FINASTRA

GPP PAYplus

Posting

Business Guide

Version Control

Version	Date	Summary of Changes
1.0		Document Created
2.0		• Changed population and usage logic around Reversed_Orig_Posting_Entry_ID • Added additional support for Earmark reference • Provided a more specific description of the Retry buttons behavior • Added additional instructions for the setup of the Interfaces profile • Addition of the separate MINF field holding the different Posting monitors
3.0	Dec 2014	Updated: • Supporting Account Model tables • Fee types • Fee Formula
4.0	Dec 2015	Updated for rebranding
5.0	Mar 2016	Updated: • Usage of P_SUSPENS_ACCT_ID in the logic of BSA posting types on the Individuals and on reversed entries for BSA posting types. • Currently implemented values of **Posting invocation context** (**D_POSTING_INVOCATION_CONTEXT**) • Removal of an incorrect action in Stop Posting queue
6.0	Jun 2016	Updated: • The additional specific functionality for the exit point of BEFORE_STOP_NON_FINAL_QUEUE – no waiting for response, and suggested configuration to handle failures in 1st step • Additional explanation regarding the option to receive the posting status of a specific entry in the Posting Response and setting it accordingly.
7.0	Jul 2016	Updated refinement of the instruction to update Posting entry value date from the response for successful entries
8.0	Apr 2017	Updated with missing System Parameters for NSF automatic retry handling
9.0	May 2017	Updated with missing action from Posting Exception queue

Table of Contents

1 Introduction

Note: This Business Guide has not yet been certified for GPP V4.6; therefore, there may be inaccuracies in this document that may require amendments in the future. For more information, please contact your Finastra Project Manager.

This document only covers the functional level of the posting logic.

Global PAYplus (GPP) performs payment processing including steps that identify the debit and credit parties, payment fees and, when required, FX conversions. During the payment processing flow, GPP interacts with other bank systems (to capture the payment, to perform an account lookup if an account is not in the GPP repository, and to send posting to the bank's ledger books). This document describes the interaction with bank ledger books during a processing step called Posting.

GPP Posting logic supports various Accounting Model options, and a way to pre-configure the behavior as per specific requirements and variations for each customer bank:

- When in the different payment processing flows (at which step) will posting entries be created and which of these will be sent,

- Which type of entries will be created and which of these will be sent in which step.

Note: See the Examples of Supported Accounting Models section for the supported Accounting Models.

The Accounting logic includes the creation of posting entries within GPP (**Msg_Posting** entries) and an invocation of an interface that includes some payment details and the posting entries' details. The interface updates the bank ledger books with relevant data. The most common payment information is:

- Debit account, amount and currency, and value date for the principal amount.

- Credit account, amount and currency, and value date for the principal amount.

- If fees apply: Fee amount and currency, the account from which fees are deducted and the P&L account to be credited. Fee entries are divided to quote fees on the credit side and fees on the debit side.

Note: In the posting logic the distinction between fees on the credit side and fees on the debit side (and their equivalent PNL entries) is based on the side of the customer that is **actually** charged and not, as in the fees logic, the customer on whom this fee is defined (rules attached).

Therefore there are cases in which a fee is defined on the debit customer, but the charged customer is actually the credit customer. Such fees are marked in the posting entries as fees on the credit side.

- If the transaction is a cross-book transaction – i.e. funds are transferred from one accounting system to another and cross-book balancing postings are required - the posting information for the cross-book balancing entries (including balancing suspense accounts used – credit and debit – and for each the amount, currency and value date)

- If the transaction involves FX conversion, and FX balancing postings are required - the posting information for cross-currency balancing entries (including balancing suspense accounts used – credit and debit – and for each, the amount , currency and value date)

In regards to Cross-Book and FX balancing entries:

Models that require Cross-Book entries can only be implemented if one of the following is true: all accounts are managed in GPP and the Booking Entity (accounting system) is set for each account entry, or an Account Lookup is used for accounts not managed in GPP and it returns the Booking Entity attribute.

In some banks the FX balancing accounts are only in one system, while in other banks, the cross-book balancing accounts are only in the base currency. This means that the decision on which currency (before or after the conversion) to do the Cross-Book balancing entries is dependent on bank specific conditions. To support such a deviation, the responsibility to properly select the suspense accounts to be used for the balancing entries is on the bank (through the ***Posting Suspense Account Selection*** rules). GPP will then perform the balancing order based on the attributes (currency, Booking entity) of these accounts.

The creation of posting entries for cross-book balancing and FX balancing is conditioned and depends on the configuration that may or may not be done. The principal posting entries and the fees entries (when fees are involved) are always created, but are also dependent on configuration as per the point in the lifetime of a message that they are created and sent.

In addition to the above, some more complex and specific requirements, again achieved via specific, available, configuration, may require the separation of posting entries into more than one step, the inclusion of separate entries for Gross principal (before fees were deducted from it) and the Fee, even if the actual payment amount was the Net from both.

Usually most of the transaction posting information is submitted to the bank accounting/booking system in real-time. There is an option to configure GPP so that some of the entries are not sent in any real-time posting request. These may be gathered and reported later (for example in the EOD report).

The interface response may be positive (i.e. posting had been done successfully) or negative (failed to perform the posting action). When a negative response is received, an error code and description are added so that GPP can take action.

The most common errors are:

- Insufficient funds in the debit account. Payments are routed to the ***Insufficient Funds*** (**NSF**) queue.

- Posting Restriction/Limitation is set on the account that should prevent using it for payments (can be set for either Debit, Credit, or both actions). The payment is routed to a manual queue (depending on the specific response and setup) – in most cases – the ***Posting Restrictions*** (**POSTREST**) queue.

- Generic error. Payments are routed to the ***Posting Exception*** (**POSTEX**) queue or other appropriate queues, as per specific requirements.

Reversed entries are supported also via configuration on specific steps within specific flows (for example processing after a Cancelation message is received and matched with an original that was already posted).

1.1 Handling Different Accounting Models and Required Behaviors

Different accounting/booking systems have different capabilities and different requirements as per the behavior of the Posting interface that is used to report the accounting postings.

GPP can support these multiple variations as per the following aspects.

Different Accounting Models

In the common and simple case, all posting entries, per payment message, are posted in a single Posting request. But GPP supports the division of posting entries into different requests in the following cases (based on pre-configured system setup).

- If it is required to send different posting entries (types - e.g. Principal amount and Fees) in different steps of payment processing, GPP can be configured to support the invocation of the Posting step at multiple points in the payment processing flow, or different flows during the

lifetime of a message. The following are some examples of the different scenarios and capabilities:

- – Different invocation points for the Principal amount and for the Fees on single/few flows through which the message in processed
- Different invocation points per debit leg and credit leg (a 2-step-accounting scenario involving a suspense account) and in each a separate set of posting entries/types
  - – An additional invocation of the Posting logic for additional fees in a specific scenario (for example a Reject is received on an original message on which regular processing fees were already computed and posted, and additional fees due to the Reject are now due)
  - – An invocation of the Posting logic on an Incoming Cover transaction message when it is matched to its Direct, but the previous posting entries generated on the Direct need to be adjusted[1], to correct the previous posting entries

Note: The available points in the flow in which this step is invoked are not configurable and are part of the flow definition, but the use of each of these invocations (whether to create the posting selected for this point or not) is configurable.

- In all of the above cases, multiple Posting requests may be sent for a single transaction, in each of the steps, as per the different posting types, if configured to be separated into different Requests.

Another variation between the different Accounting Models is in the posting entries that are required to be posted. For example: not all accounting/booking systems require GPP to calculate balancing entries (cross-book, FX etc.). Some accounting/booking systems require consolidated fee entries (when multiple fees are selected) and some require that these be received as separate entries.

See section Examples of Supported Accounting Models for the different supported Accounting Models.

Different/Specific Structure of Request/Response

Posting entries calculation logic is de-coupled from the logic of formatting of the posting information into the posting request structure. Therefore, GPP generic posting logic can support the usage of a bank specific structure and a GPP generic one.

Where there are no requirements for a specific structure, the GPP standard option is to use the Fndt Message as the structure for the Posting Request/Response, including its flexibility to exclude sub-sections from its full structure for this usage:

- From minimum – include only the posting entries details
- To maximum – include the full message details

If multiple accounting systems exist in a specific site and a different set of information items is required per the different accounting/booking systems and/or a specific accounting/booking system cannot accept any other structure than its dedicated one, the specific structure/s can be used, and different Interface profiles can be set to produce each of these different structures.

The above example is a very extreme case. In most cases, the structure can be the same, giving each accounting/booking system the option to use only the subset of information it needs from it.

Different Response Policies

---

[1] The Cover was not received, as anticipated in the Direct, either from the bank/clearing it was anticipated, or on the date it was anticipated.

Most accounting/booking systems are expected to return a response with Ack or Nak statuses as per the postings that were reported to them by the Posting request. Some accounting/booking systems are not expected to return such a response, and the postings should be considered successful upon sending them to the Host system. GPP can be pre-configured to support such a case. The interface, in this case, sets the status for the appropriate posting entries to the code that indicates posted successfully when sending the Posting request to the Host.

In addition, when a response is expected, GPP can be pre-configured to support three response methods:

- Supported:
  - One response that specifies one general Ack or a Nak for all original posting occurrences in the request. And when a Nak – including one reason code and text.
- Additional response levels that may be required:
  - A separate response per each posting occurrence, that specifies separately an Ack or a Nak for each, and when a Nak – including its specific reason code and text.
  - One response that includes separate response status information occurrence for each original posting occurrence in the request. Each occurrence specifies separately an Ack or a Nak for each, and when a Nak – including its specific reason code and text.

Specific Customer Requirements Regarding Posting

Some accounting/booking systems have specific requirements in regards to Posting, such as account replacement based on specific conditions, additional non-standard attributes on a request level or on a posting entry level.

The standard logic allows branching into a customer specific logic to support such cases, without requiring a change in the whole standard logic. An additional DB table also allows adding the additional attributes required for the specific customer bank (see section Message Attributes). The customer specific logic can be used to populate this table, and the interface handler logic can include gathering of such entries into the Posting Requests (either bank specific or the standard Fndt Message).

1.2 Examples of Supported Accounting Models

The following are some examples of supported Accounting Models that can be achieved using the configurable GPP Posting logic.

The most simple and most often used Accounting Model:

- A 1-step-accounting performed on completion of the transaction message (for the Outgoing transaction, either before sending it out, or only after receipt of an acknowledgment)

Note: A variation, per place in the flow (for Outgoing), but still as a simple one step model, would be to perform the posting step just before the Liquidity check steps. This would be used in bank sites in which there is no balance Inquiry with earmarking functionality, in which the *Liquidity Risk Management module* (**LRM**) is installed.

- Requires only the main entries for simple[2] principal credit and debit, and entries for the fees on both sides (credit and debit) with their corresponding PNL entries.

---

[2] Posting entry quotes the payment amount, even in a case it actually is not just the principal amount and a fee is actually deducted/added to it

The following example shows Posting entries as per this model, for a customer to customer payment (both customers in same current bank), a payment amount of 100 and two fees of 10 – one on the credit side customer and another on the debit side customer.

Posting Action	Amount	Posting Type
Dr debit side cust	100	Principle debit amount (**MAIN_DR**)
Cr credit side cust	100	Principle credit amount (**MAIN_CR**)
Dr debit side cust	10	Fees on debit side customer (**FEE_DR**)
Cr PNL	10	Amount in P&L account for fees on debit side customer (**PNL_DR**)
Dr credit side cust	10	Fees on credit side customer (**FEE_CR**)
Cr PNL	10	Amount in P&L account for fees on credit side customer (**PNL_CR**)

Note: Of course, if there are no fees selected for either side (or both), the relevant posting entries will not be created.

The previous, and same as per the rest of the following examples, can be sent in one Posting request, or it also can be divided into three different posting requests, if required, as per the different Posting Types.

Posting Action	Amount	Posting Type
Dr debit side cust	100	Principle debit amount (**MAIN_DR**)
Cr credit side cust	100	Principle credit amount (**MAIN_CR**)

Posting Action	Amount	Posting Type
Dr debit side cust	10	Fees on debit side customer (**FEE_DR**)
Cr PNL	10	Amount in P&L account for fees on debit side customer (**PNL_DR**)

Posting Action	Amount	Posting Type
Dr credit side cust	10	Fees on credit side customer (**FEE_CR**)
Cr PNL	10	Amount in P&L account for fees on credit side customer (**PNL_CR**)

Another variation of the above is around the consolidation option of the Fee and PNL entries[3]:

---

[3] Based on customer bank setup (see details in Business Setup section).

The following example shows Posting entries when two Fees are selected (10 and 5) and consolidation setup instructs - No consolidation of FEE and PNL entries.

Posting Action	Amount	Posting Type
Dr debit side cust	100	Principle debit amount (**MAIN_DR**)
Cr credit side cust	100	Principle credit amount (**MAIN_CR**)
Dr debit side cust	10	Fees on debit side customer (**FEE_DR**)
Cr PNL	10	Amount in P&L account for fees on credit side customer (**PNL_CR**)
Dr debit side cust	5	Fees on debit side customer (**FEE_DR**)
Cr PNL	5	Amount in P&L account for fees on credit side customer (**PNL_CR**)

The following example shows Posting entries when two Fees are selected (10 and 5) and consolidation setup instructs - consolidation of both FEE and PNL entries into one of each.

Posting Action	Amount	Posting Type
Dr debit side cust	100	Principle debit amount (**MAIN_DR**)
Cr credit side cust	100	Principle credit amount (**MAIN_CR**)
Dr debit side cust	15	Fees on debit side customer (**FEE_DR**)
Cr PNL	15	Amount in P&L account for fees on debit side customer (**PNL_DR**)

The example below shows Posting entries when two Fees are selected (10 and 5) and consolidation setup instructs - consolidation of FEE entries into one and no consolidation of PNL entries –

Posting Action	Amount	Posting Type
Dr debit side cust	100	Principle debit amount (MAIN_DR)
Cr credit side cust	100	Principle credit amount (MAIN_CR)
Dr debit side cust	15	Fees on debit side customer (FEE_DR)
Cr PNL	10	Amount in P&L account for fees on debit side customer (PNL_DR)
Cr PNL	5	Amount in P&L account for fees on debit side customer (PNL_DR)

A variation of the above Accounting Model is for if it requires including separate posting entries for the gross principal amount and the fees, even when these are defined as ***Deduct from Payment*** and in fact the Outgoing payment amount includes both (credit side fee deducted from the principal amount in a CTO or debit side fee added to the principal amount in a DDO):

The following examples show Posting entries per the simple model and its variation for separate entries - for CTO, a payment amount of 100 and credit side fees of 10 (special entries for this case are marked in Blue).

Simple Net Accounting Model

Posting Action	Amount	Posting Type
Dr debit side cust	100	Principle debit amount (**MAIN_DR**)
Cr credit side cust	90	Principle credit amount (**MAIN_CR**)
Cr PNL	10	Amount in P&L account for fees on credit side customer (**PNL_CR**)

Variation for separate entries for Gross Principal and Fee –

Posting Action	Amount	Posting Type
Dr debit side cust	100	Principle debit amount (**MAIN_DR**)
Cr credit side cust	100	Gross principle (no fee deduction) credit amount (**MAIN_GROSS_CR**)
Dr credit side cust	10	Fees on credit side customer (**FEE_CR**)
Cr PNL	10	Amount in P&L account for fees on credit side customer (**PNL_CR**)

If BEN fees are selected and no consolidation of the FEE and PNL entries, the previous example is as follows.

Simple Net Accounting Model

PNL entries are separated but all fees are deducted (once) from the MAIN_CR entry –

Posting Action	Amount	Posting Type
Dr debit side cust	100	Principle debit amount (**MAIN_DR**)
Cr credit side cust	85	Principle credit amount (**MAIN_CR**)
Cr PNL	5	Amount in P&L account for fees on credit side customer (**PNL_CR**)
Cr PNL	10	Amount in P&L account for fees on credit side customer (**PNL_CR**)

Variation for separate entries for Gross Principal and Fee

Both FEE and PNL entries are separated.

Posting Action	Amount	Posting Type
Dr debit side cust	100	Principle debit amount (**MAIN_DR**)
Cr credit side cust	100	Gross principle (no fee deduction) credit amount (**MAIN_GROSS_CR**)
Dr credit side cust	5	Fees on credit side customer (**FEE_CR**)

Posting Action	Amount	Posting Type
Dr credit side cust	10	Fees on credit side customer (**FEE_CR**)
Cr PNL	5	Amount in P&L account for fees on credit side customer (**PNL_CR**)
Cr PNL	10	Amount in P&L account for fees on credit side customer (**PNL_CR**)

When the Accounting Model requires supporting Outgoing Agent Charges (SWIFT Field 71G and its equivalent in ISO).

The following example shows Posting entries for CTO, an original payment amount of 100 and debit side agent fees of 10 (special entries for this case are marked in Blue).

Posting Action	Amount	Posting Type
Dr debit side cust	100	Principle debit amount (**MAIN_DR**)
Cr credit side cust	110	Principle credit amount (**MAIN_CR**)
Dr debit side	10	Agent fees on debit side customer (CT) (**AGT_FEE_DR**)

When the Accounting Model requires supporting Incoming Agent Charges (SWIFT Field 71G and its equivalent in ISO) –

The following example shows Posting entries for CTI, an original payment amount of 100 and debit side agent fees of 10 (special entries for this case are marked in Blue).

Posting Action	Amount	Posting Type
Dr debit side cust	110	Principle debit amount (**MAIN_DR**)
Cr credit side cust	100	Principle credit amount (**MAIN_CR**)
Cr PNL	10	Amount in P&L account for agent fees on debit side (CT) (**AGT_PNL_DR**)

When the Accounting Model requires supporting bulk posting:

- A 2-step-accounting model in which the posting steps involve a batch suspense account.
    a. The first step is performed on the message that represents the Incoming bulk (S), on the Incoming bulk flow.
    b. The second step is performed on the individual transaction messages, on their Individual flow, or on the message that represents the Outgoing bulk (A), on the Outgoing bulk flow.
- The principal credit and debit are, in this case, replaced with specific Posting Types to a *Batch Suspense Account* (**BSA**).

