Standing Payment Orders

Contents

Introduction

Standing payment orders automatically create documents serving as payment orders in the system, or automatically create macrotransactions. They can be used for a variety of operations, such as making automatic merchant payments, crediting a cardholder account, and amount normalisation, including multicurrency normalisation, and redirecting interest accrual or due normalisation.

One of the most important functions of standing payment orders is creating documents in real time, where the client sets the document amount. For example, the client may pay for public utilities online through an ATM.

This document is intended for bank or processing centre employees responsible for configuring WAY4™ and describes product creation and configuration.

While working with this document, it is recommended that users refer to the following reference material from OpenWay's documentation series:

- WAY4™ Products. Accounting Schemes
- Daily Procedures
- Events
- Acquiring Module User Manual
- WAY4™ Global Parameters
- WAY4 Client and Contract Classifiers
- Contract Functional Dates

The following conventions are used throughout this document:

- Field labels in screen forms are typed in *italics*.
- Button labels used in screen forms are placed in square brackets, such as [Approve].
- Menu selection sequences are shown with the use of arrows, such as Configuration Setup → Contract Types.
- Warnings of possible erroneous actions are marked with the  sign.
- Messages marked with the  sign contain information about important features, additional facilities, or the optimal use of certain functions of the system.

Chapter 1. Configuring Standing Payment Orders

This section describes the setup of categories of standing payment orders and features of their processing.

Categories of Standing Payment Orders

Standing payment orders can be grouped into the following types:

- General standing payment orders – payment orders of this type are created on the Accounting Scheme level. Documents are generated for all contracts using this Accounting Scheme.

  General orders are used to set up a Product's lifecycle. These payment orders are used, for example, to set up reimbursement of merchants, pay recurring merchant fees, or to calculate a minimum payment.

- Template standing payment orders. Template orders are also configured on the Accounting Scheme level and then used to create inherited orders on the contract level. The code of a template must be unique.

- Inherited payment orders are created on the contract level, based on template orders. General rules for creating inherited orders:

  - By default, inherited orders are created on demand – manually or using the Advanced Applications module. Orders are created in an inactive state (by default, the value of the *Is Active* parameter is "No"). An order is activated on the level of a specific contract.

  - Several inherited orders can be created on the basis of one template.

     If only one inherited order can be created according to a template, specify the UNIQUE_INHERITED tag in the template. The inherited order will be created with a code corresponding to the template code.

  - The codes of orders created according to templates must be unique. If a code is not set manually or using an application, a unique code will be generated automatically.

  - Some fields in an inherited payment order are synchronised with the order template and cannot be edited. Some fields of an inherited order can be edited. I.e., the main properties of an order are common for all contracts and it is convenient to define them in template order settings in the Accounting Scheme, while recipient requisites, and transfer amounts may differ from those in a template, and therefore it is convenient to define them on the contract level, that is, in inherited orders.

    Inherited standing payment orders are used, for example, to make public utilities payments, or for regular transfers to a client.

- Individual standing payment orders are completely configured on the contract's account level. These orders can only be set up for bank contracts.

Existing (created earlier than version 03.41.30) individual payment orders for issuing and acquiring contracts will work, but it is no longer possible to create new individual payment orders for these contracts. It is recommended to reconfigure existing individual payment orders for issuing and acquiring contracts.

Processing Standing Payment Orders ("Morning"/"Evening" Mode)

Documents relating to standing payment orders are created and processed according to the payment order configuration, more specifically, according to the value of the *Date Event* field (see "Determining Document Frequency and Amount").

Payment orders are processed through the "Contracts – Daily Update" procedure according to the value of global parameter "ORDER_IN_START_OF_DAY" (see the WAY4™ Global Parameters Administrator Manual). If the parameter is set to "Y" (default value), payment orders are processed as part of the "Contracts – Daily Update" procedure executed at the beginning of the next day. If the parameter is set to "N", payment orders are processed as part of the "Contracts – Daily Update" procedure executed at the end of the day, and are ignored when the procedure is executed at the beginning of the next day. This parameter can be redefined using the tag of the same name in an Accounting Scheme or in a separate order (see the description of the ORDER_IN_START_OF_DAY field in the section "Tags Used When Configuring Accouting Schemes and Accounting Templates" of the document "WAY4™ Accounting Schemes").

When processing standing payment orders, a calendar type that differs from the financial institution's calendar type can be used. To do so, set the corresponding calendar type in the payment order's *Posting Details* field (see the section "Defining Corresponding Accounts") using the CALENDAR_TYPE=<calendar name> tag (see "Tags in the *Posting Details* Field of a Standing Payment Order").

Parameters of General/Template Standing Payment Orders

The "Definition for ‹name of Accounting Scheme›" form can be used to configure parameters of a new general or template standing payment order (see the "Full Information about Accounting Scheme Templates" section in the WAY4™ Products. Accounting Schemes Administrator Manual). Select the account template and click on the [SO Full] button. Buttons [SO Due], [SO Evnt Base], [SO Interest], and [SO Norm] are used to configure certain types of standing payment orders (see "Special Grids for Entering General/Template Standing Payment Orders").

Clicking on the [SO Full] button will invoke the "SO Full for ‹name of account template›". It contains the full set of fields for configuring standing payment orders.

To enter a new standing payment order into the "SO Full for ‹name of account›" form, click on the [Ins] button and fill in the fields as needed (see Fig. 1).

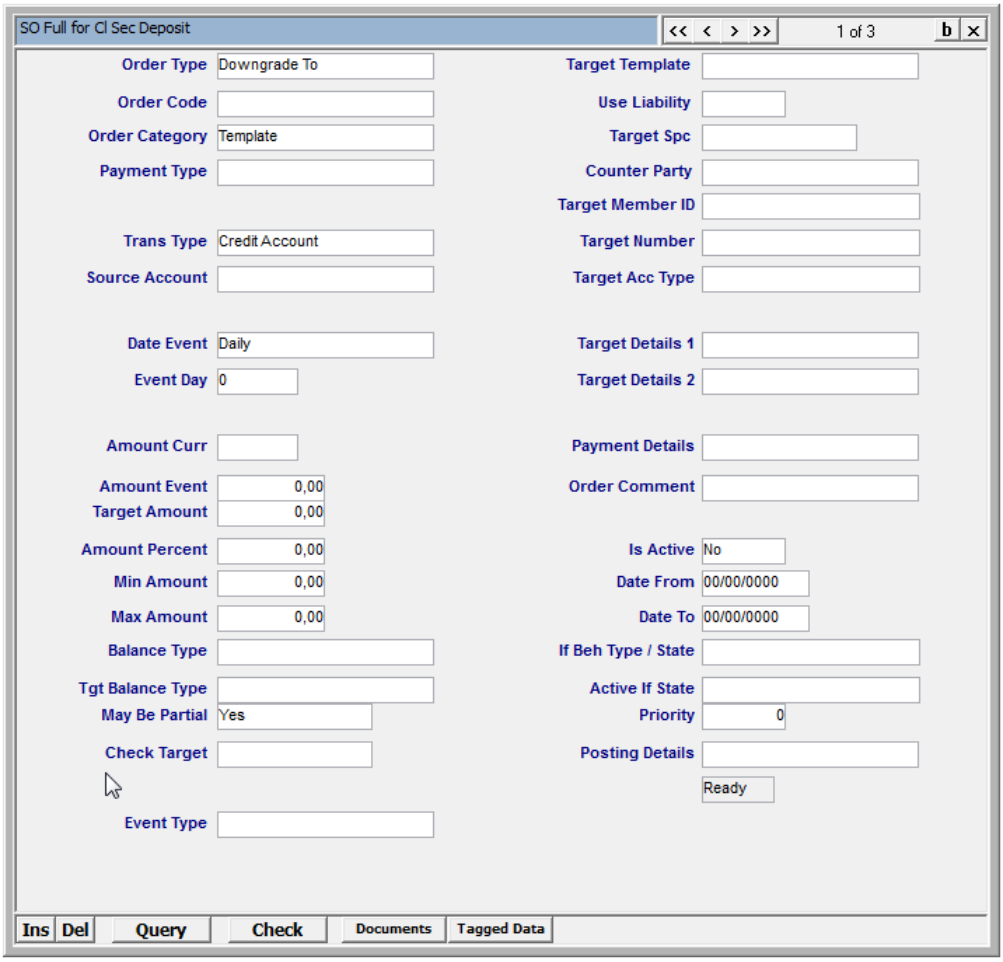

Fig. 1. Form for defining and configuring a standing payment order

The fields in the "SO Full for ‹name of account›" form can be grouped into the following categories:

- Fields for describing transactions (see "Transaction Description")
- Fields for configuring the frequency and amount of the payment order (see "Determining Document Frequency and Amount")
- Fields defining the corresponding account (see "Defining Corresponding Accounts")
- Fields defining the state of the payment order (see "Activating/Deactivating Standing Payment Orders")

The [Check] button in the system checks that the payment order's fields have been filled in correctly and gives check results. At this step, the system creates a document for the standing payment order with a zero amount. If the check is successful, the document has the "Waiting" status; otherwise, it has the "Closed" status.

A standing payment order is automatically checked when a contract is approved. To skip this stage (standing payment order parameters will not be checked during contract approval), specify the tag

IGNORE_DOC_VALIDATION in the *Posting Details* field of the standing payment order form (see Fig. 1). See "Tags in the *Posting Details* Field of a Standing Payment Order".

The [Document] button enables users to view the documents created for that payment order during its lifetime through the [Check] button.

The [Tagged Data] button is used to optimise work with the *Posting Details* field. Clicking this button opens a grid form for entering and editing tags specified in the *Posting Details* field. For a detailed description of entering and editing tags, see the "Entering and Editing Tags" section in the WAY4™ Accounting Schemes Administrator Manual.

The parameters of General standing payment orders for a contract and template standing payment orders can be viewed in the form "Gen. Orders for <account name>", opened by clicking the [Gen. Orders] button in the "Accounts for <contract name>" form.

Transaction Description

The fields that define transactions are described below:

- *Order Type* – determines the standing payment order type and the amount algorithm for this payment order. The field can take on the following values:
  - "Debit Amount" – debiting this account (or the account indicated in the *Source Account* field) and crediting the corresponding account with the amount defined in the *Target Amount* field. The amount is only transferred if the amount remaining in the account is higher than the amount specified in the *Amount Event* field. It is recommended that this standing payment order type be used when the transfer amount is known in advance.

    When this type of order is used for Supplementary orders, note that the *Amount Event* field for these orders is not checked.
  - "Credit Amount" – crediting this account (or the account indicated in the *Source Account* field) and debiting the corresponding account for the amount defined in the *Target Amount* field. The amount is only transferred if the amount remaining in the account is lower than the amount specified in the *Amount Event* field. It is recommended that this standing payment order type be used when the transfer amount is known in advance.

    When this type of order is used for Supplementary orders, note that the *Amount Event* field for these orders is not checked.
  - "Downgrade To" – debiting this account (or the account indicated in the *Source Account* field) and crediting the corresponding account. If the *Amount Percent* field is filled in, the document amount is calculated in different ways depending on the "ORDER_PCNT_RULE" global parameter value (see the WAY4™ Global Parameters Administrator Manual):
    - If the parameter value is empty (NULL), the amount of funds on the account should be reduced to the value (100 - <contents of the Amount Percent field>)%. The document amount will be <contents of the Amount Percent field>% of the amount of funds on the account.

- ♦ If the parameter value is "I", the amount of funds on the account should be reduced to the value (<contents of the Amount Percent field>)%. The document amount will be (100 - <contents of the Amount Percent field>)% of the amount of funds on the account.

If the *Amount Percent* field is not filled in, the document amount is calculated as the difference between the account balance and the value indicated in the *Target Amount* field. The amount is only transferred if the amount remaining in the account is higher than the amount specified in the *Amount Event* field and the value of the *Target Amount* field is equal to or lower than the value of the *Amount Event* field. It is recommended that this standing payment order type be used when it is necessary to debit an account to an amount known in advance (e.g. to clean out an account).

- "Upgrade To" – crediting this account (or the account indicated in the *Source Account* field) and debiting the corresponding account. If the *Amount Percent* field is filled in, the document amount is calculated depending on the value of the global parameter "ORDER_PCNT_RULE" (the parameter influences calculation of the document amount for an "Upgrade To" type payment order in the same way as for a "Downgrade To" type, see the description of a "Downgrade To" type payment order). If the *Amount Percent* field is not filled in, the document amount is calculated as the difference between the account balance and the value indicated in the *Target Amount* field. The amount is only transferred if the amount remaining in the account is lower than the amount specified in the *Amount Event* field and the value of the *Target Amount* field is higher than or equal to the value of the *Amount Event* field. If both accounts belong to the same contract and the balance of the corresponding account is lower than the calculated document amount, the amount will be adjusted to be equal to the balance of the corresponding account. It is recommended that this standing payment order type be used when it is necessary to credit an account to an amount known in advance (e.g. to fully pay off an account debt).
- "Downgrade To (Single)" – the same as "Downgrade To", but can be only be used in an account from an active-passive pair. That is, if this account is one of an active-passive pair, the document amount will not depend on the balance of the pair account.
- "Upgrade To (Single)" – the same as "Upgrade To", but can be only be used in an account from an active-passive pair. That is, if this account is one of an active-passive pair, the document amount will not depend on the balance of the pair account.
- "Upgrade Advice" – the same as "Upgrade To", but if both accounts belong to the same contract, the balance of the corresponding account is not considered when calculating the document amount (general acceptance payment). It is recommended that this standing payment order type be used when it is necessary to fully pay off a certain account debt even if it makes the resulting balance of the corresponding account negative.
- "Normalisation" – the same as "Upgrade To", but the document amount is calculated considering the amount remaining in all accounts with this

account as an upper-/lower-limit normalisation account (see an example in the "Multicurrency Normalisation" section). It is impossible to create an individual payment order of this type.

