Flat-Rate Remuneration & Flat-Rate Run in SAP FS-ICM

1 Introduction

Flat-rate payments are non-performance-based remunerations granted independently of specific sales transactions and typically follow a fixed schedule (e.g., monthly). The Flat-Rate Run is used to automatically create these flat-rate payments and record the result in a document.

Flat-rate payments can be divided into salary-like components and flat-rate compensations. Salary-like compensations include, for example, social allowances or payments during sickness and vacation. Flat-rate compensations, on the other hand, cover specific expenses such as office or advertising costs.

A special type of flat-rate payment is the start-up subsidy, designed for newly joining distribution partners. Especially at the beginning of their activities, commission partners often cannot rely solely on their sales income. A start-up subsidy – either as a fixed payment or as a guaranteed promise – secures a basic income during this initial phase.

Besides positive flat-rate agreements, there are also negative flat-rate components, such as costs that the commission partner must bear (e.g., insurance premiums or specific administrative fees or cost for using company hard- and/or software). These negative components are also considered during the Flat-Rate Run and are offset against the rest of the compensation.

The framework for these types of flat-rate compensation is defined in standard commission contracts, while specific amounts are negotiated individually with each partner. The Flat-Rate Run ensures that all relevant flat-rate payments are automatically calculated and posted at the end of each period.

2 Customizing, Standard Flat-Rate Agreement and Commission Contract

To ensure that the Flat-Rate Run can create flat-rate payments, the following preparatory steps are required:

2.1 Customizing for Flat-Rate Remuneration

Customizing Path:

CACSIMG > Incentive and Commission Management > Basic and Master Data
> Standard Commission Contract > Remuneration Agreements
> Agreement for Flat-Rate Remunerations

Define Account Assignment Types

The Account Assignment Type is a technical key that contains all booking-relevant information (e.g., posting key, cost type). It also includes settings for both forms of account assignment: Settlement Account Assignment and Cost Account Assignment.

Settlement Account Assignment is linked to the settlement type in the configuration (customizing for Settlement Type). The Cost Account Assignment is linked to the flat-rate compensation types (customizing for Flat-Rate Remuneration Type).

Maintain remuneration type groups

A remuneration type group acts as a collective container for different remuneration types. It is assigned to the remuneration type.

Edit Flat-Rate Remuneration Types

Remuneration types define procedures and parameters for calculating the remuneration. The configuration includes:

- Remuneration Type
- Performance Group
- Compensation Indicator (direct, indirect, or statistical)
- Account Assignment Type or Combination Number Payment Timing (in advance, in arrears, or none)
- Calculation Method (full or differential commission)

Maintain Standard Agreement for Flat-Rate Remuneration

A standard agreement for flat-rate remunerations defines the parameters under which a flat-rate remuneration is granted. This includes the assigned remuneration types, the validity period, a fixed amount if applicable, and the frequency of remuneration.

A fixed amount can optionally be stored in the standard agreement. Additionally, a minimum and a maximum value can be specified.

The agreement defines how often the remuneration is to be granted (e.g. monthly). This frequency determines the intervals at which the Flat-Rate Run generates remuneration.

The parameters defined here form the basis for the individual agreement in the commission contract.

2.2 Assign Standard Flat-Rate Agreement to the Standard Commission Contract

TC:

NWBC (Role ICM Analyst)

1. Select or create a standard commission contract.
2. Use the correct change date and navigate to optional agreements.
3. Choose agreement type "08 Flat-Rate Remuneration".
4. Add available standard flat-rate agreements to the active ones.
5. Save, verify, and activate the changes.

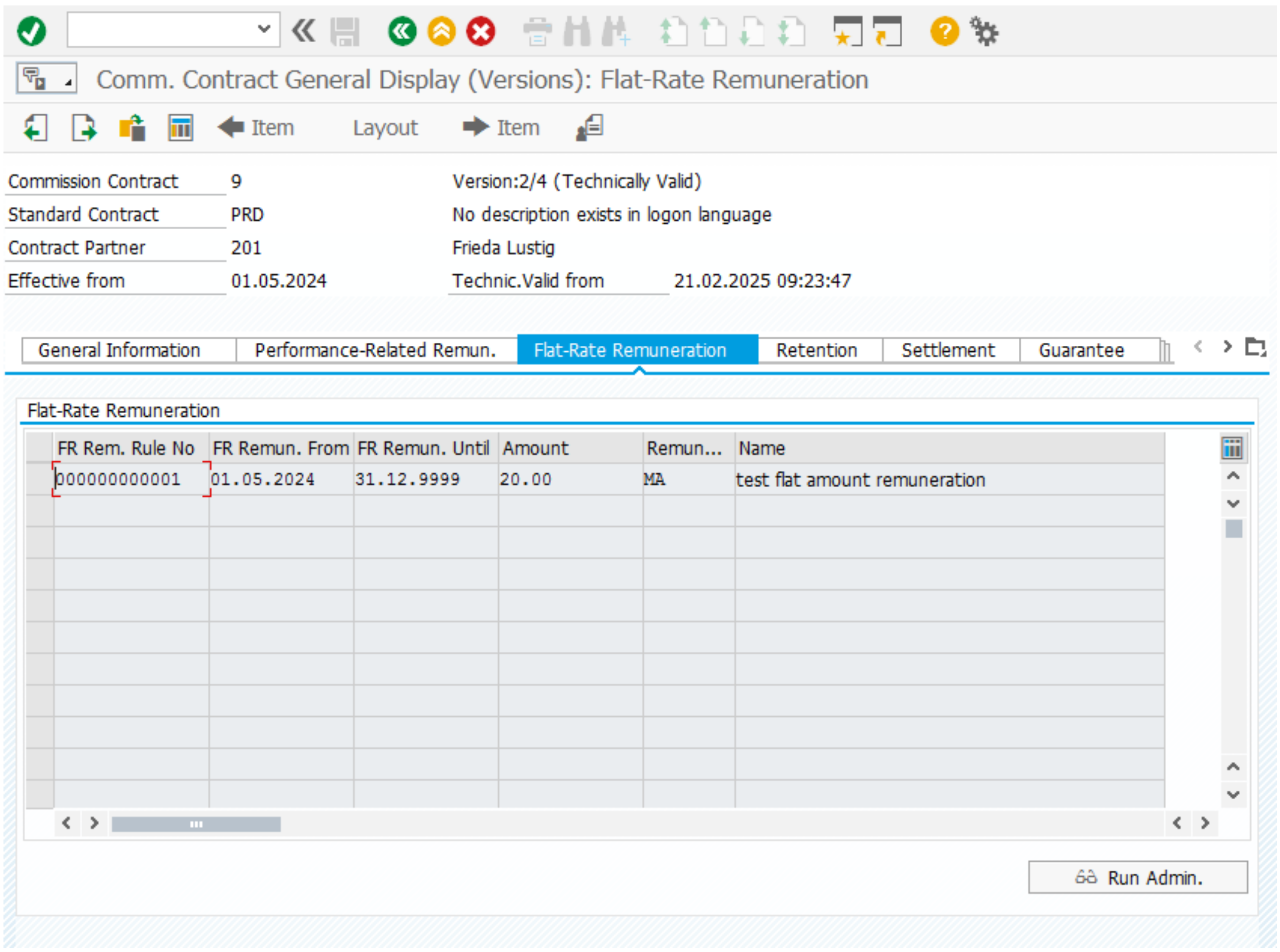
If the assignment via NWBC fails, tables like CACS_CONTR, CACS_STAGASS, CACS_STDCTR, and CACS_STDCTRT must be updated manually.

2.3 Creating or Editing a Commission Contract

Once the standard flat-rate agreement (Std. Agreement) has been configured and assigned to the standard commission contract, a corresponding flat-rate remuneration agreement (Agreement) can be created in the actual commission contract (under the tab "Flat-Rate Remuneration").

To do this, the following must be entered:

- A flat-rate remuneration rule number
- Validity period (from/to)
- A remuneration amount (must lie between min/max values from Customizing, if defined) The remuneration type



Commission Contract	9	Version:2/4 (Technically Valid)	
Standard Contract	PRD	No description exists in logon language	
Contract Partner	201	Frieda Lustig	
Effective from	01.05.2024	Technic.Valid from	21.02.2025 09:23:47

Flat-Rate Remuneration

FR Rem. Rule No	FR Remun. From	FR Remun. Until	Amount	Remun...	Name
000000000001	01.05.2024	31.12.9999	20.00	MA	test flat amount remuneration

Picture 1: commission contract with flat-rate agreement

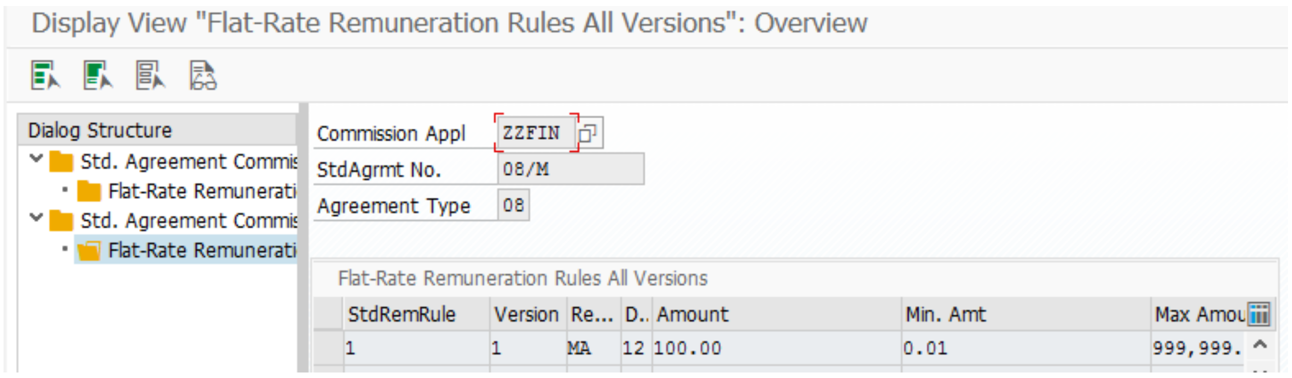
Special Case: Flat-Rate Amount

If no amount is entered in the commission contract's flat-rate agreement, and a fixed amount is defined in the standard agreement, the system displays a message indicating that values (specifically the amount) will be inherited from the standard agreement.

Example:
If the standard agreement contains a fixed amount of €100 (see Picture 2), and a

flat-rate agreement without amount is added to commission contract 10 (see Picture 3), the system prompts a message. Upon confirmation, the amount from Customizing (€100) is applied to the agreement.

However, if no fixed amount is stored in the standard agreement, then the flat-rate amount becomes a mandatory field in the commission contract.