Note: GPP does not support in bulk processing the passing of the fees to the Individual transaction messages and their side customer. For that reason, no posting type allows credit side fees in the example below.

The example below shows Posting entries per this model where an Outgoing CTO bulk from the customer is processed and posted separately on the credit side (bulk processing only on the customer's side).

In the specific scenario, 10 Individuals in bulk, each with a payment amount of 100 and fees of 9 per Individual on the debit side customer (special entries for this case, in addition to the separation into 2 steps, are marked in Blue).

First Posting step on the Bulk (S) message[4] – Principal amount is sum of all in file

Posting Action	Amount	Posting Type
Dr debit side cust	1000	Principle debit amount (**MAIN_DR**)
Cr BSA	1000	Credit to batch susp. account for batch posting (**BSA_CR**)
Dr debit side cust	90	Fees on debit side customer (**FEE_DR**)
Cr PNL	90	Amount in P&L account for fees on debit side customer (**PNL_DR**)

Second Posting step on each Individual transaction message – Principal amount is per individual

Posting Action	Amount	Posting Type
Dr BSA	100	Debit to batch susp. account for batch posting (**BSA_DR**)
Cr credit side	100	Principle credit amount (**MAIN_CR**)

When the Accounting Model requires supporting bulk posting on both sides –

In the specific scenario, the 10 Individuals in Incoming bulk are bulked into two Outgoing files (as per the credit side customer/account).

First Posting step on the Bulk (S) message

Same as in the above example.

Second Posting step on bulk (A) message[5] – Principal amount is sum of all in file

Posting Action	Amount	Posting Type
Dr BSA	500	Debit to batch susp. account for batch posting (**BSA_DR**)
Cr credit side	500	Principle credit amount (**MAIN_CR**)

[4]The S message is a separate message that is processed in GPP (via the bulk flow) until completion (including the Posting step), and is GPP's way of represents the whole Incoming file

[5] The A message is a separate message that is created when the different Individual messages are bulked into different Outgoing files per the account to post on. This message is also processed in GPP until completion (including the Posting step), and is GPP's way of represents the whole Outgoing file

When the Accounting Model requires supporting 2-step-accounting for various reasons:

- It is required to do the debit side posting as soon as possible (after debit side processing + fees and FX steps) and the rest of the posting upon completion.

- It is required to do 1-step-accounting if STP, but dynamically change to 2-step-accounting model if the transaction is stopped in a manual queue:
    - Do the first leg just before the transaction is stopped and the second leg once the transaction is re-submitted and can be completed

- It is required to do 1-step-accounting if STP, but dynamically change to 2-step-accounting model if in an Outgoing Direct&Cover scenario. The Debit Value date is different than the Credit Value date (future date) and the bank's accounting system does not support future dated posting entries.
    - Do the first leg on the Direct and the second leg on the Cover, once it can be completed

- Other customer specific reasons, such as in G3 CTO flows, the need to post the debit leg before sending the transaction to G3, but do the credit side posting only after a response from G3 (to be able to use the date/cycle as recorded in G3).
    - In all of the above cases this Accounting Model would involve

- A 2-step-accounting model in which the posting steps involve a suspense account through which the funds are transferred in the first leg, to the second leg.

- The principal credit and debit, and entries for the fees passed to the second step are, in this case, replaced with specific Posting Types to a Suspense Account (SA).

The following example shows Posting entries per this model, for an Outgoing CT from the customer, payment amount of 100 and fees of 10 on debit side customer (special entries for this case, in addition to the separation into two steps, are marked in Blue).

First Posting step

Posting Action	Amount	Posting Type
Dr debit side cust	100	Principle debit amount (**MAIN_DR**)
Cr SA	100	Credit principal amount to susp. account for 2-stp-acctg (**SA_MAIN_CR**)
Dr debit side cust	10	Fees on debit side customer (**FEE_DR**)
Cr PNL	10	Amount in P&L account for fees on debit side customer (**PNL_DR**)

Second Posting step

Posting Action	Amount	Posting Type
Dr SA	100	Debit principal amount to susp. account for 2-stp-acctg (**SA_MAIN_DR**)
Cr credit side	100	Principle credit amount (**MAIN_CR**)

A variation of the above Accounting Model is, again, if it requires including separate posting entries for the gross principal amount and the fees, even when these were defined as ***Deduct from Payment***

and in fact the Outgoing payment amount includes both. In this variation – fees are to be posted on the first step.

The following examples shows Posting entries per the above model and its variation for separate entries – for CTO, a payment amount of 100 and credit side fees of 10.

Simple Net Accounting Model

First Posting step

Posting Action	Amount	Posting Type
Dr debit side cust	100	Principle debit amount (**MAIN_DR**)
Cr SA	90	Credit principal amount to susp. account for 2-stp-acctg (**SA_MAIN_CR**)
Cr PNL	10	Amount in P&L account for fees on credit side customer (**PNL_CR**)

Second Posting step

Posting Action	Amount	Posting Type
Dr SA	90	Debit principal amount to susp. account for 2-stp-acctg (**SA_MAIN_DR**)
Cr credit side	90	Principle credit amount (**MAIN_CR**)

Variation for separate entries for Gross Principal and Fee (special entries for this case, in addition to the separation into two steps, are marked in Blue).

First Posting step

Posting Action	Amount	Posting Type
Dr debit side cust	100	Principle debit amount (**MAIN_DR**)
Cr SA	100	Credit gross principal amount (no fee deduction) to susp. account for 2-stp-acctg (**SA_MAIN_GROSS_CR**)
Dr SA	10	Credit fee amount on credit side to susp. account for 2-stp-acctg (**SA_FEE_CR**)
Cr PNL	10	Amount in P&L account for fees on credit side customer (**PNL_CR**)

Second Posting step

Posting Action	Amount	Posting Type
Dr SA	90	Debit principal amount to susp. account for 2-stp-acctg (**SA_MAIN_DR**)
Cr credit side	90	Principle credit amount (**MAIN_CR**)

An additional variation of the above Accounting Model is for separate posting entries for the gross principal amount and the credit side *Deduct from Payment* fees. However in this variation, fees are to be posted in the second step.

The following examples show Posting entries per the above model and its variation for separate entries - for CTO, a payment amount of 100 and credit side fees of 10.

Simple Net Accounting Model

First Posting step

Posting Action	Amount	Posting Type
Dr debit side cust	100	Principle debit amount (**MAIN_DR**)
Cr SA	100	Credit principal amount to susp. account for 2-stp-acctg (**SA_MAIN_CR**)

Second Posting step

Posting Action	Amount	Posting Type
Dr SA	100	Debit principal amount to susp. account for 2-stp-acctg (**SA_MAIN_DR**)
Cr credit side	90	Principle credit amount (**MAIN_CR**)
Cr PNL	10	Amount in P&L account for fees on credit side customer (**PNL_CR**)

Variation for separate entries for Gross Principal and Fee (special entries for this case, in addition to the separation into two steps, are marked in Blue).

First Posting step

Posting Action	Amount	Posting Type
Dr debit side cust	100	Principle debit amount (**MAIN_DR**)
Cr SA	100	Credit principal amount to susp. account for 2-stp-acctg (**SA_MAIN_CR**)

Second Posting step

Posting Action	Amount	Posting Type
Dr SA	100	Debit principal amount to susp. account for 2-stp-acctg (**SA_MAIN_DR**)
Cr credit side	100	Gross principle (no fee deduction) credit amount (**MAIN_GROSS_CR**)
Dr credit side	10	Fees on credit side customer (**FEE_CR**)

Posting Action	Amount	Posting Type
Cr PNL	10	Amount in P&L account for fees on credit side customer (**PNL_CR**)

Balancing Entries:

Note: The balancing entries (all types) always quote the Fee and the Principal amounts (original) separately. There is no balancing entry that would use a Net amount (Fee deducted/added to the Principal).

When the Accounting Model requires supporting the posting of cross-booking balancing entries in the cases that a debit entry is posted to an account on one accounting system and the corresponding credit entry is posted to an account on another accounting system.

Note: These entries need to be generated, if required, per step/invocation, as per the above cases, to allow full balancing.

The following example shows Posting entries per this model for a simple Accounting Model – an Outgoing CT from the customer, a payment amount of 100 and fees of 10 on the debit side customer, in which the principal credit account and the debit customer account are managed in different accounting systems, and so are the PNL account and the debit customer fee account (additional entries for the balancing entries are marked in Blue).

Posting Action	Amount	Posting Type	Accounting System
Dr debit side cust	100	Principle debit amount (**MAIN_DR**)	AccSys1
Cr credit side cust	100	Principle credit amount (**MAIN_CR**)	AccSys2
Cr SA1	100	Principal amount- Cross-booking-systems balancing on debit side system (**SA_CRSS_BOOK_MAIN_DBT_SD**)	AccSys1
Dr SA2	100	Principal amount- Cross-booking-systems balancing on credit side system (**SA_CRSS_BOOK_MAIN_CDT_SD**)	AccSys2
Dr debit side cust	10	Fees on debit side customer (**FEE_DR**)	AccSys1
Cr PNL	10	Amount in P&L account for fees on debit side customer (**PNL_DR**)	AccSys3
Cr SA1	10	Fee amount- Cross-booking-systems balancing on debit side system (**SA_CRSS_BOOK_FEE_DBT_SD**)	AccSys1
Dr SA3	10	Fee amount- Cross-booking-systems balancing on credit side system (**SA_CRSS_BOOK_FEE_CDT_SD**)	AccSys3

When the Accounting Model requires supporting the posting of FX balancing entries in the cases that a debit entry is posted to an account in one currency and the corresponding credit entry is posted to an account in another currency.

Note: These entries need to be generated, if required, per step/invocation, as per the above cases, to allow full balancing.

The example below shows Posting entries per this model for an Outgoing CT from customer, payment amount of Debit amount of 100 EUR and fees of 10 EUR on the debit side customer, while the

principal credit account the PNL account are USD accounts (additional entries for the balancing entries are marked in Blue).

Posting Action	Amount	Posting Type
Dr debit side cust	100 EUR	Principle debit amount (**MAIN_DR**)
Cr credit side cust	128 USD	Principle credit amount (**MAIN_CR**)
Cr SA1	100 EUR	Principal amount- FX balancing on debit CCY (**SA_FX_BAL_MAIN_DBTCCY**)
Dr SA2	128 USD	Principal amount- FX balancing on credit CCY (**SA_FX_BAL_MAIN_CDTCCY**)
Dr debit side cust	10 EUR	Fees on debit side customer (**FEE_DR**)
Cr PNL	12.8 USD	Amount in P&L account for fees on debit side customer (**PNL_DR**)
Cr SA1	10 EUR	Fee amount- FX balancing on debit (Fee) CCY (**SA_FX_BAL_FEE_DBTCCY**)
Dr SA2	12.8 USD	Fee amount- FX balancing on credit (PNL) CCY (**SA_FX_BAL_FEE_CDTCCY**)

The future roadmap will also include the additional functionality elements for the Accounting Models that require supporting the posting of cross-branch[6] balancing entries, cross-date[7] balancing entries, separate entries for tax amounts (on fees and principal amount), dedicated entries for rebate[8] and separate entry for Return of Funds.

2 Processing

2.1 Service Invocation

In each flow that GPP supports there are specific places in the flow in which the Posting step is invoked.

As per the specific requirements for each site, and per the different transaction scenarios, GPP can be configured to create or not create posting entries (original or reversed) for this invocation. If configured to create posting entries, this configuration also determines which posting types are created in each of these invocations. This flexibility is achieved using the predefined system setup of the Posting Type Selection rules.

Just before the rules are invoked, the Posting invocation context (D_POSTING_INVOCATION_CONTEXT) field is updated with the relevant value, for the point in processing workflow where this invocation is being performed (values are configured in Fields_Values for Field_Type=POSTING_INVC_CONTEXT). This field can then be used within the conditions of the Posting Type Selection rules to differentiate what and when is included in each invocation. For details on Posting invocation context see Message Attributes.

The following are the currently supported values, per the supported exit points:

Value	Exit Point Description
BEFORE_IN_COMPLETE_OR _OUT_SEND	The most common exit point, used in High Value flow, right before sending a transaction out, or just before an Incoming transaction is set as completed. Used for one leg posting, or for second leg on target side posting in Optimistic mode.
AFTER_OUT_ACK	The exit point within the flow invoked after receiving an ACK NAK from the system that the transaction was sent to. Used for one leg posting, or for second leg on target side posting in Pessimistic mode for Outgoing transactions.
AFTER_NAK_RETURN_CANC ELATION	The exit point within the negative flows, after cancelation, rejection, return, NAK from system that the transaction was sent to, or a failure in posting of 2nd leg for a scenario of more than one leg posting. Mainly used for reversing already posted entries, and, if required, adding reversal fees.
BEFORE_STOP_NON_FINAL_ QUEUE	The exit point within the termination flow, when a transaction is stopped in a manual or wait queue. Used when the Accounting model requires to dynamically change the normal behavior of one leg posting (in STP scenarios), into a twosteps model, The first step performs source side posting before stopping.
BULK_SUBBATCH	The exit point on the flow of the S message. Used when source side lump sum posting is required (either Incoming file from clearing or file received from customer with lump-sum posting required instruction (Batch-Booking=True)), so lump-sum posting was performed on the S message.
INDIVIDUAL_SUBBATCH_CO MPL	The exit point on the flow of the individual transaction in the Mass Payment flow. Used when individuals are required to be posted as itemized, either when the other side is required to be posted as lump sum posting, or when posting should be performed in one leg per each individual.

Value	Exit Point Description
BULK_OUT_FILE	The exit point on the flow of the A message. Used when target side lump sum posting is required and done on the A message.

In general, in the case of a Direct & Cover, the posting entries are on the Direct for Incoming transactions and on the Cover for Outgoing transactions (based on the system setup).

The exception for this would be:

- If the Incoming Cover is received from clearing but as an Inter-office message, through another Office. In this case the Direct debits the other office's account, and the real debit to the clearing settlement account is on the Incoming Cover (to the other Office), along with a credit to the destination's Office account.

- If the specific bank's Accounting Model requires that in the case Dr Value date is different than Cr Value date, a 2-step-accounting is performed: the first leg (Debit leg) is on the Direct, and the second leg, the (Credit leg) is on the Cover (when ready to be posted).

In the case of an Incoming Direct & Cover, if the Direct is processed to completion, without waiting for the cover (due to Sender Credit Line, or for Directs that their Cover is suspected to arrive through clearing – and debit account is enriched as the clearing debit account to allow early appearance in the position of the account), there may be a case in which the Cover, once received, may differ than Direct info, such as a different value date, an additional party added to chain that changes the Debit party/account or different amount. In such scenarios posting adjustments may be required on the Cover (Future).

2.2 Posting Calculation

When reaching the step in processing where accounting entries need to be created (prior to building the Posting Interface Request file), the inner step, creating the selected entries for current message and current step, is performed.

- For each of the accounts that are used for postings:
    - If the Accounts entry does not have any value in the *Booking Entity attribute* (**Accounts.Booking_Ent**) –
        › The system looks for the default **Booking Entity** in the **Office** profile (**Banks.Def_Bookng_Ent**) and uses it as this account's **Booking_Entity**
        › If such a default Booking Entity does not exist
            o The posting process is stopped with a failure to post for this message
            o An error is logged in the *System Log* (**ErrorLog**) and **MsgErr**
            o None of the posting entries are created for this message
            o Interface returns control to GPP Core with a Request Failure indication
            o Message falls to *Repair* queue
- For each entry prepared for posting
    - If the Amount value for posting is 0[9]

---

[9] Such a case can occur when preparing customer fee debit entry for which a 100% discount was given to the customer

- If this is not an entry related to Fee (either FEE or PNL) or the entry is related to Fee and the configuration in the Office profile does not instruct to create an entry for a 0 fee (***Include posting entry for 0 amount fee is not set*** (**Banks.Post_Zero_Fee**<>1)) [10]
  - Posting Entry is not created
- If the Currency value for posting is not the Base Currency for the relevant ***Office*** (checking **Banks.Currency**)
  - Interface computes the **Base_Amount** by invoking the GPP Core conversion function at the moment of creation of each **Msg_Posting entry**, and does not rely on the rates and Equivalent Amounts computed during prior GPP computations[11]

The logic of generating the different posting entries is derived from the Posting Types (and the rule sub-type) that are selected by the ***Posting Type Selection*** rules (system)

- If the Posting Type was selected as Reverse posting, GPP searches for the relevant original entry, per Posting Type, either on current transaction message or on its related lump-sum message (S for MP flows and MT102 for a child MT103).
  - When the original is not on the current message but on the lump-sum message[12]
    - If the P_SUSPENSE_ACCT_UID is not empty (meaning the original entry to be reversed from first step is a lump-sum posting on the relevant S or MT102)
      - GPP creates a new entry with values copied from original entry to be reversed (the one with same Posting Type from the lump-sum message of first step), with these differences – **Amount** is taken from current message[13], Reversed_Orig_Posting_Entry_ID copied from Posting_Entry_ID of original, Cr_Dr_Flag is populated with the opposite of value in original, I**s_Reversal_Ind**=1 and, a new Posting_Entry_ID
      - As there is a relation of 1:n between the original posting entry (on the S) and the possible reversed (per Individual), the **Posting_Entry_ID** of the new reversed entry is NOT copied into the **Reversed_Orig_Posting_Entry_ID** of the original entry. In this case the **Reversed_Orig_Posting_Entry_ID** in the original (of the S) remains empty.
    - If the P_SUSPENSE_ACCT_UID is empty (meaning the posting was not done as a lump sum on the first step, and original entry is on same individual transaction message
      - GPP uses same logic as for regular reverse on same message (next bullet).

      Note: Above logic applies also for a scenario of MT102 (behaving like an S from Posting perspective) and its related MT103 individuals (behaving like I's).
  - Else (the original postings to be reversed are on current message)
    - GPP creates a new entry with values copied from original entry to be reversed (the one with same Posting Type), with these differences: **Reversed_Orig_Posting_Entry_ID** copied from **Posting_Entry_ID** of original, **Cr_Dr_Flag** is populated with the opposite of value in original, **Is_Reversal_Ind**=1 and, of course, a new **Posting_Entry_ID**
    - The **Posting_Entry_ID** of the new reversed entry is copied into the **Reversed_Orig_Posting_Entry_ID** of the original entry

---

[10] There are cases that a bank requires a 0 posting entry to be created for such a fee

[11] This will allow overcoming non-balanced totals of the postings (as per the equivalent amounts), since rates may have changed during the GPP processing.