- "Norm Advice" – the same as "Upgrade Advice", but the document amount is calculated considering the amount remaining in all accounts with this account as an upper-/lower-limit normalisation account. It is impossible to create an individual payment order of this type.

- *Trans Type* – determines the transaction type for the document created by the payment order. It is a drop-down list of transaction types registered in the system dictionary ("Full → Configuration Setup → Transaction Types → Transactions - All") with the following field values: *Source* = "Account", *Chain Type* = "Original", *Category* = "Individual".

  If the *Trans Type* field is filled in, its value is related to the value of the *Order Type* field in the following way:

  - If "Debit Amount", "Downgrade To" or "Downgrade To (Single)" is specified in the *Order Type* field, the "Credit Account" value should be selected in the *Trans Type* field.
  - If "Credit Amount", "Upgrade To", "Upgrade To (Single)", "Upgrade Advice", "Normalisation", or "Norm Advice" is specified in the *Order Type* field, the "Debit Account" value should be selected in the *Trans Type* field.

  When generating a document according to a standing payment order for a contract, the corresponding macrotransaction is generated according to the contract Service configured for the transaction type specified in the *Trans Type* field. For instance, a fee may be changed for this transaction.

  If the *Trans Type* field is not filled in, no transaction type is specified in a document generated for the standing payment order.

- *Payment Type* – specifies the type of the public utility payment. The types of utility payments can be accessed through the "Payment on Account Types" form (Full → Configuration Setup → Transaction Types → Payment on Account Types).

  Tags (and their values) set in the *Add Info* field of the "Payment on Account Types" form can be inherited to a document created by a payment order with this payment type. To do so, the corresponding tags must be specified as the value of the global parameter PAYMENT_TYPE_TAGS_TO_SO (see the section "PAYMENT_TYPE_TAGS_TO_SO" of the document "WAY4™ Global Parameters").

- *Order Code* – order code:

  - This field is mandatory for template orders. The code of a template order must be unique in the Accounting Scheme. The code's uniqueness is checked when the template is checked and when approving the corresponding Accounting Scheme.
  - This field is optional for general orders. It is not necessary for the code of a general order to be unique. For example, orders of this category are used to set up a Supplementary order whose code must match the code of

another order (see the section "Supplementary Standing Payment Orders").

- *Priority* – determines the order in which standing payment orders configured for the contract's accounts are processed. First, the system processes standing payment orders with non-negative field values in ascending order (e.g. 0, 10, 20, 30). Then, amount normalisation is performed. Last, the system processes standing payment orders with negative field values in ascending order (e.g. –30, –20, –10).

  Note that orders related to a contract (inherited and individual orders are processed on a first-come, first served basis, as defined by priorities) are processed first, and then "General" orders (according to their priorities).

- *Source Account* – this field is used to redefine the source account from which the payment order draws its funds. Thus, the order amount can be calculated for one account, but posting can take place from another account of the contract.

  When interest accrual is redefined, this field is used to redefine the account to which interest is accrued.

- *Order Category* – payment order category (see "Categories of Standing Payment Orders"). The field can take on one of the following values:
  - "General" – a general standing payment order.
  - "Template" – a template standing payment order.

Determining Document Frequency and Amount

Fields for configuring creation frequency and amount of documents generated for standing payment orders are described below.

- *Date Event* – used in conjunction with the *Event Day* field to configure the frequency of document creation for standing payment orders, as well as the order of acceptance for these documents:
  - "Daily" – daily generation of a document for a payment order. The *Event Day* field is not filled in. The document is generated and processed by the "Contracts – Daily Update" procedure.
  - "Weekly" – a document for the standing payment order is created once a week. The date is determined as follows: the *Event Day* field shows the day of the week on which the document is created and posted by the "Contracts – Daily Update" procedure (see the "Contracts Daily Update Procedure" section in the Daily Procedures User Manual); the day of the week is indicated by numbers (1, 2, …, 7), where "1" = "Monday".
  - "Monthly" – a document for the standing payment order is created once a month. The creation date is determined as follows: the *Event Day* field shows the calendar day on which the document is created and posted by the "Contracts – Daily Update" procedure.
  - "Monthly – a document for a standing payment order is generated once a month. The order's activation time (date) depends on the "Morning"/"Evening" mode set by the ORDER_IN_START_OF_DAY

parameter (see the section "Processing Standing Payment Orders ("Morning"/"Evening" Mode)").

- ♦ For "Morning" mode, a payment order activates when the day after the day specified in the *Event Day* field is opened. If the date for activation falls on a weekend/holiday, the order will activate when the first working day opens after the weekend/holiday (when "Monday" opens).
- ♦ For "Evening" mode, an order activates when the day specified in the *Event Day* field closes. If the date for activation falls on a falls on a weekend/holiday, the order will activate when closing the last working day before the weekend/holiday (when closing "Friday"). If activation of an order must be shifted to when the first working day after the weekend/holiday is closed (when closing "Monday"), the DUE_TO_WRK_DAY=Y tag must be set in the payment order's *Posting Details* field.

- "Quarterly" – a document is created once a quarter. The creation date is determined as follows: the *Event Day* field indicates the quantity of days at the beginning of the quarter after which the document is created and processed through the "Contracts – Daily Update" procedure. If the date falls on a non-working day, document creation and posting take place on the next banking day.
- "Yearly" – a document is created once a year. The creation date is determined as follows: a year plus the number of months specified in the *Event Day* field must elapse after the day the contract is opened until the document is created and processed through the "Contracts – Daily Update" procedure. If the indicated date falls on a non-working day, document creation and posting take place on the next banking day. For instance, if the contract is opened on 10 January, 2009 and the value entered in the *Event Day* field is "5", a document as to the payment order will be created on 10 June, 2010.
- "Billing Date" – a document is created once a given billing cycle. The creation date is determined as follows: the *Event Day* field indicates the quantity of calendar days at the beginning of the billing cycle after which the document is created and processed through the "Contracts Daily Update" procedure. The length of the billing cycle itself is defined through the Accounting Scheme configuration. If the indicated date falls on a non-working day, document creation and posting take place on the next banking day.
- "Single" – used to describe a "single" payment, which, unlike regular payments, is made whenever it is required, for example, through an ATM.
- "Interbranch" – used to configure special standing payment orders for fee transfers in interbranch operations (see an example under subheading "Transferring Fees to Other Financial Institutions"). It works as follows: during an interbranch operation, a document is created to transfer the fee amount from the source branch revenue account to the target branch revenue account. Initially, it has the "Waiting" status. Until the document is posted and has the "Posted" status, every identical operation will increase the document amount. After the document is posted, a new

document will be generated when executing the next interbranch operation. It is not possible to create an individual standing payment order of this type.

- "Event Opened" – used to activate a standing payment order through an opened Event indicated in the *Event Type* field. A document is created and posted when the Event is opened. It is not possible to create an individual standing payment order of this type.
- "Event Closed" – used to activate a standing payment order through a closed Event indicated in the *Event Type* field. A document is created and posted when the Event is closed. It is not possible to create an individual standing payment order of this type.
- "Normalisation" – in this case, the payment order serves to redefine amount normalisation (see the "Limit Normalisation" section of the WAY4™ Accounting Schemes Administrator Manual). A payment order of this type only generates a macrotransaction. A macrotransaction is created and processed prior to the normalisation procedure (for an example, see "Multicurrency Normalisation"). For the processing to take place after normalisation, a negative value must be assigned to the *Priority* field (see the description of this field under subheading "Transaction Description"). Payment orders of this type must contain the "General" value in the *Order Category* field.
- "Account Due" – redefines due normalisation for all due normalisation types except for "End Cycle Due" and "Quarter". For instance, field *Target Template* of a standing payment order redefines field *Due Template* of an account template. A payment order of this type only generates a macrotransaction. It is not possible to create an individual payment order of this type.
- "Account Interest" – – redefines interest accrual. For instance, the *Source Account* field of a standing payment order redefines the *Interest Template* field of an account template. A bank contract and a type of an account of a bank contract used for interest accrual can be redefined in fields *Target Number* and *Target Acc Type*, respectively. If they do not need to be redefined, specify in fields *Target Number* and *Target Acc Type* the bank contract and account type specified in the account template. A payment order of this type only generates a macrotransaction. It is not possible to create an individual payment order of this type.
- "Supplementary" – this value is used when configuring supplementary payment orders (see "Supplementary Standing Payment Orders"). A payment order of this type only generates a macrotransaction.
- "Interest by Credit" – this type of payment order is used to accrue interest for the elapsed part of the billing period when transferring money from this account to another account of the same contract. A payment order of this type only generates a macrotransaction.
- "End Cycle Due" – a document for the standing payment order is generated and processed by the "Contracts – Daily Update" procedure executed when opening a billing cycle. The document is processed before due normalisation of accounts with due normalisation types

"End Cycle Due" and "Quarter". To process it after normalisation, specify a negative value in the *Priority* field of the standing payment order (see a description of the field in section "Transaction Description").

- "Use Contract Date" – a document for a payment order is generated for a specific contract functional date. The functional date that will be used is specified in the order's *Posting Date* field using the USE_DUE_DATE=<date code> tag. For example, USE_DUE_DATE=DUE_DATE; by default (if the tag is not set, "DueDate" (DUE_DATE) is used to process orders. See the section "Functional Dates" of the document "Contract Functional Dates".
- "Custom" – the frequency of generating documents for a standing payment order is defined using a custom procedure.

- *Event Day* – used in conjunction with the *Date Event* field to configure the frequency of document creation for standing payment orders.
  - If *Date Event* contains the "Daily" value, this field is not filled in
  - If the *Date Event* field contains "Weekly", "Monthly", "Quarterly", "Yearly", or "Billing Date", then the *Event Day* field indicates the quantity of calendar days at the beginning of the week, month, quarter, year, or billing cycle after which a document should be created.
- *Amount Event* – used in conjunction with the *Order Type* field to specify a criterion for generating a document according to the standing payment order.
- *Target Amount* – defines a fixed document amount.
- *Amount Percent* – defines a document amount as a percentage of the amount remaining in the account.
- *Amount Curr* –currency of the document amount.
- *Event Type* – the type of Event that initiates the payment (for more details on working with system Events, see the Events Administrator Manual).

  Note that a payment is initiated for a contract for which an Event was opened. I.e. if a payment order uses an account balance amount (for example, when the value of the *Order Type* field is "Downgrade To"), when the Event specified in the *Event Type* field opens, the account balance of the contract for which the Event was called will be examined. If the Event is called for a subordinate contract, the balance will be taken from the subordinate contract.
- *Check Target* – used to check whether it is possible to generate a document in the event that this account is replenished from an account belonging to another contract registered in the system. The field can take on one of the following values:
  - "Check Available" – a document will be generated only if the calculated amount of the document does not exceed the Amount Available of the debited contract.
  - "Check Balance" – a document will be generated only if the calculated amount of the document does not exceed the Amount Available minus the credit amount.

  - "Advice" – a document will be generated regardless of the debited contract's Amount Available and balance.
  - "No Check" – no check of the debited contract's Amount Available or balance is executed.

- *May Be Partial* – specifies whether or not a partial transfer is possible. The field can take on either "Yes" or "No". It is used in the following way: if "Yes" is specified in the field and the calculated document amount is higher than the Amount Available of the debited contract, the amount of the document will be equal to the Amount Available of the debited contract.
- *Balance Type* – drop-down list of balance types registered in the system. The value of the balance type specified in the field will be used to calculate the standing payment order amount instead of the balance of the account.
- *Tgt Balance Type* – drop-down list of balance types registered in the system, used as follows: if "Check Available" or "Check Balance" is selected in the *Check Target* field, a document will only be generated if the calculated document amount does not exceed the value of the specified balance type.
- The *Max Amount* and *Min Amount* fields allow the maximum and minimum amount of a standing payment order to be limited.

 The *Target Amount* and *Amount Percent* fields must not be used at the same time.

 If the *Event Type* field indicates a name of an Event and the *Date Event* field indicates a value other than "Event Opened" or "Event Closed", then the indicated Event will be opened when the standing payment order is activated.

 If flexible dates of activation will be used for a general payment order (for example, in different situations, an order may be activated daily, weekly, or monthly), set up several orders and use a classifier to activate the required order.

Defining Corresponding Accounts

There are a few set ways to define corresponding accounts for standing payment orders. Accordingly, fields in the form for defining the corresponding account are also divided into groups. When using one group of fields to define a corresponding account, the other groups of fields should be left blank.

In determining the corresponding account of a standing payment order, the following instructions are used:

- If the corresponding account is an account of the same contract for which the standing payment order is configured, the *Target Template* field is filled in with the name of the corresponding account.
- If the corresponding account is an account of a contract linked to the current contract through a Liability, Main/Sub or Related link, the following fields are used:
  - The *Target Spc* field indicates a related contract if the corresponding account belongs to one; if not, this field is not filled in.
  - The *Use Liability* field may contain one of the following values:

- ♦ "Yes", if the corresponding account belongs to a higher liability contract (Full Liability Main or Affiliated Main)
- ♦ "To Sub", if the corresponding account belongs to a lower liability contract; to select a specific contract from the list of lower liability contracts, specify the code of the Product used by the required contract in the *Target Number* field.

 The Liability contract's Product code must be specified in the *Target Number* field, otherwise the order will be activated. If a contract has several subordinate Liability contracts, the order will be activated once for an arbitrary subordinate Liability contrat with this Product.