Display View "Flat-Rate Remuneration Rules All Versions": Overview

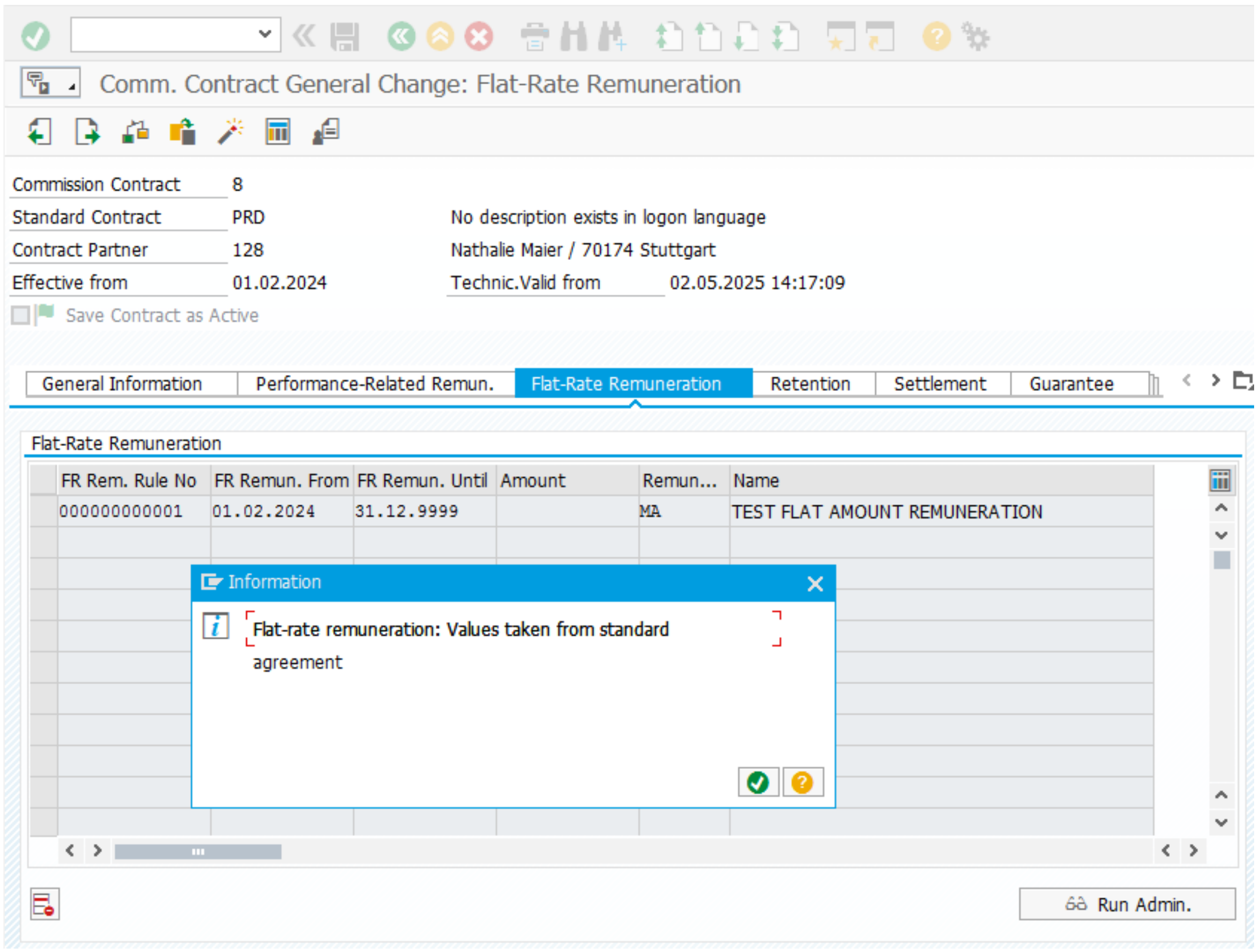
Dialog Structure
- Std. Agreement Commis
  - Flat-Rate Remunerati
- Std. Agreement Commis
  - Flat-Rate Remunerati

Commission Appl	ZZFIN
StdAgrmt No.	08/M
Agreement Type	08

Flat-Rate Remuneration Rules All Versions

StdRemRule	Version	Re...	D..	Amount	Min. Amt	Max Amou
1	1	MA	12	100.00	0.01	999,999.

Picture 2: standard agreement with fix amount


Comm. Contract General Change: Flat-Rate Remuneration

Commission Contract	8		
Standard Contract	PRD	No description exists in logon language	
Contract Partner	128	Nathalie Maier / 70174 Stuttgart	
Effective from	01.02.2024	Technic.Valid from	02.05.2025 14:17:09

Save Contract as Active

Flat-Rate Remuneration

FR Rem. Rule No	FR Remun. From	FR Remun. Until	Amount	Remun...	Name
000000000001	01.02.2024	31.12.9999		MA	TEST FLAT AMOUNT REMUNERATION

Information

Flat-rate remuneration: Values taken from standard agreement

Picture 3: commission contract with flat-rate agreement without individual amount

2.4 Customizing for Run Administration and Process Control

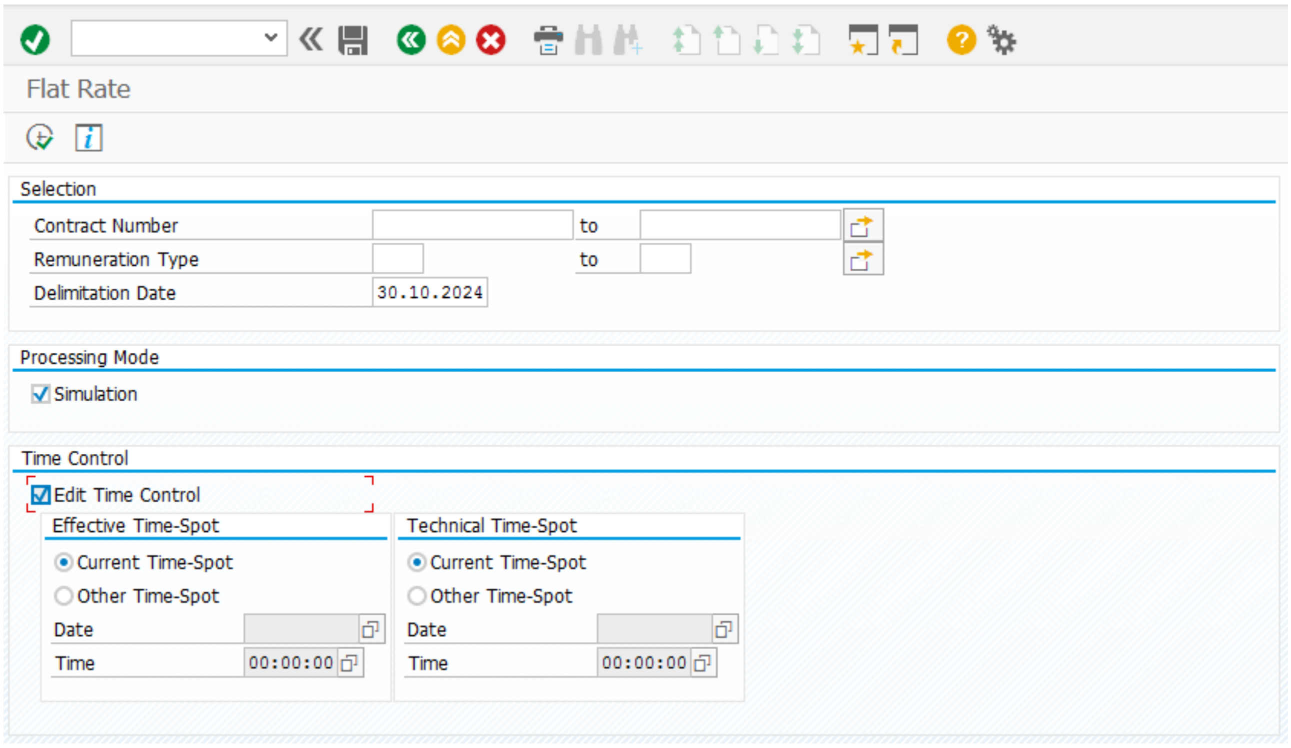
Customizing Path:
CACSIMG > Incentive and Commission Management > Basic Functions > Prepare Mass Processes

The number range of the run ID is created here, the process types are defined, and the process types are allocated to the process methods.

3 Flat-Rate Run

The Flat-Rate Run is part of the periodic settlement processes in ICM. As previously mentioned, it serves to calculate and process flat-rate remunerations for sales partners.

You can launch the Flat-Rate Run using transaction CACSFR1 (Package: CACSF1; Report: CACS_PRC_FLAT_RATE; Generated class: ZZFIN_CL_RPC_PRD_FR1 for application ZZFIN)


Flat Rate

Selection

Contract Number		to	
Remuneration Type		to	
Delimitation Date	30.10.2024		

Processing Mode

☑ Simulation

Time Control

☑ Edit Time Control

Effective Time-Spot		Technical Time-Spot	
● Current Time-Spot		● Current Time-Spot	
○ Other Time-Spot		○ Other Time-Spot	
Date		Date	
Time	00:00:00	Time	00:00:00

Picture 4: selection screen of the Flat-Rate Run

3.1 Selection Screen

Contract Number

Specify the commission contracts to be processed. If left blank, all contracts will be included.

Remuneration Type

One or more flat-rate remuneration types can be specified. If left blank, all types are processed.

Delimitation Date

Defaults to the system date and determines up to which date the remuneration is calculated.

Example: Contract 9 includes a flat-rate agreement starting 01.05.2024. With a delimitation date of 08.11.2024, remuneration is calculated through 31.10.2024. With a delimitation date of 08.10.2024 or 31.10.2024, the system only processes remuneration through 30.09.2024.

Simulation Mode

If selected, remuneration is only simulated – no postings are made.

Time Control

Defines which version of the commission contract and flat-rate agreement is used for calculation.

3.2 Processing

The Flat-Rate Run follows a defined sequence of steps that ensures all flat-rate remunerations are systematically checked and processed in accordance with the applicable agreements. Each step has a specific function and forms the basis for the next one, to ensure smooth and traceable processing of the remunerations.

The following describes the essential processing phases of the Flat-Rate Run.

Identification of commission contracts

At the beginning of the Flat-Rate Run, the system identifies the commission contracts that contain flat-rate agreements and are relevant for the current period. This selection is based on criteria such as contract partner, validity of the agreement, and the date of the next due remuneration. The commission contracts are then processed one by one in the subsequent process steps.

Review of the remuneration agreement and period rules

For each selected commission contract, the system checks the stored remuneration agreements. The corresponding remuneration rule is used to determine which amounts and services need to be considered. If there are different settlement periods, the period rule ensures that the correct remuneration period is determined. This guarantees that only remunerations relevant for the current processing period are included.

Determining the due date

After checking the period rule, the system analyses whether the flat-rate remuneration is due in the current period. Flat-rate remunerations become due:

- if they have not yet been paid in the current or even in previous periods
- if they are defined as retroactive payments for the previous period
- if they are defined as an advance for the upcoming period.

This step determines whether a flat-rate remuneration is included in the processing.

Creation of a posting document

If a flat-rate remuneration is recognized as due, the system creates a document. This document contains the remuneration amount and the related information from the flat-rate remuneration agreement and serves as the basis for further processing or later verification.

Logging of the processing

The entire Flat-Rate Run is logged. The system creates a detailed log that records every processed contract and the applied remuneration rules. This logging ensures transparency and allows for easy tracking of all steps performed, including the criteria checked and the determined due payments.

Through the structured processing of the Flat-Rate Run, the system ensures that all relevant flat-rate remunerations are accurately processed and documented in accordance with the contractual agreements.

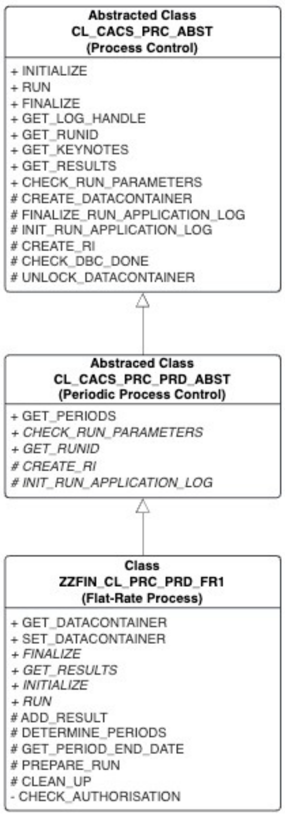
3.3 Technical Components

The technical implementation of the Flat-Rate Run relies on core functions that ensure flexible and efficient processing. The two most important technical components in this context are process control and run administration.

3.3.1 Process Control

Process control is used to automate and coordinate the various steps within the remuneration processes. While run administration focuses on the execution of specific calculation runs, process control ensures that the correct sequence of the individual process steps is followed. It manages and monitors all steps from data initialization to final posting and is therefore responsible for the orderly and successful execution of the processing workflow.

The following model illustrates the inheritance relationship between process control and the Flat-Rate Run.



Picture 5:
inheritance process
control

The italicized methods have each been redefined in the inheriting class. This way, the Flat-Rate Run is controlled using the methods

INITIALIZE (initialize process), RUN (start process), and FINALIZE (complete process). The other inherited methods support the individual process steps.

All objects related to process control are contained in the package CACSPR (ICM process control).

The class ZZFIN_CL_PRC_PRD_FR1 was generated through the application generation of ZZFIN. Examples and screenshots shown on the page refer to the application ZZFIN.