[12] The original was posted on another message representing the Incoming file

[13] This would be the amount to be reversed per this message (I or A), as opposed to the **Amount** in the original posting entry (on the S) which was the full bulk amount

Principal entries:

- If *Principle debit amount* (**MAIN_DR**) Posting Type is selected as **Original posting** in this step

    – The system creates a Debit entry – *Principle debit amount* (**MAIN_DR**)

    – If Balance Inquiry was invoked prior to this Posting invocation and an earmark reference was returned for this debit amount (**P_BI_MAIN_EARMARK_REF** is not empty) – this reference is quoted in the posting entry

- If *Principle credit amount* (**MAIN_CR**) Posting Type is selected as *Original posting* in this step

    – The system creates a Credit entry – *Principle credit amount* (**MAIN_CR**)[14]

- If *Gross principle (no fee deduction) credit amount* (**MAIN_GROSS_CR**) Posting Type is selected as *Original posting* in this step (Accounting Model requires separate Fee entry in case of fees deducted from the Outgoing payment amount)

    – In a case there were credit fees deducted from the Outgoing payment –

        › The system creates a Credit entry for Principal Amount for credit side, with the only difference from a regular *Principle credit amount* (**MAIN_CR**) entry – **Amount** is taken from *Cdt amt* (**MINF.P_CDT_AMT**)[15] instead of *Sttlm amt* (**MINF.X_STTLM_AMT**)

    – Else (there were no credit fees deducted from the Outgoing payment) –

        › Interface changes Posting Type to *Principle credit amount* (**MAIN_CR**) and continues the logic as above (with **MAIN_CR**)

- If *Debit to batch susp. account for batch posting* (**BSA_DR**) Posting Type is selected as *Original posting* in this step (message is processed with bulk posting, at least on one side)

    – Interface prepares a Debit entry for Principal Amount for debit side into batch suspense account

- If **Credit to batch susp. account for batch posting** (**BSA_CR**) Posting Type is selected as *Original posting* in this step (message is processed with bulk posting, at least on one side) –

    – Interface prepares a Credit entry for Principal Amount for credit side into batch suspense account

Note: *Debit to batch susp account for batch posting* (**BSA_DR**) and *Credit to batch susp account for batch posting* (**BSA_CR**) Posting Type is selected either on the message that represents the Incoming file (S), or on the individual (in case of In-Bulk) that originated from this file, or the message that represents the Outgoing file (A) that includes such individual (in case of In-Out bulk or only Out bulk).

- If *Debit principal amount to susp. account for 2-stp-acctg* (**SA_MAIN_DR**) Posting Type is selected as *Original posting* in this step (2-step-accounting not due to bulk is required)

    – If *Posting Suspense Account Selection* rule with sub-type *SA_MAIN_DR* was not caught – setup problem

---

[14] The Amount can be – In CT message - a Net Amount after deduction of immediate fees defined in the Fee Formula as *Deduct From Payment*, or the original payment amount if Fee Formula was defined as *Deduct From Account.* The deduction, when applicable, is performed in Core GPP during Fee Calculation stage. This amount can also include Agent Fees added to be passed to the next bank. . In DDI case – this amount can also include the Incoming Agent Fees

[15] That quotes the original amount

> › The posting process is stopped with a failure to post for this message

> › An error is logged in the **MsgErr** indicating a missing setup for suspense account of type *SA_MAIN_DR*

> › None of the posting entries are saved for this message

> › Interface returns control to GPP Core with a Request Failure indication

> › Message falls to *Repair* queue

- – Else (a suspense account was selected for this entry)

> › The system creates a Debit entry for Principal Amount for debit side into a suspense account. The suspense account for this scenario will be selected using the *Posting Suspense Account Selection* rules with sub-type *SA_MAIN_DR*

- If *Credit principal amount to susp. account for 2-stp-acctg* (**SA_MAIN_CR**) Posting Type is selected as *Original posting* in this step (2-step-accounting not due to bulk is required) –

  - – If *Posting Suspense Account Selection* rule with sub-type *SA_MAIN_CR* was not caught – setup problem –

    > › The posting process is stopped with a failure to post for this message

    > › An error is logged in the *MsgErr* indicating a missing setup for suspense account of type *SA_MAIN_CR*

    > › None of the posting entries are saved for this message

    > › Interface returns control to GPP Core with a Request Failure indication

    > › Message falls to *Repair* queue

  - – Else (a suspense account was selected for this entry)

    > › The system creates a Credit entry for Principal Amount for credit side into a suspense account. The suspense account for this scenario will be selected using the *Posting Suspense Account Selection* rules with sub-type *SA_MAIN_CR*.

- If *Credit gross principal amount (no fee deduction) to susp. account for 2-stp-acctg* (**SA_MAIN_GROSS_CR**) Posting Type is selected as *Original posting* in this step (Accounting Model requires 2-step-accounting and separate Fee entry in case of fees deducted from the payment amount) –

  - – If credit fees are deducted from the payment

    > › If *Posting Suspense Account Selection* rule with sub-type *SA_MAIN_GROSS_CR* was not caught – setup problem –

    > o The posting process is stopped with a failure to post for this message

    > o An error is logged in the **MsgErr** indicating a missing setup for suspense account of type *SA_MAIN_GROSS_CR*

    > o None of the posting entries are saved for this message

    > o Interface returns control to GPP Core with a Request Failure indication

    > o Message falls to *Repair* queue

    > › Else (a suspense account was selected for this entry)

    > o The system creates a Credit entry for Principal Amount for credit side, with the only difference from a *Credit principal amount to susp. account for 2-stp-acctg* (**SA_MAIN_CR**) entry – **Amount** is taken from *Principle credit amount* (**MINF.P_CDT_AMT**) instead of *Sttlm amt* (**MINF.X_STTLM_AMT**)

Note: The suspense account for this scenario will be selected using the *Posting Suspense Account Selection* rules with sub-type *SA_MAIN_CR*, as it needs to select same account as per the *SA_MAIN_CR*.

- – Else (there were no credit fees deducted from the payment)

    - › The system changes Posting Type to *Credit principal amount to susp. account* for 2-stp-acctg (**SA_MAIN_CR**) and continues the logic below (for **SA_MAIN_CR**)

Fee entries:

Note: In the posting logic the distinction between *Fees on the debit side customer* (**FEE_DR**) and *Fees on the credit side customer* (**FEE_CR**) (and their equivalent *Amount in profit P&L account for fees on debit side customer* (**PNL_DR**)/*Amount in profit P&L account for* **fees on credit side customer** (**PNL_CR**)) is based on the side of the customer that actually is charged and not, as in the fees logic, the customer on whom this fee is defined (rules attached).

Therefore there are cases in which a fee is defined on the Dr customer, but the charged customer is actually the Cr customer. Such fees will be marked in the posting entries as *Fees on the credit side customer* (**FEE_CR**).

Also, if no setup is found with consolidation instructions in either level, the default behavior is NO (no consolidation).

- • If either of the **Fees** or **PNL Posting Types** are selected as *Original posting* for this step (the fees involved in this transaction should be posted in this step), and these fees are immediate fees[16] and were not already posted (there are **Msg_Fees** entries in which **Msg_Fees.Fee_Posting_Entry_ID** is empty) –

    - – If *Fees on debit side customer* (**FEE_DR**) Posting Type is selected for fees on the debit side customer –

        - › The systems uses the **Msg_Fees** entries for this MID, in which **Msg_Fees.Fee_Posting_Entry_ID** is empty, that should be posted immediately (**Msg_Fees.Apply** = NOW) and that are for debit side (**Msg_Fees.Paying_Party** = DR, and are defined to be deducted from the customer fee account (*Fee Formula* instructed to *Deduct From Account* (**Msg_Fees.Deduct_From** = A))[17] to create one consolidated **FEE_DR** posting entry or more entries, based on the owning customer (*Parties* profile) configuration or, if empty, based on the default configuration per *Office* –

            - o If the configuration instructs consolidation of all fees to one entry (**Customrs.Cnsld_Fee_Post/Banks.Def_Cnsld_Fee_Post** ='Yes- to one') – one Debit entry will be created, summing fees[18] from the above entries, for this side customer

            - o If the configuration instructs consolidation of fees per Fee Type Category (**Customrs.Cnsld_Fee_Post/Banks.Def_Cnsld_Fee_Post** ='Yes- per fee type ctgy') – a few Debit entries will be created depending on the *Fee Type Category*, summing fees from the above entries grouped by the **Fee_Types.Fee_Type_Group** for this side customer

---

[16] Fee postings are not sent for fees that were marked as either **LATER** or **WAIVE**

[17] In the case of Outgoing CT message and DR side Fee rules (rules on the Debit side customer) if the *Fee Formula* is defined as *Deduct From Payment* (**Msg_Fees.Deduct_From = P**) - the fees will actually be charged to the Credit side - as the payment amount will be reduced. Current logic does not cover the DDO case in which fees are added to payment. Will need to be included once GPP supports such behavior. In case of Incoming CT message with Agent Fees – the DR side **Msg_Fees** will be marked as *Deduct From Payment* (**Msg_Fees.Deduct_From = P**) as well – so these will not be included in the **FEE_DR** posting (as these were paid by debit customer that is not in current bank)

[18] Fee amount taken always from the field in which it is already quoted in the appropriate customer's fee account's currency

- If the configuration instructs not to consolidate fees (**Customrs.Cnsld_Fee_Post/Banks.Def_Cnsld_Fee_Post** =No) – separate Debit entry will be created for each of the above **Msg_Fees** entries

› If no such **Msg_Fees** entries are found[19] – no **Msg_Posting** entries will be created for this type

- If *Amount in profit P&L account for fees on debit side customer* (**PNL_DR**) Posting Type is selected for this step

Note: The system setup would be defined so that either **Amount in profit P&L account for *fees on debit side customer*** (**PNL_DR**) or *Amount in P&L account for agent fees on debit side (CT)* (**AGT_PNL_DR**) is selected but never both, as they are looking at the same **Msg_Fees** entries and should never be created together. (**AGT_PNL_DR** is only relevant in Incoming CTs in which 71G in SWIFT terminology (or its ISO equivalent) is not empty).

› The system uses the same subset of **Msg_Fees** entries as for the *Fees on debit side customer* (**FEE_DR**)[20] to create one consolidated **PNL_DR** entry or more entries, based on whether same or different PNL account is defined for the different fees, and for each of the defined PNL accounts – based on each P&L *Accounts* entry configuration or, if empty, based on the default configuration per *Office* –

- If the configuration instructs consolidation of all fees to one entry (**Accounts.Cnsld_PNL_Post/Banks.Def_Cnsld_PNL_Post** = 'Yes- per P&L account') – one Credit entry per each P&L account involved will be created, summing fees[21] from the above entries, for this side customer

- If the configuration instructs consolidation of fees per Fee Type Category (**Accounts.Cnsld_PNL_Post / Banks.Def_Cnsld_PNL_Post** = 'Yes- per fee type ctgy') – a number of Credit entries will be created depending on the *Fee Type Category* and each P&L account involved, summing fees from the above entries grouped by the **Fee_Types.Fee_Type_Group** for this side customer

- If the configuration instructs not to consolidate fees (**Accounts.Cnsld_PNL_Post/Banks.Def_Cnsld_PNL_Post** = No) – separate Credit entry will be created for each of the above **Msg_Fees** entries for this side customer

› If no such **Msg_Fees** entries are found – no **Msg_Posting** entries will be created for this type

- If *Credit fee amount on credit side to susp. account for 2-stp-acctg* (**SA_FEE_CR**) Posting Type is selected for fees on the credit side customer (along with *Credit gross principal amount (no fee deduction) to susp. account for 2-stp-acctg* (**SA_MAIN_GROSS_CR**))[22]

› If *Posting Suspense Account Selection* rule with sub-type *SA_FEE_CR* was not caught – setup problem -

- The posting process is stopped with a failure to post for this message.

---

[19] No fees to be applied now as deduct from account on the debit side customer

[20] As per current logic (see above footnote) - the **FEE_DR** and the **PNL_DR** posting entries should use the same **Msg_Fees** entries.

[21] Fee amount taken always from the field in which it is already quoted in the appropriate P&L account's currency (**Fee_PNL_Amount**)

[22] The case the accounting model requires to post the credit side fees in a different posting entries even when they are really deducted from the payment amount

- o An error is logged in the **MsgErr** indicating a missing setup for suspense account of type *SA_FEE_CR*.
- o None of the posting entries are saved for this message.
- o Interface returns control to GPP Core with a Request Failure indication.
- o Message falls to *Repair* queue.
- › Else (a suspense account was selected for this entry)
- o The systems uses the Msg_Fees entries for this **MID**, in which **Msg_Fees.Fee_Posting_Entry_ID** is empty, that should be posted immediately (**Msg_Fees.Apply** = NOW) and are for credit side –
    - ▪ Fees defined by rules on the credit side customer (**Msg_Fees.Paying_Party** = CR) as should be deducted from the customer fee account (*Fee Formula* instructed to *Deduct From Account* (**Msg_Fees.Deduct_From** = A))

    Or

    - ▪ Fees defined by rules on either side customer[23] as should be deducted from the payment (*Fee Formula* instructed to *Deduct From Payment* (**Msg_Fees.Deduct_From** = P except the ones related to Incoming Agent Fees[24]))
- › If no such **Msg_Fees** entries are found – no **Msg_Posting** entries will be created for this type

- – If *Fees on credit side customer* (**FEE_CR**) Posting Type is selected for fees on credit side customer
  - › The systems uses the **Msg_Fees** entries for this **MID**, in which **Msg_Fees.Fee_Posting_Entry_ID** is empty, that should be posted immediately (**Msg_Fees.Apply** = NOW) and are for credit side –
  - › If *Gross principle (no fee deduction) credit amount* (**MAIN_GROSS_CR**) is selected, along with this *Fees on credit side customer* (**FEE_CR**) (and not *Principle credit amount* (**MAIN_CR**))[25]
    - ▪ Fees defined by rules on the credit side customer (**Msg_Fees.Paying_Party** = CR) as should be deducted from the customer fee account (Fee Formula instructed to Deduct From Account (**Msg_Fees.Deduct_From** = A))

    Or

    - ▪ Fees defined by rules on either side customer[26] as should be deducted from the payment (*Fee Formula* instructed to *Deduct From Payment*

---

[23] There are cases in which on an Outgoing CT message a DR side fee is defined as *Deduct from Payment* (either in the original Fee Formula, or converted if account is an Asset account) – which actually passes it to the credit side customer

[24] These will be defined with **Msg_Fees.Deduct_From** = P but should not be included in the **FEE_CR** entry as they were charged on the debit customer in the previous bank

[25] The case the accounting model requires to post the fees in a different posting entries even when they are really deducted from the payment amount

[26] There are cases in which an Outgoing CT message a DR side fee is defined as *Deduct from Payment* (either in the original Fee Formula, or converted if account is an Asset account) – which actually passes it to the credit side customer

<div style="text-align: right">

(**Msg_Fees.Deduct_From** = P except the ones related to Incoming Agent Fees[27]))

</div>

- o Else (***Principle credit amount*** (**MAIN_CR**) is selected along with this ***Fees on credit side customer*** (**FEE_CR**)) –

- o The system only uses the **Msg_Fees** entries above that should be deducted from the customer fee account (***Fee Formula*** instructed to ***Deduct From Account*** (**Msg_Fees.Deduct_From** = A))

  › The system creates one consolidated **FEE_CR** entry per the credit account[28] or more entries based on the owning customer (***Parties*** entry) configuration or, if empty, based on the default configuration per ***Office***. Logic for consolidation is the same as per ***Fees on debit side customer*** (**FEE_DR**)

  › If no such **Msg_Fees** entries are found – no **Msg_Posting** entries will be created for this type

- – If **Amount in profit P&L account for *fees on credit side customer*** (**PNL_CR**) Posting Type is selected for this step –

Note: The system setup would be defined so that either the ***Amount in profit P&L account for fees on credit side customer*** (**PNL_CR**) or the ***Amount in P&L account for agent fees on credit side (DD)*** (**AGT_PNL_CR**) is selected but never both, as they are looking at the same Msg_Fees entries and should never be created together. (**AGT_PNL_CR** is only relevant in Incoming DDs in which 71G in SWIFT terminology (or its ISO equivalent) is not empty).

  › The systems uses the same subset of **Msg_Fees** entries as for the ***Fees on credit side customer*** (**FEE_CR**) to create one consolidated ***Amount in profit P&L account for fees on credit side customer*** (**PNL_CR**) entry or more entries, based on whether same or different PNL account is defined for the different fees. For each of the defined PNL accounts, based on the P&L ***Accounts*** entry configuration or, if empty, based on the default configuration per ***Office***.

If no such **Msg_Fees** entries are found – no **Msg_Posting** entries will be created for this type.

- If ***Agent fees on debit side customer*** (**CT**) (**AGT_FEE_DR**)[29] Posting Type is selected as ***Original posting*** for agent fees on debit side customer –

  – The systems uses the **Msg_Fees** entries for this **MID**, in which **Msg_Fees.Fee_Posting_Entry_ID** is empty, that should be posted immediately (**Msg_Fees.Apply** = NOW) and that are for debit side (**Msg_Fees.Paying_Party** = AF[30], to create one consolidated Fee posting entry or more entries, based on the owning customer (***Parties*** entry) configuration, or, if empty, based on the default configuration per ***Office***.

  – If no such **Msg_Fees** entries are found, no **Msg_Posting** entries will be created for this type.