- ♦ "No" (or blank field) – in other cases

▪ The *Target Acc Type* field should indicate a name of the payment target contract's account type. The field value is selected from the list of all account types registered in the system (see the "Account Types" form found at the "Full → Configuration Setup → Accounting Setup → Account Types" menu path).

- To configure a standing order to make a direct payment (for example, a public utility payment to a standard payee through an ATM, the *Payment Type* and *Counter Party* fields must be filled in. For more information, see the section "Direct Payments with Standing Payment Orders".

   For standing payment orders on public utilities, the *Date Event* field should read "Single".

- If the payee is registered in the WAY4 database and it is possible to identify its contract in the system by its ID, then the following fields are filled in:
  - ▪ The *Target Number* field indicates the number of the target contract in WAY4.
  - ▪ The *Target Acc Type* field indicates the account type name of the target contract. It is selected from a list of all account types registered in the system (see the "Account Types" form, which is opened by clicking the "Full → Configuration Setup → Accounting Setup → Account Types" menu item).
- If the payee is identified in an external payment system by the contract's RBS number, the following fields are filled in:
  - ▪ The *Target Member ID* field should indicate the value of the client bank identifier corresponding to the bank ID contained in the *Bank ID Code* field of the "RBS Bank Identification Codes" table (see the "BIC Table" section in the Acquiring Module User Manual).
  - ▪ The *Target Number* field should indicate the RBS number of the payee's contract (for example, the client's settlement account in the bank).

The following fields are also used to define the corresponding account:

- *Target Details 1*, *Target Details 2*, *Payment Details*, and *Posting Details* – fields for entering additional information on the corresponding account, e.g. the payee's tax identification number.

If the *Payment Details* field is filled in for a standing payment order, its value is copied to the *Transactions Details* field of the document generated for the payment order. Otherwise, the *Transactions Details* field of the generated document reads "Standing payment order".

The *Posting Details* field is used exclusively to enter tagged information (see the section "Tags in the *Posting Details* Field of a Standing Payment Order"). In particular:

- The *Posting Details* field is used if it is necessary to redefine the corresponding account of a general payment order on the contract level. This means that in template generation of documents for all contracts using this Accounting Scheme, the target account can be redefined for a certain contract. This setting is made using contract tagged parameters: the tags TGT_NUM_TAG=<TAG_NAME>;TGT_MBR_TAG=<TAG_NAME>; are set in the *Posting Details* field (for more information, see the section "Tags in the *Posting Details* Field of a Standing Payment Order"). These tags with the corresponding values are specified on the contract level.
- The *Posting Details* field can be used to configure supplementary standing payment orders (see the section "Supplementary Standing Payment Orders") and to determine the amount of a document.

The value of the *Posting Details* field is copied to the *Reason Details* field of the document generated for the standing payment order.

- The *Order Comment* field is used to enter general comments explaining the purpose of a payment order.

Activating/Deactivating Standing Payment Orders

Inherited standing payment orders are created in an inactive state. They can only be activated on the account level.

If the template payment order indicates the "No" value for activation, the inherited standing payment order also becomes inactive, although it may contain the "Yes" value in the *Is Active* field.

In addition to manual activation/deactivation of standing payment orders, the system also allows for activating/deactivating them automatically in cases where:

1. A specified date arrives
2. A contract is moved from one behaviour type to another
3. A specified State is activated/deactivated
4. A specified Event is opened/closed.

In the first case, standing payment order activation and deactivation dates are specified in fields *Date From* and *Date To*, respectively. If the field *Date From* is filled in, but the field *Date To* is left blank, the standing payment order will be activated on the date specified in the field *Date From* and remain active until it is deactivated through other mechanisms. If the field *Date To* is filled in, but the field *Date From* is left blank, the standing payment order will be active since it

is activated through other mechanisms until the date specified in the field *Date To*.

For the second case, the *If Beh Type* field is used. The standing payment order will be activated when the contract is moved to the behaviour type indicated in this field. The field filled in by selecting a behaviour type from a behaviour group set up for the Accounting Scheme (see the "Form "Account Schemes"" section of the WAY4™ Accounting Schemes Administrator Manual).

For the third case, the *Active if State* field is used. It indicates the State whose activation will activate the standing payment order (see the States Administrator Manual). This field is filled in by selecting a State from the list, which is generated depending on the contract category. For private issuing contracts, for instance, this list is generated using the "Full → Configuration Setup → Products → Issuing Private Products → Issuing Event States" menu item. For acquiring contracts, the "Full → Configuration Setup → Products → Acquiring Products → Acquiring Event States" menu item is used.

⚠ Note that value "Yes" must be specified in the *Is Active* field of standing payment orders whose field *Active if State* is filled in.

For the fourth case, the *Posting Details* field of the standing payment order (see Fig. 1 in section "Parameters of General/Template Standing Payment Orders") is used. It should contain the "‹Event Status›:‹Event Code›:‹Sign›;" value, where:

- ‹Event Status› takes the "P" value if the standing payment order is activated/deactivated when the Event is opened or the "C" value if the standing payment order is activated/deactivated when the Event is closed
- ‹Event Code› is the code of the Event whose opening/closing activates/deactivates the standing payment order.
- ‹Sign› takes the "+" value if the standing payment order is activated when the Event is opened/closed or the "–" value if the standing payment order is deactivated when the Event is opened/closed.

For example, to activate a standing payment order when an Event with the "E1" code is closed, it is necessary to specify the "C:E1:+;" value in the *Posting Details* field. To deactivate a standing payment order if an Event with the "E2" code is opened, it is necessary to specify the "P:E2: –;" value in the *Posting Details* field.

Supplementary Standing Payment Orders

The system allows documents to be created for standing payment orders as a result of a document being created for some other standing payment order. In this case, the values of the *Order Code* fields of the two payment orders should be the same.

By default, supplementary payment orders use the original order's amount and currency. If the amount and currency of the supplementary payment order itself must be used for a supplementary payment order, set the OWN_AMOUNT tag in the order (see the section "Tags in the *Posting Details* Field of a Standing Payment Order").

Supplementary orders are created with the "General" category.

"Credit Amount" or "Debit Amount" may be specified in a Supplementary order's *Order Type* field.

It is possible to create documents for supplementary standing orders through the following processes:

- Interest accrual on the amount remaining in the account; for this, indicate "INT=‹Debit Account Type Code›‹Credit Account Type Code›;" in the *Posting Details* field of the standing payment order (see Fig. 1 in section "Parameters of General/Template Standing Payment Orders").
- Due normalisation, for this, indicate "DN=‹Debit Account Type Code›‹Credit Account Type Code›;" in the *Posting Details* field of the standing payment order (see Fig. 1 in section "Parameters of General/Template Standing Payment Orders").
- Limit normalisation, for this, indicate "LN=‹Debit Account Type Code›‹Credit Account Type Code›;" in the *Posting Details* field of the standing payment order (see Fig. 1 in section "Parameters of General/Template Standing Payment Orders").
- Change of the credit limit amount; for this, indicate "CL=+;" or "CL=−;" in the *Posting Details* field of the standing payment order (see Fig. 1 in section "Parameters of General/Template Standing Payment Orders").

See the section "Tags in the *Posting Details* Field of a Standing Payment Order".

In the above examples, ‹Debit Account Type Code› and ‹Credit Account Type Code› refer to the debit account type code and credit account type code respectively (see the "Account Types" section of the WAY4™ Products. Accounting Schemes Administrator Manual).

Configuring Redefinition of Template Order Parameters in a Contract

When generating an order according to a template, WAY4 allows a number of template order parameters to be redefined with values set on the level of a specific contract (or on the tariff level). To do so:

- Special predefined tags (see below) are used in a payment order's *Posting Details* field. As their values, arbitrary tags are specified that are checked on the contract level and the values of which are used instead of template parameters. The list of predefined tags:
  - ORDER_PCNT_PARM – sets an arbitrary tag used in a contract to redefine the value of the *Amount Percent* (AMOUNT_PERCENT) field of the template order.
  - ORDER_MIN_PARM – sets an arbitrary tag used in a contract to redefine the value of the *Min Amount* (MIN_AMOUNT) field of the template order.
  - ORDER_MAX_PARM – sets an arbitrary tag used in a contract to redefine the value of the *Max Amount* (MAX_AMOUNT) field of the template order.ORDER_AMNT_PARM – sets an arbitrary tag used in a

contract to redefine the value of the *Amount Event* (AMOUNT_EVENT) field of the template order.

- ORDER_TGT_AMNT_PARM – sets an arbitrary tag used in a contract to redefine the value of the *Target Amount* (TARGET_AMOUNT) field of the template order.
- TGT_NUM_TAG and TGT_MBR_TAG – sets arbitrary tags tags used on the contract level to redefine the corresponding account (the *Target Number* and *Target Member ID* fields of the template payment order, respectively).

  See the section "Tags in the *Posting Details* Field of a Standing Payment Order"

- On the contract level, in the ADD_INFO field, arbitrary tags are specified that were defined as values in the previous step, and their values are set.

For example, the ORDER_PCNT_PARM=ORDER_PCNT; tag is set in the *Posting Details* field of a message template. The contract's ADD_INFO field contains the ORDER_PCNT tag with a value for redefining the *Amount Percent* (AMOUNT_PERCENT) parameter of the order. This value is used when generating an order for a contract.

Tariff type codes of tariffs with the "Threshold" role can be set as ORDER_PCNT_PARM, ORDER_MIN_PARM, ORDER_MAX_PARM, ORDER_AMNT_PARM, and ORDER_TGT_AMNT_PARM tag values. In this case, the values of the corresponding fields will be redefined on the tariff level. For more information, see the document "WAY4™ Advanced Tariff Management".

Special Grids for Entering General/Template Standing Payment Orders

To create and configure different types of standing payment orders, it is convenient to use special grids. Buttons that invoke each grid are presented in the "Definition for ‹name of Accounting Scheme›" grid (see the "Full Information about Accounting Scheme Templates" section of the WAY4™ Products. Accounting Schemes Administrator Manual):

- The [SO Due] button invokes the "SO Due" grid, which is used to redirect due normalisation configured for the account template in the Accounting Scheme.
- The [SO Norm] button invokes the "SO Norm" grid, which is used to redirect amount normalisation configured for the account template in the Accounting Scheme. An example of how this type of standing payment order is used can be found under subheading "Multicurrency Normalisation".
- The [SO Evnt Base] button invokes the "SO Evnt Base" grid, which is used to configure a standing payment order for which the system creates a document when an Event is opened or closed. An example of how this type of standing payment order is used can be found under subheading "Activating Standing Payment Orders by Events".

- The [SO Interest] button invokes the "SO Interest" grid, which is used to configure a standing payment order for redirecting interest accrual. An example of how this type of standing payment order is used can be found under subheading "Instead Orders".

Parameters of Inherited Standing Payment Orders Created on the Basis of a Template

Inherited standing payment orders are created on the basis of a template on the contract level manually or using the Advanced Applications module.

An inherited standing payment order is manually created in a contract's form (for example, Full → Issuing → Contract Input & Update → Issuing Contract (Private)):

- Click the [Activate] button in the contract form and execute the "Create Order by Template" context menu command. The "Create New Order" form will open (see Fig. 2).

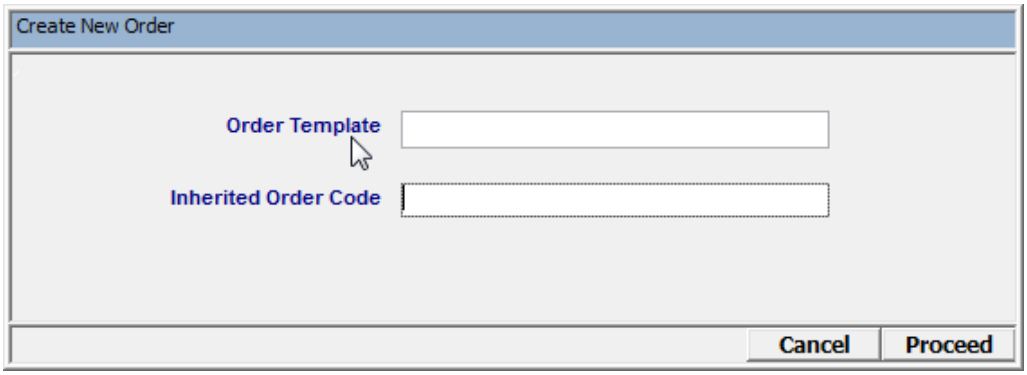

Fig. 2. Choosing a template for creating an inherited standing payment order

- Fill in the following fields in the "Create New Order" form:
  - In the *Order Template* field's list, choose the code of the template in the Accounting Scheme, based on which the payment order is being created.

    If one template order is set up in the Accounting Scheme, the *Order Template* field can be left empty. In this case, the template order code will be determined automatically.
  - In the *Inherited Order Code* field, specify a unique code that will be assigned to the inherited payment order. This code will be used to search for the order, for example, to make changes using the Advanced Applications module.

    If this field is left empty, a unique code for the order will be generated automatically.

  The form's fields are filled in similarly to a general/template payment order form's fields (see the section "Parameters of General/Template Standing Payment Orders").

- After filling in the fields, click the [Proceed] button. An inherited payment order will be created for the contract. Inherited standing payment orders are created in an inactive state. The form for the new inherited order will open.
- Inherited standing payment orders are additionally configured and activated in the "Create Order by Template" form that opens automatically after creation of an inherited payment order, or in the "Pers Orders for <account name>" form opened by clicking the [Pers Orders] button in the contract form or in the "Accounts for <contract name>" form. "Create Order by Template" and "Pers Orders for <account name>" form fields are the same, see Fig. 3.

  Existing (created before version 03.41.30) individual payment orders for issuing and acquiring contracts are shown when the [Pers Orders] button is clicked. Starting from version 03.41.30, individual payment orders can only be created for bank contracts.