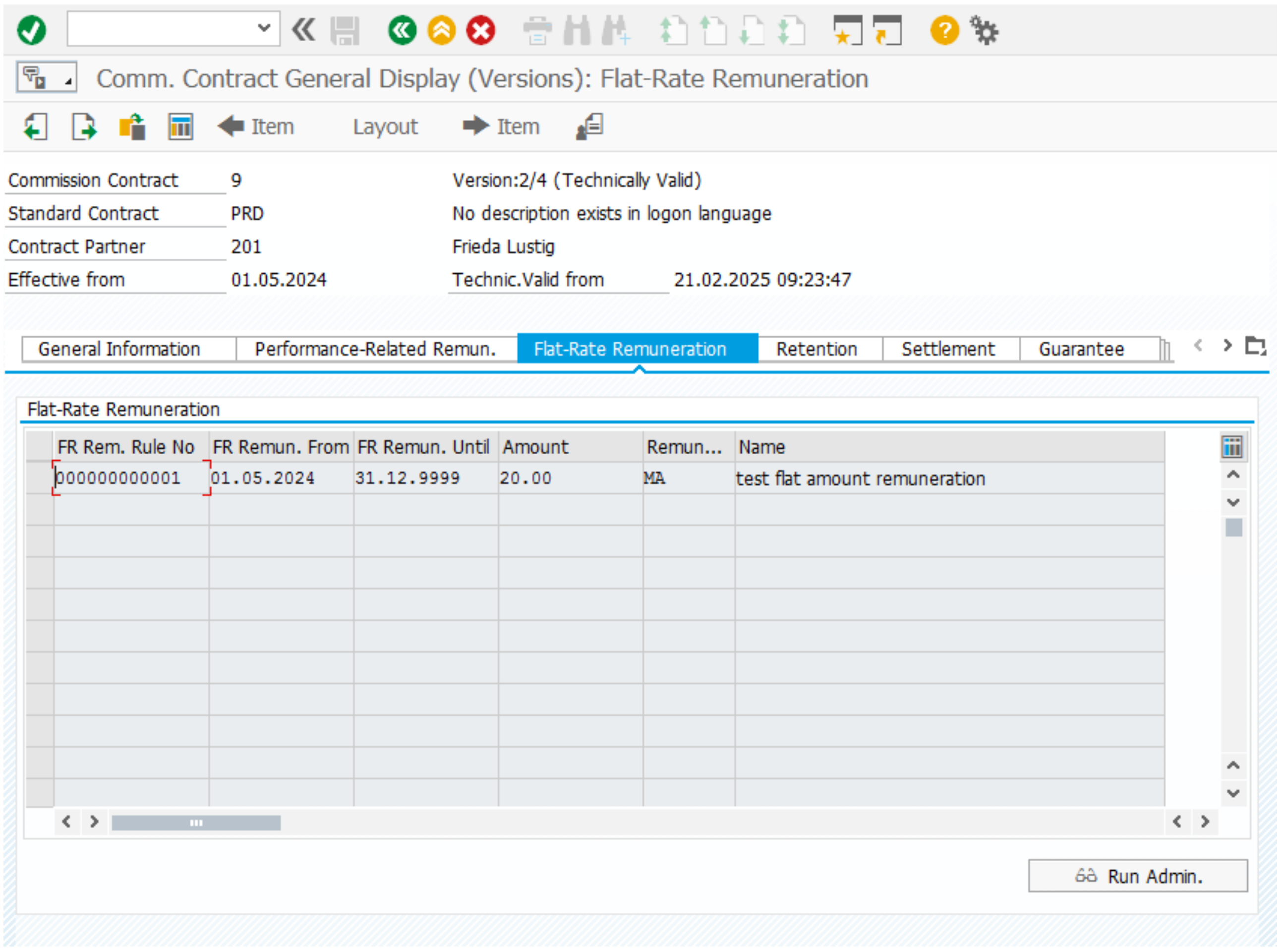
3.3.2 Run Administration

Run administration focuses on managing and executing remuneration calculations. A "run" refers to a completed unit of calculations and processing steps that is applied to a specific set of remuneration data, usually within a defined period or for specific records.

Run administration centrally logs the technical information for a periodic run (or also for non-periodic runs) and documents which process ran in which period for the respective commission contract.

In addition, it ensures that for periodic processes, only one run is executed per period, and that all information is up-to-date and transparently displayed.

The following picture shows a screenshot of commission contract 9, in which the tab "Flat-Rate Remuneration" is selected. In the bottom right corner, there is a button labelled "Run Admin." Clicking this button opens the run administration view.



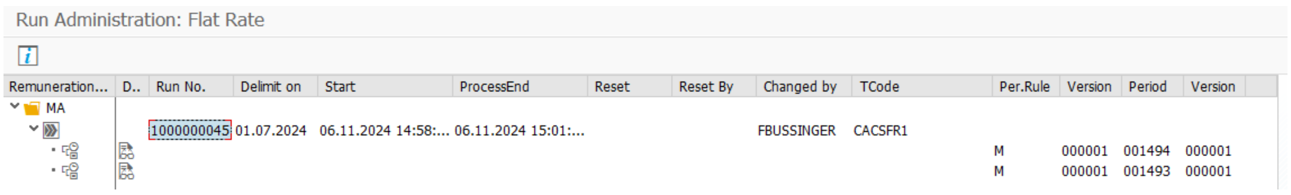
Comm. Contract General Display (Versions): Flat-Rate Remuneration

Commission Contract	9	Version:2/4 (Technically Valid)	
Standard Contract	PRD	No description exists in logon language	
Contract Partner	201	Frieda Lustig	
Effective from	01.05.2024	Technic.Valid from	21.02.2025 09:23:47

Flat-Rate Remuneration

FR Rem. Rule No	FR Remun. From	FR Remun. Until	Amount	Remun...	Name
000000000001	01.05.2024	31.12.9999	20.00	MA	test flat amount remuneration

Picture 6: navigation to run administration



Run Administration: Flat Rate

Remuneration...	D..	Run No.	Delimit on	Start	ProcessEnd	Reset	Reset By	Changed by	TCode	Per.Rule	Version	Period	Version
MA													
		1000000045	01.07.2024	06.11.2024 14:58:...	06.11.2024 15:01:...			FBUSSINGER	CACSFR1				
										M	000001	001494	000001
										M	000001	001493	000001

Picture 7: run administration for flat-rates

Using run administration, it is possible to check directly from within the commission contracts which periodic processes have already been executed or reversed. From the run administration view, you can also navigate to the related documents, as well as to the applied period rules and periods.

In the example, commission contract 9 was processed for periods 5 and 6 within the Flat-Rate Run.

Technically, run administration is linked to process control and therefore to the periodic processes. Through the inheritance of the CREATE_RI method, an instance of run administration is created, which is used at many points in the periodic processes.

Interaction between Process Control & Run Administration

As part of a flat-rate remuneration process, run administration initiates the Flat-Rate Run and carries out all related calculations within a single run. Process control, in this case, ensures that prerequisites are fulfilled (e.g. data entry completed) and that all subsequent processes are correctly executed after the calculation.

Both components work together to ensure that the Flat-Rate Run is executed efficiently and in an orderly manner.

3.3.3 Commission Document

All remuneration processes, including the Flat-Rate Run, document the remuneration results in the commission document. A commission document always consists of a document header, which contains general data applicable to the entire document, and, if applicable, of document line items that contain data relevant to the individual document types.

Document types are used to classify documents and are stored in the document header. They apply to the entire document. The document type has the following functions:

- It controls how the document is stored,
- defines the structure and layout of a document,
- differentiates the types of business transactions to be posted (e.g. posting, reversal),
- and controls field behaviour.

The document types are contained in table TCACS_DOCT.

Commission documents can be reversed depending on the business transaction to be posted. This reversal is handled in the commission system through reversal documents. A commission document cannot be changed; therefore, it does not carry a version date. Instead, a document can only be reversed and newly created.

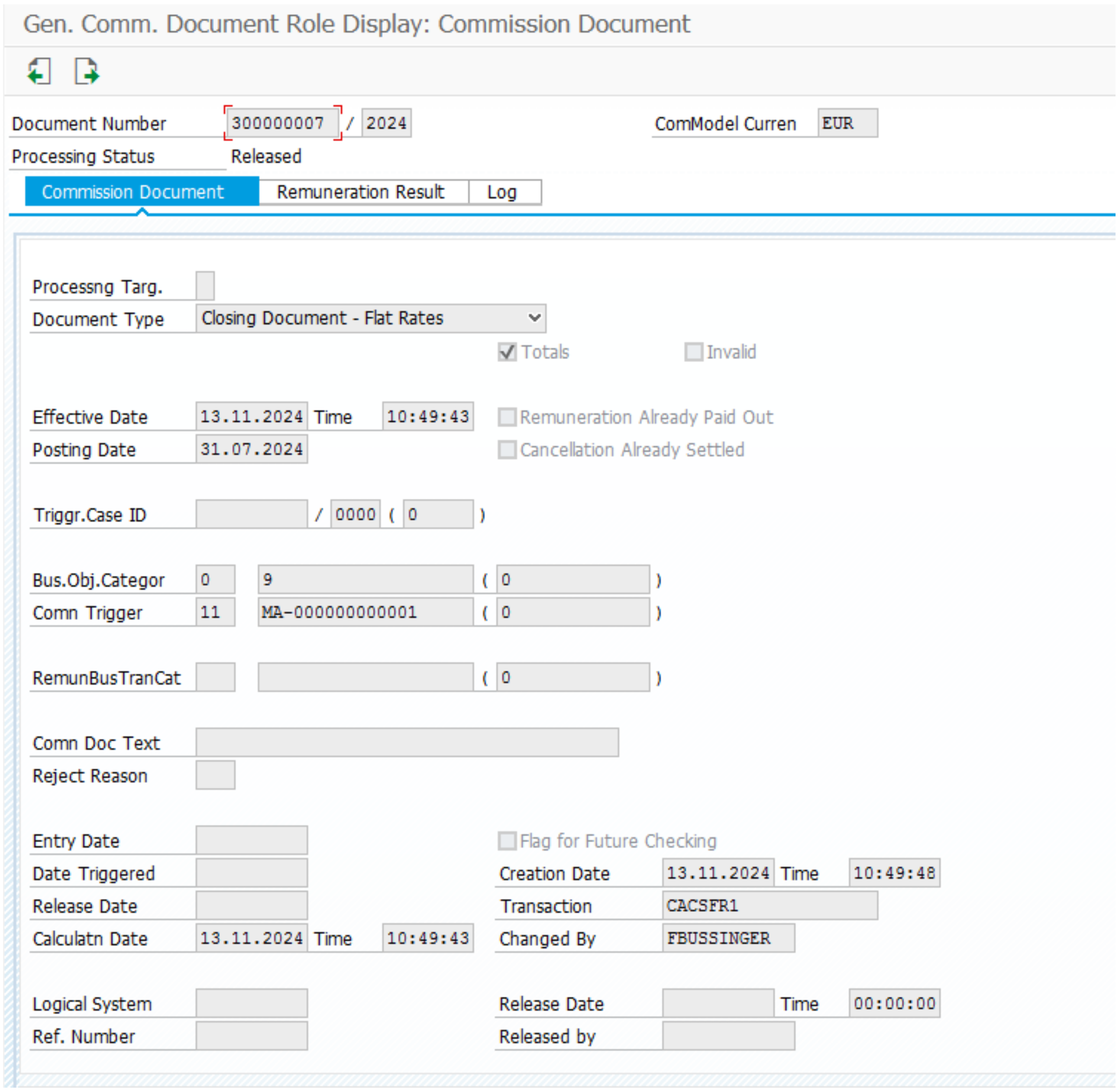
Each remuneration determined during the process leads to a remuneration line, in which the basis for the entitlement and the contractual provisions underlying the entitlement are recorded.

In addition to the actual remuneration amount (entitlement), each remuneration line also contains additional amount fields for liability, offset, and payout. All these amount fields are derived from the results of the individual process steps.

The Flat-Rate Run results in documents with document type 04 (= Flat-Rates).

The document of type 04 contains the following information:

The document header includes general data that applies to the entire document (document-specific data, information about the commission trigger, etc.).