---

[27] These will be defined with **Msg_Fees.Deduct_From** = P but should not be included in the **FEE_CR** entry as they were charged on the debit customer in the previous bank

[28] If some of the fees are defined as ***Deduct from Payment*** and some as ***Deduct from Account*** – there could be 2 separated Credit accounts

[29] Only be relevant for Outgoing CT messages - Fees defined as Agent Fee on debit side customer (would be passed in payment amount (and quoted in 71G in SWITF terminology) to be taken by the next bank in credit chain). This type of entry will be created on the originating side with no related credit entry to a P&L account – as it is intended for another bank. The fee amount will be added to the payment amount. The **AGT_PNL_DR** posting type will be recorded in the next bank when the message will be received as an Incoming CT.

[30] In the case of Outgoing Agent Fees in CT message the fees will always be defined as ***Deduct from Account***

- If *Amount in P&L account for agent fees on debit side (CT)* (**AGT_PNL_DR**)[31] Posting Type was selected as Original posting for the agent fees on the debit side customer –

  – The systems uses the **Msg_Fees** entries for this **MID**, in which **Msg_Fees.PNL_Posting_Entry_ID** is empty, that should be posted immediately (**Msg_Fees.Apply** = NOW) and that are for debit side (**Msg_Fees.Paying_Party** = DR[32] for all the attributes that relate to the PNL account, but will take the charge amount from the Incoming transaction – Field 71G in SWITF terminology (or its ISO equivalent). Only one entry will be created for this type, regardless of the consolidation setup.

  – If no such **Msg_Fees** entries are found, no **Msg_Posting** entries will be created for this type

- If *Agent fees on credit side customer* (**DD**) (**AGT_FEE_CR**)[33] Posting Type is selected as Original posting for agent fees on debit side customer

  – The system uses the **Msg_Fees** entries for this **MID**, in which **Msg_Fees.Posting_Entry_ID** is empty, that should be posted immediately (**Msg_Fees.Apply** = NOW) and that are for debit side (**Msg_Fees.Paying_Party** = AF**[34]**, to create one consolidated Fee posting entry or more entries, based on the owning customer (*Parties* entry) configuration or, if empty, based on the default configuration per *Office*

  – If no such **Msg_Fees** entries are found – no **Msg_Posting** entries will be created for this type

- If *Amount in P&L account for agent fees on credit side (DD)* (**AGT_PNL_CR**)[35] Posting Type is selected as *Original posting* for agent fees on debit side customer –

  – The systems uses the Msg_Fees entries for this **MID**, in which **Msg_Fees.PNL_Posting_Entry_ID** is empty, that should be posted immediately (**Msg_Fees.Apply** = NOW) and that are for debit side (**Msg_Fees.Paying_Party** = DR[36] for all the attributes that relate to the PNL account, but will take the charge amount from the Incoming transaction – Field 71G in SWITF terminology (or its ISO equivalent). Only one entry will be created for this type, regardless of the consolidation setup.

  – If no such **Msg_Fees** entries are found, no **Msg_Posting** entries will be created for this type

- Whenever a **Msg_Fees** entry is taken into account within a **Msg_Posting** entry, the **Msg_Posting.Posting_Entry_ID** is copied into the **Msg_Fees.Fee_Posting_Entry_ID**. Whenever a **Msg_Fees** entry is taken into account within a **Msg_Posting** PNL entry, the **Msg_Posting.Posting_Entry_ID** is copied into the **Msg_Fees.PNL_Posting_Entry_ID**[37]

---

[31] Only be relevant for Incoming CT messages - Fees defined as Agent Fee on original debit side customer (that were passed in payment amount (and quoted in 71G in SWITF terminology) from previous bank to be taken by the current bank). This type of entry will be created on Incoming CT with no debit entry to a customer fee account – as it was already charged on originating debit customer and was intended for current bank.

[32] These **Msg_Fees** entries are computed on the current bank for the purpose of deriving the PNL account and to allow validation that the charge amount passes in the payment amount (and quoted in Field 71G (or ISO equivalent)) does not exceed computed DR fees + accepted tolerance

[33] Only be relevant for Outgoing DD messages - Fees defined as Agent Fee on credit side customer (would be passed in payment amount (and quoted in 71G in SWITF terminology) to be taken by the next bank in credit chain). This type of entry will be created on the originating side with no related credit entry to a P&L account – as it is intended for another bank. The fee amount will be deducted from the settlement amount. The **AGT_PNL_CR** posting type will be recorded in the next bank when the message will be received as an Incoming DD.

[34] In the case of Outgoing Agent Fees in DD message the fees will always be defined as *Deduct from Account*

[35] Only be relevant for Incoming DD messages - Fees defined as Agent Fee on original credit side customer (that were passed in settlement amount (and quoted in 71G in SWITF terminology) from previous bank to be taken by the current bank). This type of entry will be created on Incoming DD with no debit entry to a customer fee account – as it was already charged on originating credit customer and was intended for current bank.

[36] These **Msg_Fees** entries are computed on the current bank for the purpose of deriving the PNL account and to allow validation that the charge amount passes in the settlement amount (and quoted in Field 71G (or ISO equivalent)) does not exceed computed CR fees + accepted tolerance

[37] To allow identifying which of the fee entry was already taken for posting (to prevent this fee entry from being re-calculated/updated after the posting step).

- At this point, after the above Principal and Fees entries have been created, the interface takes care of any adjustment that might be required so that the Total Equivalent Amount of debit entries and Total Equivalent Amount of credit entries is balanced[38] –

  - The interface sums all **Base_Amount** values from the entries with **CR_DR_Flag** = DR, sums all **Base_Amount** values from the entries with **CR_DR_Flag** =CR, and compares them

  - If they are not exactly the same (some 0.0X of difference can occur due to rounding)

    › The system calculates the difference

    › The system looks for the first P&L account involved in these postings (in the postings with Posting Type *Amount in profit P&L account for fees on debit side customer* (**PNL_DR**) or *Amount in profit P&L account for fees on credit side customer* (**PNL_CR**))

    › If it finds such an entry, it adjusts the **Base_Amount** with the difference, to make the totals balance (deducts if CR Total is greater than DR total, or adds if less than)

    › If no Fee posting/s and their parallel Fee PNL posting/s exist –

      o The system checks whether one of the accounts in of the Principal Amount is an Asset or a Suspense account (not a customer account) – *Posting Type Credit principal amount to susp. account for 2-stp-acctg* (**SA_MAIN_CR**)/*Debit principal amount to susp. account for 2-stp-acctg* (**SA_MAIN_DR**) or *Credit to batch susp. account for batch posting* (**BSA_CR**)/*Debit to batch susp. account for batch posting* (**BSA_DR**).

      o If it finds such an own bank account, the interface adjusts the **Base_Amount** of the Principal Amount on the side with the difference, to make the totals balance (if on DR side - deducts if DR Total is greater than CR total, or adds if opposite; If on CR side - deducts if CR Total is greater than DR total, or adds if opposite;)

      o Else (both Principal Amount postings are on customer accounts – Posting Types **MAIN_CR** and **MAIN_DR**) – Interface adjusts the **Base_Amount** of the Principal Amount on the Debit customer side (Posting Type **MAIN_DR**) with the difference – to make the totals balance (deducts if **DR** Total is greater than **CR** total, or adds if opposite).

Balancing entries:

General:

In some banks the currency balancing accounts are only in one system, while in other banks, the cross-book balancing accounts are only in the base currency. This means that the decision on which currency (before or after the conversion) to do the Cross-Book entries is dependent on bank specific conditions. To support such deviation, the responsibility to properly select the suspense accounts to be used for the balancing entries is on the bank (through the *Posting Suspense Account Selection* rules).

Once such suspense accounts are selected, the order of transferring the relevant amounts through these two types of balancing entries is based on their currency and the Booking Entity attributes.

The following are some guidelines regarding the valid combinations:

---

[38] There could be cases, due to FX conversions and rounding, in which the Total of all Equivalent Amounts for debit postings and the Total of all Equivalent Amounts for credit postings are not balanced. The rest of the Posting types do not involve FX conversions, and their Equivalent Amount is either in Base Currency (in all cases of Cross-Booking) or is based on the original posting entry (in the case of SPOT FX). Therefore – the balancing should be performed at this point.

FX Balancing – The suspense accounts selected (using the ***Posting Suspense Account Selection*** rules) to be used for the FX balancing entries should be should be – one in the credit and the other in the debit currency, of the posting entries to be balanced.

Cross-Book Balancing – The suspense accounts selected (using the ***Posting Suspense Account Selection*** rules) to be used for the Cross-Book balancing entries should be – one managed in the credit and the other in the debit Booking Entity (accounting system), of the posting entries to be balanced.

In addition:

a. If, for either the principal or the fee amounts, both FX and Cross-Book balancing entries are required, the currency of the Cross-Book balancing suspense accounts should be in the same currency[39] and must be either the debit or the credit side currency, but then the Booking Entity (accounting system) of the FX balancing suspense accounts must be managed within the same Booking Entity (accounting system)[40], and the same as the other side's original entry account.

Example – If the currency of the Cross-Book balancing suspense accounts (both) is the debit side currency then the Booking Entity (accounting system) of the FX balancing suspense accounts (both) needs to be the same as the Booking Entity of the credit side's account, and vice versa

b. When no FX balancing is required and Cross-Book balancing is – the Cross-Book balancing is limited to be performed either on the currencies of the original entries or in base currency – so the currencies of the Cross-Book suspense accounts can be either base currency or the currencies that each fit the currency of its original entry to be balanced.

c. When FX balancing is required but Cross-Book balancing is not – the FX balancing is not limited to be performed in specific booking entity/ies (accounting system/s) – so there is no limit and validation on the booking entity of the selected suspense accounts.

Note: The balancing entries (all types) always quote separately the Fee and the Principal (original). There will not be a balancing entry that would use a Net amount (Fee deducted/added to the Principal).

- **FX Balancing:**

Note: when validating against the Booking Entity of the balancing suspense accounts – if the **Booking_Ent** attribute is not set in this specific **Accounts** entry – **Banks.Def_Booking_Ent** should be used as the suspense account's ***Booking Entity***

– **FX Balancing Principal** - If ***Principal amount- FX balancing on credit CCY*** (**SA_FX_BAL_MAIN_DBTCCY**)/***Principal amount- FX balancing on debit CCY*** (**SA_FX_BAL_MAIN_CDTCCY**) Posting Types are selected as ***Original posting*** for this step (Accounting Model requires the creation of FX balancing entries for Principal amount entries)[41]:

› If the principal debit account and the principal credit account have different currencies, an FX conversion was performed on principal amount –

o If no ***Posting Suspense Account Selection*** rule with sub-type of at least one of the above, required, posting entries was caught

---

[39] Otherwise there would need to be a separate FX balancing pair of entries that would need to balance these cross-book balancing entries.

[40] Otherwise there would need to be a separate cross-book balancing pair of entries that would need to balance these FX balancing entries.

[41] These Posting Types will either be selected together or none selected.

Or

If the currencies of the selected suspense accounts do not exactly fit the currencies to be balanced (**Msg_Posting.Acc_Ccy** of the principal credit and debit entries to be balanced) – one account per currency

Or

(

If Cross-Book balancing – Principal is also required (***Principal amount- Cross-booking-systems balancing on credit side system*** (**SA_CRSS_BOOK_MAIN_CDT_SD**)/***Principal amount- Cross-booking-systems balancing on debit side system*** (**SA_CRSS_BOOK_MAIN_DBT_SD**) Posting Types were selected for this step)–

And

(

If the Booking Entity of the selected suspense accounts is not the same

Or

The Booking Entity of the selected FX balancing suspense accounts is on the same side (debit/credit) as the currency of the Cross-Book balancing suspense accounts – i.e. the Booking Entity of the selected FX balancing suspense accounts is the same as the Booking Entity of debit side principal entry and the currency of the Cross-Book balancing suspense accounts is also same as the debit side principal entry[42].

)

)

Setup problem –

- The posting process is stopped with a failure to post for this message.
- An error is logged in the **MsgErr** indicating a missing/wrong setup for suspense account of appropriate type (concatenation of a few, if a few are missing/wrong).
- None of the posting entries are saved for this message.
- The interface returns control to GPP Core with a Request Failure indication.
- The message falls to ***Repair*** queue.

o Else (validations did not fail) –

- If FX occurred on Principal Amount – The system creates a Debit entry for FX Balancing for Posting Principal Amount in credit side currency and a Credit entry for FX Balancing for Posting Principal Amount in debit side currency

- **FX balancing Fee** – If ***Fee amount- FX balancing on credit (PNL) CCY*** (**SA_FX_BAL_FEE_DBTCCY**)/***Fee amount- FX balancing on debit (Fee) CCY*** (**SA_FX_BAL_FEE_CDTCCY**) Posting Types are selected as ***Original posting*** for this step (Accounting Model requires the creation of FX balancing entries for Fee amount entries): [43]

› If the debit and credit entries for either side Fee Amount include accounts with different currencies – FX conversion was performed on fee amount

---

[42] This case (and the vice versa one) cannot be supported as we cannot pass the amount from debit currency to credit currency in the debit Booking Entity and then still use the debit currency to pass from the Booking Entity to the credit Booking entity

[43] These Posting Types will either be selected together or none selected.

- If no ***Posting Suspense Account Selection*** rule with sub-type of at least one of the above, required, posting entries was caught

  Or

  If the currencies of the selected suspense accounts do not exactly fit the currencies to be balanced (**Msg_Posting.Acc_Ccy** of the principal credit and debit entries to be balanced) – one account per currency

  Or

  (

  If Cross-Book balancing - Fee is also required (***Fee amount- Cross-booking-systems balancing on credit side system*** (**SA_CRSS_BOOK_FEE_CDT_SD**)/***Fee amount- Cross-booking-systems balancing on debit side system*** (**SA_CRSS_BOOK_FEE_DBT_SD**) Posting Types are selected for this step)

  And

  (

  If the Booking Entity of the selected suspense accounts is not the same

  Or

  The Booking Entity of the selected FX balancing suspense accounts is on the same side (debit/credit) as the currency of the Cross-Book balancing suspense accounts – i.e. the Booking Entity of the selected FX balancing suspense accounts is the same as the Booking Entity of debit side principal entry and the currency of the Cross-Book balancing suspense accounts is also same as the debit side principal entry.

)

)

  Setup problem –
  - The posting process is stopped with a failure to post for this message.
  - An error is logged in the **MsgErr** indicating a missing/wrong setup for suspense account of appropriate type (concatenation of a few, if a few are missing/wrong).
  - None of the posting entries are saved for this message.
  - The interface returns control to GPP Core with a Request Failure indication.
  - The message falls to ***Repair*** queue.
- Else (validations did not fail) –
  - If FX occurred on Fee Amount – The system creates a Debit entry for FX Balancing for Posting Fee Amount in credit side currency and a Credit entry for FX Balancing for Posting Fee Amount in debit side currency

- **Cross-Book Balancing:**
  - **Cross-Book Principal –** If ***Principal amount- Cross-booking-systems balancing on credit side system*** (**SA_CRSS_BOOK_MAIN_CDT_SD**)/***Principal amount- Cross-booking-systems balancing on debit side system*** (**SA_CRSS_BOOK_MAIN_DBT_SD**)

Posting Types are selected as *Original posting* for this step (Accounting Model requires the creation of cross book (accounting systems) balancing entries for Principal amount entries) [44]:

› If the principal debit account and principal credit accounts have different Booking Entities – principal amount was transferred from one Booking Entity to another –

    o If no *Posting Suspense Account Selection* rule with sub-type of at least one of the above, required, posting entries was caught

    Or

    If the Booking Entities of the selected suspense accounts do not exactly fit the Booking Entities to be balanced (as per the **Msg_Posting.Booking_Ent** of the relevant entries to be balanced) – one account per Booking Entity

    Or

        (

    If FX balancing - Principal is also required (*Principal amount- FX balancing on credit CCY* (**SA_FX_BAL_MAIN_DBTCCY**)/*Principal amount- FX balancing on debit CCY* (**SA_FX_BAL_MAIN_CDTCCY**) Posting Types were selected for this step)

    And

        (

    If the currency of the selected suspense accounts is not the same

    Or

    The Booking Entity of the selected FX balancing suspense accounts is on the same side (debit/credit) as the currency of the Cross-Book balancing suspense accounts – i.e. the Booking Entity of the selected FX balancing suspense accounts is the same as the Booking Entity of debit side principal entry and the currency of the Cross-Book balancing suspense accounts is also same as the debit side principal entry.[45]

)

)

    Or

     (

    If FX balancing - Principal is **not** required (**SA_FX_BAL_MAIN_DBTCCY'** / '**SA_FX_BAL_MAIN_CDTCCY'** Posting Types were not selected for this step)

    And

    The currency of the selected suspense account on credit side is neither base currency nor fits the currency of the original credit entry, or same no fit on the debit side[46] (as per the **Msg_Posting.Acc_Ccy** of the relevant entries to be balanced)

)

---

[44] These Posting Types will either be selected together or none selected.

[45] This case (and the vice versa one) cannot be supported as we cannot pass the amount from debit currency to credit currency in the debit Booking Entity and then still use the debit currency to pass from the Booking Entity to the credit Booking entity

[46] When no FX balancing is required – the Cross-Book balancing is limited to be performed either on the currencies of the original entries or in base currency

Setup problem –

- The posting process is stopped with a failure to post for this message.

- An error is logged in the **MsgErr** indicating a missing/wrong setup for suspense account of appropriate type (concatenation of both, if both are missing/wrong).

- None of the posting entries are saved for this message.

- The interface returns control to GPP Core with a Request Failure indication.

- The message falls to *Repair* queue.

- Else (validations did not fail) –

- The system creates a Debit entry for Cross-Booking-Systems for the Principal Amount on the Credit side system and a Credit entry for Cross-Booking-Systems for the Principal Amount on the Debit side system.