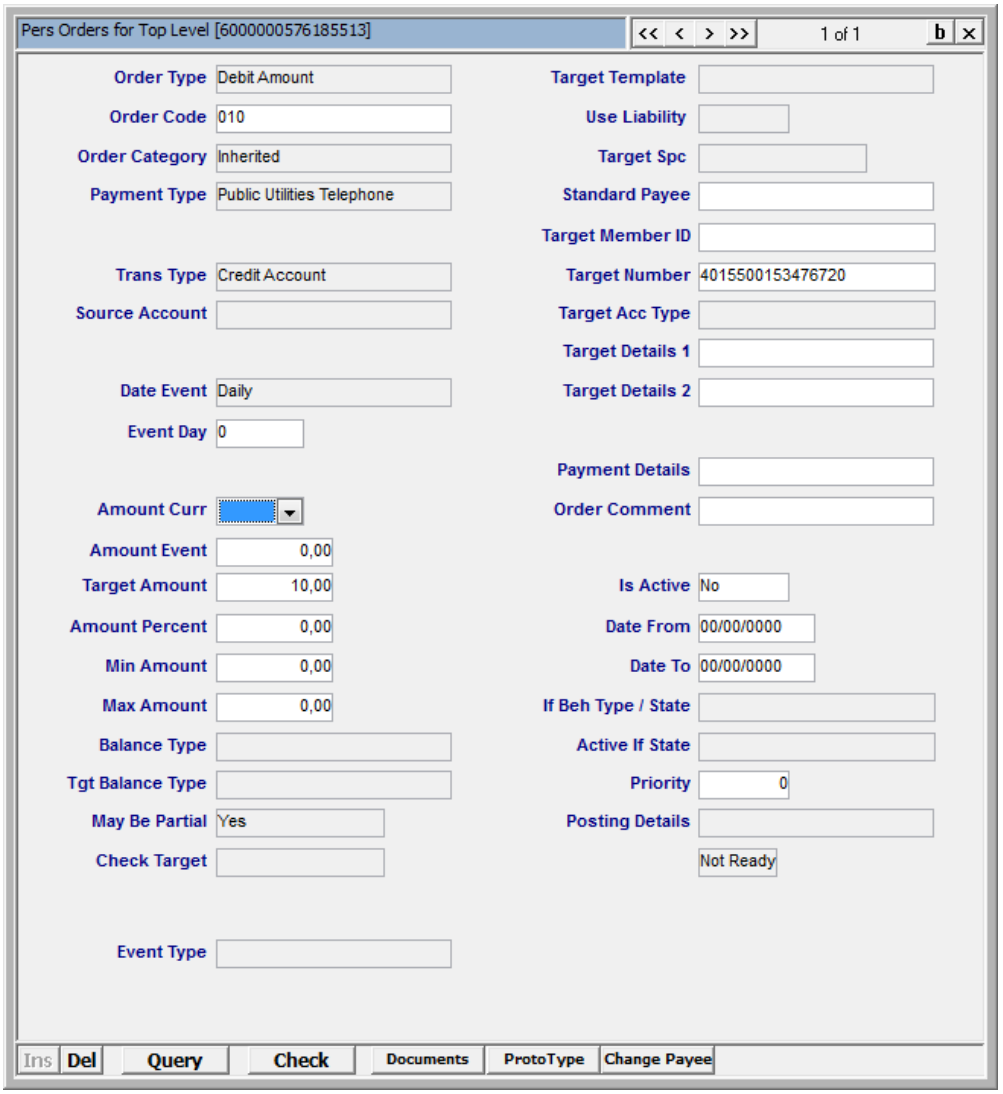

Fig. 3. Form for an inherited standing payment order

Fields for inherited standing payment orders are separated into two groups:

- Standing payment order custom fields. This group consists of the following fields:

- ♦ Some fields describing a payment's corresponding account: *Counter Party, Target Member ID, Target Number, Target Details 1, Target Details 2, Payment Details*.
- ♦ Fields determining a document amount (*Amount Curr*, *Amount Event*, *Target Amount*, *Amount Percent*, *Min Amount*, *Max Amount*).
- ♦ Field determining the frequency for generating a document – *Event Day*.
- ♦ The fields *Order Code*, *Order Comment*, *Is Active*, *Date From*, *Date To*, *Priority*.

Changes to these fields made in template payment orders are not reflected in inherited orders.

- ♦ The *Payment Type* and *Date Event* fields can be edited if the UNIQUE_INHERITED; tag is not set in the template used to create the order. If the values of these fields were redefined in an inherited order, they will not be synchronized with the template. If they were not redefined, when the values of these fields are changed in a template order, they will be updated in the inherited order.

▪ The rest of the form's fields are template (inherited) fields of a standing payment order. This group of fields cannot be edited on the contract level (these fields are filled in when generating a template payment order). When the values of these fields are changed in the Accounting Schme, they are updated in inherited standing payment orders.

The *Posting Details* field is included in the set of (inherited) fields synchronized with the template. I.e. if the *Posting Details* field is filled in for an application to create/change an inherited order, this data will not be saved.

The [Prototype] button is used to view the template payment order based on which a given inherited payment order was created.

When the [Check] button is clicked, WAY4 checks that standing payment order fields have been filled in correctly, and informs the user of the check results. WAY4 creates a standing payment order document for a null amount. If the check is successful, the document will have the "Waiting" status; otherwise the document will have the "Closed" status.

The frequency for generating a standing payment order can be redefined on the contract level using the following settings:

- The *Date Event* field value (type of frequency for activating an order) is redefined using the tag DATE_EVENT_TAG. See the section "Tags in the *Posting Details* Field of a Standing Payment Order".
- The *Event Day* field value is redefined using the tag DATE_EVENT_DAY_TAG. See the section "Tags in the *Posting Details* Field of a Standing Payment Order".

  An inherited payment order will activate with a specified frequency on specified days. For example, the same payment order with the "Monthly"

value of the *Date Event* field may activate on different days of the month, depending on the value of the tag in the contract.

For inherited orders created before version 03.41.30, when the *Order Code* field of a template and inherited order was not mandatory, note the following:

- To save requisites of inherited payment orders defined on the contract level when changes are made in Accounting Schemes (i.e. when changing the template of an inherited order), the template's *Order Code* field must be filled in. When changing an Accounting Scheme if there is a similar template in the target scheme, with the same code, inherited payment order requisites will be saved. Otherwise, inherited payment orders are deactivated and requisites are not saved.
- To save requisites of inherited payment orders when transferring a contract to another financial institution. If there is a similar template in the target institution, with the same code, inherited payment orders requisites will be saved. Otherwise, inherited payment orders are deactivated and requisites are not saved.

To set up standing payment orders for paying utilities to an individual payee at an ATM see the section "Direct Payments with Standing Payment Orders".

Parameters of Individual Standing Payment Orders

Individual standing payment orders created without a link to a template are entered and edited in the "Indv Orders for <name of account>" form opened by clicking the [Indv Orders] button in a bank contract's "Accounts for <name of contract>" form.

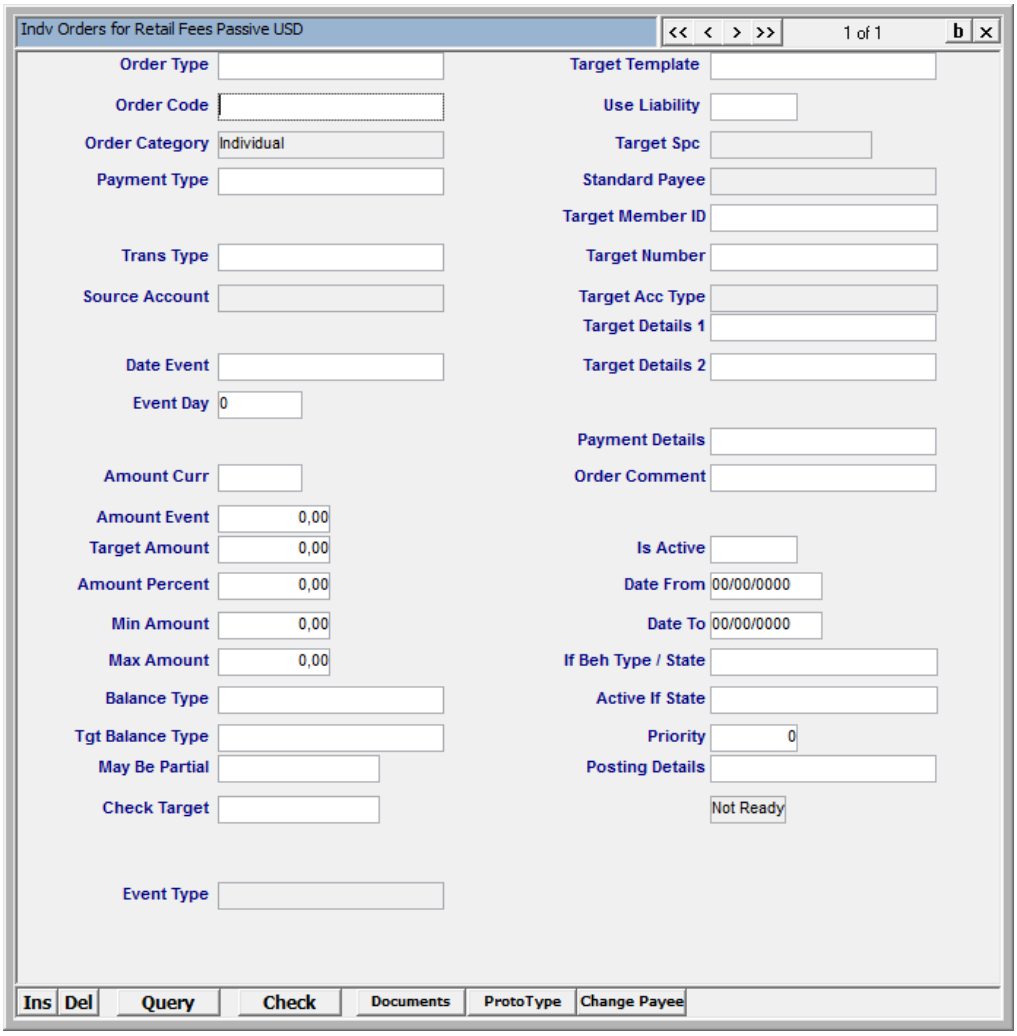

Fig. 4. Form for an individual standing payment order

The fields of the form are filled in the same way as the fields of general/template payment order form (see "Parameters of General/Template Standing Payment Orders").

Some of the fields displayed in the general/template standing payment order are not available in the personal payment order form due to limitations on creation of personal standing payment orders of specific types (see section "Transaction Description", description of field *Order Type*).

Direct Payments with Standing Payment Orders

Direct payments are made, for example, when paying utilities at an ATM. In doing so, financial documents to the payee are generated on the basis of payment orders; funds are transferred bypassing the acquirer on whose device the payment is being made.

For more information about direct payments, see the section "Payment (Transaction) Party Bank Requisites" in the document "Documents".

In standing payment orders, recipients of a direct payment are specified using the *Payment Type* and *Counter Party* fields:

- *Payment Type* – specifies the payment type (for example, type of utility). To work with the list of payment types, use the form "Payment on Account Types" (Full → Configuration Setup → Transaction Types → Payment on Account Types).
- The *Counter Party* field is filled in from a special list of payment parties registered in WAY4 as follows:
  - For a general/template standing payment order, a payment party is selected from the list "Payees for <name of payment type>" (standard payment parties) configured according to payment type in the form "Payment on Account Types" (Full → Configuration Setup → Transaction Types → Payment on Account Types → [Payees]).
  - Within inherited standing payment orders, custom lists of payment parties can be used. Payment parties are registered:
    - In the form "Payees for <name of client>", subordinate to the form with client parameters (for example, "Full → Issuing → Contracts Input & Update → Clients (Private) → [Client – Edit] → [Payees]").
    - In the "Payees for <contract name>" form, in the child form with contract parameters (for example, "Full → Issuing → ContractsInput & Update → Issuing Contracts (Private) → [Payees]").

    For more information about configuring payment party requisites, see the section "Payment (Transaction) Party Bank Requisites" in the document "Documents".

When processing a document created according to a standard payment order referring to a payment party from the list of individual or standard payment parties, information about payee and payer requisites is registered under the document.

- Transaction target data is taken from the "Parties for <name of payment recipient>" form referred to by the payment order.

  When processing a general order, if the appropriate party is not found for the contract, a search will be made for a party for a client record.
- Transaction source requisite data is automatically taken from the system.

This data is viewed by clicking the [Parties] button in the document form (Full → Documents Input & Update → Doc - General Form → Doc – General → [Parties]).

For more information, see the section "Payment (Transaction) Party Bank Requisites" in the document "Documents".

Chapter 2. Examples of Standing Payment Order Use

This section contains typical examples of the use of standing payment orders.

Retail Payments

Standing payment orders can be used to create daily documents through which funds in the form of issuer bankcard payments that have accumulated in the contract accounts of retail organisations will be transferred to the settlement accounts of those organisations.

To configure a standing payment order using a template, proceed as follows:

- For a merchant account template (for example, "Merchant Current") in the Accounting Scheme, configure a general standing payment order with the following parameters:
  - In the *Order Type* field, indicate value "Downgrade To".
  - In the *Trans Type* field, indicate value "Credit Account".
  - In the *Order Category* field, indicate "General".
  - In the *Date Event* field, select the "Daily" value from the list.
  - In the *Is Active* field, indicate "Yes".
- The payee is specified on the contract level (in the ext_data field) or in the Product (custom_data field). For more information, see the description of the tags TGT_MBR_TAG and TGT_NUM_TAG in the sections "Defining Corresponding Accounts" and "Tags in the *Posting Details* Field of a Standing Payment Order".