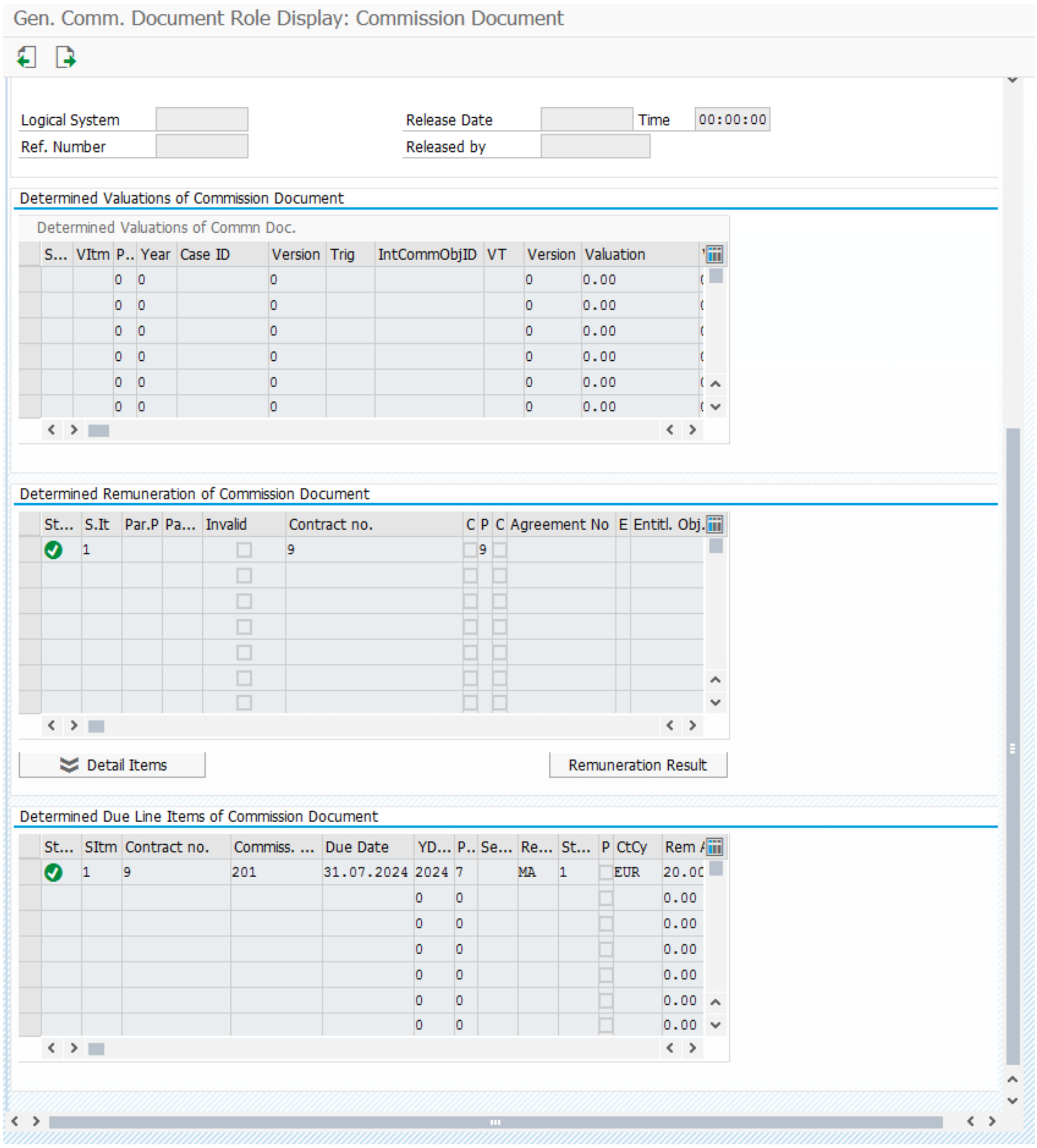
Document Number	300000007 / 2024	ComModel Curren	EUR
Processing Status	Released		

Processng Targ.			
Document Type	Closing Document - Flat Rates	✓ Totals	☐ Invalid
Effective Date	13.11.2024 Time 10:49:43	☐ Remuneration Already Paid Out	
Posting Date	31.07.2024	☐ Cancellation Already Settled	
Triggr.Case ID	/ 0000 (0)		
Bus.Obj.Categor	0 9 (0)		
Comn Trigger	11 MA-000000000001 (0)		
RemunBusTranCat	(0)		
Comn Doc Text			
Reject Reason			
Entry Date		☐ Flag for Future Checking	
Date Triggered		Creation Date	13.11.2024 Time 10:49:48
Release Date		Transaction	CACSFR1
Calculatn Date	13.11.2024 Time 10:49:43	Changed By	FBUSSINGER
Logical System		Release Date	Time 00:00:00
Ref. Number		Released by	

Picture 8: document header

A document line item includes data that applies only to that specific item. The types of document line items are valuations, remunerations and due dates.

For documents of type 04, line items of type valuation are not relevant.



Gen. Comm. Document Role Display: Commission Document

Logical System		Release Date	
Ref. Number		Released by	

Time 00:00:00

Determined Valuations of Commission Document

Determined Valuations of Commn Doc.

S...	VItm	P..	Year	Case ID	Version	Trig	IntCommObjID	VT	Version	Valuation	'
		0	0		0				0	0.00	(
		0	0		0				0	0.00	(
		0	0		0				0	0.00	(
		0	0		0				0	0.00	(
		0	0		0				0	0.00	(
		0	0		0				0	0.00	(

Determined Remuneration of Commission Document

St...	S.It	Par.P	Pa...	Invalid	Contract no.	C	P	C	Agreement No	E	Entitl. Obj.
✓	1			☐	9	☐	9	☐			
				☐		☐		☐			
				☐		☐		☐			
				☐		☐		☐			
				☐		☐		☐			
				☐		☐		☐			
				☐		☐		☐			

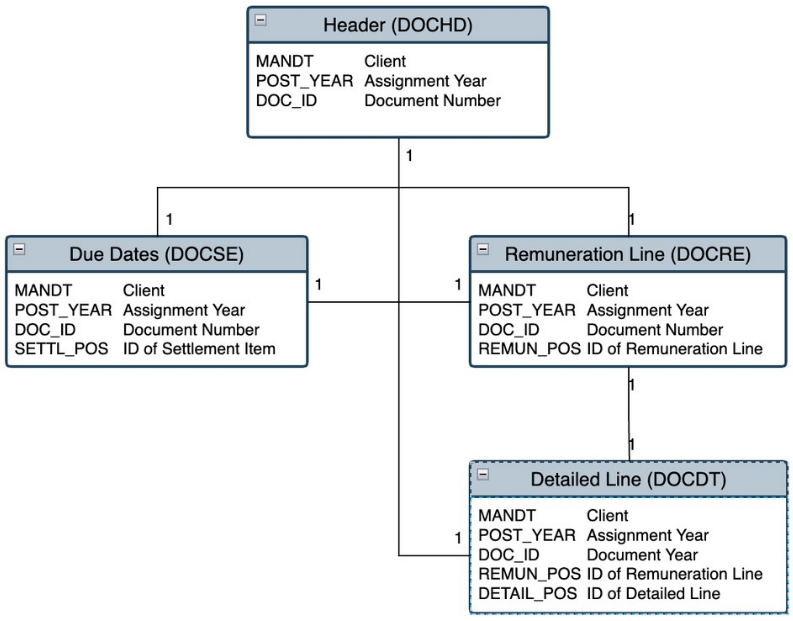
Determined Due Line Items of Commission Document

St...	SItm	Contract no.	Commiss. ...	Due Date	YD...	P..	Se...	Re...	St...	P	CtCy	Rem A
✓	1	9	201	31.07.2024	2024	7		MA	1	☐	EUR	20.0C
					0	0				☐		0.00
					0	0				☐		0.00
					0	0				☐		0.00
					0	0				☐		0.00
					0	0				☐		0.00
					0	0				☐		0.00

Picture 9: document line items

The data model of the commission document includes several tables that are generated as part of the application generation process (e.g. ZZFIN_DOCHD).

For a flat-rate document, however, only the tables shown in the following data model are relevant:



Header (DOCHD)
MANDT Client
POST_YEAR Assignment Year
DOC_ID Document Number
Due Dates (DOCSE)
MANDT Client
POST_YEAR Assignment Year
DOC_ID Document Number
SETTL_POS ID of Settlement Item
Remuneration Line (DOCRE)
MANDT Client
POST_YEAR Assignment Year
DOC_ID Document Number
REMUN_POS ID of Remuneration Line
Detailed Line (DOCDT)
MANDT Client
POST_YEAR Assignment Year
DOC_ID Document Year
REMUN_POS ID of Remuneration Line
DETAIL_POS ID of Detailed Line
1
1
1
1
1
1
1
1

Picture 10: data model of a flat-rate document

3.4 Technical Process of Flat-Rate Run

To ensure the Flat-Rate Run is technically transparent and comprehensible, the following chapters describe each process step in detail.

3.4.1 Prepare Process

The Flat-Rate Run begins with a few preparatory steps before the processes defined by the process control are started.

These preparatory steps are described below in bullet form:

1. Business and technical timestamps are checked. If the fields are empty (no entries were made on the selection screen), they are filled with current timestamps.
2. The CREATE method of the factory class CL_CACS_PRC_FACTORY is called.
    a. Relevant variables (APPL, PRC_TYPE and PRC_METH) are checked.
    b. Using the class CL_CACS_PRD_DB_ITF, the process data (process class, type, and method) for the application ZZFIN is determined.
    c. The process class ZZFIN_CL_PRC_PRD_FR1 is instantiated (go_prc).
        i. In the constructor, it is checked whether the database implementation for the Flat-Rate Run with application ZZFIN has already been carried out (class CL_CACS_DBC_HELPER).
        ii. Then, the application log of the Flat-Rate Run is instantiated (BAL; object CACS; sub-object ZZFIN_FR1).
3. The contract numbers and remuneration types to be processed are transferred from the select-option fields to global tables.

If one of the checks or the instantiation of the process class was not successful, the processing is cancelled, and an appropriate error message is issued.

3.4.2 Fill Data Container

The data container serves as a central storage for all relevant information that is processed and required during the Flat-Rate Run. It consolidates and organizes the various data and makes them available during the individual processing steps.

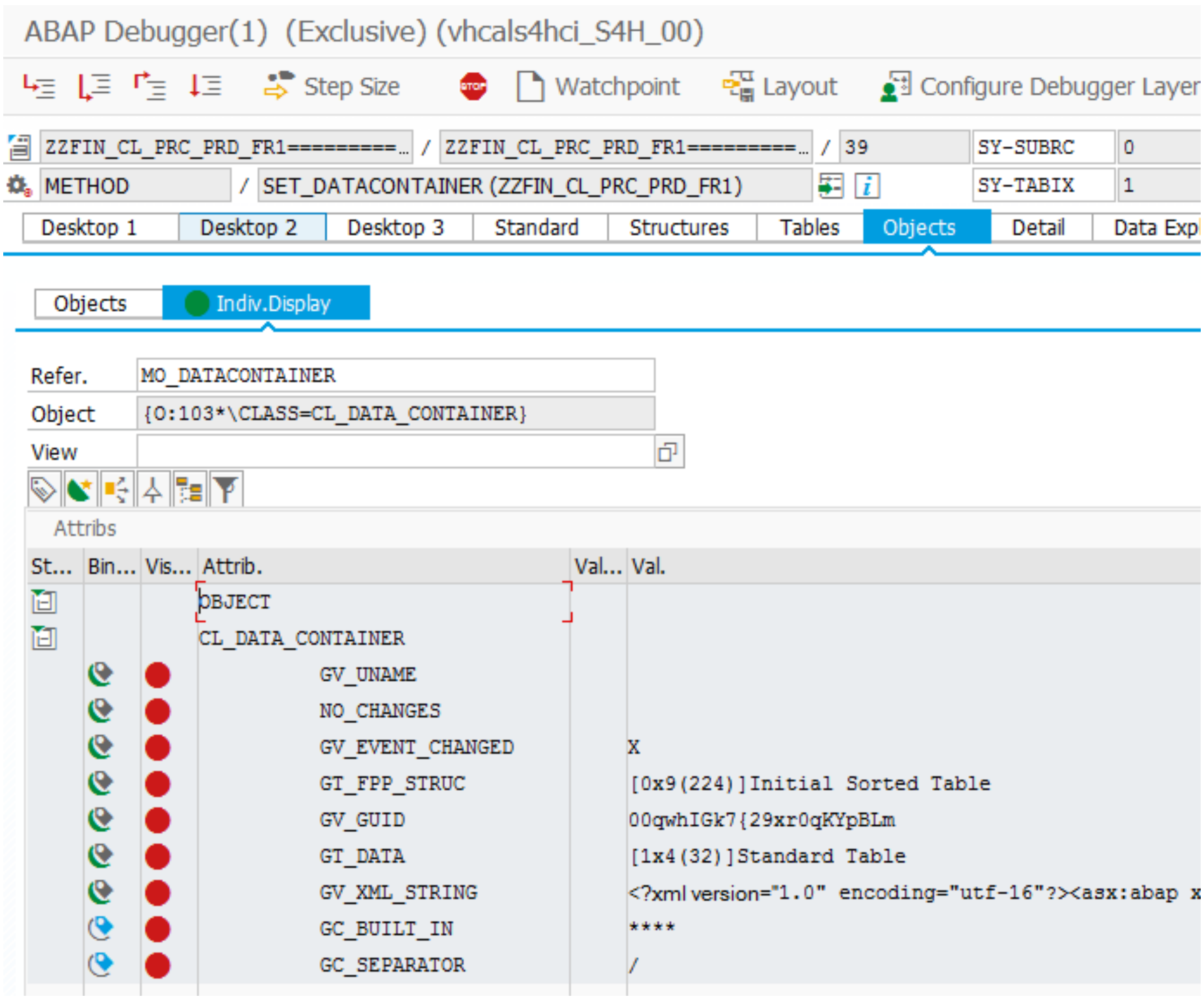
The process step is described below in bullet form:

Call of the method SET_DATACONTAINER by the process instance.