- **Cross-Book Fee Amount** – If *Fee amount- Cross-booking-systems balancing on credit side system* (**SA_CRSS_BOOK_FEE_CDT_SD**)/*Fee amount- Cross-booking-systems balancing on debit side system* (**SA_CRSS_BOOK_FEE_DBT_SD**) Posting Types are selected as *Original posting* for this step (Accounting Model requires the creation of cross book (accounting systems) balancing entries for Fee amount entries)[47]:

  › If the debit and credit entries for either side Fee Amount include accounts with different Booking Entities – principal amount was transferred from one Booking Entity to another –

  - If no *Posting Suspense Account Selection* rule with sub-type of at least one of the above, required, posting entries was caught

    Or

    If the Booking Entities of the selected suspense accounts do not exactly fit the Booking Entities to be balanced (as per the **Msg_Posting.Booking_Ent** of the relevant entries to be balanced) – one account per Booking Entity

    Or

    (

    If FX balancing - Fee is also required (*Fee amount- FX balancing on credit (PNL) CCY* (**SA_FX_BAL_FEE_DBTCCY**)/*Fee amount- FX balancing on debit (Fee) CCY* (**SA_FX_BAL_FEE_CDTCCY**) Posting Types were selected for this step)

    And

    (

    If the currency of the selected suspense accounts is not the same

    Or

    The Booking Entity of the selected FX balancing suspense accounts is on the same side (debit/credit) as the currency of the Cross-Book balancing suspense accounts – i.e. the Booking Entity of the selected FX balancing suspense accounts is the same as the Booking Entity of debit side principal entry and the currency of the Cross-Book balancing suspense accounts is also same as the debit side principal entry

)

---

[47] These Posting Types will either be selected together or none selected.

)

Or

(

If FX balancing - Principal is **not** required
(**SA_FX_BAL_FEE_DBTCCY/SA_FX_BAL_FEE_CDTCCY** Posting Types were
not selected for this step)

And

The currency of the selected suspense account on credit side is neither base
currency nor fits the currency of the original credit entry, or same no fit on the
debit side[48] (as per the **Msg_Posting.Acc_Ccy** of the relevant entries to be
balanced)

)

Setup problem –

- The posting process is stopped with a failure to post for this message.

- An error is logged in the **MsgErr** indicating a missing/wrong setup for suspense
  account of appropriate type (concatenation of both, if both are missing/wrong).

- None of the posting entries are saved for this message.

- The interface returns control to GPP Core with a Request Failure indication.

- The message falls to *Repair* queue.

o Else (validations did not fail) –

- The system creates a Debit entry for Cross-Booking-Systems for the Fee Amount
  on the Credit side system and a Credit entry for Cross-Booking-Systems for the
  Fee Amount on the Debit side system

Note: The following logic is performed to deal with a case of *Repair* after successful posting (full or
partial) and re-run of the message through the full flow.

- After the creation of the posting entries in memory – for each posting entry the system checks if
  an entry with this **Posting_Type** already exists in the **Msg_Posting** table for the current message

  – If the **Posting_Type** is for either of the fee amount types and the *Fee_Type_Group* is
    different between the entry in DB and the entry in memory[49] –

    › Exit this logic

  – Else –

    › If it exists in DB and its **Posting_Sts** is any of the negative response statuses - N, L, E,
      R, O –

      o If the current entry in memory is set with **Is_Reversal_Ind** and the entry in DB is not[50]
        –

---

[48] When no FX balancing is required – the Cross-Book balancing is limited to be performed either on the currencies of the original entries or in
base currency

[49] This is a case of a new fee that should be added on top of the previous ones and is not a new version of the previous Fee/PNL entry re-
computed in same step as old one as message runs again through the full flow

[50] This case would probably not happen as *Posting Type Selection* rules should not select this Posting Type in Reverse mode

- - The entry in memory is redundant and is deleted from posting entries array, as there is no need to reverse a failed entry
  - o Else (both are original posting or both are reversed entries)
    - The system deletes the entry from the DB[51] (**Msg_Posting** and **Msg_Posting_Ext**)
    - If the related entry of the DB entry is with **Posting_Sts**=B and the **Interface_Name** is empty[52] the system deletes this entry as well (**Msg_Posting** and **Msg_Posting_Ext**)
- › If it exists in DB and its **Posting_Sts** is Successful – S –
  - o If the current entry in memory is set with **Is_Reversal_Ind** and the entry in DB is not[53] –
    - Exit this logic
  - o Else (both are original posting or both are reversed entries) [54]–
    - The system compares all attributes of the entry from the DB and the entry in memory
    - If exactly the same[55]–

      The system deletes the entry in memory (as entry in DB is still correct)
  - o Else (at least one attribute is different[56]) –
    - If both are original posting entries –

      The system creates a new reversed entry from the entry in DB and adds it to the posting entries array in memory to be posted in this step.

      If the related entry of the DB entry does not exist in the current array in memory (will happen in a case first run undergone 2-step-accounting (when message was about to fall to *Repair*), and second run is done in 1-step-accounting[57] –

      The system also creates a new reversed entry from this related entry in DB as well.

    - Else – both are reversed posting entries –

      The message is routed to *Posting Exception* (**POSTEX**) queue with an appropriate error logged in **MsgErr**.

---

[51] It is not yet registered in the bank accounting system and the newly created entry would replace it in DB once current process will complete the posting request generation

[52] A case the related posting entry was not sent out

[53] This is the case in which current entry in memory is the reversal of the entry in DB. Both are valid and need to remain

[54] This is the case we need to take care of to select the entry that should remain – one in DB or one in memory

[55] Nothing was changed in the message to cause a need to re-post this entry

[56] Probably due to the repair the user has performed, or the time that past since previous posting of the entry

[57] Current entry is either of the MAIN_DR/MAIN_CR or PNL_DR/PNL_CR, and the related entry in DB is a DR/CR to an SA while the related in memory is the normal MAIN_CR/MAIN_DR (with their variations)

2.3 Customer Specific Logic

After all **Msg_Posting** entries are created – the interface invokes a function in which specific logic can be incorporated[58]. Whenever such logic is not required this function will be empty.

2.4 Posting Request Formatting and Transmission

Grouping into posting request/s

For each posting invocation step the system can generate one request that includes all the posting entries created in this step, or multiple requests, each grouping posting entries of specific Posting Types. Another option allows not including some of the created entries in any posting request[59].

The above is achieved, as per bank's requirements, using one of the parameters of the ***Posting Type Selection*** rules (sub-action).

Routing

Another type of variation allows the usage of different Interface entries for specific scenarios (using configuration via the Interface Selection rules) to allow usage of different queues/destinations[60], or even different structures.

Structure

GPP can use the Fndt message proprietary format (this format is described in GPP Technical Guide Fndt Message Format) or a customer specific format, as required by the bank.

If the Fndt message proprietary format is used, it can be configured to be used as Full (the full Fndt message structure with all the payment attributes). It can also be configured to exclude some parts of its structure and as minimum to include only the section with the posting entries information. This configuration is performed as per the bank's requirements.

Content

Using either format options, the information that is passed in the Posting Request commonly includes some general information from the transaction message, and a section with multiple occurrences, one per each posting entry, that each includes the attributes of one posting instruction – including (if supported by the request structure) posting entry type, account number, account currency, account booking entity, Cr/Dr indicator, posting amount, posting value date, Force Post indicator[61], Reversal indicator[62], and if reversed – the ID of the reversed original entry.

Additional attributes required by a specific accounting system but not included in the standard **Msg_Posting** entry in GPP can be added using the Customer Specific code and an additional extension for bank specific information (**Msg_Posting_Ext**).

---

58 Logic that was identified as really specific, and therefore was not generalized to be used as part of the Standard logic.

59 In some cases not all entries are required to be sent real-time during the payment processing. These may be gathered later on a separate report prepared as part of EOD tasks.

60 Distribution to different queues to allow Host system to prioritize its handling of such requests per payment types/scenarios or to allow sending different requests to different destinations (e.g. different accounting systems)

61 This attribute allows passing to the Host system the indication that a user requested (in advance) to not fail this posting.

62 This attribute allows passing to the Host system the indication that this posting is a reversed one (a reverse of another that was sent before for this message that since then has changed/been canceled etc.).

Transmission aspects

As per posting interface configuration, if the interface is asynchronous, the payment is set to *Wait Posting* status and the appropriate **Msg_Posting.Posting_Sts** and message level posting monitors are set with the value P – for Pending response.

If the interface is synchronous, an immediate response is returned.

The interface also can be configured as Publish (message does not wait for the response and continues processing). When invoked from the termination flow, for the model of dynamically splitting the posting into 2 steps upon stopping the message, the interface is configured not to wait for a response and continue routing the message to its instructed status. If the interface is asynchronous he appropriate **Msg_Posting.Posting_Sts** and message level posting monitors are still set with the value P – for Pending response, and when a response is received it is handled and the appropriate **Posting_Sts** and monitors are set accordingly, but message status is not affected by such response.

Note: The configuration of *Posting Type selection* rules for the final leg, in this model, needs to make sure that if 1st leg entries failed – the full list of posting needs to be created, and if 1st leg posting was successful – need to only create the 2nd leg entries at that point.

Note: Mix of an asynchronous on first step of the 2-step-accounting model and Publish in the second step can be supported, by configuring as two different Interface entries with different behavior.

If the interface is down (for example, the interface status is inactive), the payment is held in the inactive status queue that is defined under the posting interface.

2.5 Posting Response Handling

Structure

As the posting request, GPP can be configured to use as the posting response structure, the Fndt message proprietary format, full or limited, or a customer specific format as required by each customer.

Processing -

- When a Posting Response is received:
  - The system performs the standard validations for an incoming response
  - If the response cannot be parsed – and hence the system could not identify for sure to which message it belongs – or cannot find the matching request
    - › The system only logs the error in the general *System Log* (**ErrorLog**) table
    - › The message, in this case, remains in the *Wait Posting* status on the payment level and also the different posting occurrences status level remain waiting for a response (P)[63]

---

[63] A user will have the option to use either the *Release - As Positive Posting* or the *Release – As Negative Posting* buttons to mark the posting entries accordingly, and allow the continuation of the flow as if a response was received (either).

– If Posting response was received, parsed and a match was found or in the case a user just clicked either of the release as positive/negative posting buttons in a manual queue[64]:

  › If the user instructed posting response action was already performed on this message for the appropriate step (as the appropriate 1st or 2nd ***Posting response manually instructed*** (**MU_MAN_1ST_PSTNG_RSPNS/ MU_MAN_2ND_PSTNG_RSPNS**) indicates) [65] :

    o An appropriate Follow-up (**P_FOLLOW_UP** = **LATE_POST_RSPNS_AFTR_MAN_INSTR** (should be setup as user code) flag is set on the transaction

    o If the status returned in the Posting response is successful[66], either from the one return code, or specifically from the tag returned with the status of each entry (F_POSTING_INFO_STATUS if using the Fndt Message structure), and only if manually instructed status was also successful[67] (relevant **Msg_Posting.Posting_Sts**=S) –

      ▪ The system compares each **Msg_Posting.Value_Date** with posting date from the related entry in the Response, and if different – updates the **Msg_Posting.Value_Date** with posting date from the Response.

  › Else (no user instructed posting response action was performed on this message)

    o The system updates each of the relevant Msg_Posting entries and the relevant Posting Monitors (MF_<posting type>_POSTING_STS), either from the one return code, or specifically from the tag returned with the status of each entry (F_POSTING_INFO_STATUS if using the Fndt Message structure):

Notes:

It is under the scope of each Posting Response Handler to use the generic GPP functionality of translation from the interface/bank system specific values into the generic GPP values using the **Error_Mapping** table[68]

If using the Fndt Message as the structure of the Posting interface (Response) – GPP supports the response code location either within the dedicated <ResponseDetails> sub-tree or from the <**TxInfAndSts**> sub-tree (using the ISO success/failure codes) within a pacs.002, returned as the <**Pmnt**> content, within this Fndt Message response.

      ▪ If it is a positive Response – **Msg_Posting.Posting_Sts** updated to S (for Successful).

      ▪ If it is a negative Response – **Msg_Posting.Posting_Sts** updated to either of the following depending on the type of negative response -

      N (for NSF - insufficient funds posting response from Host)

      E (for Processing/Technical error – occurred either at the Host or within the integration layer in between GPP and the Host systems, when applicable)

---

[64] ***Release - As Positive Posting*** (emulating a positive posting Response) or ***Release - As* Negative Posting** (emulating a negative posting Response)

[65] If such an instructed posting action was performed, we need to mark this message for additional investigation, as we are looking at a late response after manual resolution

[66] Only for successful posting the value date needs to be

[67] The Response re-assures the positive status of these entries

[68] GPP mechanism for translating between internal and external codes

L (for posting limitations/restrictions response from Host)

O (for Optional specific other Reject that the bank wants to specifically identify from other reject reasons for special handling, when applicable)

R (for Reject posting response from Host (all other failures that are not NSF, technical or specifically, posting limitations/restrictions or marked as O for specific handling)

- ▪ If the status returned in the Posting response is successful, either from the one return code, or specifically from the tag returned with the status of each entry (F_POSTING_INFO_STATUS if using the Fndt Message structure) -

  - • The system compares each **Msg_Posting.Value_Date** with posting date from the related entry in the Response, and if different – updates the **Msg_Posting.Value_Date** with posting date from the Response
- o The system checks if this is the last Response for which there are Msg_Posting entries with **Posting_Sts** as P. if it is the last

  - ▪ The system sets the Posting Request Response status (**MI_POSTING_REQ_RSPNS_STS**) is set to F[69]

  - ▪ If the *Follow up* (**P_FOLLOWUP**) attribute of the message is set with appropriate code for "Timeout - no response"[70] - this attribute will be cleared to its 'Alert off' code (to be taken out of the custom queue view)

Note: Through the Business setup, it is possible to setup a custom queue to identify transactions for which no response was received[71] (criteria: Posting Request Response status (**MI_POSTING_REQ_RSPNS_STS**)= P[72]), and trigger an alert on this custom queue (filter).

- o The system also checks if the related posting entry was included in a specific Request and sent out

  - ▪ If it was not (it was configured not to be included in any request) –

    The System makes the same updates as per the above sent and responded entry to this related one – status, values date.

Notes:

- • In the case there are two related Posting Types – *Principle credit amount* (**MAIN_CR**) and *Fees on credit side customer* (**FEE_CR**)[73], that may have been sent in two separate Requests, and both are related to the one *Amount in profit P&L account for fees on credit side customer* (**PNL_CR**) entry – there is no way other than to define a hard coded priority as to which Response should be the one to take the status and date from. In this case – this should be the one that was received for the **MAIN_CR**. Same as in the case of a **PNL_CR** entry being consolidated from the multiple Fee entries, while the **FEE_CR** entries are not – the hard coded priority to use the response of the first **FEE_CR** entry for the update of the not-sent **PNL_CR** entry.

---

[69] Indicating all Responses was received and posting step (for this step) is finalized.

[70] Meaning a Timeout already occurred for the relevant Request, but in this case the user did not already initiate any manual response emulation

[71] For cases the message continues processing and does not wait for the response in *Wait Posting* (**MP_WAIT**) queue (e.g. Offline mode, 2nd step performed as Publish (not waiting for the response))

[72] Posting response was not yet received and is still pending

[73] A case when some of the fees were defined as *Deduct from Payment* and some as not

- There are no validations in GPP on the value date provided in the Response. It is the responsibility of the accounting system to provide a correct and comprehensive value, if wishing to override the provided date in the Request.

  - If the message status is different than the List of Sts For Posting Response[74]

    - If the message is already in Complete or Canceled[75] –

      If the response was positive – no additional step is required.

      If the response was negative and message was processed during Offline mode – an appropriate Follow-up (**P_FOLLOW_UP** = FAILED_OFFLINE_POSTING (should be setup as user code) flag will be set on the transaction.

Note: Through the Business setup, it is possible to setup a custom queue for these transactions (criteria: *Follow up flag* = **FAILED_OFFLINE_POSTING**, Posting state or cancelation posting state) = Offline or Prepare for Online, Message Status in (Complete, Canceled), Transaction category/Product is CTI or DDI), and trigger an alert on this custom queue (filter).

    - Else, if the status is in the List of Sts For Posting Response (message was waiting for the Response in another valid queue)

      If no **Msg_Posting** entries for this message are still in P **Posting_Sts**[76] – it is sent back to continue the processing, while the flow is selected based on the response status.

      If positive response the –

      Continues the processing flow taking into account, when deciding on the flow, a linked Cancelation message, if exists[77].

      If negative response – as per the below specification (as per specific interface capabilities – can use different failure cases that dictates different behaviors).

      For the code that represents No sufficient funds (**Msg_Posting.Posting Sts** is marked with N).

      The message is routed to *Insufficient Funds* (**NSF**) queue with the appropriate error logged in **MsgErr**.

      For a code that represents Posting limitations/restrictions (**Msg_Posting.Posting Sts** marked with L).

---

[74] A variable that can be set differently in each implementation – in current implementation - *Wait Posting* (**MP_WAIT**) or *Posting Exception* (**POSTEX**)

[75] The case the message is already in Complete can occur in cases the original posting step was performed with no stopping of the message processing, and the case it is already in Canceled can occur if cancelation was performed with no stopping of the message processing – e.g. on Non-Active state of the Posting interface – Offline mode, or optimistic mode

[76] A case the posting entries of current step were sent in a few requests, and current response is not the last one expected.

[77] A case a Cancelation request was received on the original message when it was still waiting for posting response. In this case – the original message needs to be Canceled, instead of continuing on the positive flow.

The response details are translated into Stop Flags values and the appropriate fields in the message are updated with these values (as per Stop Flags functionality).

The message is routed to *Posting Restrictions* (**POSTREST**) queue with the appropriate error logged in **MsgErr**.

For a code that represents Other NAK reasons for special marking (**Msg_Posting.Posting Sts** marked with O) –

The system will invoke a customer specific code/rule to separate this case from the other failures, with the appropriate error logged in **MsgErr**.

For any other code, that represents a generic failure, the message is routed to the *Posting Exception* (**POSTEX**) queue[78]. If failure response received from a 'real' response (is not the result of a user clicking the either of the release buttons) an appropriate error logged in **MsgErr**.