Public Utility Payments

Standing payment orders can be used to create payment orders that may be used to make online payments for public utilities.

The cardholder makes a payment through an ATM. As a result, the system creates a document in the database. Processing the document generates a charge from the contract account to the public utility payee's account. The funds will be blocked on the cardholder's account from the time the cardholder uses the ATM to the time the indicated amount is withdrawn from the account.

To configure a standard payment order using a template, proceed as follows:

- For a client deposit account template (for example, "Cl Deposit") in the Accounting Scheme, configure a template standing payment order with the following parameters:
  - In the *Order Type* field, indicate "Debit Amount".
  - In the *Trans Type* field, indicate "Credit Account".
  - In the *Payment Type* field, select a payment type from the list.

- In the *Order Category* field, indicate "Template".
- Set the *Date Event* field to "Single".
- Set the *Is Active* field to "Yes".

- Configure the inherited standing payment order. To do this, click on the [Pers Orders] button in the contract's account. The screen will display the "Pers Orders for Cl Deposit" form, which allows the inherited payment order to be edited. In this form, do the following:
  - Indicate the payee (see "Defining Corresponding Accounts").
  - Set the *Is Active* field to "Yes".

Multicurrency Normalisation

Multicurrency normalisation is used if the Accounting Scheme contains account templates in different currencies.

When there are normalising standing payment orders, multicurrency normalisation with these payment orders takes place regardless of the value of the global parameter MULTICURRENCY_NORMALIZATION (see the document "WAY4™ Global Parameters"). When performing multicurrency normalisation this way, currency is converted either at the main rate or at an FX Type rate specified in the Accounting Scheme.

To take an example: in the Accounting Scheme whose basic currency is EUR, there are two accounts, Cl Deposit (USD) and Cl Deposit (EUR) (see Fig. 5).

For Cl Deposit (EUR), it is necessary to configure a standing payment order with the following parameters:

- Field *Order Type* – Upgrade To
- Field *Trans Type* – Debit Account
- Field *Date Event* – Normalisation
- Field *Target Template* should show the template of the account that will be used as a backup account for insufficient funds; in this case, Cl Deposit (USD)
- Field *Is Active* should read "Yes"

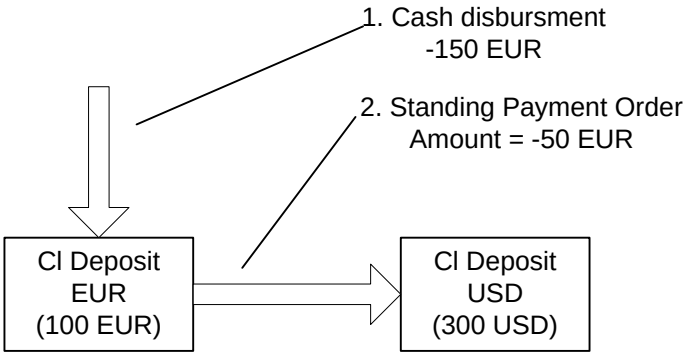

Fig. 5. Multicurrency normalisation through standing payment orders

When a transaction is executed, a charge is generated to the Cl Deposit (EUR) account. If the transaction amount exceeds limits set in the *Low Limit Amount* field, the overflow amount is withdrawn through a payment order from the Cl Deposit (USD) account. It should be kept in mind that the *Priority* field must indicate a positive value so that funds withdrawal takes place before the standard account normalisation procedure.

Multicurrency normalisation is also possible in Accounting Schemes with more than two currencies. For example, an Accounting Scheme may contain currencies EUR, USD and JPY. In this case, normalisation may include all three accounts. Their repayment priority for loans will depend on the defined standard payment order priority (see the description of the *Priority* field under subheading "Transaction Description").

In an example of multicurrency normalisation, it would be helpful to see how standing payment orders of the "Normalisation" type are used The example described above used a standing order of type "Upgrade To". In that case, normalisation was not considered for the Cl Deposit (EUR) account when calculating the charge amount to the Cl Deposit (USD) account.

For example, account "Cl Deposit (EUR)" is configured for account normalisation as shown in Fig. 6. In this case, if a standing payment order is configured with the parameter values indicated earlier, a charge for only 250 euros is generated to the "Cl Deposit (USD)" account when executing the transaction. For the "Cl Deposit (USD)" account to be charged with the amount including normalisation for the "Cl Deposit (EUR)" account (250 euros + 200 euros + 17 euros), a standing order with the following parameters should be configured for this account:

- Field *Order Type* – "Normalisation"
- Field *Trans Type* – "Debit Account"
- Field *Date Event* – "Normalisation"
- Field *Target Template* should show the template of the account that will be used as a backup account for insufficient funds; in this case, "Cl Deposit (USD)"
- Field *Is Active* should be set to "Yes"

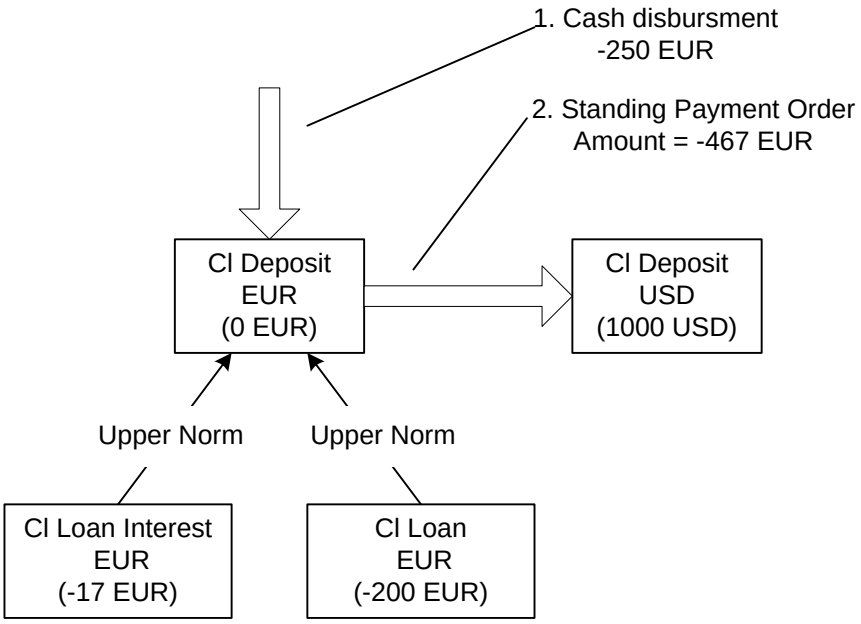

Fig. 6. Multicurrency normalisation through standing orders, with account normalisation included

When the transaction document is processed, a charge is generated to the Cl Deposit (EUR) account. A document is created for the standing order replenishing the Cl Deposit (EUR) account with funds from Cl Deposit (USD). The document amount is a composite of the overdraft amount and the loan amount of the accounts associated with the Cl Deposit (EUR) account through upper-limit normalisation. In this example, they are Cl Loan (EUR) and Cl Loan Interest (EUR). If there are not enough funds in the Cl Deposit (USD) account, the document amount will be adjusted to be equal to the amount remaining in the Cl Deposit (USD) account. The funds will be distributed between the Cl Deposit (EUR), Cl Loan (EUR) and Cl Loan Interest (EUR) accounts according to the amount normalisation configuration.

Transferring Fees to Other Financial Institutions

When executing interbranch operations, fees often need to be transferred to another branch's account. In this case, a special standing payment order is configured for the branch's fee account.

For example, the system registers financial institutions FI 1 and FI 2. The system must be configured so that when a card from FI 2 is acquired in an ATM belonging to FI 1, the appropriate fee is transferred to FI 1.

For this, a standing payment order with the parameters described below needs to be configured for a bank contract account belonging to FI 2. This is the account to which fees will be transferred according to the card contract's Target service (for example, the Client Fees Passive account of contract "002-Client Fee").

- *Order Type* – Debit Amount
- *Trans Type* – Credit Account
- *Order Category* – General

- If the *Date Event* field contains value "Interbranch", a document with the "Waiting" status will be created when executing an interbranch operation. This document will transfer fee funds to the target branch revenue account. Until posting occurs and the document takes on status "Posted", every subsequent operation will increase the document amount. When the next interbranch operation is executed, a new document will be created.
- *Target Number* – the number of the bank contract belonging to FI 1; fees will be transferred to this contract's account. The insitution's code in the contract number may be arbitrary (for example, "001-BRANCH_ACQ_FEES"). When a document is created, the institution's code in the contract number will automatically be replaced by the actual code of the counterparty financial institution. Therefore, it is not the document's target_number that falls in the contract number 001-BRANCH_ACQ_FEES (see above), but, for example, the number 123-BRANCH_ACQ_FEES (if the device belongs to an institution with the code 123).
- *Target Acc Type* – the type of the account of FI 1's bank contract to which fees will be transferred (for example, "Cash Fees Passive").
- *Is Active* – "Yes".

Changing Contract Behaviour Types

Consider the following example. A contract is moved from the first behaviour type to the second or higher. The following actions need to be executed automatically:

- Loan interest already accrued to the loan interest account must be transferred to an off-balance loan interest account (see "Activating Standing Payment Orders by Events").
- From that time to the time the contract returns to the first behaviour type, loan interest should be accrued to the off-balance interest account (see "Instead Orders").

Activating Standing Payment Orders by Events

For a client loan interest account (for example, "Cl Loan Int"), a standing payment order with the following parameters is configured:

- Field *Order Type* indicates the "Upgrade Advice" order type
- Field *Trans Type* contains the "Debit Account" value
- Field *Data Event* indicates "Event Opened"
- Field *Event Type* contains a name of an Event described earlier
- Field *Target Template* indicates an off-balance loan interest account (account template)
- Field *Is Active* is set to "Yes"

As a result of transferring a contract from the first behaviour type to the second or higher one, a preconfigured Event opens (see the Events Administrator Manual).

When the Event opens, the standing payment order is activated. This creates a document transferring the loan interest already accrued to the loan interest account (for example, "Cl Loan Int") to the off-balance loan interest account (for example, "Cl Loan Int Off").

Instead Orders (Redirecting Interest Accrual)

To redirect loan interest accrual when moving a contract from the first behaviour type to the second or higher one, an inactive standing payment order with the following parameters must be configured for the account from which interest is accrued (for example, "Cl Loan"):

- Field *Order Type* indicates the "Credit Amount" order type.
- Field *Source Account* indicates the off-balance loan interest account or account template (for example, "Cl Loan Int Off").
- Field *Date Event* indicates "Account Interest".
- Field *If Beh Type* indicates a behaviour type name. The standing payment order will be activated when the contract is moved to this behaviour type.
- Field *Target Acc Type* indicates the type of the off-balance loan interest account (for example, "Loan Int Accrual Off").
- Field *Is Active* reads "No".
- The *Target Number* field must be left empty.

When a contract is transferred from the first behaviour type to the second or higher one, the standing payment order will be activated. Until the contract returns to the first behaviour type, interest will be accrued to the off-balance loan interest account (for example, "Cl Loan Int Off").

When there are several Instead orders in the same account, an order's priority can be set using the *Priority* field. Orders will be processed in the following order (for example when there are orders with positive and negative *Priority* field values): 30, –20, –10, 0, 10, 20, 30.

Chapter 3. Tags in the *Posting Details* Field of a Standing Payment Order

Name	Value	Description
Tags in the *Posting Details* field of a standing payment order:		
ON_REQUEST		The tag is specified in a template payment order. This tag allows disabling of automatic generation of inherited payment orders on the contract level based on template payment orders when an Account Scheme is approved. In this case, a payment order is created on demand.
CUSTOM_AMOUNT		Allow the CUST_ORDER_AMOUNT procedure to be used for calculating the amount of a payment order. For detailed information, contact WAY vendor representatives.
OWN_AMOUNT		Supplementary payment orders by default use the amount and currency of the original payment order. This tag allows the supplementary payment order's amount and currency to be used for it.
SAVE_AMOUNT		If this tag is set, when a payment order with the *Event Type* field filled in activates, data on the amount and currency of a payment order will be saved in the *Posting Detail* field of this Event. This amount can be used to calculate the fee charged when the Event opens (for example, a fee can be set as a percentage of the payment order amount).
REQUEST	"Y" "N"	When this parameter is set ("Y"), the process for generating a document for a payment order analyses the amount available and the status of the target contract. If a document is generated successfully, it is assigned the category (REQUEST_CATEGORY) "Request". When the value is "N", no check is made, an "Advice" category document is generated.
REDEF_TARGET		This tag allows redefinition of a standing payment order's *Target Number, Target Spc, Target Member Id* fields. The tag is used, for example, to implement services for transferring funds from one card to another at an ATM (the transfer is made using a payment order). When configuring the order, the corresponding field (for example, *Target Number*) is not filled in, the number of the target card is entered with the ATM keypad.