The following parameters are passed to the method: application, range with contracts, range with remuneration types, delimitation date, simulation flag, business and technical timestamp.

1. The import parameters are passed to an internal structure ls_parameters.
2. An instance of the data container class CL_DATA_CONTAINER (mo_datacontainer) is created. If an error occurs, the processing is cancelled, and an error message is issued.
3. The parameters in the ls_parameters structure are passed to the data container.
4. The unique data container ID (GUID) is determined, and the container is saved.
5. The data container instance is unlocked so that it can be reused later.
6. Finally, the import parameters are passed to class attributes of the process class.

The data container instance mo_datacontainer is an attribute of the class CL_CACS_PRC_ABST and is inherited into the process class ZZFIN_CL_PRC_PRD_FR1.



ABAP Debugger(1) (Exclusive) (vhcals4hci_S4H_00)

ZZFIN_CL_PRC_PRD_FR1=========...	ZZFIN_CL_PRC_PRD_FR1=========...	39	SY-SUBRC	0
METHOD	SET_DATACONTAINER (ZZFIN_CL_PRC_PRD_FR1)		SY-TABIX	1

Refer. MO_DATACONTAINER

Object {O:103*\CLASS=CL_DATA_CONTAINER}

View

Attribs

St...	Bin...	Vis...	Attrib.	Val...	Val.
			OBJECT		
			CL_DATA_CONTAINER		
		●	GV_UNAME		
		●	NO_CHANGES		
		●	GV_EVENT_CHANGED		X
		●	GT_FPP_STRUC		[0x9(224)]Initial Sorted Table
		●	GV_GUID		00qwhIGk7{29xr0qKYpBLm
		●	GT_DATA		[1x4(32)]Standard Table
		●	GV_XML_STRING		<?xml version="1.0" encoding="utf-16"?><asx:abap x
		●	GC_BUILT_IN		****
		●	GC_SEPARATOR		/

Picture 11: instance of data container mo_datacontrainer

3.4.3 Initialize Process

Process initialization is the first step of the entire processing procedure. Its main task is to define the necessary prerequisites and parameters for the Flat-Rate Run and to ensure that all relevant data and settings are correctly provided for the subsequent processing.

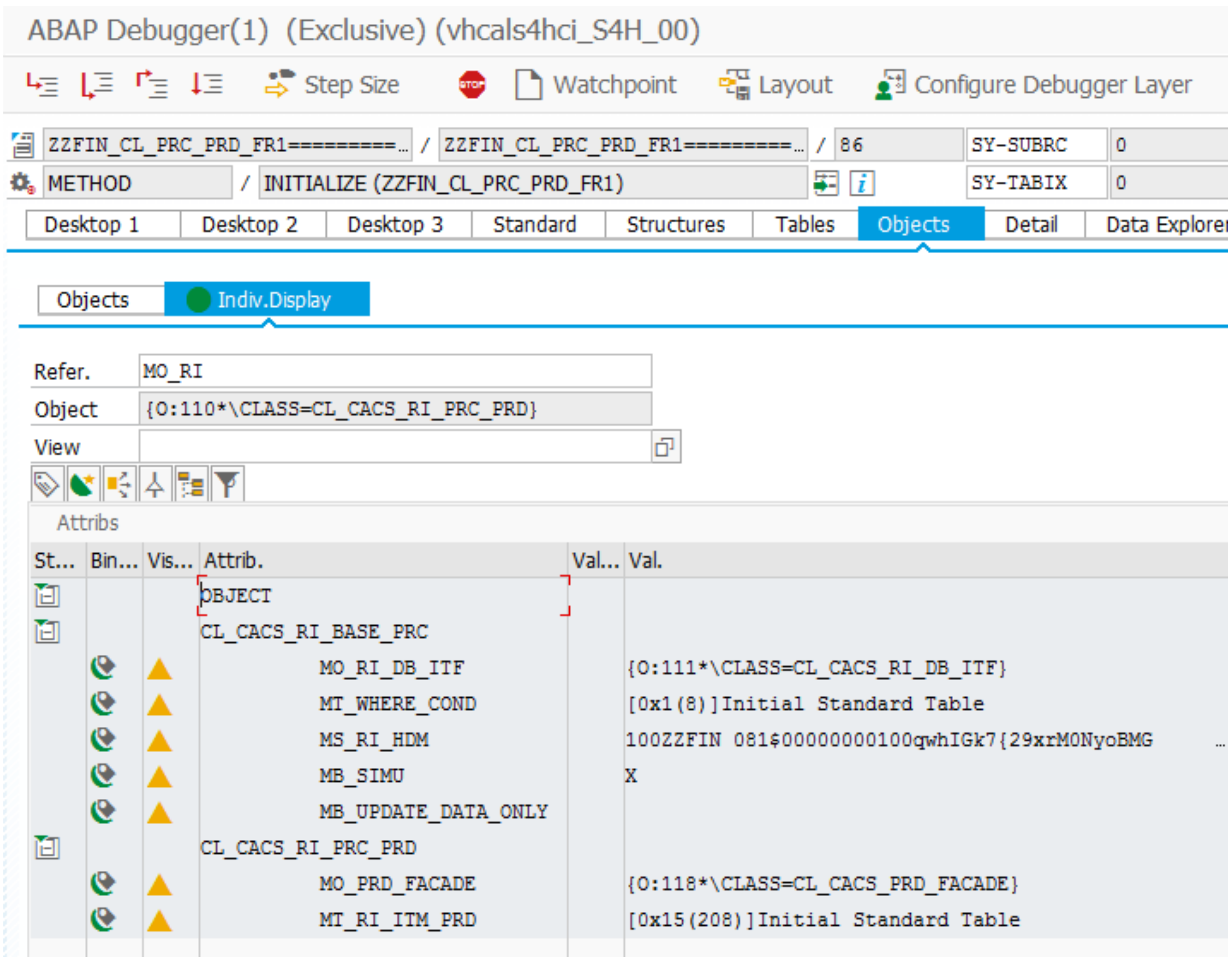
The process step is described below in detail:

The method INITIALIZE is called by the process instance.

1. Authorization check (authorization object E_CACS_MAS; activity 23) is performed.
2. The authorization check can be extended at this point using a BAdI (CACS_PRC_AUTHORITY).
3. If an error occurs during the authorization check, processing is aborted, and an error message is issued.
4. Run administration (mo_ri) is created. If an error occurs during this process, processing is aborted, and an error message is issued.

5. The run ID from run administration is transferred to the application log of the Flat-Rate Run.
6. Run parameters are checked.
7. The run parameter check can be extended at this point using a BAdI (CACS_PRC_PRD_CHECK_RUN).
8. Finally, the application log of the Flat-Rate Run is populated with run information (run ID, delimitation date, business and technical validity dates, simulation or productive run).

The instance of run administration mo_ri is an attribute of the class CL_CACS_PRC_PRD_ABST and is inherited into the process class ZZFIN_CL_PRC_PRD_FR1.


ABAP Debugger(1) (Exclusive) (vhcals4hci_S4H_00)

Step Size | Watchpoint | Layout | Configure Debugger Layer

ZZFIN_CL_PRC_PRD_FR1==========... / ZZFIN_CL_PRC_PRD_FR1==========... / 86 | SY-SUBRC | 0

METHOD / INITIALIZE (ZZFIN_CL_PRC_PRD_FR1) | SY-TABIX | 0

Desktop 1 | Desktop 2 | Desktop 3 | Standard | Structures | Tables | Objects | Detail | Data Explorer

Objects | Indiv.Display

Refer. MO_RI
Object {O:110*\CLASS=CL_CACS_RI_PRC_PRD}
View

Attribs

St...	Bin...	Vis...	Attrib.	Val...	Val.
			OBJECT		
			CL_CACS_RI_BASE_PRC		
			MO_RI_DB_ITF		{O:111*\CLASS=CL_CACS_RI_DB_ITF}
			MT_WHERE_COND		[0x1(8)]Initial Standard Table
			MS_RI_HDM		100ZZFIN 081$00000000100qwhIGk7{29xrM0NyoBMG
			MB_SIMU		X
			MB_UPDATE_DATA_ONLY		
			CL_CACS_RI_PRC_PRD		
			MO_PRD_FACADE		{O:118*\CLASS=CL_CACS_PRD_FACADE}
			MT_RI_ITM_PRD		[0x15(208)]Initial Standard Table

Picture 12: instance of run administration mo_ri

3.4.4 Start Process

The process step "Start Process" is the central processing step in the Flat-Rate Run. In this step, the actual processing of the selected commission contracts and their flat-rate agreements is carried out.

The process step is described below in bullet form:

The method RUN is called by the process instance.

1. Preparation of the run
    a. A list of commission contracts to be processed is created using the class CL_CACS_INDCTR_PRC_LIST (lo_indctr_list) (selection from CACS_CTRTBU).
    b. An iterator (pointer) is created that points to the list of commission contracts (lo_indctr_iterator).
    c. A facade instance for standard commission contracts is created using the class CL_CACS_STD_FACADE_FLAT and the application (mo_stdctr_facade).
    d. The same is done for the periods using the class CL_CACS_PRD_FACADE (mo_prd_facade).
    e. Finally, a database interface is created using the generated class ZZFIN_CL_DOCUMENT_DB_ITF (lo_doc_db_itf).
    f. If an error occurs during one of these preparatory steps, processing is aborted, and an error message is issued.
2. The pointer is set to the first commission contract in the list.
3. A WHILE loop is started, which is not exited until the pointer has gone through all commission contracts.
4. The commission contract to which the pointer refers is identified (lo_indctr). If the commission contract cannot be identified, the pointer is set to the next one in the list and the loop starts again from the beginning.
5. An application log is created for the current pointer (sub-object CACS_TEMP), and a message is written to the log indicating which commission contract is being processed.
6. If the Flat-Rate Run is executed in productive mode, the commission contract is locked. This step is not performed for simulation runs. If an error occurs, the exception is written to the application log (pointer), the log is cleaned up (handle updated), the list and pointer to the commission contract list are cleaned up. The pointer is then set to the next commission contract, and the loop is restarted from the beginning.
7. A list of flat-rate agreements for the commission contract is created (lo_indagr_list) (selection from CACS_REMARU). If an error occurs, the exception is written to the application log (pointer), the log is cleaned up (handle updated), the list and the pointer to the commission contract list are cleaned up. The pointer is then set to the next commission contract, and the loop is restarted from the beginning.
8. A pointer to the list of flat-rate agreements is then created (lo_indagr_iterator). If an error occurs, the exception is written to the application log (pointer), the log is cleaned up (handle updated), the list and the pointer to the commission contract list are cleaned up.
   The pointer is then set to the next commission contract, and the loop is restarted from the beginning.
9. The pointer is then set to the first agreement in the list.
10. Another WHILE loop is started, which is not exited until the pointer has gone through all agreements of the commission contract.

11. The current agreement to which the pointer refers is identified (lo_indagr). If this agreement cannot be identified, the pointer is set to the next agreement, and the WHILE loop (agreements) is restarted from the beginning.
12. The number of the flat-rate remuneration rule (ld_rule_id) and the remuneration type (ld_type_id) are determined.
13. A message is written to the application log (pointer) indicating which agreement and remuneration type have been identified.
14. Determination of periods (mt_period)
    a. The period rule (ld_prdrule_id) and the version of the period rule (ld_prdrule_version) are determined using the facade mo_stdctr_facade.
    b. The start date of the last calculated period is determined using run administration (mo_ri).
    c. If the start date of the previous period could be determined, the corresponding period and the start date of the next period are determined.
    d. If no start date for the next period could be determined (only the case if no flat-rates have been calculated for the commission contract yet), then the start date of the agreement is used as the start date of the next period.
    e. If the start date of the next period is later than the delimitation date, the determination of the periods is exited immediately, and no periods are determined.
    f. All periods from the determined start date up to the delimitation date are determined (mt_period).
    g. If an error occurs during any of the above steps, the determination of the periods is aborted. The exception is written to the application log (pointer), the log is cleaned up (handle updated), the list and the pointer to the agreement list are cleaned up. The pointer is then set to the next agreement, and the WHILE loop (agreements) is restarted from the beginning.
15. LOOP over all determined periods (ld_prd_end_date).
16. The end date of the period is determined. If an error occurs during this process, the exception is written to the application log (pointer), the log is cleaned up (handle updated), the list and the pointer to the agreement list are cleaned up. The pointer is then set to the next agreement, and the WHILE loop (agreements) is restarted from the beginning.
17. A message is written to the application log indicating which period (start to end) is being processed.
18. It is then checked whether the commission contract has already been terminated in the current period. If so, a message is written to the log with the information that the commission contract has ended. The next period in the LOOP is then processed from the beginning.
19. Calculation of the flat-rate remuneration using the end date of the period (class CL_CACS_INDAGR_PRC_FLAT_RATE, method CALCULATE)

a. The calculation date (= end date of the period) is passed into the calculation. First, it is checked whether it is filled.
b. It is then checked whether the calculation date is earlier than the start date of the agreement.
c. Next, it is checked whether the calculation date is later than or equal to the end date of the agreement.
d. Finally, it is checked whether the amount of the agreement is empty or zero.
e. If all of these checks are passed (i.e. none of them applies), the amount of the agreement is transferred (ld_flat_rate_amount).
f. If one of these checks applies, the exception is written to the application log (pointer), and the LOOP continues with the next period.