2.6 Retry Posting Invocation

- Following are the cases in which a retry (not the automatic one done via the interface timeout functionality) can be invoked for the posting –

  - User clicked the *Force Ins. Funds* or *Retry Ins. Funds* buttons in the *Insufficient Funds* (**NSF**) queue or the *Override* button in *Posting Restrictions* (**POSTREST**) queue[79] –

    › For all the relevant entries marked with **Msg_Posting.Posting_Sts**=N[80] or L[81] respectively

      o Set Msg_Posting.Posting_Sts and their corresponding posting monitors (**MF_..._POSTING_STS**) to T[82]

      o If *Monitor user force from NSF* (**MU_NSF_FORCE_STS**)=1 or *Override stop flags* (**MU_STOP_FLAGS_OVERRIDE_STS**)=1 – set **Msg_Posting.Force_Post_Ind** to 1

    › Send the appropriate Posting Request (for the entries that were marked with **Msg_Posting.Posting_Sts**=N or L respectively)

  - User clicked the *Resend* button in the *Processing Communications* page for a specific Posting Request line –

    › If this interface is set to allow retry and the message status is in the allowed statuses for retry list –

---

[78] In case a specific other behavior is required – such as in Immediate flows –

- Sending the message to Rejected when needs to stand in SLA – an automatic routing from this queue to the Rejected queue would be achieved using Message Workflow Determination - STP (15) rule

- Sending the message to Repair - in cases there is no need to stand in SLA and manual repair is allowed – an automatic routing from this queue to the Repair queue would be achieved using another Message Workflow Determination - STP (15) rule. Same solution would be applied when message need to go to Rejected (in case need to stand in SLA).

[79] For an overidable Stop Flag

[80] For negative posting response of type Insufficient Funds

[81] For negative posting response of type Posting Limitations/ restrictions

[82] For Retry

o   No update of any monitor or posting status upon request sending[83]

o   **Message_External_Ineraction** entry is copied, marked as resend and file from the **Message_External_Ineraction.Interface_Content** is resent

Note: From the *Insufficient Funds* (**NSF**) queue, automatic attempts are made to re-post the payment at predefined time intervals. If the response remains Insufficient Funds, there is no need to audit each attempt as a new error because the previous error remains.

2.7 Reverse Posting

If any account, amount, value date, processing date has changed or the transaction has been canceled/rejected after posting step was performed, the posting must be reversed. This is done by creating reverse entries (for the entries that need to be reversed) with a Reverse indication to the posting interface.

These entries are also created using a pre-defined configuration of the *Posting Type Selection* rules on negative flows and just before adding new posting entries[84] details).

2.8 Bulk Posting

The majority of the functionality for bulk posting is actually within the way GPP processes bulk files. As GPP creates a message that represents the different files or posting related consolidation units (S for Incoming file and A for Outgoing file), and these are processed to derive the appropriate Batch suspense account as their debit or credit account (depending on the type of the file) – the posting logic just treats these messages as any other message and no specific posting logic is required, apart from the specific invocation points of the Posting step within the processing flows of these messages.

3   Manual Handling

Manual handling may be required due to the following events:

- Insufficient funds in the debit account – The message is routed to the *Insufficient Funds* (**NSF**) queue. User can manually force the check (*Force Ins. Funds*) – to initiate another request for same entries but with *Force Post flag* set; try to post again (*Retry Ins. Funds*) – to try reprocessing the message again, eventually invoking another request; or wait for the periodic automatic retry. (See section Insufficient Funds for more details).

Note: When a separate balance inquiry (**BI**) is used and the message falls to the *Insufficient Funds* (**NSF**) queue and is forced by the operator, the posting is done with an indication to the interface to skip the balance inquiry check and continue posting. The posting interface must ensure that the posting is performed even without sufficient funds. Otherwise, the payment is routed to the **NSF** queue again.

- Posting restrictions/limitations of the account – The message is routed to the *Posting Restrictions* (**POSTREST**) queue. The User can manually override (*Override*) the Stop Flag (if overridable) – to initiate another request for same entries but with *Force Post flag* set; try to post again (*Retry Post. Rest.*) – to try reprocessing the message again, eventually invoking another request; (see section Posting Restriction for more details).

---

[83] Once a response is received – it will invoke the statuses update logic (may or may not include the message processing logic within the response handling – in final statuses – it will not)

[84] In case new entry have same Posting Type as successfully posted existing one, both are either Original and new one differs at least in one of the attribute of the posting entry (the message was changed from time it was posted)

- Error in the posting system – The message is routed to the **Posting Exception** (**POSTEX**) queue. User can manually emulate a posting response (**Release – As Positive Posting**/**Release – As Negative Posting**) – to release the message to continue the flow (as per response type). (See section Posting Exception for more details).

- In the case no response is received from the posting system and the message is still in the **Wait Posting** (**MP_WAIT**) queue – user can manually emulate a posting response (**Release – As Positive Posting**/**Release – As Negative Posting**) – to release the message to continue the flow (as per response type). (See section Wait Posting for more details).

Note: The **Release – As Positive Posting**/**Release – As Negative Posting** buttons may also be available in additional (final) statuses[85] based on specific configuration per specific sites.

---

[85] Will be used if posting interface will be defined with Publish transmission mode (message does not wait for the response)

4 Static Data

The following section defines the business guide building blocks that are used in setting up the Posting Service. These building blocks include:

- Business Setup
  - System Parameters
  - Profiles
  - Business Rules
- System Setup
  - Profiles
  - System Rules
- Message Attributes
- Statuses
- Access Class Entitlements

Note: Previous Accounting logic allowed configuring of different Accounting behavior per different MOPs for Outgoing and Incoming single transactions, to perform posting at an earlier stage, if a payment is received from or sent to RTGS (parameters for configuring this feature are in the Accounting pane of the Method of Payments profile). Also, this previous logic was based on configuration in the system parameters - **SEND_BEFORE_POSTING** and **ACCBYMOP**.

This differentiation in the new design is implemented using the ***Posting Type Selection*** system rules and allows more flexibility.

The two system parameters and some GUI attributes under the ***Accounting*** title (***Outgoing*** and ***Incoming***, and the **Clearing Suspense account**), in the ***Processing*** tab of the ***MOP*** profile, are still in view, but implementations that use the new Accounting logic do not use them.

These will be removed from the GUI once there are no more banks using the previous logic.

4.1 Business Setup

4.1.1 System Parameters

System Parameters

Name	Description	Default
SEND_BEFORE_POSTING	**NOT IN USE IN NEW ACCOUNTING LOGIC** Specifies whether to send the message before posting it or to post the message before sending it.	No
ACCBYMOP	**NOT IN USE IN NEW ACCOUNTING LOGIC** Specifies whether accounting generation is based on system parameter SEND_BEFORE_POSTING or the values in the Outgoing and Incoming fields of the MOP profile.	Yes
ASAP_POST_REST_CHECK	Specifies whether to perform the stop flags check as soon as possible. Values: - Yes - stop flags check is initially performed for debit and credit sides separately as soon as the relevant account is loaded, either from GPP or via Feeder Request. The final	No

Name	Description	Default
	check for both sides is also performed, in this case, on the later point in the flow (as performed if the value is No). No - consolidated stop flags check is performed for both sides later in the flow, after MOP selection, STP override rules, Fee calculation and Time Hold rules, but before sending future dated transactions to Scheduled queue. Default	
MAXNSFRETRY	Defines the maximum count for insufficient funds check retries. After which an automated retry is not performed.	-1
NSF_CHECK_FREQ	Specifies the frequency, in milliseconds, that transaction in the Insufficient Funds (NSF) status re-checked. The frequency should not be less than the NSF task frequency as depicted in the EVENT_DEFINITION.FREQUENCY_IN_MINUTES column. Example: 300,000 milliseconds equal to 5 minutes	

4.1.2 Profiles

4.1.2.1 System Users

Users profile - Posting related parameters

Field Name	Description	Format
Force limit – insufficient funds	Maximum amount in base currency that the user can "force pay" from an account that has insufficient funds to cover the payment.	[AMOUT]{0..19}

4.1.2.2 MOP

MOP profile - Posting related parameters

Field Name	Description	Format
Clearing suspense account: Office	**NOT IN USE IN NEW ACCOUNTING LOGIC** The clearing suspense account office details: • Account office • Account number • Account currency – Auto-populated for the account selected • Account owner – Concatenates and displays the BIC and account owner's name. Read Only.	Optional selection from list or Data Search results
Return of funds account: Office	The return of funds account details– used whenever a Return message is received and cannot directly find the customer's account: • Account office • Account number	Optional selection from list or Data Search results

Field Name	Description	Format
	• Account currency – Auto-populated for the account selected • Account owner – Concatenates and displays the BIC and account owner's name. Read Only.	
Settlement accounting	Specifies how settlement accounting is to be executed for the MOP. Values: • None – No settlement accounting required • Payment – Settlement accounting is executed for each payment • Net settlement – Settlement accounting is executed only if triggered by a net settlement message The Outgoing and Incoming fields are enabled only if the value of Settlement accounting is set to **Payment**	Mandatory selection from drop-down list
Outgoing	**NOT IN USE IN NEW ACCOUNTING LOGIC** Specifies at what point in the processing flow settlement accounting is performed for a payment whose credit MOP is set to this MOP. Values: • Before send – settlement accounting on the payment will be executed before its transmission • After confirmation – settlement accounting on the payment will be executed following receipt of final confirmation	Mandatory selection from drop-down list if **Settlement accounting** is set to **Payment**
Incoming	**NOT IN USE IN NEW ACCOUNTING LOGIC** Specifies at what point in the processing flow settlement accounting is performed for a payment whose debit MOP is set to this MOP. Values: • ASAP – Settlement accounting for the payment occurs after account and debit MOP determination • Before finalization – Settlement accounting for the payment occurs when payment processing is completed	Mandatory selection from drop-down list if **Settlement accounting** is set to **Payment**

4.1.2.3 Office

Office profile - Posting related parameters

Field Name	Description	Format
Default booking entity	The default booking/Accounting system for accounts in the system (used if no Booking entity is set per each Accounts entry).	Optional selection from drop-down list
Default behavior for posting fees		
Consolidated fee posting entries	Default on the Office level to be used if not set at Parties level. Indicates whether fee posting entries (on customer side) should be consolidated, in case multiple fees selected per transaction. Available values are: Yes- to one – for consolidation to one entry (**ONE**)	Optional selection from drop-down list

Field Name	Description	Format
	Yes- per fee type ctgy – for consolidation per fee type category (**FEE_TP_CTGY**) No – for no consolidation (**NO**) empty If no selection in Parties profile – default needs to be defined	
Consolidated P&L posting entries	Default on the Office level to be used if not set at PNL Accounts level. Indicates whether P&L posting entries (to P&L side) should be consolidated. Available values are: Yes- per P&L account – for consolidation per P&L account (**PNL_ACC**) Yes- per fee type ctgy – for consolidation per fee type category (**FEE_TP_CTGY**) No – for no consolidation (**NO**) empty If no selection in Accounts profile – default needs to be defined	Optional selection from drop-down list
Include posting entry for 0 amount fee	An indication (if set to 1) that posting entry should be created for a fee of amount 0 (default GPP behavior does not create posting entry for an amount of 0)	Check box

4.1.2.4   Parties

Parties profile - Posting related parameters

Field Name	Description	Format
Consolidated fee posting entries	Indicates whether fee posting entries (on customer side) should be consolidated, in case multiple fees selected per transaction. Available values are: • Yes- to one – for consolidation to one entry (**ONE**) • Yes- per fee type ctgy – for consolidation per fee type category (**FEE_TP_CTGY**) • No – for no consolidation (**NO**) • empty If no selection in Parties profile – system will use default from Offices profile	Optional selection from drop-down list

4.1.2.5   Accounts

Accounts profile - Posting related parameters

Field Name	Description	Format
Booking Entity	The relevant Booking Entity under which postings for this account are managed (available values are from the *User Codes* for *Type* BOOKING_ENT. Should be previously defined)	Optional selection from

Field Name	Description	Format
	If customer site includes multiple accounting/booking systems - each Accounts entry managed in GPP must be set up with a value in this field	drop-down list
Consolidated P&L posting entries	Indicates whether P&L posting entries (to P&L side) should be consolidated. Available values are: • Yes- per P&L account – for consolidation per P&L account (**PNL_ACC**) • Yes- per fee type ctgy – for consolidation per fee type category (**FEE_TP_CTGY**) • No – for no consolidation (**NO**) • empty If no selection in Accounts profile – system will use default from Offices profile	Optional selection from drop-down list

4.1.2.6 Account Types

Account Types profile - Posting related parameters

Field Name	Description	Format
Customer account indicator	An indication that the account type is a type for customer accounts (copied to the **Msg_Posting** entry relevant for the posting account as per its type)	Check box

4.1.2.7 Fee Type

Fee Type profile - Posting related parameters

Field Name	Description	Format
Fee type category	A categorization code that may be required to be sent per fee to an accounting system, and on which fee posting entries can be consolidated	Optional selection from drop-down list (Populated with values from **Fields_Values** whith **Field_Type**='FEE_TYPE_CTGRY')

4.1.2.8 Fee Formula

Fee Formula profile - Posting related parameters

Field Name	Description	Format
Fee transaction code	A categorization code that may be required to be sent per fee to an accounting system	Optional selection from drop-down list. Possible values defined in the Transaction Codes profile

4.1.2.9 User Codes

New entries need to be defined in this profile for each booking entity/accounting system in the system. Use the *Type* of BOOKING_ENT to define these codes[86].

4.1.3 Business Rules

A rule consists of a set of conditions and an action. The conditions refer to attributes of the transaction or other data in the system. The action determines what should be done if the conditions are met.

---

[86] These codes will then be available for selection as the Booking Entity in the Accounts profile

Rules must be attached to an object (such as a processing office or customer) before they can become effective. For example, Fee Formula Selection rules are attached to an Office profile. If more than one rule is attached to an object, the rules are evaluated in the order that they are listed.

4.1.3.1   Posting Suspense Account Selection

Description: The *Posting Suspense Account Selection* rule type is a business rule type that is dedicated to the posting logic, and it allows the selection of different suspense accounts to be used for the different Posting Types required for posting. A rule of this rule type determines which suspense accounts are to be used for each Posting Type entry.

The rule type has sub-types that fit the Posting Type codes (as configured using the system rules - *Posting Types Selection*) and that each represents a posting to a suspense account of a specific type (sub-list of the codes defined in **Fields_Values** (**Field_Type**=POSTING_TYPE – the ones that have an SA_ prefix).

Specific guidance in regards to suspense accounts used for Cross-Book and FX balancing entries:

In some banks the currency balancing accounts are only in one system, while in other banks, the cross-book balancing accounts are only in the base currency. This means that the decision on which currency (before or after the conversion) to do the Cross-Book entries is dependent on bank specific conditions. To support such deviation, the responsibility to properly select the suspense accounts to be used for the balancing entries is the bank (through the *Posting Suspense Account Selection* rules).

Once such suspense accounts are selected, the order of transferring the relevant amounts through these two types of balancing entries is based on their currency and the Booking Entity attributes.

The following are some guidelines regarding the valid combinations:

- FX Balancing - The suspense accounts selected (using the *Posting Suspense Account Selection* rules) to be used for the FX balancing entries should be one in the credit currency and another in the debit currency of the posting entries they are to balance.

- Cross-Book Balancing - The suspense accounts selected (using the *Posting Suspense Account Selection* rules) to be used for the Cross-Book balancing entries should be one managed in the credit *Booking Entity* (accounting system) and another in the debit *Booking Entity* (accounting system) of the posting entries they mean to balance.

- In addition -

    - If for either the principal or the fee amounts, both FX and Cross-Book balancing entries are required, the Cross-Book balancing suspense accounts currency should be in the same currency[87] and needs to be either the debit or the credit side currency, but then the *Booking Entity* (**accounting system**) of the FX balancing suspense accounts needs to be managed within the same Booking Entity (accounting system)[88], and the same as the other side's original entry account.

    Example: If the currency of the Cross-Book balancing suspense accounts (both) is the debit side's currency, then the *Booking Entity* (accounting system) of the FX balancing suspense accounts (both) needs to be the same as the Booking Entity of the credit side's account and vice versa.

    - When no FX balancing is required and Cross-Book balancing is – the Cross-Book balancing is limited to be performed either on the currencies of the original entries or in base currency –

---

[87] Otherwise there would need to be a separate FX balancing pair of entries that would need to balance these cross-book balancing entries.

[88] Otherwise there would need to be a separate cross-book balancing pair of entries that would need to balance these FX balancing entries.

so the currencies of the Cross-Book suspense accounts can be either base currency or the currencies that each fit the currency of its original entry to be balanced.

 - When FX balancing is required but Cross-Book balancing is not – the FX balancing is not limited to be performed in specific booking entity/ies (accounting system/s) – so there is no limit and validation on the booking entity of the selected suspense accounts.

Rule conditions: The bank will be able to set the conditions of this rule (probably mainly on the currency and Booking Entity of the credit/debit account or the currency of the fee account, but can also be on different message attributes), to allow selection of a specific suspense account to be used for the posting.

Note: When setting up the *Posting Suspense Account Selection* rules for FX balancing it is recommended to use the *Ccy of cdt posting entry to balance* (**D_CDT_POSTNTRY_TOBAL _CCY**) and *Ccy of dbt posting entry to balance* (**D_DBT_POSTNTRY_TOBAL _CCY**) to determine the accounts in the correct currency as per the entries that should be balanced.

When setting up the *Posting Suspense Account Selection* rules for Cross-Book balancing it is recommended to use the *Booking entity of cdt posting entry to balance* (**D_CDT_POSTNTRY_TOBAL _BOOKENT**) and *Booking entity of dbt posting entry to balance* (**D_DBT_POSTNTRY_TOBAL _BOOKENT**) to determine the accounts in the correct booking entity (accounting system) as per the entries that should be balanced.

These recommendations are usually necessary for the balancing of fee amount entries, as no other field can provide the currency and booking entity of the accounts involved in fee transfer.

Rule Action: An Account in the system.

Rule Attachment: *Posting Suspense Account Selection* rules are attached to the local office. Multiple rules may be attached. First matching rule applies.

Rule Invocation and Evaluation: The *Posting Suspense Account Selection* rule type is invoked as part of the posting logic, after the *Posting Type Selection* rules (system) select the appropriate Posting Types to be posted in a specific step in the flow. The *Posting Suspense Account Selection* rules that will be invoked are the ones whose sub-type fits the selected Posting Type which is a suspense account posting Type (code pre-fixed by SA).