Name	**Value**	**Description**
IN_WRK_DAY		If the tag is set, for orders for which the "Monthly", Quarterly" or "Billing Date" value is specified in the *Date Event* field, the "Event Day" value will be set in working days. For example, in a payment order, the "Month" value is specified in the *Date Event* field, and the value *"-1"* is specified in the *Event Day* field. In this configuration, a payment order will activate on the last day of the month (including on a non-working day). When the IN_WRK_DAY tag is specified, the payment order will be activated on the last working day of the month.
DATE_EVENT_DAY_TAG	<tag name>	Makes it possible to redefine the value of a standing payment order's *Event Day* field (date of order activation frequency): • The tag is specified in inherited payment order parameters in the format DATE_EVENT_DAY_TAG=<tag name>;, where <tag name> is an arbitrary tag name. • Using *DATE*_EVENT_DAY_TAG, the tag can be set on the contract level (ACNT_CONTRACT table) in the ADD_INFO field (ADD_INFO_01, ADD_INFO_02, ADD_INFO_03, ADD_INFO_04) or in the ext_data field, or on the Product level in the CUSTOM_DATA field. The tag is specified in the format <tag name>=<day (date)>;, where: ♦ <tag name> is a tag set with DATE_EVENT_DAY_TAG. ♦ <day (date)> is the day or date the corresponding inherited payment order is activated. Used, for example, so that the same payment order with the "Monthly" frequency, depending on the tag value (<tag name>), is activated on different days of the month.
DATE_EVENT_TAG	<tag name>	Makes it possible to redefine the value of the *Date Event* field (payment order activation frequency type): • The tag is specified in inherited payment order parameters DATE_EVENT_TAG=<tag name>;, where <tag name> is an arbitrary tag name.. • Using DATE_EVENT_TAG the tag can be set on the contract level (ACNT_CONTRACT table) in the ADD_INFO field (поля ADD_INFO_01, ADD_INFO_02, ADD_INFO_03, ADD_INFO_04) or in the ext_data field, or on the Product level in the CUSTOM_DATA field. The tag is specified in the format <tag name>=<frequency type code>;, where: ♦ <tag name> – is the tag set with DATE_EVENT_TAG. ♦ <frequency type code> – the frequency type code for the frquency with which the corresponding inherited payment order must be activated: "U" – "Use Contract Date"

Name	Value	Description
		"D" – "Daily " "W" – "Weekly" "M" – "Monthly" "Y" – "Yearly" "B" – "Billing Date" "Q" – "Quarterly" Used, for example, so that the same payment order, depending on the value of the tag (<tag name> in the document's ADD_INFO field, is activated with different frequency - monthly, quarterly or daily.
IF_CS_TYPE	<Classifier type code>	This tag sets the type of classifier checked. Used together with the IF_CS_VALUE, IF_NOT_CS_VALUE, IF_CS_TYPE_FOR tags. For more information, see the section "Executing Actions Depending on Classifier Values" of the document "WAY4™ Client and Contract Classifiers". It is possible to set a check of several classifiers for a contract by using the IF_CS_NUMB, IF_CS_TYPE<N>, IF_CS_VALUE<N>, IF_NOT_CS_VALUE<N>, IF_CS_TYPE_FOR<N> tags. See the description of the IF_CS_NUMB tag.
IF_CS_VALUE	<CS_STATUS_VALUE.CODE 1>, <CS_STATUS_VALUE.CODE 2>,…,<CS_STATUS_VALUE.CODE N>	A payment order is only activated if clients and their contracts correspond to the set value of this classifier. Several codes separated by commas may be specified as the value of this tag. Used together with the IF_CS_TYPE tag. For more information, see the section "Executing Actions Depending on Classifier Values" of the document "WAY4™ Client and Contract Classifiers". It is possible to set a check of several classifiers for a contract by using the IF_CS_NUMB, IF_CS_TYPE<N>, IF_CS_VALUE<N>, IF_NOT_CS_VALUE<N>, IF_CS_TYPE_FOR<N> tags. See the description of the IF_CS_NUMB tag.
IF_NOT_CS_VALUE	<CS_STATUS_VALUE.CODE 1>, <CS_STATUS_VALUE.CODE 2>,…,<CS_STATUS_VALUE.CODE N>	A payment order is not activated if clients and their contracts correspond to the set value of this classifier. Several codes separated by commas may be specified as the value of this tag. Used together with the IF_CS_TYPE tag. For more information, see the section "Executing Actions Depending on Classifier Values" of the document "WAY4™ Client and Contract Classifiers". It is possible to set a check of several classifiers for a contract by using the IF_CS_NUMB, IF_CS_TYPE<N>, IF_CS_VALUE<N>,

Name	Value	Description
		IF_NOT_CS_VALUE<N>, IF_CS_TYPE_FOR<N> tags. See the description of the IF_CS_NUMB tag.
IGNORE_DOC_VALIDATION		A standing payment order is automatically checked when approving a contract. To skip this step (to not check standing payment order parameters when approving a contract), specify the IGNORE_DOC_VALIDATION tag.
TGT_NUM_TAG	<name of tag for redefining target contract number "Contract#">	Used with TGT_MBR_TAG. These tags allow redefinition of the target contract number ("Contract #") and target ID ("Target Member ID") in a payment order. Tags set with TGT_MBR_TAG are specified in the contract (in the EXT_DATA, ADD_INFO_01/02/03/04 field) or in the Product (CUSTOM_DATA field) and determine the corresponding contract parameters. A payment order with these settings will transfer funds from the current contract to the contract specified in the contract's EXT_DATA field or Product's CUSTOM_DATA field.
TGT_MBR_TAG	<name of tag for redefining Target ID "Target Member ID">	see the definition of TGT_NUM_TAG
CALENDAR_TYPE	<name of business calendar type>	The tag is used so that a payment order is activated according to a calendar differing from the financial institution’s calendar.
FILL_SOURCE_FEE		This tag is used to show the amount of the fee charged to the merchant in a document generated for a standing payment order. The fee is reflected in the *Source Fee Amount* field of the document.
INT	‹Debit Account Type Code›‹Credit Account Type Code›	The tag makes it possible to generate documents for additional payment orders resulting from accrual of interest on an account balance.
DN	‹Debit Account Type Code›‹Credit Account Type Code›	The tag makes it possible to generate documents for additional payment orders resulting from due normalisation.
LN	‹Debit Account Type Code›‹Credit Account Type Code›	The tag makes it possible to generate documents for additional payment orders resulting from volume normalisation.
CL	"+" "-"	The tag makes it possible to generate documents for additional payment orders resulting from a change in credit limit.
STATUS_CATEGORY	"V" "I" "D"	A payment order will only activate if the contract is in the status with the specified category (the value of the *Is Valid* field of the "Contract Statuses" form "Full → Configuration Setup → Contract Types → Contract Statuses"):

Name	Value	Description
		• "V" – Valid • "D" – Decline • "I" – Invalid
IF_PARM	<tagged parameter name> CLIENT.SHORT_NAME CLIENT.FIRST_NAME CLIENT.LAST_NAME CLIENT.COMPANY_NAME CLIENT.AGE	This tag sets a checked classifier (tagged parameter) for a contract. Used together with the IF_PARM_VALUE tag. For more information, see the section "Classifiers without a Fixed List of Values" of the document "WAY4™ Client and Contract Classifiers". A number of client parameters can be checked. To do so, the following values can be specified as the tag's value: CLIENT.SHORT_NAME – check the *Short Name* field in the client form. CLIENT.FIRST_NAME – check the *First Name* field in the client form. CLIENT.LAST_NAME – check the *Last Name* field in the client form. CLIENT.COMPANY_NAME – check the *Company Name* field in the client form (place of work) CLIENT.AGE – client age. Determined according to date of birth (*Date of Birth* field) and the system date (i.e. age at the time of the request is determined). These client attributes do not have to be registered as contract custom parameters. It is possible to set a check of several parameters for a contract by using the IF_PARM_NUMB, IF_PARM<N>, IF_PARM_VALUE<N>, IF_PARM_FOR<N> tags. See the description of the IF_PARM_NUMB tag.
IF_PARM_VALUE	"<tagged parameter value>" "NOT_EMPTY" "EMPTY" "LIST_WITH:<list of values separated by commas>" "BETWEEN<value> AND <value>"	A payment order is activated only if clients and their contracts correspond to the set classifier value. Used together with the IF_PARM tag. The IF_PARM_VALUE=NOT_EMPTY; tag is used to check for the existence of a parameter (the tag set using the IF_PARM tag) without checking its value. When the IF_PARM_VALUE=EMPTY; tag is set, the order is activated if the specified parameter doesn't have a value. Note that with this value, the order will also be activated if the parameter is not present in the contract. When "IF_PARM_VALUE=LIST_WITH:<list of values separated by commas>" is set, an order is activated if even one value from the list (LIST_WITH) matches that set in the contract. When "IF_PARM_VALUE=BETWEEN<value> AND <value>" the order is activated if the value in the contract falls in the range set by the tag (for example, IF_PARM_VALUE=BETWEEN40 AND 50;). The value must be set with spaces, as shown in the example. It is possible to set a check of several parameters for a contract by using the

Name	Value	Description
		IF_PARM_NUMB, IF_PARM<N>, IF_PARM_VALUE<N>, IF_PARM_FOR<N> tags. See the description of the IF_PARM_NUMB tag.
ORDER_PCNT_PARM	<tag name>	The parameter ORDER_PCNT_PARM sets an arbitrary tag used in a contract (in the add_info field) to redefine the value of the *Amount Percent* field (amount_percent) of a template payment order. This field can be redefined on the tariff level; in this case, the tariff type code of a tariff with the "Threshold" role is set as the tag value (for more information, see the document "WAY4™ Advanced Tariff Management").
ORDER_MIN_PARM		The parameter ORDER_MIN_PARM sets an arbitrary tag used in a contract (in the add_info field) to redefine the value of the *Min Amount* field (min_amount) of a template payment order. This field can be redefined on the tariff level; in this case, the tariff type code of a tariff with the "Threshold" role is set as the tag value (for more information, see the document "WAY4™ Advanced Tariff Management"). For payment orders with the BY_BATCH property, the ORDER_MIN_PARM tag is also used to redefine the order's *Min Amount* (min_amount) field.
ORDER_MAX_PARM		The parameter ORDER_MAX_PARM sets an arbitrary tag used in a contract (in the add_info field) to redefine the value of the *Max Amount* field (max_amount) of a template payment order. This field can be redefined on the tariff level; in this case, the tariff type code of a tariff with the "Threshold" role is set as the tag value (for more information, see the document "WAY4™ Advanced Tariff Management").
ORDER_AMNT_PARM		The parameter ORDER_AMNT_PARM sets an arbitrary tag used in a contract (in the add_info field) to redefine the value of the *Amount Event* field (amount_event) of a template payment order. This field can be redefined on the tariff level; in this case, the tariff type code of a tariff with the "Threshold" role is set as the tag value (for more information, see the document "WAY4™ Advanced Tariff Management").

Name	Value	Description
ORDER_TGT_AMNT_PARM		The parameter ORDER_TGT_AMNT_PARM sets an arbitrary tag used in a contract (in the add_info field) to redefine the value of the *Target Amount* field (target_amount) of a template payment order. This field can be redefined on the tariff level; in this case, the tariff type code of *a* tariff with the "Threshold" role is set as the tag value (for more information, see the document "WAY4™ Advanced Tariff Management").
DUE_TO_WRK_DAY	"Y"	This tag is set for payment orders with the "Monthly" indicator. When the tag DUE_TO_WRK_DAY=Y; is set and the value of the global parameter ORDER_IN_START_OF_DAY is "N", if the date for activating the payment order falls on a non-working day (according to the Event Day parameter), the payment order will be processed on closing the first working day after the non-working days (when closing "Monday"). If the tag is not set, the payment order will be processed when closing the last working day before the non-working days (when closing "Friday").
CHECK_ROUTING	"Y"	The CHECK_ROUTING=Y; tag is set in a payment order *transferring funds* from a WAY4 contract account to an external account (to the account of a contract that is not registered in WAY4). The amount of such a payment order is calculated with consideration of the fee according to the corresponding Service of the source contract (the Service set up for the transaction type from the *Trans Type* field of the order). If CHECK_ROUTING=Y; is set, when calculating the order amount, the amount available in the source contract is checked. If the amount in the source contract is sufficient to make the payment for the order including the fee, a document is generated for the order. If funds are insufficient, an order is not activiated and no document is generated. If CHECK_ROUTING=Y; is not set, if the main amount for the order is present, a document is always generated, no check is made for sufficient funds in the source contract to make the payment including the fee. If when posting the document it turns out that funds on the account are not sufficient to debit the fee, the correspsonding amount goes to an OVL account.
SHIFT_MTR_GL_DATE	"+" "-"	The tag SHIFT_MTR_GL_DATE redefines the global parameter of the same name. Makes it possible to shift the date of posting entries to GL accounts resulting from activation of a payment order to the previous (when the value is "-") or next (value is "+") working day if the payment order activates on a non-

Name	Value	Description
		working day. For more information, see the description of the global parameter SHIFT_MTR_GL_DATE in the document "WAY4 Global Parameters".
BASE_DATE	CLIENT.BIRTH_DATE CLIENT.DATE_EXPIRE CLIENT.DATE_OPEN CLIENT.ADD_DATE_01 CLIENT.ADD_DATE_02 CONTRACT.FIRST_ACTIVITY_DATE YEAR "CONTRACT.DATE_OPEN" "CONTRACT.LAST_BILLING" "CONTRACT.NEXT_BILLING"	For a payment order with the "Yearly" value of the Date Event parameter, this tag makes it possible to redefine the start date for the period after which the payment order will be activated: "CLIENT.BIRTH_DATE" – the period starts from the client's birth date (from the date specified in the birth_date field of the CLIENT table). • "CLIENT.DATE_EXPIRE" – the period starts from the date_expire field of the CLIENT table. • "CLIENT.DATE_OPEN" – the period starts from the date of registering the client in the database (from the date specified in the date_open field of the CLIENT table). • "CLIENT.ADD_DATE_01" – the period starts from the date specified in the add_data_01 field CLIENT table. • "CLIENT.ADD_DATE_02" – the period starts from the date specified in the add_data_02 field of the CLIENT table. • "CONTRACT.FIRST_ACTIVITY_DATE" – the period starts from the date of the contract's first financial activity. The contract's FIRST_ACTIVITY_DATE tag is analysed, in which the date of the first financial activity is automatically specified. • "YEAR" – the period starts from the first day of the current year. • "CONTRACT.DATE_OPEN" – the period starts from the contract opening date. • "CONTRACT.LAST_BILLING" – the period starts from the start date of the last billing cycle. • "CONTRACT.NEXT_BILLING" – the period starts from the start date of hte next billing cycle. For example, when a contract is created, the last_billing date is set equal to the contract opening date (for example, 01.09.2016), the next_billing date is 01.10.2016. When BASE_DATE=CONTRACT.LAST_BILLING; (if no shift in calendar days is set), the order will be activated for the first time on 01.09.2016 and in the next year on 01.09.2017.
UNIQUE_INHERITED		If only one inherited order can be created according to a template, specify the UNIQUE_INHERITED tag in the template. The inherited order will be created with a code corresponding to the template code