20. The currency of the amount is determined (ld_indctr_curr).
21. The remuneration type is then determined (ld_type_id).
22. Using the facade mo_stdctr_facade, the attributes of the remuneration agreement are determined (ls_rem). If an error occurs during this process, the exception is written to the application log (pointer), the log is cleaned up (log copied), and processing is aborted, and an error message is issued.
23. Instantiation of a document using the class CL_CACS_DOCUMENT_FACTORY (lo_doc).
    a. The input parameters (application, document type, posting date, business and technical validity dates) are checked for completeness. If any of these attributes is not filled, document creation is aborted, the exception is written to the application log (pointer), the log is cleaned up (log copied), processing is aborted, and an error message is issued.
    b. Depending on the document type, a different method is called to create the document instance. For document type 04 (= Flat-Rates), the method CREATE_DOCUMENT_FLAT_RATE is used.
        i. The template class CACSFR_CL_DOCUMENT_FLAT_RATE for document creation is determined.
        ii. Then, the string CACSFR in the class name is replaced by the application: ZZFIN_CL_DOCUMENT_FLAT_RATE.
        iii. Finally, an instance of the document class is created (lo_doc).
    c. If the instance could not be created, the exception is written to the application log (pointer), the log is cleaned up (log copied), processing is aborted, and an error message is issued.
24. Next, a Flat-Rate document is created via the document instance (lo_doc).
    a. The input parameters (commission contract number, amount, amount currency, agreement start date, and the attributes of the remuneration type) are checked for completeness. If any of these attributes is not filled, document creation is aborted, the exception is written to the application log (pointer), the log is cleaned up (log copied), processing is aborted, and an error message is issued.
    b. Document creation is prepared (period start md_prd_begin_date and period end md_prd_end_date are determined).

c. The document header is created and populated with data (ms_dochdm).
d. The document header can be extended at this point using a BAdI (CACS_FR_ENRICH_DOCUMENT).
e. If the document header could not be created, the exception is written to the application log (pointer), the log is cleaned up (log copied), processing is aborted, and an error message is issued.
f. Next, the remuneration line is created (ls_dorcrem).
    i. Possible data from the header is copied into the docrem structure.
    ii. The document line item is then determined (remun_pos).
    iii. Additional docrem fields are populated (object status and processing status, commission contract number, remuneration type and its version).
    iv. The settlement group is then determined (settle_group).
    v. If any of the above values could not be determined, the exception is written to the application log (pointer), the log is cleaned up (log copied), processing is aborted, and an error message is issued.
g. The result type is determined using the function module CACS_DETERMINE_UPD_ID (ms_upd).
h. Next, the amount fields of the remuneration line are completed.
    i. Depending on the values of the result type, fields are either included or excluded.
    ii. If the claim flag is set, the fields rem_conamnt and rem_ledamnt are filled with the amount. If the flag is not set, these fields and rem_quan are cleared (all fields related to remuneration claim).
    iii. If the offset flag is set, the fields offset_conamnt and offset_ledamnt are filled with the amount. If the flag is not set (which is the case in the Flat-Rate Run), these fields and offset_quan are cleared (all fields related to offsetting).
    iv. If the due flag is set, the fields due_conamnt and due_ledamnt are filled with the amount. If the flag is not set, these fields and due_quan are cleared (all fields related to payout).
    v. If the resp flag is set, the fields resp_conamnt and resp_ledamnt are filled with the amount. If the flag is not set (which is the case in the Flat-Rate Run), these fields and resp_quan are cleared (all fields related to liability).
i. The remuneration line can be extended at this point using a BAdI (CACS_FR_ENRICH_DOCUMENT).
j. The detailed line for the remuneration line is created (ls_docdtm).
    i. Possible data from the remuneration line is copied into the docdtm structure.
    ii. The number of the detailed line is determined (detail_pos).
    iii. If the number of the detailed line cannot be determined, the exception is written to the application log (pointer), the log is

cleaned up (log copied), and processing is aborted, and an error message is issued.

   iv. Additional field contents are added to the docdtm structure (proc_step = identification of a process sub-step, and resulttype = result type of the remuneration).

k. The detailed line can be extended at this point using a BAdI (CACS_FR_ENRICH_DOCUMENT).
l. The settlement item is created (ls_docsem).
   i. Possible data from the remuneration line is copied into the docsem structure.
   ii. The business partner number is added to the docsem structure (gpart).
   iii. The number of the settlement item is determined (settl_pos).
   iv. If the settlement item cannot be determined, the exception is written to the application log (pointer), the log is cleaned up (log copied), and processing is aborted, and an error message is issued.
   v. The due dates are determined (due_date = due date, due_year = due year, and due_month = due period).
m. The settlement item can be extended at this point using a BAdI (CACS_FR_ENRICH_DOCUMENT).
n. The filled structures ls_docrem (remuneration line), ls_docdtm (detail item), and ls_docsem (settlement item) are transferred to the respective attribute tables: mt_docrem, mt_docdtm, mt_docsem.

25. Post document
    a. Using the database interface mo_doc_db_itf, the document is registered for posting. The document header (ms_dochdm) and the attribute tables mt_docrem, mt_docdtm, and mt_docsem are transferred.
    b. It is checked whether the document header itself is filled, whether the key fields of the document header are filled, and whether the document has already been registered.
    c. If any of the above values are missing, the exception is written to the application log (pointer), the log is cleaned up (log copied), processing is aborted, and an error message is issued.
    d. All parameter values are transferred to internal structures and tables.
    e. Finally, the application log for the document is saved (sub-object ZZFIN_DOC).
26. The application log (document) is copied to the application log of the pointer.
27. Now the run information is added to the run administration.
    a. The import parameters are checked (commission contract number, remuneration agreement, remuneration type, remuneration rule, version of the remuneration rule, period number, version of the period).

    b. If any of the above values are not filled, the exception is written to the application log (pointer), the log is cleaned up (log copied), and processing is aborted, and an error message is issued.
    c. It is checked whether a data record for the current run already exists in table mt_ri_itm_prd. If so, the addition of information to run administration is aborted. If no record is found, a new entry is added.
28. Run information is added to the result table of the Flat-Rate Run (mt_result).
29. The current period end and the period information are saved in a variable for the next period (ld_prd_prev_end_date and ls_prd_prev).
30. ENDLOOP over periods: the next period is processed.
31. The list and the pointer to the agreement list are cleaned up.
32. The pointer is set to the next agreement.
33. ENDWHILE agreement: the next agreement is processed.
34. The application log (pointer) is transferred to the application log of the Flat-Rate Run.
35. Using the database interface lo_doc_db_itf, it is checked whether a COMMIT is necessary.
    a. It is checked whether the table mt_docrem is filled and how many entries it contains.
    b. Then, the table mt_dochdm_reversed is checked for number of entries.
    c. If either of the two tables contains one or more entries, the flag commit_necessary is set to X.
36. If the check has shown that a COMMIT must be performed, the method POST_DOCUMENTS is called using the database interface lo_doc_db_itf and the flag mb_update_task to post the documents.
    a. It is first checked whether the document header table mt_dochdm is empty or whether the Flat-Rate Run was executed in simulation mode. If yes, the table contents of the document tables are cleared, and the posting of the documents is aborted.
    b. If the document header is filled and the Flat-Rate Run was executed productively, the function module ZZFIN_DOC_POST2 is called to post the documents.
    c. Afterwards, the contents of the document tables are cleared.
    d. If document creation was not successful, the exception is written to the application log (Flat-Rate Run), processing is aborted, and an error message is issued.
37. The run management is saved.
    a. The header data (key fields, log handle, and time fields) are checked. If the check of the header data is not successful, the exception is written to the application log of the Flat-Rate Run, the application log is saved, and processing is aborted (the final process step "Follow-up Process" is still executed).
    b. If the Flat-Rate Run is executed productively and a database update is not required, then run administration is saved (either in update task or not, depending on configuration).
38. A COMMIT WORK AND WAIT follows.

39. The list and the pointer to the commission contracts list are cleaned up.
40. The commission contract is unlocked.
41. The pointer is set to the next commission contract.
42. ENDWHILE commission contract: the next commission contract is processed.
43. Documents are posted again via the database interface lo_doc_db_itf and method POST_DOCUMENTS (see points under 36).

3.4.5 Finalize Process

The process step FINALIZE is the final step in the Flat-Rate Run and serves to permanently store the results and data of the processing and to cleanly conclude the Flat-Rate Run. This step ensures that all data generated during processing is correctly saved and the process is properly closed.

The process step is described below in bullet form:

The method FINALIZE is called by the process instance.

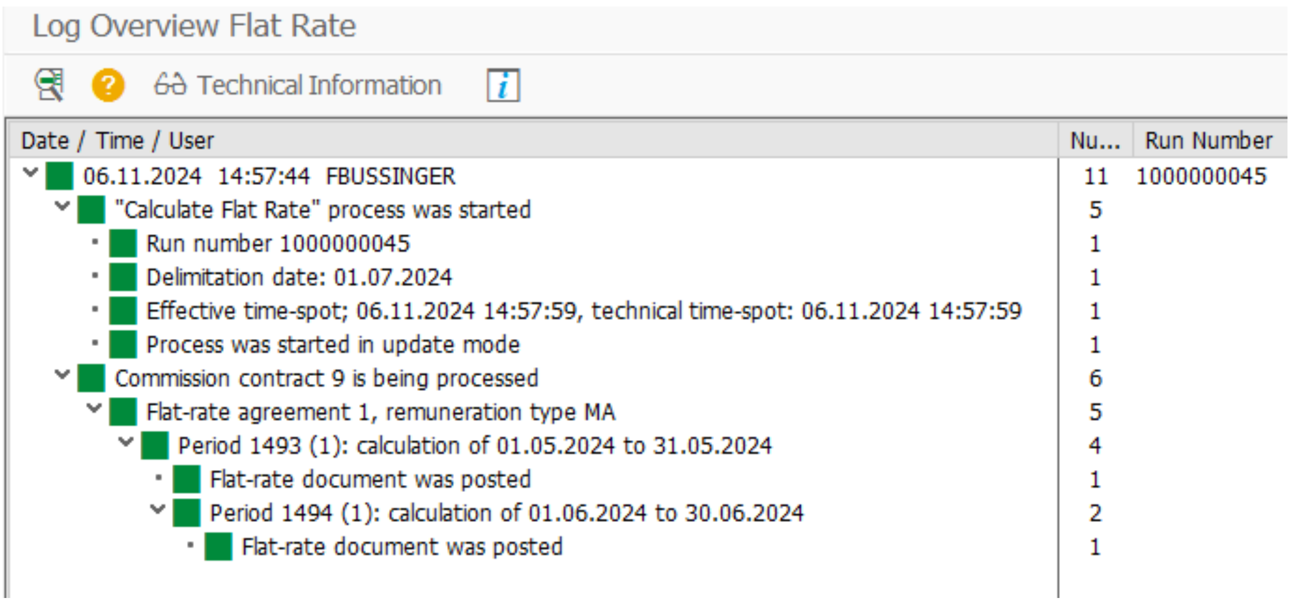
1. The application log of the Flat-Rate Run is finalized (messages are saved).
2. A current timestamp is determined, which is defined as the end time.
3. The run administration is supplemented with the end time.
4. The run administration is saved.
    a. The header data (key fields, log handle, and time fields) are checked. If the header data check is not successful, the exception is written to the application log of the Flat-Rate Run, the application log is saved, and processing is aborted (the last process step “Follow-up” is still executed).
    b. If the Flat-Rate Run is executed productively and a database update is not required, then run administration is saved (in update task or not, depending on configuration).
5. A COMMIT WORK follows.