Rule Usage: The *Posting Suspense Account Selection* rule type is used to allow the setup of specific suspense accounts to be used in the posting entries (not as part of the message processing accounts[89]) for the different selected Posting Types that represent a posting to a type of a suspense account (e.g. special suspense account used for cross-book-balancing). These rules allow the selection of different accounts as per the message attributes. The bank users need to setup these rules in advance to be used whenever a specific type of suspense account is required for the posting logic (as per the configuration (Accounting Model) per the specific customer – *Posting Type Selection* system rules).

---

[89] Like a BSA – Batch Suspense Account – that is really the credit/debit account in the S message

Example:

If the *Posting Type Selection* rule selected the *Principle debit amount* (**MAIN_DR**), *Credit principal amount to susp. account for 2-stp-acctg* (**SA_MAIN_CR**)[90], *Principal amount- Cross-booking-systems balancing on debit side system* (**SA_CRSS_BOOK_MAIN_DBT_SD**) and *Principal amount- Cross-booking-systems balancing on credit side system* (**SA_CRSS_BOOK_MAIN_CDT_SD**) Posting types in a certain step – 3 invocations of the *Posting Suspense Account Selection* rules will take place with each of the sub-types that fit the 3 last relevant Posting Types above (**SA_MAIN_CR, SA_CRSS_BOOK_MAIN_DBT_SD** and **SA_CRSS_BOOK_MAIN_CDT_SD**).

- In the invocation with the **SA_MAIN_CR** sub-type the system would use the selected rule's Action as the suspense account to do the credit leg posting entry in the 2-step-accounting

- In the invocation with the **SA_CRSS_BOOK_MAIN_DBT_SD** sub-type the system would use the selected rule's Action as the suspense account to use for the leg of the cross-booking-system balancing entries on the debit side system

In the invocation with the **SA_CRSS_BOOK_MAIN_CDT_SD** sub-type the system uses the selected rule's Action as the suspense account to use for the leg of the cross-booking-system balancing entries on the credit side system.

4.2 System Setup

4.2.1 System Rules

4.2.1.1 Posting Type Selection

GPP allows variations within its posting logic as per different Accounting Models required for different banks. This variation is achieved via the pre-defined inclusions of posting logic invocation/s within the processing flow/s via pre-defined setup of *Posting Type Selection* system rules.

Description: A rule of this rule type determines which posting types are required for this invocation and which posting type are sent in which request. This selection can be conditioned depending on the transaction's attributes (for example, different set of posting types per different invocations (places in the flow); inclusion of different posting types into different requests.

Note: The option to send different requests (in different structures) per different accounting system can be achieved by using different request grouping names as sub-actions for the entries related to different accounting systems, in conjunction with the additional rules of Interface Selection.

Rule Sub-Type: Two rule sub-types are supported, Original posting and Reverse posting. The, values are defined in **Fields_Values** (**Field_Type**=POSTING_SUB_TYPE). The Reverse posting sub-type is only invoked in negative flows (e.g. reject from G3).

Note: Future thoughts – We may add another Sub-Type to be used for balance Inq (will be caught only for banks that require separate balance inquiry). Will allow creating the Posting entries (only for types for which balance check is required – e.g. in some banks Fees will not require balance check), saving reservation references from response, when supported, and then usage of these same entries (with the reservation references) in the Posting step.

Rule Conditions: The fields exposed to this rule type include all the message attributes, and related derived attributes. The conditions may be based on the point in the flow the posting logic is triggered for the message (**D_FLOW_CONTEXT**), on its transaction category (CTO, CTI. DDO etc.), on the

---

[90] As the bank requires to do the postings in a 2-step-accounting mode

steps already performed as per the different monitors (including the posting monitors, per Posting Type), etc.

Rule Attachment: *Posting Type Selection* rules are attached to either the global office or local office. GPP search for the attached rules first, at global office level, and then at local office. Multiple rules may be attached. All matching rules apply (per rule sub type).

Notes:

Since implementation bases the posting entries order, within the *Posting* tab, on the order of attachment of the *Posting Type Selection* rules (and order of invocation steps), the guidelines for this order are as follows:

- The entries should be in pairs of DR/CR

- Debit entries (for principal and fee amount) should always come before the credit entries. In case of different types of FEE entries, fees on debit side (FEE and PNL entries) should appear before fees on credit side (FEE and PNL entries)

- MAIN entries should always come before FEE (and PNL) entries

- If either of the balancing Posting Types is required, the MAIN ones should come after the MAIN entries that should be balanced, and the FEE ones after the FEE entries that should be balanced

Rule Action: A name of a posting type. Possible values are defined in Fields_Values (**Field_Type**=POSTING_TYPE). Also supported - the STOP action.

Rule Sub-Action (Usage): Request grouping name – values defined in **Fields_Values** (**Field_Type**= REQST_GRP_NM). This allows the grouping/no-grouping of posting types into one or more requests. Specific values –

- **NOT_TO_POST_IN_REQST** – can be used if the Posting Type should only be used to create a Msg_Posting entry in GPP but should not be included in a Posting request sent to the bank systems.

- **RLTD_IN_SEPRT_REQSTS** – can be used if multiple entries can be created with the same Posting Type (Fee entries are the common example) but should be sent in separate Posting requests to the bank systems (related pairs, for example *Fees on debit side customer* (**FEE_DR**) and its related *Amount in P&L account for fees on debit side customer* (**PNL_DR**), as long as both were selected with this same value).

Note: The usage of **RLTD_IN_SEPRT_REQSTS** should only be used for a Posting Type for which multiple posting entries may be created. For other Posting Types, each resulting in a single entry, the grouping of the pairs, if required to send by pairs and not in one group, should be done by using a unique sub-action per each pair (or single in the case the related should not be posted out).

Rule Invocation and Evaluation: The rule type can be invoked in a few steps in the flow, but these are pre-defined per the different flows. The Original sub-type is always invoked (in every point in flow that posting is invoked), and the Reverse sub-type is only invoked on negative flows (e.g. reject from G3) in addition to the Original.

Usage: GPP scans all the rules attached to the global office, and then to the local office according to their attachment order. Once evaluation of all rules is complete, all selected posting types are gathered and grouped according to the different Request Grouping Name selected per the different posting types as instructed in the rule's sub-action. Each of these groups is sent in a different Posting Request.

For the model of dynamically splitting the posting into two steps upon stooping the message, the configuration of *Posting Type selection* rules for the final leg, needs to make sure that if 1st leg entries failed – the full list of posting needs to be created, and if 1st leg posting was successful – need to only create the 2nd leg entries at that point.

GPP then invokes the appropriate posting interface/s as per the Interface Selection rules. The specific interface handler/s build/s the Posting Request/s and sends it/them.

4.2.1.1.1 Inward RTGS Payments to SWIFT

Inward RTGS to SWIFT will perform posting upon receipt of SWIFT ACK.

The posting entries required for MT103/PACS.008 in local and foreign currencies payments where crediting book account (Vostro) but forwarding SWIFT message are:

Principal

- Debit Scheme Settlement
- Credit Customer (receiver)
- Fee (Gross model SHA/BEN)
- Debit Customer (receiver)
- Credit P&L

Tax on Fee (India)

- Debit Customer (receiver)
- Credit Tax Payable
- Tax on Tax and Tax on tax P&L when applicable

The posting entries required for MT202 local and foreign currency payments where crediting book account (Vostro) but forwarding SWIFT message are:

Principal

- Debit Scheme Settlement
- Credit Customer (receiver)

4.2.1.1.2 Inward SWIFT payments with Agent Fees

Inward SWIFT to BOOK/INTERO will perform posting upon Completion.

The posting entries required for MT103/PACS.008 will be performed on a NET basis, where:

Principal + Fee

- Debit Customer (sender - Whole amount)
- Credit Customer (receiver - Net of Fee)
- Credit Agent P&L
- Credit Tax Payable (for India)
- Tax on tax P&L when applicable

4.2.1.1.3 BOOK Payment to RTGS with Fees

Outward payments to RTGS will perform 2-step posting - Debit side will be generated before sending to scheme and Credit side will be generated after response from scheme.

The 1st step accounting stage would include the gross method:

Debit entries before sending:

Principal

- [Debit Customer](#) (originator)

- [Credit Suspense account](#)

Fees

- [Debit Customer](#)

- [Credit Fee P&L](#)

- India would have Tax entries:

- [Debit Customer](#) (receiver)

- [Credit Tax payable](#)

- [Tax on Tax](#) and [Tax on tax P&L](#) when applicable.

4.2.2 System Profiles

4.2.2.1 Interfaces

Posting interface profile setup is required. This setup is a system setup done by Finastra as per specific bank requirements.

- Posting may be configured as a synchronous or an a-synchronous interface (Wait behavior). For the a-synchronous mode only, a wait queue is specified.

- In both cases, an inactive behavior (SKIP, STOP_UNTIL_ACTIVE, PERMANENT_STOP or STORE) is defined. In STOP cases an inactive queue is specified (for cases where the interface is down and inactive).

- The format used as the Posting request is defined – PROPRIETRY, Fndt Message or any other standard message type structure

- Additional technical definitions e.g. the connection point of the MQ queue used.

- It is recommended to define separate Interface Type entries for different invocation points (if required as per the Accounting Model). This way GPP code can use the same approach as used for other interfaces in regards to using the MI_..._STS monitors to make the decision whether the invocation should take place, or it already was done and current processing through the relevant point in the flow should skip it.

See the *GPP Online Help* or the *GPP Interfaces – Technical Guide* for a description of the relevant fields.

Note: When the Fndt Message format is defined as the structure to use for this interface , the system can also be setup to control which sections of the generic Fndt Message structure are to be sent out for this Request/Response, from maximum – full structure (including all the message related attributes), to minimum – only the section quoting the posting information.

4.3 Message Attributes

Message Attributes – **MINF** - Posting related parameters

Field ID	Name	Tree Name	Description
A set of monitors – 2 per posting type - **MF**_<posting type>_**POSTING_STS** and a	Flow monitor _<posting type> Post sts	Flow monitor _<posting type> posting sts	Flow monitors' status for each of the posting types. These monitors are saved in a string

Field ID	Name	Tree Name	Description
separate one for the reversal status **MF_**<posting type>**_R_POSTING_STS** Example: **MF_CRD_FEES_POSTING_STS** and **MF_CRD_FEES_R_POSTING_STS** for the posting statuses of the **FEE_CR** posting entries – original and reversed	Flow monitor _<posting type> rvrsl Post sts	Flow monitor _<posting type> rvrsl posting sts	format – each character representing a different monitor in P_POSTING_STATE_MONITOR.Values: • B - (Before Waiting) for posting entry that was prepared but not yet sent to Host, • P - (Pending Response) for waiting for Posting response from Host, • S - (Successful) for posted successfully, • N - (NSF) for insufficient funds posting response from Host, • E - (Processing/Technical error) for posting for which a Nak Host Response was received, indicating that a processing/technical error occurred either at the Host or within the integration layer in between GPP and the Host systems (when applicable), • L - for failure Host Response due to posting limitations/restrictions • O - (Optional specific other Reject) – for reject that the bank wants to specifically identify from other reject reasons for special handling. • R - (Reject) for reject posting response from Host (all other failures that are not NSF, technical or specifically marked as O for specific handling), • C - (Cancelled) for a posting that was canceled, • T - (Retry) for a posting entry that was resent again • X - when not applicable or not yet calculated (default)
MU_NSF_FORCE_STS	User monitor force from NSF	User force from NSF	Monitor of user action to force from *Insufficient funds* (**NSF**) status
MU_STOP_FLAGS_OVERRIDE_STS	Override stop flags	Override stop flags	Monitor of user action to override a stop flag on an account - from *Posting restrictions* (**POSTEX**) status

Field ID	Name	Tree Name	Description
D_CDT_POSTNTRY_TOBAL _CCY	Ccy of cdt posting entry to balance	Ccy of cdt posting entry to balance	The currency of the credit side posting entry for which an FX balancing entry is about to be created
D_DBT_POSTNTRY_TOBAL _CCY	Ccy of dbt posting entry to balance	Ccy of dbt posting entry to balance	The currency of the debit side posting entry for which an FX balancing entry is about to be created
D_CDT_POSTNTRY_TOBAL_BOOKENT	Booking entity of cdt posting entry to balance	Booking entity of cdt posting entry to balance	The booking entity of the credit side posting entry for which cross-book balancing entry is about to be created
D_DBT_POSTNTRY_TOBAL_BOOKENT	Booking entity of dbt posting entry to balance	Booking entity of dbt posting entry to balance	The booking entity of the debit side posting entry for which cross-book balancing entry is about to be created
D_POSTING_INVOCATION _CONTEXT	Posting invocation context	Posting invocation context	The context of the place in the processing workflow in which current posting invocation is done. Allows differentiating when and what will be included in each invocation

The following are the details available in the generic Posting logic, to be included in the posting request.

Posting information per message – **Msg_Posting**

Attribute DB Name	Posting Tab Alias	Description
MID		The MID of the related payment
OFFICE		The office of the payment. Quoted in this table to allow filtering according to the view office without the need to perform a join with the MINF table
DEPARTMENT		The department of the payment
IS_REVERSAL_IND	Reverse	Indicates whether the current posting entry is a reversal of another entry
POSTING_TYPE	Posting Type	The entry posting type, can have the values as setup in **Fields_Values** for **Field_Type**=POSTING_TYPE
CR_DR_FLAG	Cr/Dr	The Credit Debit flag
ACC_NO	Account Number	Posting account
ACC_CCY	Currency	Posting account currency

Attribute DB Name	Posting Tab Alias	Description
ACC_OFFICE	Office	Posting account office
ACC_TYPE		Posting account type
CUST_ACC_IND		An indication that the posting account is a customer account (copied from the relevant attribute of the **ACCTYPES** entry relevant for the posting account)
BOOKING_ENT		The booking entity (representing the destination posting/booking system) on which the account for this posting is managed. Note that this attribute may be used in the filtering of posting entries into different accounting request XMLs and timing.
ACC_TRANS_CODE		Posting account Transaction Code • For Principal debit entry – populated with **P_DBT_TX_CD** (that is populated using the Debit transaction code rules) • For Principal credit entry – populated with **P_CDT_TX_CD** (that is populated using the Credit transaction code rules) • For fee entries - populated with the transaction code of the relevant *Fee Formula*.
ACC_COST_CTR		The *Cost Center* to which the account belongs. Used by Anti-Money Laundering, mainly for Posting and mainly in US.
FEE_TYPE_GROUP		Fee Type Group (Category). Possible values defined in **Fields_Values** with **Field_Type**=FEE_TYPE_CTGRY. Common examples are: RETURN_FUND, PROCESSING.
VALUE_DATE	Value Date	the relevant Dbt/Cdt value date of the posting entry
AMOUNT	Amt	Posted amount for this entry
BASE_AMOUNT		Posted amount quoted in base currency
POSTING_STS	Posting Status	The posting status (would be equal to the relevant monitor on the payment). Possible values: • B – (Before Waiting) for posting entry that was prepared but not yet sent to Host, • P – (Pending Response) for waiting for Posting response from Host, • S – (Successful) for posted successfully, • N – (NSF) for insufficient funds posting response from Host, • E – (Processing/Technical error) for posting for which a Nak Host Response was received, indicating that a processing/technical error occurred either at the Host or within the integration layer in between GPP and the Host systems (when applicable),

Attribute DB Name	Posting Tab Alias	Description
		• L – for failure Host Response due to posting limitations/restrictions • O – (Optional specific other Reject) – for reject that the bank wants to specifically identify from other reject reasons for special handling. • R – (Reject) for reject posting response from Host (all other failures that are not NSF, technical or specifically marked as O for specific handling), • T – (Retry) for a posting entry that was resent again • X – when not applicable or not yet calculated (default)
EARMARK_REF		Earmark reference that was returned from the Balance Inquiry interface (if returned), and should be sent back with the posting entry to allow the linkage between the earmarked funds and this posting entry
FORCE_POST_IND		Indicates whether the current posting entry was sent to accounting system with an instruction to force post. • If 1 – Force was instructed • If empty or 0 – no Force was instructed
INTERFACE_NAME		The Interface name (from interface type table) through which posting entry has been transmitted, if it was sent out. If it was not sent out – field will be empty
REVERSED_ORIG_POSTING_ENTRY_ID		In the original posting entry this tag quotes the reversed posting entry id. In the reversed posting entry this tag quotes the posting entry id of the original posting entry that is being reversed. **Note:** When the first step accounting is a lump sum (on the S): • Only in the reversed posting entries of the individual messages, the original posting is quoted. • This tag is not quoted in the original lump posting entry, as there is a relation of 1:n between the original posting entry (on the S) and the possible reversed (per Individual)
POSTING_REQST_ID		The unique ID of the Posting request through which this entry was posted to the bank accounting system
POSTING_ENTRY_ID		A unique ID of the posting entry
MAN_POST_RSPNS_IND		Indicates whether the current posting status is the result of a manually instructed posting response a user had requested – either Positive or Negative. • If 1 – manually instructed posting response was performed • If empty or 0 – no manually instructed posting response was performed

Attribute DB Name	Posting Tab Alias	Description
POSTING_INVOCATION_CONTEXT		Holds the posting invocation context of the posting entry[91]
OFFICE_BSNESSDATE		The business date of the bank/Office in which this record was created
CREATE_DATE		The DateTime when the Posting entry was created (in Local Office time zone terms)
TIMESTAMP		The DateTime of last update of this Posting entry (in server time zone terms)

In addition to the above, it is possible to compute/derive additional information that is specific per a specific customer bank and include it in the posting request. This can be achieved using the **Msg_Posting_Ext** table that may be populated as part of the Customer Specific logic step (see section **Error! Reference source not found.**).

Customer specific posting information per message – **Msg_Posting_Ext**

Attribute DB Name	Description
POSTING _ID	A unique ID of either the posting entry (**Msg_Posting.Posting_Entry_ID**) in the generic **MSG_POSTING** table (if **Ext_Type**=ENTRY) – to allow additional customer specific information on the entry level, or the Request ID (**Msg_Posting.Posting_Reqst_ID**) (if **Ext_Type**=REQST) – to allow additional customer specific information on the request level.
EXT_TYPE	The type of the extension this entry represents (which ID is quoted in the **Posting_ID** field). Values: ENTRY – the extension is for additional entry level attribute REQST – the extension is for additional request level attribute
FIELD_NAME	The name of the additional customer specific attribute of the posting entry
FIELD_VALUE	The value of the related Field_Name for the specific posting entry

---

[91] The value in the D_POSTING_INVOCATION_CONTEXT when entry created.

