your organization. At this point, you should have a solid understanding of BAM's place within the broader SAP S/4HANA ecosystem and be well prepared to continue building a comprehensive, integrated treasury framework.



Adrian Matys, Jean-Michele Szczecina

Payment and Bank Communication Management with SAP

- Implement in-house banking, advanced payment management, cash management, and more with step-by-step instructions
- Deploy bank connectivity methods and tools
- Establish a payment factory in your SAP S/4HANA landscape



www.sap-press.com/6053

We hope you have enjoyed this reading sample. You may recommend or pass it on to others, but only in its entirety, including all pages. This reading sample and all its parts are protected by copyright law. All usage and exploitation rights are reserved by the author and the publisher.

Adrian Matys is an SAP finance professional with more than 10 years of experience. His focus areas include treasury, advanced payment management, cash management, and collections and dispute management. Jean-Michele Szczecina is a subject matter expert in the areas of payments and cash management at Zanders. He has been involved in numerous payment management and cash management projects.

ISBN 978-1-4932-2682-5 • 653 pages • 08/2025

E-book: \$94.99 • Print book: \$99.95 • Bundle: \$109.99



Rheinwerk
Publishing