3.4.6 Follow-up

After the final process step FINALIZE, a few remaining tasks are performed.

These are described below in bullet form:

1. The run number and the log handle of the process are determined.
2. A COMMIT WORK follows.
3. A message indicating that the process “Calculate Flat-Rate” is completed is written to the application log of the Flat-Rate Run.
4. The result table is determined (gt_result).
5. Finally, the result of the Flat-Rate Run is displayed using the function module CACS_PR_UI_2000_SHOW_RESULTS.


Log Overview Flat Rate

```
Date / Time / User                                                              Nu...  Run Number
06.11.2024  14:57:44  FBUSSINGER                                                 11    1000000045
  "Calculate Flat Rate" process was started                                       5
    Run number 1000000045                                                         1
    Delimitation date: 01.07.2024                                                 1
    Effective time-spot; 06.11.2024 14:57:59, technical time-spot: 06.11.2024 14:57:59   1
    Process was started in update mode                                            1
  Commission contract 9 is being processed                                        6
    Flat-rate agreement 1, remuneration type MA                                   5
      Period 1493 (1): calculation of 01.05.2024 to 31.05.2024                    4
        Flat-rate document was posted                                             1
        Period 1494 (1): calculation of 01.06.2024 to 30.06.2024                  2
          Flat-rate document was posted                                           1
```

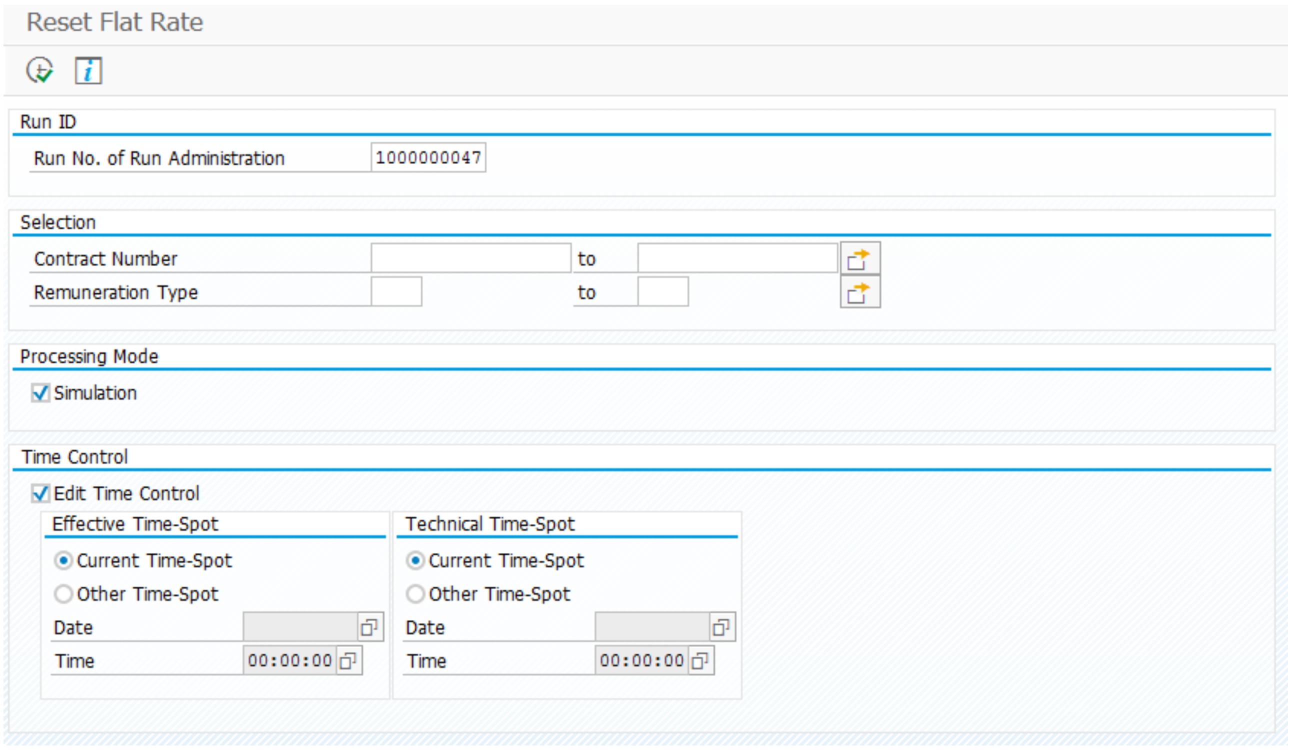
Picture 13: Log of Flat-Rate Run for commission contract 9 (periods 5 and 6)

4 Reversal of Flat-Rate Remunerations

Flat-rate remunerations generated through the Flat-Rate Run can be reversed. However, only the most recent run can be reversed, and only if the resulting documents from the last run have not yet been settled.

The application for reversing flat-rate remunerations is launched using transaction CACSFR2 (Package: CACSF1; Report: CACS_PRC_FLAT_RATE_REV; Generated class: ZZFIN_CL_RPC_PRD_FR2 for application ZZFIN).

4.1 Selection Screen


Reset Flat Rate

Run ID

Run No. of Run Administration: 1000000047

Selection

Contract Number		to	
Remuneration Type		to	

Processing Mode

[✓] Simulation

Time Control

[✓] Edit Time Control

Effective Time-Spot	Technical Time-Spot
● Current Time-Spot	● Current Time-Spot
○ Other Time-Spot	○ Other Time-Spot
Date	Date
Time 00:00:00	Time 00:00:00

Picture 14: selection screen of reset flat-rate

Run Number of Run Administration

This is the run number of the Flat-Rate Run to be reversed. This field is mandatory.

Contract Number

To restrict the reversal to a specific commission contract of the run, the contract number must be entered.

If no contract number is specified, all commission contracts that were processed in the selected run will be considered for reversal.

Remuneration Type

To restrict the reversal to specific flat-rate remuneration types from the run, a remuneration type must be entered.

If no remuneration type is specified, all remuneration types from the run will be reversed.

Simulation Mode

If this checkbox is selected, the reversal is simulated only. No postings are performed.

Time Control

Time control determines which Customizing data is referenced during the process.

4.2 Processing

The essential processing phases of the reversal of flat-rate remunerations are described below. (Only the process step RUN is considered here.)

Check Run Number

At the beginning, it is checked whether the specified run can be reversed (i.e. whether it is the most recent run and has not already been reversed).

Determine Documents

After a successful check, all documents that have not yet been reversed are retrieved for the run. It is then checked whether the document has already been settled.

Check Contract Number & Remuneration Type

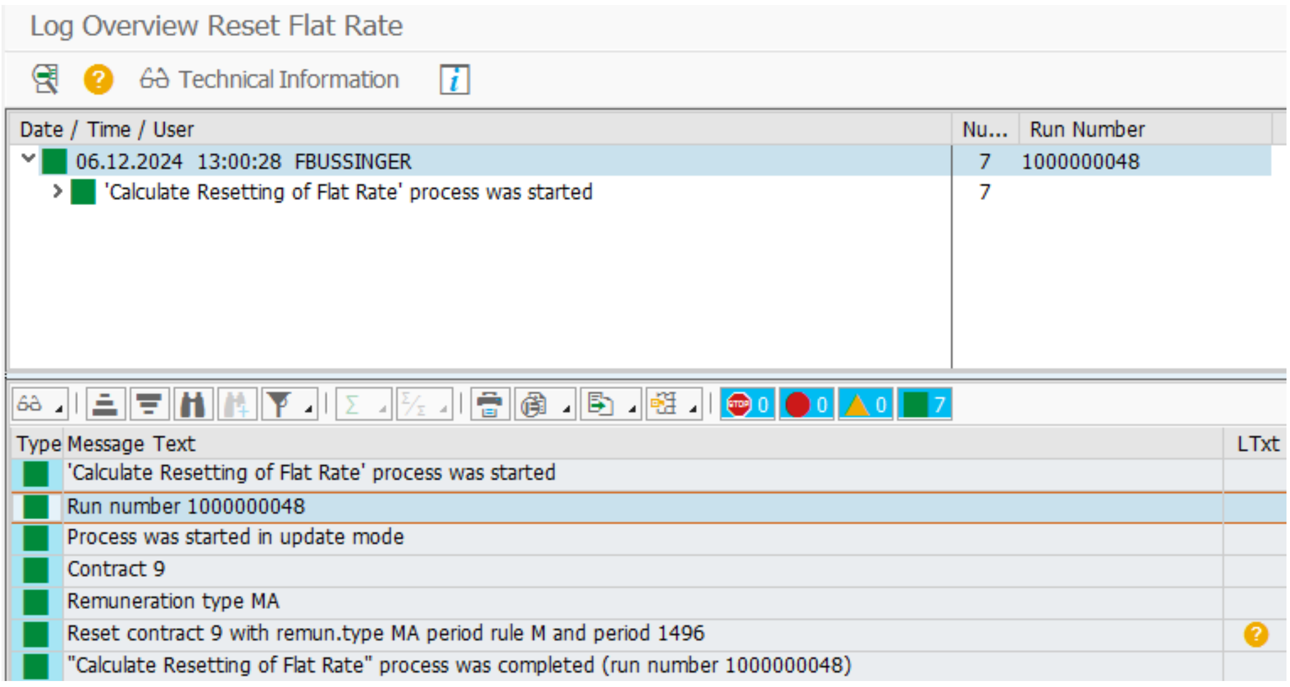
If a contract number and/or remuneration type has been entered on the selection screen, any documents that do not match these specifications are skipped during the reversal.

Reverse Document

A reversal document is created, which references the original document being reversed. The original document is marked as invalid and is linked to the newly created reversal document via the cancellation reference.

Output of Results & Log

As the final step, the result and the processing log are displayed.



Date / Time / User	Nu...	Run Number
🟩 06.12.2024 13:00:28 FBUSSINGER	7	1000000048
🟩 'Calculate Resetting of Flat Rate' process was started	7	

Type	Message Text	LTxt
🟩	'Calculate Resetting of Flat Rate' process was started	
🟩	Run number 1000000048	
🟩	Process was started in update mode	
🟩	Contract 9	
🟩	Remuneration type MA	
🟩	Reset contract 9 with remun.type MA period rule M and period 1496	?
🟩	"Calculate Resetting of Flat Rate" process was completed (run number 1000000048)	

Picture 15: log of reset flat-rate

The processing is described in bullet form (only for process step RUN):

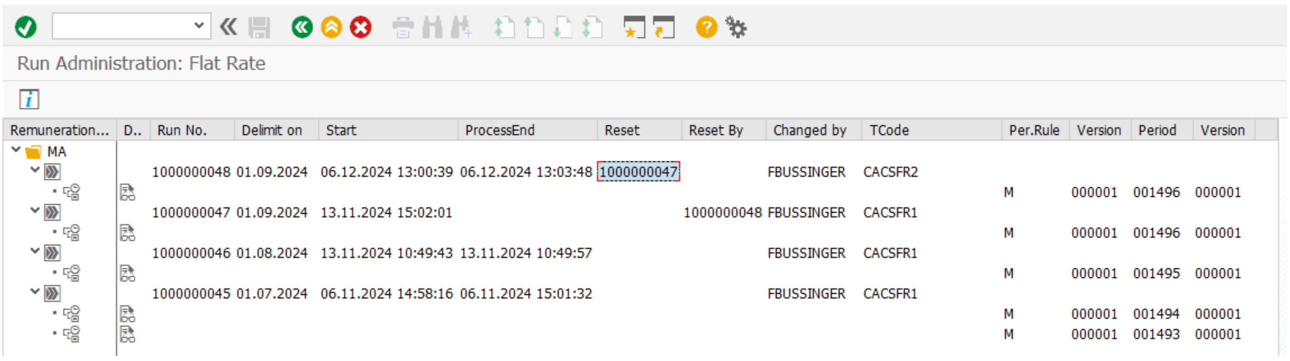
1. Check whether there is a more recent run, or whether the specified run is not the latest.
2. Check whether the specified run has already been reversed.
3. Create a list of documents belonging to the specified run.
4. Create a pointer to this list of documents.
5. Set the pointer to the first document.
6. Start a WHILE loop that continues until all documents have been processed.
7. Determine the current document instance.
8. Retrieve the run information from the document header.
    a. Determine remuneration rule and remuneration type.
    b. Determine period rule and version of the period rule.
    c. Determine period and version of the period.
9. Check the contract number and remuneration type from the selection screen and, if they do not match, continue with the next document.
10. Determine the contract number and remuneration type of the current document.
11. Create application log (object CACS; sub-object ZZFIN_FR2).
12. Write contract number and remuneration type of the current document to the application log.
13. Check whether the document has already been settled.
    a. Determine document ID and validity year.
    b. Read due items from DOCSE.
    c. If the fields SETTLE_DATE, SETTLE_POST_YEAR, and SETTLE_DOC_ID are empty, the document has not been settled. Otherwise, processing is stopped.
14. Reverse the document.
    a. Fill the document header for the reversal document and register the document for reversal.
15. Collect and transfer run information for the reversal to the run Administration.
16. Save result in the result table.
17. Check whether a commit is necessary. If yes, the reversal document is created in the update task.
18. Set pointer to the next document.
19. Continue the WHILE loop with the next document.
20. Save current run administration.
21. Save run administration for the reversal.
22. Create reversal document in the update task (if not already done before).