4.4 Statuses

4.4.1 Wait Posting

Group it belongs to	Description	Available actions
Waiting	Messages that are waiting for posting to be done/response to be received (for a-synchronous posting interface)	Available posting related buttons for Wait Posting

Available posting related buttons for *Wait Posting*

Action	Description
Release – As Positive Posting	If configured for dual approval – message is routed to the *Approve Manual Posting Response* queue for approval/rejection.
	Else (or after approval within the above queue) – Releases the payment to continue processing, even though a posting request was previously sent for it, but no response received, emulating a positive Posting Response. Only entitled users can use this action button.
	Sets all the relevant **Msg_Posting.Posting_Sts** and **MF_POSTING_[..] STS** to S. Also sets the appropriate (per posting leg) *Posting response manually instructed* (**MU_MAN_1ST_PSTNG_RSPNS/MU_MAN_2ND_PSTNG_RSPNS**) to P for positive emulation, and *Request Response status* (**MI_POSTING_REQ_RSPNS_STS**) to M for manually instructed response, and clears the *Follow up* (**P_FOLLOWUP**) attribute of the message if it is set with appropriate code for "Timeout - no response"[92].
Release – As Negative Posting	If configured for dual approval – message is routed to the *Approve Manual Posting Response* queue for approval/rejection.
	Else (or after approval within the above queue) – Releases the payment to continue processing, even though a posting request was previously sent for it, but no response received, emulating a negative Posting Response. Only entitled users can use this action button.
	Sets all the relevant **Msg_Posting.Posting_Sts and MF_POSTING_[..] STS** to R. Also sets the appropriate (per posting leg) *Posting response manually instructed* (**MU_MAN_1ST_PSTNG_RSPNS/MU_MAN_2ND_PSTNG_RSPNS**)) to N for negative emulation, and *Request Response status* (**MI_POSTING_REQ_RSPNS_STS**) to M for manually instructed response, and clears the *Follow up* (**P_FOLLOWUP**) attribute of the message if it is set with appropriate code for "Timeout - no response".
Cancel	Initiates the cancelation of the payment.

---

[92] Meaning a Timeout already occurred for the relevant Request

4.4.2 Stop Posting

Group it belongs to	Description	Available actions
Inactive	Messages that are held due to interface inactivity	Available posting related buttons for Stop Posting

Available posting related buttons for **Stop Posting**

Action	Description
Send to Repair	Sends the message to the **Repair** queue.

4.4.3 Insufficient Funds

Group it belongs to	Description	Available actions
Manual process	When the transaction debit account is short of funds, the payment is held in a specific queue until the debit account balance is sufficient to cover the transaction. Insufficient funds decision making is performed as part of the balance inquiry, if invoked, or the posting interface phase where, instead of performing the posting, the interface sets the status to **Insufficient funds**.	Available posting related buttons for Insufficient Funds

Available posting related buttons for **Insufficient Funds**

Action	Description
Force Ins. Funds	Allows payment processing to continue even though the debit account has insufficient funds. Only entitled users whose user profile maximum force limit amount is greater than the message amount can use this action button. Sets **Monitor user force from NSF** (**MU_NSF_FORCE_STS**), skips another Balance Inquiry, if invoked and failed due to short of funds, reaching the posting step, or redo posting once more. If the failure received per posting invocation, the new Posting request includes within it an indication that these entries should be forced this time. In both cases, allows the posting interface to take this monitor into account.
Retry Ins. Funds	Sends the message back to re-processing (skipping already successful steps) until point of Balance Inquiry and/or Posting invocation and performs relevant interface call again.
Send to Repair	Sends the message to the **Repair** queue. If partial posting has been successfully performed, the accounting will be re-evaluated after submission from **Repair**, and if different than already successful posting entries – these would be reversed and new ones would be posted.
Cancel	Initiates the cancelation of the payment. If partial posting has been successfully performed, the successful posting entries are reversed.

4.4.4 Posting Restriction

Group it belongs to	Description	Available actions
Manual process	Messages that have a posting restriction (stop flag) that needs to be acknowledged or overridden	See Available posting related buttons for Posting Restriction

Available posting related buttons for *Posting Restriction*

Action	Description
Override	Allows overriding the restriction (only active for a message that was stopped due to an overridable Stop Flag), and re-processing (skipping already successful steps) until point of Account Lookup or Posting invocation and performs relevant interface call again. If the failure received per posting invocation, the new Posting request includes within it an indication that these entries should be forced this time.
Retry Post. Rest.	Sends the message back to re-processing (skipping already successful steps) until point of Account Lookup or Posting invocation and performs relevant interface call again.
Send to Repair	Sends the message to the *Repair* queue. If partial posting has been successfully performed, the accounting will be re-evaluated after submission from *Repair*, and if different than already successful posting entries – these would be reversed and new ones would be posted.
Cancel	Initiates the cancelation of the payment. If partial posting has been successfully performed, the successful posting entries are reversed.

4.4.5 Posting Exception

Group it belongs to	Description	Available actions
Exception	Messages that failed posting with neither of the special handling failure responses (NSF/Posting Restrictions)	See Available posting related buttons for Posting Exception

Available posting related buttons for *Posting Exception*

Action	Description
Release – As Positive Posting	If configured for dual approval – message is routed to the Approve Manual Posting Response for approval/rejection. Else (or after approval within the above queue) – Releases the payment to continue processing, even though a negative posting response was received (probably after manually handling it in the bank's accounting system), emulating an overriding positive Posting Response. Only entitled users can use this action button.

Action	Description
	Sets all the relevant **Msg_Posting.Posting_Sts and MF_POSTING_[..] STS** to S. Also sets the appropriate (per posting leg) *Posting response manually instructed* (**MU_MAN_1ST_PSTNG_RSPNS/MU_MAN_2ND_PSTNG_RSPNS**) to P for positive emulation, and *Request Response status* (**MI_POSTING_REQ_RSPNS_STS**) to M for manually instructed response, and clears the *Follow up* (**P_FOLLOWUP**) attribute of the message if it is set with appropriate code for "Timeout - no response".
Send to Repair	Sends the message to the *Repair* queue. If partial posting has been successfully performed, the accounting will be re-evaluated after submission from *Repair*, and if different than already successful posting entries – these would be reversed and new ones would be posted.
Cancel	Initiates the cancelation of the payment. If partial posting has been successfully performed, the successful posting entries are reversed.

4.4.6 Approve Manual Posting Response

Group it belongs to	Description	Available actions
Manual Process	Messages, for which a manually instructed posting response was performed, and are waiting for a second user to approve/reject the instruction	See Available posting related buttons for Approve Manual Posting Response

Available posting related buttons for *Approve Manual Posting Response*

Action	Description
Approve Manual Positive Response	Approves the manual instructions to emulate positive posting response, invoking the appropriate logic. Button appears only if user clicked a *Release – As Positive Posting Response* button.
Reject Manual Positive Response	Rejects the manual instructions to emulate positive posting response, routing the message back to the previous queue. Button will only appear if user clicked a *Release – As Positive Posting Response* button.
Approve Manual Negative Response	Approves the manual instructions to emulate negative posting response, invoking the appropriate logic. Button will only appear if user clicked a *Release – As Negative Posting Response* button
Reject Manual Negative Response	Rejects the manual instructions to emulate negative posting response, routing the message back to the previous queue. Button appears only if user clicked a *Release – As Negative Posting Response* button

4.5 Access Class Entitlements

4.5.1 Required Access Class for Using Buttons

Required Access Class for using buttons

Access Level	Button	Comment
Release – As Positive Posting button	Release – As Positive Posting	In *Wait posting* and *Posting Exception* statuses
Release – As Negative Posting button	Release – As Negative Posting	In *Wait posting* status
Force Insufficient Funds button	Force NSF	In *Insufficient funds* status
Retry Insufficient Funds button	Retry NSF	In *Insufficient funds* status
Override button	Override	In *Posting Restrictions* status
Retry Posting Restrictions button	Retry Post. Rest.	In *Posting Restrictions* status
Approve Manual Positive Response button	Approve Manual Positive Response	In *Approve Manual Posting Response* status
Reject Manual Positive Response button	Reject Manual Positive Response	In *Approve Manual Posting Response* status
Approve Manual Negative Response button	Approve Manual Negative Response	In *Approve Manual Posting Response* status
Reject Manual Negative Response button	Reject Manual Negative Response	In *Approve Manual Posting Response* status

4.6 Audit Trail and Errors

Audit Trail and Errors

Code	Description	Notes
28854	Posting response is missing mandatory field: [Field Name]	
40091	Party account number (or a related account number) does not have sufficient funds. Payment routed to NSF.	Balance inquiry failure; Insufficient funds
Pending	Reversed posting entry of same posting Type – [posting type] - already posted are different than entries currently calculated - posted reversed entry – [Posting_Entry_Id of posted entry from DB], currently calculated entry – [Coma separated concatenation of values from the calculated reversed posting entry in memory]	An error for the case a message reached (again) a point in the flow in which reversed posting was required, but the current outcome is different than an already successful reversed posting of same Posting Type

Code	Description	Notes
Pending	Missing / improper setup of Posting Suspense Account Selection rules, as no rule was caught for sub type [posting type] to point to the suspense account that should be used for this type of posting	No rule was caught for required sub type to be point to the suspense account that should be used for this type of posting
Pending	Missing setup for Booking Entity (account specific or default). Posting cannot continue	Missing setup for Booking Entity (account specific or default)
Pending	Posting process failed for interface [Interface name]. Provided reason is: [Response failure description from bank system] ([Response failure code from bank system])	
Pending	Non valid Account Type for [Posting Type] posting entry. Posting cannot continue	Probably missing setup for the account type passed in Account lookup

5 Recommended Setup

The required setup will be:

- User Codes to define the accounting systems used in each deployment – use the *Type* of BOOKNG_ENT

- Set the *Default Consolidated P&L posting entries* in the *Banks* profile or the *Consolidated P&L posting entries* in the different P&L *Accounts* profiles as per bank's preferences

- Set the *Default Consolidated fee posting entries* in the *Banks* profile *or Consolidated fee posting entries* in the different *Parties* profiles as per the customer's preferences

- Define the *Posting Suspense Account Selection* rules with different sub-types (e.g. *SA_MAIN_CR, SA_MAIN_GROSS_CR, SA_FEE_CR. SA_MAIN_DR, SA_CRSS_BOOK_MAIN_CDT_SD,* S*A_CRSS_BOOK_MAIN_DBT_SD* – as per the required Accounting Model) selecting the suspense account to be used for the different cases

Notes:

1. For Bulk CTO there is no need to setup a *Posting Suspense Account Selection* rule. The batch suspense account is selected by GPP as setup, required for bulk processing, in the *Currency Preference* profile (per relevant currency).

2. The *Posting Suspense Account Selection* rules with sub-type *SA_MAIN_CR, SA_MAIN_GROSS_CR, SA_FEE_DR* and *SA_MAIN_DR* can select, if so required, the same account as used as the batch suspense account.

Appendix A: Glossary

The table lists the terms used in this document.

Term	Description
Asset Account	An account which reflects funds owned by the bank but held at another institution. An asset account mirrors the balance of the account held by another institution
Asynchronous Interface	GPP does not wait for the on-line response from an external system. Depending on specific interface behavior and configuration, the payment message is either put in a queue and once a response is received, GPP continues its payment processing, or continues processing and once a response is received the appropriate actions, per the response, take place
BI	Balance Inquiry
Booking Entity	An attribute of an account that identifies what booking entity of the bank holds the account. Used to identify multi-book scenarios
Credit	An accounting entry which results in either a decrease in assets or an increase in liabilities
Cross-Booking-System Entries	Cross-Accounting-System entries are special accounting entries generated to move funds between booking entities. The entries can be incorporated into the main Accounting Interface or can be handled by an additional dedicated interface
Currency Balancing Entries	When the requirement is that accounting entries in foreign currency must be balanced per currency, additional accounting entries are posted to special currency balancing suspense accounts to balance per each currency
Debit	An accounting entry which results in either an increase in assets or a decrease in liabilities
External Account	An account held outside of GPP
FRD	Refers to the iteration of an FRSD dealing with Requirements but not Solutions
FRSD	Functional Requirements and Solution Document
GPP	Global PAYplus
Internal Account	An account held within GPP
Multiple Books	For an office, more than one set of accounting books are maintained. In many cases, one book is in local currency and other books are based in foreign currency
Non-Asset Account	An account which reflects funds owned by another party but held by the bank
P&L	Profit and loss
Posting	A generic term for generating accounting entries which are usually sent outside of GPP
STP	Straight Through Processing. Automated processing without operator intervention between the time the payment is dropped into the system until it reaches completion

Term	Description
Synchronous Interface	GPP waits for the on-line response from an external system before payment processing continues

Appendix B: Supported Posting Types

The following are the supported Posting Types each with a description on when it would, or should, be used.

- The main and more often used[93]:
  - **MAIN_CR** – principle credit amount
  - **MAIN_DR** – principle debit amount
  - **FEE_DR** – fees on debit side customer
  - **FEE_CR** – fees on credit side customer
  - **PNL_DR** – amount in P&L account for fees on debit side
  - **PNL_CR** – amount in P&L account for fees on credit side

- In case aged fees should be supported:
  - **AGT_FEE_DR** - Agent fees on debit side customer (CT) – for Outgoing messages
  - **AGT_PNL_DR** - Amount in P&L account for agent fees on debit side (CT) – for Incoming messages
  - **AGT_FEE_CR** - Agent fees on credit side customer (DD) – for Outgoing messages
  - **AGT_PNL_CR** - Amount in P&L account for agent fees on credit side (DD) – for Incoming messages

- In case bulk posting is supported and the posting steps are involving a batch suspense account - the following Posting Types will also be included:
  - **BSA_CR** – credit to batch suspense account for batch posting
  - **BSA_DR** – debit to batch suspense account for batch posting

Note: GPP does not support the fees defined as ***Deduct from Payment*** in bulk processing. This is the reason there are no posting type to allow separating the entries to **BSA_GROSS_CR** and **FEE_CR**.

- In case 2-step-accounting is required, for various reasons[94], and hence - involving a suspense account used to pass the funds between the steps - the following Posting Types will also be included:
  - **SA_MAIN_CR** - credit principal amount to suspense account of principal amount for bank specific reasons to use 2-step-accounting
  - **SA_MAIN_DR** - debit principal amount to suspense account of principal amount for bank specific reasons to use 2-step-accounting

---

[93] Note that there are additional, alternative, Posting Types representing the principal credit, principal debit and the fee types, for more complex scenarios such as 2-step-accounting required, or case fees were deducted from payment (lower credit for receiving side) and requirement asks for posting as Net (one entry includes the Principal and the Fee) or as Gross (detailed in separate entries). These would be described below in this list

[94] - An accounting model that requires to do the debit side posting as-soon-as-possible and the rest of the posting on completion;

- An accounting model that requires to do 1step accounting if STP, but move to 2-step-accounting model if message is stopped in a manual queue;

- Other customer specific reasons – such as – in G3 CTO flows – the need to post the debit leg before sending to G3 but do the credit side posting only after a response from G3 (to be able to use the date/cycle as recorded in G3

Note: the following 2 types will be used in cases the fee amount would need to pass through the suspense account[95]

- **SA_FEE_CR** - credit fee amount on credit side to suspense account for bank specific reasons to use 2-step-accounting

- **SA_FEE_DR** - debit fee amount on debit side to suspense account for bank specific reasons to use 2-step-accounting

- In case the accounting model requires to report cross-booking[96] balancing entries – the following Posting Types will also be included:

  - **SA_CRSS_BOOK_MAIN_CDT_SD** – Principal amount - cross-booking-systems balancing on the credit side system (through specific suspense account)

  - **SA_CRSS_BOOK_MAIN_DBT_SD** – Principal amount - cross-booking-systems balancing on the debit side system (through specific suspense account)

  - **SA_CRSS_BOOK_FEE_CDT_SD** – Fee amount - cross-booking-systems balancing on the credit side system (through specific suspense account)

  - **SA_CRSS_BOOK_FEE_DBT_SD** – Fee amount - cross-booking-systems balancing on the debit side system (through specific suspense account)

- In case the accounting model requires to report FX[97] balancing entries – the following Posting Types will also be included:

  - **SA_FX_BAL_MAIN_CDTCCY** – Principal amount - FX balancing (using credit CCY specific suspense account)

  - **SA_FX_BAL_MAIN_DBTCCY** – Principal amount - FX balancing (using debit CCY specific suspense account)

  - **SA_FX_BAL_FEE_CDTCCY** – Fee amount - FX balancing on customer side (using credit (PNL) CCY specific suspense account)

  - **SA_FX_BAL_FEE_DBTCCY** – Fee amount - FX balancing on customer side (using debit (FEE) CCY specific suspense account)

- In case the accounting model requires to report the credit[98] fees in separate entries even when they are actually deducted from the payment amount – the following Posting Types will also be included:

  - **MAIN_GROSS_CR** – Gross principle (no fee deduction) credit amount

  - **SA_MAIN_GROSS_CR** – Credit gross principal amount (no fee deduction) to suspense account of bank specific reasons to use 2-step-accounting

---

[95] - An accounting model that requires to do 1step accounting if STP, but move to 2-step-accounting model if message fails due to customer not identified, in which case the fee (and principal in above posting type) would be posted to a suspense account until resolution with appropriate customer;

 - A 2-step-accounting accounting model that requires to show separate fee and principal entries even in a case of a fee defined as Deduct from Payment

[96] When configured to use these Posting Types - entries will be created if a debit entry is posted to an account on accounting system and the corresponding credit entry is posted to an account on another accounting system

[97] When configured to use these Posting Types - entries will be created if a debit entry is posted to an account in one currency and the corresponding credit entry is posted to an account on another currency

[98] As per current phase only logic on Deduct from Payment as per CTOs is supported, and hence only credit fees are mentioned in this group. Once GPP will support adding fees to payment amount for DDOs – this solution will also need to be enriched for such debit fees behavior