4.3 Technical Components

The reversal of flat-rate remuneration also uses both process control and run administration.

4.3.1 Run Administration

In the run administration view, accessible via the commission contract, the reversal is visible (a new run number 10…48 appears with information indicating which run was reversed). Likewise, the reversed run is updated with the run number of the reversal.

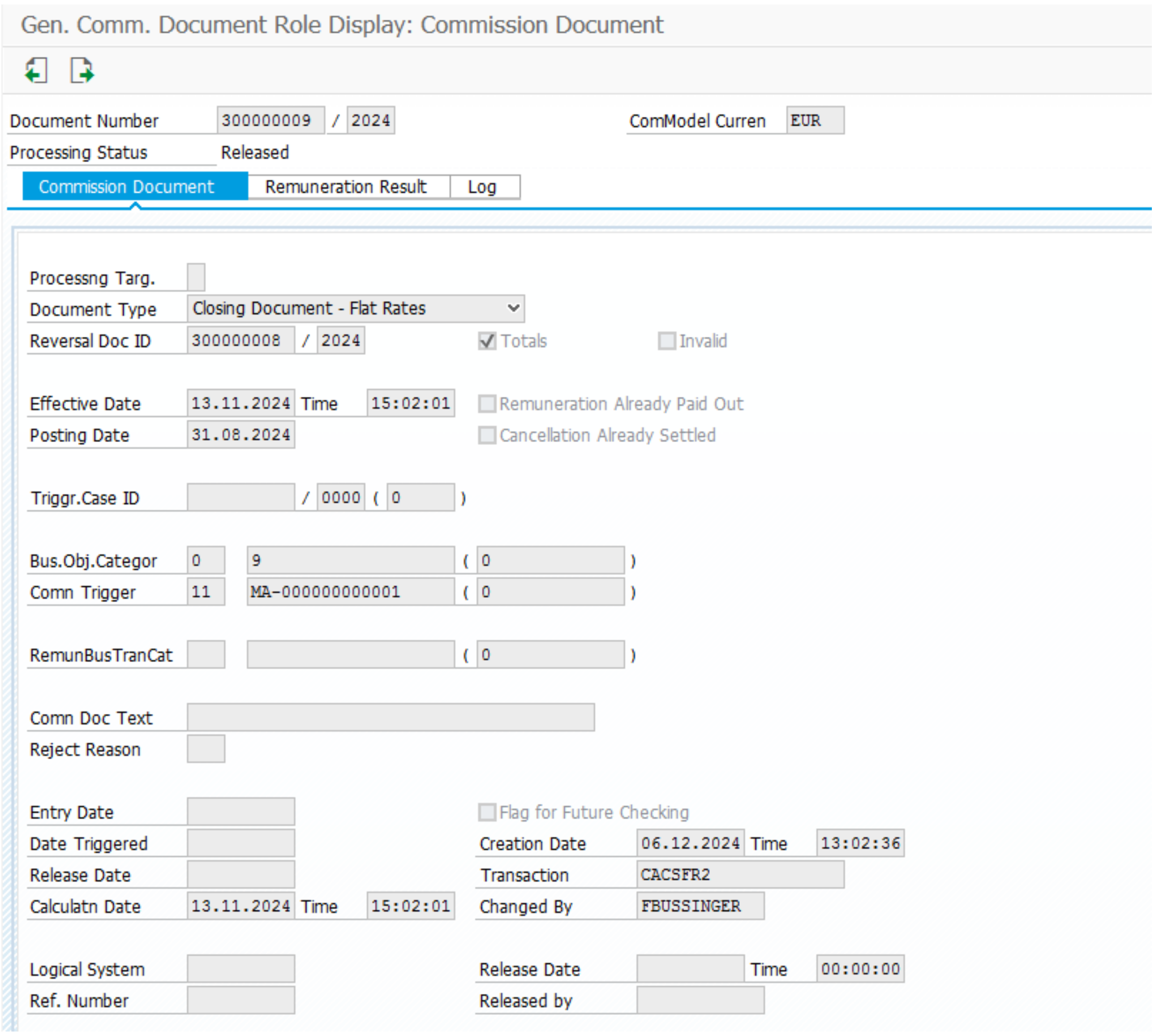

Run Administration: Flat Rate

Remuneration...	D..	Run No.	Delimit on	Start	ProcessEnd	Reset	Reset By	Changed by	TCode	Per.Rule	Version	Period	Version
MA													
		1000000048	01.09.2024	06.12.2024 13:00:39	06.12.2024 13:03:48	1000000047		FBUSSINGER	CACSFR2				
										M	000001	001496	000001
		1000000047	01.09.2024	13.11.2024 15:02:01			1000000048	FBUSSINGER	CACSFR1				
										M	000001	001496	000001
		1000000046	01.08.2024	13.11.2024 10:49:43	13.11.2024 10:49:57			FBUSSINGER	CACSFR1				
										M	000001	001495	000001
		1000000045	01.07.2024	06.11.2024 14:58:16	06.11.2024 15:01:32			FBUSSINGER	CACSFR1				
										M	000001	001494	000001
										M	000001	001493	000001

Picture 16: run administration after reversal

4.3.2 Commission Document

As mentioned above, a new document was created – a reversal document. This document refers to the document that was reversed using the Reversal Document ID field.


Gen. Comm. Document Role Display: Commission Document

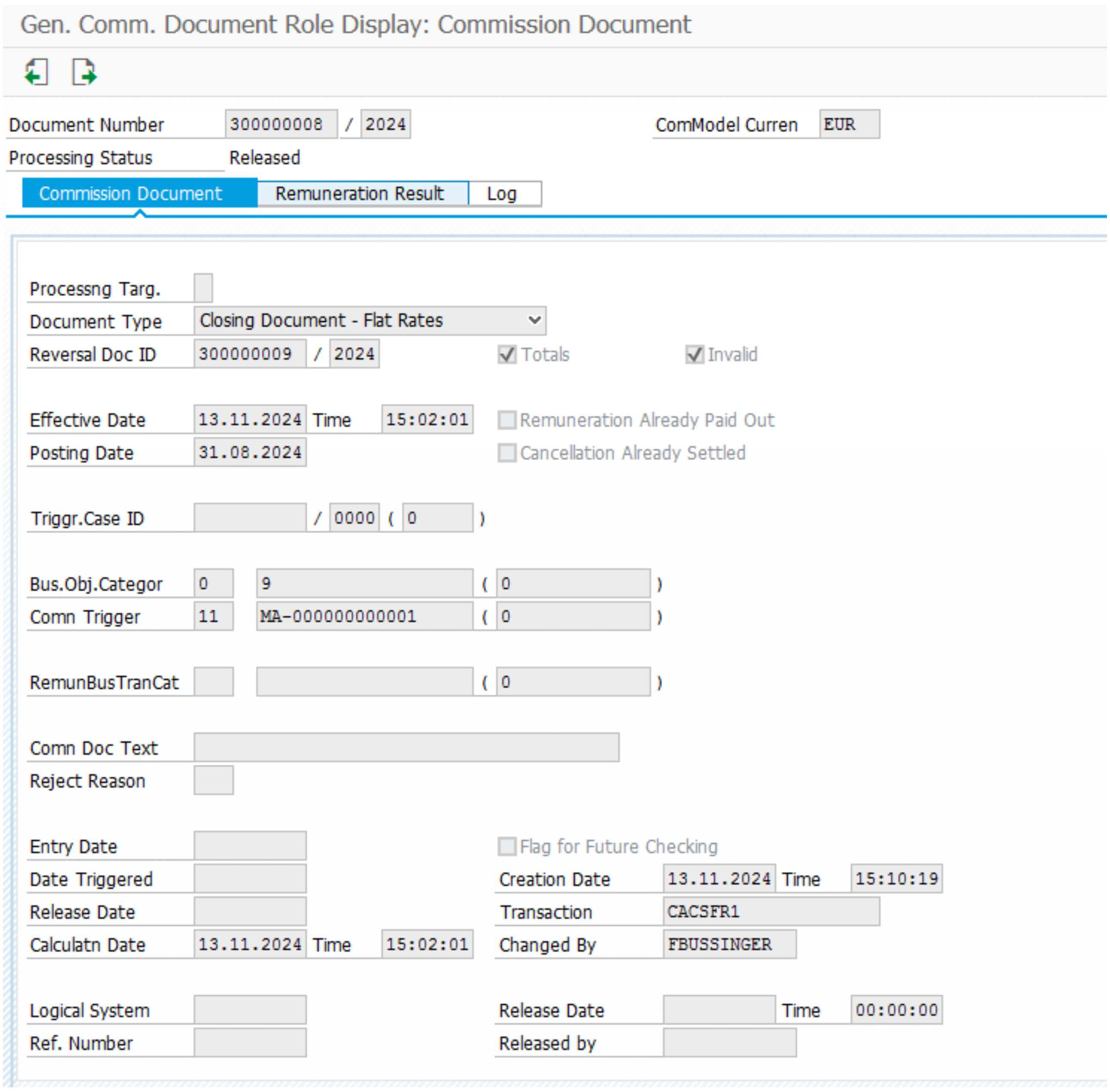
Document Number	300000009 / 2024	ComModel Curren	EUR
Processing Status	Released		

Commission Document | Remuneration Result | Log

Processng Targ.			
Document Type	Closing Document - Flat Rates		
Reversal Doc ID	300000008 / 2024	✓ Totals	☐ Invalid
Effective Date	13.11.2024 Time 15:02:01	☐ Remuneration Already Paid Out	
Posting Date	31.08.2024	☐ Cancellation Already Settled	
Triggr.Case ID	/ 0000 (0)		
Bus.Obj.Categor	0 9 (0)		
Comn Trigger	11 MA-000000000001 (0)		
RemunBusTranCat	(0)		
Comn Doc Text			
Reject Reason			
Entry Date		☐ Flag for Future Checking	
Date Triggered		Creation Date	06.12.2024 Time 13:02:36
Release Date		Transaction	CACSFR2
Calculatn Date	13.11.2024 Time 15:02:01	Changed By	FBUSSINGER
Logical System		Release Date	Time 00:00:00
Ref. Number		Released by	

Picture 17: reversed document

The reversed document is marked as invalid (checkbox: invalid). This document also refers to the reversal document via the same Reversal Document ID.



Gen. Comm. Document Role Display: Commission Document

Document Number	300000008 / 2024	ComModel Curren	EUR
Processing Status	Released		

Commission Document | Remuneration Result | Log

Processng Targ.			
Document Type	Closing Document - Flat Rates		
Reversal Doc ID	300000009 / 2024	✓ Totals	✓ Invalid
Effective Date	13.11.2024 Time 15:02:01	Remuneration Already Paid Out	
Posting Date	31.08.2024	Cancellation Already Settled	
Triggr.Case ID	/ 0000 (0)		
Bus.Obj.Categor	0 9 (0)		
Comn Trigger	11 MA-000000000001 (0)		
RemunBusTranCat	(0)		
Comn Doc Text			
Reject Reason			
Entry Date		Flag for Future Checking	
Date Triggered		Creation Date	13.11.2024 Time 15:10:19
Release Date		Transaction	CACSFR1
Calculatn Date	13.11.2024 Time 15:02:01	Changed By	FBUSSINGER
Logical System		Release Date	Time 00:00:00
Ref. Number		Released by	

Picture 18: invalid document after reversal

This article was written by Franzi Bussinger
We look forward to your feedback at: wissen@vanatix.com
Learn more about us at www.vanatix.com
Vanatix AG · Hemmelrather Weg 201 · 51377 Leverkusen · Germany