Name	Value	Description
STORNO_ACTION	"DEFAULT" "RECALC" "SKIP" "REAPPLY"	The tag makes it possible when adjusting transactions to configure actions with orders and entries generated according to these orders. For more information, see the section "Working with the Reversal Management Module" of the document "Reversal Management.
BATCH_INFO_ENTRIES		When the BATCH_INFO_ENTRIES tag is set, invoice batches generated for a payment order are marked as containing informational records. Invoices from this batch cannot be debited or paid separately, one-by-one. Used, for example, when transferring funds from a cumulative "High Prescision" account (see the section "High Precision Accounting" of the document "WAY4 Accounting Schemes").
IF_CURRENCY	<tag name> TRANS_CURR SETTL_CURR RECONS_CURR CONTRACT <tag or contract custom parameter> <classifier code> <three-digit numeric currency code or currency name>	This tag sets the document currency that must be checked. Possible values: • Document tag containing the currency code. • TRANS_CURR – transaction currency from the document. • SETTL_CURR – settlement currency from the document. • RECONS_CURR – reconciliation currency from the document (in the currency in which the source bank provides transaction information to the payment system). • CONTRACT – contract currency. • Tag or contract custom parameter with a currency code. A contract's custom parameter can be registered as a "Primary" classifier. •Code of the classifier with a currency code. •Explicit specification of currency – the tag value may be a code or the name of a certain currency. Used together with the IF_CURRENCY_VALUE, IF_CURRENCY_RULE, IF_CURRENCY_FOR tags.
IF_CURRENCY_VALUE	EMPTY NOT_EMPTY USE_IN_BANK <currency code1>, <currency code2>,…,<currency codeN> <currency name1>, <currency name2>,…,<currency nameN> <tag name>	A payment order activates only if the document currency specified with the IF_CURRENCY tag corresponds to parameters set in the IF_CURRENCY_VALUE tag. Possible values: • EMPTY – this currency is not specified. • NOT_EMPTY – this currency is specified. • USE_IN_BANK – this currency has the Use in Bank parameter value of "Yes" (*i.e*. for this currency, conversion can be performed in WAY4 and the rate of the currency to the financial institution's local currency can be entered during the daily *opening* procedure). • A code or name of a certain currency or several currencies, separated by

Name	Value	Description
		commas can be used as the tag value. • Tag or contract custom parameter containing a currency code or list of currencies separated by commas can be used as the tag value. A contract custom parameter can be registered as a "Primary" classifier.
IF_CURRENCY_RULE	ACC_SCHEME EQUAL NOT_EQUAL NOT_IN_LIST	To set additional conditions for checking a currency, use the IF_CURRENCY_VALUE tag together with the IF_CURRENCY_RULE tag: • To *check the* currency set in the IF_CURRENCY for correspondence to the Accounting Scheme it is necessary to set the tags *IF_CURRENCY*_RULE=ACC_SCHEME;IF_CURRENCY_VALUE=<list of account codes separated by commas>;. If a list of account codes is set, the currency is searched for among the specified Accounting Scheme account templates. If a list of account codes is not set, the currency is searched for among all Accounting Scheme account templates. • To check a currency for correspondence (or non-correspondence) to another document currency, it is necessary to set the tags IF_CURRENCY_RULE=EQUAL; or IF_CURRENCY_RULE=NOT_EQUAL; the value of IF_CURRENCY_VALUE is also set as the value of the IF_CURRENCY tag. • When IF_CURRENCY_RULE=NOT_IN_LIST; is specified, a check will be made that the currency is not included in the list of currencies specified explicitly or through the parameter in the tag IF_CURRENCY_VALUE.

Name	Value	Description
XF_ROUND;		The XF_ROUND; tag together with the PRECISION tag defines rules for rounding and transferring the amount accumulated in a “High Precision” cumulative account to a standard account. For more information, see the section “High Precision Accounting” of the document “WAY4 Accounting Schemes”. !For "High Precision" accounts with the "Value Date Due" value of the *Due Type* field, the XF_ROUND; tag cannot be set in the account template and in payment orders used for due normalisation (with "Account Due" in the *Date Event* field).
DATE_EVENT_TYPE_MODE	FORTNIGHTLY SEMIANNUAL	Provides additional functionality to determine the frequency for activation of a standing payment order with the "Weekly" and "Yearly" values in the *Charge Event* field: ▪ FORTNIGHTLY – the order activates once every two weeks. When the tag has this value, the *Value Days* field works as follows: when a value from 1 to 7 is specified, the order activates on the specific day of each oddly-numbered week; when a value from 8 to 14 is specified (where "8" = "Monday", etc.) the order activates on the specified day of each evenly-numbered week. ▪ SEMIANNUAL – the order activates semi-annuallly.
IF_PARM_FOR	"BILLING" "LIABILITY" "TOP" "BASE" "DOC_SOURCE" "DOC_TARGET" "FROM_DOC" "CONTRACT_ROLE" CONTRACT_ROLE_PARM=<tag name> "LIAB_CATEGORY"	The IF_PARM_FOR tag is used together with the IF_PARM/IF_PARM_VALUE tags to redefine the contract for which these checks are made. • "BILLING" – for the account contract from which settlement is made. • "LIABILITY" – for a higher-ranking contract in a "Liability" hierarchy. • "TOP" – for the top contract in a hierarchy. • "BASE" – for the main contract in a "Main/Sub" hierarchy, with which this contract is related. • *"DOC*_SOURCE" – for the contract specified in the document's *Source Contract* (source_contract) field (see the "Doc-Brief" form). • "DOC_TARGET" – for the contract specified in the document's *Target Contract* (target_contract) field (see the "Doc-Brief" form). • "FROM_DOC" – the contract is taken from the document's *Add Data* (add_info) field, according to the tag specified with the CONTRACT_TAG tag (in the same field). • "CONTRACT_ROLE" – for this value, specify the CONTRACT_ROLE=<role code> tag in this field; for example:

Name	Value	Description
		IF_PARM_FOR=CONTRACT_ROLE;CONTRACT_ROLE=PAYMENT_LEVEL;. In this case, a search will be made for a contract with the CONTRACT_ROLE=<role code>; tag upward in a Liability hierarchy (in our example, a search for the CONTRACT_ROLE=PAYMENT_LEVEL tag). The check is made for the contract with the specified tag. If no contract with this tag is found in the Liability contract, the top contract in the hierarchy will be selected. • When a search for a contract must be made in a Liability hierarchy according to the value of an arbitrary tag, the CONTRACT_ROLE_PARM=<tag name>; tag must be additionally used in the configuration: For example, when the following settings are specified in the configuration: CONTRACT_ROLE=LEVEL1;CONTRACT_ROLE_PARM=LEVEL; WAY4 will search upward in the Liability hierarchy for a contract with the LEVEL= LEVEL=LEVEL1; tag. The check is made for the contract with the specified tag • "LIAB_CATEGORY" – in this case, a search for a contract will be made upward in a Liability hierarchy within the category/categories set in this field using the LIAB_CATEGORY=<category1>,<category2> tag. Liability category codes, separated by commas, are specified as the LIAB_CATEGORY tag value: "Y" – "Full Liability" category "N" – "Affiliated" category "R" – "Reporting" category "A" – "Only Check Balance" category. For example, when the IF_PARM_FOR=LIAB_CATEGORY;LIAB_CATEGORY=Y,N; tags are specified, a search is made for the top contract in a "Liability" hierarchy, starting from the current contract within the "Full Liability" ("Y") and "Affiliated" ("N") categories. If a contract has no higher-ranking contracts, it will be used. If a higher-ranking contract belongs to another LIAB_CATEGORY, for example "Reporting" ("R")), no move up the hierarchy will be made. I.e. a contract from the middle of the Liability hierarchy will be used, the top one in in the "Full Liability" ("Y") and "Affiliated" ("N") categories. It is possible to set a check of several parameters for a contract by using the IF_PARM_NUMB, IF_PARM<N>, IF_PARM_VALUE<N>, IF_PARM_FOR<N> tags. See the description of the IF_PARM_NUMB tag.

Name	Value	Description
IF_CS_TYPE_FOR	"BILLING" "LIABILITY" "TOP" "BASE" "CONTRACT_ROLE" CONTRACT_ROLE_PARM=<tag name> "LIAB_CATEGORY"	The IF_CS_TYPE_FOR tag is used together with IF_CS group tags to redefine the contract for which these checks are made. • "BILLING" – for the account contract from which settlement is made. • "LIABILITY" – from a higher-ranking contract in a "Liability" hierarchy. • "TOP" – from the top contract in a hierarchy. • "BASE" – from the main contract in a "Main/Sub" hierarchy, with which this contract is related. • "CONTRACT_ROLE" – for this value, specify the CONTRACT_ROLE=<*role* code> tag in this field; for example: IF_CS_TYPE_FOR=CONTRACT_ROLE;CONTRACT_ROLE=PAYMENT_LEVEL;. In this case, *a* search will be made for a contract with the CONTRACT_ROLE=<role code>; tag upward in a Liability hierarchy (in *our example*, a search for the CONTRACT_ROLE=PAYMENT_LEVEL tag). The check is made for the contract with the specified tag. If no contract with this tag is found in the Liability contract, the top contract in the hierarchy will be selected. • When a search for a contract must be made in a Liability hierarchy according to the value of an arbitrary tag, the CONTRACT_ROLE_PARM=<tag name>; tag must be additionally used in the configuration: For example, when the following settings are specified in the configuration: CONTRACT_ROLE=LEVEL1;CONTRACT_ROLE_PARM=LEVEL; WAY4 will search upward in the Liability hierarchy for a contract with the LEVEL= LEVEL=LEVEL1; tag. The check is made for the contract with the specified tag • "LIAB_CATEGORY" – in this case, a search for a contract will be made upward in a Liability hierarchy within the category/categories set in this field using the LIAB_CATEGORY=<category1>,<category2> tag. Liability category codes, separated by commas, are specified as the LIAB_CATEGORY tag value: "Y" – "Full Liability" category "N" – "Affiliated" category "R" – "Reporting" category "A" – "Only Check Balance" category. For example, when the IF_CS_TYPE_FOR=LIAB_CATEGORY;LIAB_CATEGORY=Y,N; tags are specified, a search is made for the top contract in a "Liability" hierarchy, starting from the current contract within the "Full Liability" ("Y") and "Affiliated" ("N") categories. If a contract has no higher-ranking contracts, it will be used. If a higher-ranking contract belongs to another LIAB_CATEGORY, for example

Name	Value	Description
		"Reporting" ("R")), no move up the hierarchy will be made. I.e. a contract from the middle of the Liability hierarchy will be used, the top one in in the "Full Liability" ("Y") and "Affiliated" ("N") categories. It is possible to set a check of several parameters for a contract by using the IF_CS_NUMB, IF_CS_TYPE<N>, IF_CS_VALUE<N>, IF_NOT_CS_VALUE<N>, IF_CS_TYPE_FOR<N>. See the description of the IF_CS_NUMB tag.
IF_CURRENCY_FOR	"BILLING" "LIABILITY" "TOP" "BASE" "RELATED" "CONTRACT_ROLE" CONTRACT_ROLE_PARM=<tag name> "LIAB_CATEGORY"	Redefines the contract for which a check is made using the IF_CURRENCY tag. Possible values: • "BILLING" – for the account contract from which settlement is made. • "LIABILITY" – for a higher-ranking contract in a "Liability" hierarchy. • "TOP" – for the top contract in a hierarchy. • "BASE" – for the main contract in a "Main/Sub" hierarchy, with which this contract is related. • "RELATED" – related contract, with the relation *specified in* the RELATION tag. • "CONTRACT_ROLE" – for this value, specify the CONTRACT_ROLE=<role code> tag in this field; for example: IF_*CURRENCY*_FOR=CONTRACT_ROLE;CONTRACT_ROLE=PAYMENT_LEVEL;. In this case, a search will be made for a contract with the CONTRACT_ROLE=<role code>; tag upward in a Liability hierarchy (in our example, a search for the CONTRACT_ROLE=PAYMENT_LEVEL *tag)*. The check is made for the contract with the specified tag. If no contract with this tag is found in the Liability contract, the top contract in the hierarchy will be selected. • When a search for a contract must be made in a Liability hierarchy according to the value of an arbitrary tag, the CONTRACT_ROLE_PARM=<tag name>; tag must be additionally used in the configuration: For example, when the following settings are specified in the configuration: CONTRACT_ROLE=LEVEL1;CONTRACT_ROLE_PARM=LEVEL; WAY4 will search upward in the Liability hierarchy for a contract with the LEVEL= LEVEL=LEVEL1; tag. The check is made for the contract with the specified tag • "LIAB_CATEGORY" – in this case, a search for a contract will be made upward in a Liability hierarchy within the category/categories set in this field using the LIAB_CATEGORY=<category1>,<category2> tag. Liability category codes, separated by commas, are specified as the LIAB_CATEGORY tag value:

Name	Value	Description
		"Y" – "Full Liability" category "N" – "Affiliated" category "R" – "Reporting" category "A" – "Only Check Balance" category. For example, when the IF_CURRENCY_FOR=LIAB_CATEGORY;LIAB_CATEGORY=Y,N; tags are specified, a search is made for the top contract in a "Liability" hierarchy, starting from the current contract within the "Full Liability" ("Y") and "Affiliated" ("N") categories. If a contract has no higher-ranking contracts, it will be used. If a higher-ranking contract belongs to another LIAB_CATEGORY, for example "Reporting" ("R")), no move up the hierarchy will be made. I.e. a contract from the middle of the Liability hierarchy will be used, the top one in in the "Full Liability" ("Y") and "Affiliated" ("N") categories.
IF_CS_NUMB	<number of additional checks>	Sets the number of additional checks for the IF_CS_TYPE tag. The tag is used together with the IF_CS_TYPE<N>, IF_CS_VALUE<N>, IF_NOT_CS_VALUE<N>, IF_CS_TYPE_FOR<N> tags. I.e. by default the IF_CS_TYPE *tag* is checked. If the IF_CS_NUMB=<number of additional checks>; tag is set, the specified number of IF_CS_TYPE<N> tags is checked. For example, if IF_CS_NUMB=3;, the IF_CS_TYPE1, IF_CS_TYPE2, IF_CS_TYPE3 tags are checked in addition to the IF_CS_TYPE tag.
IF_PARM_NUMB	<number of additional checks>	Sets the number of additional checks for the IF_PARM tag. The tag is used together with the IF_PARM<N>, IF_PARM_VALUE<N>, IF_PARM_FOR<N> tags. I.e. by default the IF_PARM tag is checked. If the IF_PARM_NUMB=<number of additional checks>; tag is set, the specified number of IF_PARM<N> tags is checked. For example, when IF_PARM_NUMB=3;, the IF_PARM1, IF_PARM2, IF_PARM3 tags are checked in addition to the IF_PARM tag.
SET_TO_BALANCE	Y	The SET_TO_BALANCE=Y; tag is used when configuring an order to move the amount of unauthorised overdraft to another account. When this tag is set, the order amount is calculated to correct the account balance to the amount defined by the balance type defined in the order's *Balance Type* field (to the balance amount with the hardcoded OVL code). I.e. after the document for the order is posted, the account balance will correspond to the amount of the specified balance.

Name	Value	Description
ZERO_BAL_TYPE	<balance type code>	An order will only be activated if the value of the balance specified in the tag is equal to zero. The tag can be used together with the ZERO_BAL_TYPE_FOR tag.
ZERO_BAL_TYPE_FOR	"BILLING" "LIABILITY" "TOP" "BASE" "RELATED" "CONTRACT_ROLE" "CONTRACT_ROLE_PARM=<tag name>" "LIAB_CATEGORY"	Redefines the contract for which a check is made using the ZERO_BAL_TYPE tag. Possible values: • "BILLING" – for the account contract from which settlement is made. • "LIABILITY" – for the higher ranking contract in the Liability hierarchy. • "TOP" – for the top contract in the hierarchy. • "BASE" – for the main contract in a "Main/Sub" hierarchy, with which this contract is related. • "RELATED" – a related contract with the relation specified in the RELATION tag. • "CONTRACT_ROLE" – for this value, specify the CONTRACT_ROLE=<role code> tag in this field; for example: ZERO_BAL_TYPE_FOR=CONTRACT_ROLE;CONTRACT_ROLE=PAYMENT_LEVEL;. In this case, a search will be made for a contract with the CONTRACT_ROLE=<role code>; tag upward in a Liability hierarchy (in our example, a search for the CONTRACT_ROLE=PAYMENT_LEVEL tag). The check is made for the contract with the specified tag. If no contract with this tag is found in the Liability contract, the top contract in the hierarchy will be selected. • When a search for a contract must be made in a Liability hierarchy according to the value of an arbitrary tag, the CONTRACT_ROLE_PARM=<tag name>; tag must be additionally used in the configuration: For example, when the following settings are specified in the configuration: CONTRACT_ROLE=LEVEL1;CONTRACT_ROLE_PARM=LEVEL; WAY4 will search upward in the Liability hierarchy for a contract with the LEVEL= LEVEL=LEVEL1; tag. The check is made for the contract with the specified tag • "LIAB_CATEGORY" – in this case, a search for a contract will be made upward in a Liability hierarchy within the category/categories set in this field using the LIAB_CATEGORY=<category1>,<category2> tag. Liability category codes, separated by commas, are specified as the LIAB_CATEGORY tag value:

Name	Value	Description
		"Y" – "Full Liability" category "N" – "Affiliated" category "R" – "Reporting" category "A" - "Only Check Balance" category. For example, when the ZERO_BAL_TYPE_FOR=LIAB_CATEGORY;LIAB_CATEGORY=Y,N; tags are specified, a search is made for the top contract in a "Liability" hierarchy, starting from the current contract within the "Full Liability" ("Y") and "Affiliated" ("N") categories. If a contract has no higher-ranking contracts, it will be used. If a higher-ranking contract belongs to another LIAB_CATEGORY, for example "Reporting" ("R")), no move up the hierarchy will be made. I.e. a contract from the middle of the Liability hierarchy will be used, the top one in in the "Full Liability" ("Y") and "Affiliated" ("N") categories.
NON_ZERO_BAL_TYPE_FOR	"BILLING" "LIABILITY" "TOP" "BASE" "RELATED" "CONTRACT_ROLE" "CONTRACT_ROLE_PARM=<tag name>" "LIAB_CATEGORY"	Redefines the contract for which a check is made using the NON_ZERO_BAL_TYPE tag. Possible values: • "BILLING" – for the account contract from which settlement is made. • "LIABILITY" – for the higher ranking contract in the Liability hierarchy. • "TOP" – for the top contract in the hierarchy. • "BASE" – for the main contract in a "Main/Sub" hierarchy, with which this contract is related. • "RELATED" – a related contract with the relation specified in the RELATION tag. • "CONTRACT_ROLE" – for this value, specify the CONTRACT_ROLE=<role code> tag in this field; for example: NON_ZERO_BAL_TYPE_FOR=CONTRACT_ROLE;CONTRACT_ROLE=PAYMENT_LEVEL;. In this case, a search will be made for a contract with the CONTRACT_ROLE=<role code>; tag upward in a Liability hierarchy (in our example, a search for the CONTRACT_ROLE=PAYMENT_LEVEL tag). The check is made for the contract with the specified tag. If no contract with this tag is found in the Liability contract, the top contract in the hierarchy will be selected. • When a search for a contract must be made in a Liability hierarchy according to the value of an arbitrary tag, the CONTRACT_ROLE_PARM=<tag name>; tag must be additionally used in the configuration: For example, when the following settings are specified in the configuration: CONTRACT_ROLE=LEVEL1;CONTRACT_ROLE_PARM=LEVEL; WAY4 will search upward in the Liability hierarchy for a contract with the LEVEL= LEVEL=LEVEL1; tag. The check is made for the contract with the specified tag

Name	Value	Description
		• "LIAB_CATEGORY" – in this case, a search for a contract will be made upward in a Liability hierarchy within the category/categories set in this field using the LIAB_CATEGORY=<category1>,<category2> tag. Liability category codes, separated by commas, are specified as the LIAB_CATEGORY tag value: "Y" – "Full Liability" category "N" – "Affiliated" category "R" – "Reporting" category "A" - "Only Check Balance" category. For example, when the NON_ZERO_BAL_TYPE_FOR=LIAB_CATEGORY;LIAB_CATEGORY=Y,N; tags are specified, a search is made for the top contract in a "Liability" hierarchy, starting from the current contract within the "Full Liability" ("Y") and "Affiliated" ("N") categories. If a contract has no higher-ranking contracts, it will be used. If a higher-ranking contract belongs to another LIAB_CATEGORY, for example "Reporting" ("R")), no move up the hierarchy will be made. I.e. a contract from the middle of the Liability hierarchy will be used, the top one in in the "Full Liability" ("Y") and "Affiliated" ("N") categories.
NON_ZERO_BAL_TYPE	<balance type code>	An order will only be activated if the value of the balance specified in the tag is not equal to zero. The tag can be used together with the NON_ZERO_BAL_TYPE_FOR tag.
MIN_BAL_TYPE_FOR	"BILLING" "LIABILITY" "TOP" "BASE" "RELATED" "CONTRACT_ROLE" "CONTRACT_ROLE_PARM=<tag name>" "LIAB_CATEGORY"	Redefines the contract for which a check is made using the MIN_ZERO_BAL_TYPE tag. Possible values: • "BILLING" – for the account contract from which settlement is made. • "LIABILITY" – for the higher ranking contract in the Liability hierarchy. • "TOP" – for the top contract in the hierarchy. • "BASE" – for the main contract in a "Main/Sub" hierarchy, with which this contract is related. • "RELATED" – a related contract with the relation specified in the RELATION tag. • "CONTRACT_ROLE" – for this value, specify the CONTRACT_ROLE=<role code> tag in this field; for example: MIN_BAL_TYPE_FOR=CONTRACT_ROLE;CONTRACT_ROLE=PAYMENT_LEVEL;. In this case, a search will be made for a contract with the CONTRACT_ROLE=<role code>; tag upward in a Liability hierarchy (in our example, a search for the CONTRACT_ROLE=PAYMENT_LEVEL tag). The

Name	Value	Description
		check is made for the contract with the specified tag. If no contract with this tag is found in the Liability contract, the top contract in the hierarchy will be selected. • When a search for a contract must be made in a Liability hierarchy according to the value of an arbitrary tag, the CONTRACT_ROLE_PARM=<tag name>; tag must be additionally used in the configuration: For example, when the following settings are specified in the configuration: CONTRACT_ROLE=LEVEL1;CONTRACT_ROLE_PARM=LEVEL; WAY4 will search upward in the Liability hierarchy for a contract with the LEVEL= LEVEL=LEVEL1; tag. The check is made for the contract with the specified tag • "LIAB_CATEGORY" – in this case, a search for a contract will be made upward in a Liability hierarchy within the category/categories set in this field using the LIAB_CATEGORY=<category1>,<category2> tag. Liability category codes, separated by commas, are specified as the LIAB_CATEGORY tag value: "Y" – "Full Liability" category "N" – "Affiliated" category "R" – "Reporting" category "A" - "Only Check Balance" category. For example, when the MIN_BAL_TYPE_FOR=LIAB_CATEGORY;LIAB_CATEGORY=Y,N; tags are specified, a search is made for the top contract in a "Liability" hierarchy, starting from the current contract within the "Full Liability" ("Y") and "Affiliated" ("N") categories. If a contract has no higher-ranking contracts, it will be used. If a higher-ranking contract belongs to another LIAB_CATEGORY, for example "Reporting" ("R")), no move up the hierarchy will be made. I.e. a contract from the middle of the Liability hierarchy will be used, the top one in in the "Full Liability" ("Y") and "Affiliated" ("N") categories.
MIN_BAL_TYPE	<balance type code>	The MIN_BAL_TYPE tag sets the code of the balance type being checked. This balance is compared with the amount set in a tariff with the Threshold role (the checked tariff is set with the LIMIT_AMOUNT=<tariff type code>; tag). If the balance type is less than the amount set in the tariff, an order is not activated. Tariff setup is described in the document "WAY4 Advanced Tariff Management".

Name	Value	Description
MAX_BAL_TYPE_FOR	"BILLING" "LIABILITY" "TOP" "BASE" "RELATED" "CONTRACT_ROLE" "CONTRACT_ROLE_PARM=<tag name>" "LIAB_CATEGORY"	Redefines the contract for which a check is made using the MAX_BAL_TYPE_FOR tag. Possible values: • "BILLING" – for the account contract from which settlement is made. • "LIABILITY" – for the higher ranking contract in the Liability hierarchy. • "TOP" – for the top contract in the hierarchy. • "BASE" – for the main contract in a "Main/Sub" hierarchy, with which this contract is related. • "RELATED" – a related contract with the relation specified in the RELATION tag. • "CONTRACT_ROLE" – for this value, specify the CONTRACT_ROLE=<role code> tag in this field; for example: MAX_BAL_TYPE_FOR=CONTRACT_ROLE;CONTRACT_ROLE=PAYMENT_LEVEL;. In this case, a search will be made for a contract with the CONTRACT_ROLE=<role code>; tag upward in a Liability hierarchy (in our example, a search for the CONTRACT_ROLE=PAYMENT_LEVEL tag). The check is made for the contract with the specified tag. If no contract with this tag is found in the Liability contract, the top contract in the hierarchy will be selected. • When a search for a contract must be made in a Liability hierarchy according to the value of an arbitrary tag, the CONTRACT_ROLE_PARM=<tag name>; tag must be additionally used in the configuration. For example, when the following settings are specified in the configuration: CONTRACT_ROLE=LEVEL1;CONTRACT_ROLE_PARM=LEVEL; WAY4 will search upward in the Liability hierarchy for a contract with the LEVEL= LEVEL=LEVEL1; tag. The check is made for the contract with the specified tag • "LIAB_CATEGORY" – in this case, a search for a contract will be made upward in a Liability hierarchy within the category/categories set in this field using the LIAB_CATEGORY=<category1>,<category2> tag. Liability category codes, separated by commas, are specified as the LIAB_CATEGORY tag value: "Y" – "Full Liability" category "N" – "Affiliated" category "R" – "Reporting" category "A" - "Only Check Balance" category. For example, when the MAX_BAL_TYPE_FOR=LIAB_CATEGORY;LIAB_CATEGORY=Y,N; tags are specified, a search is made for the top contract in a "Liability" hierarchy, starting from the current contract within the "Full Liability" ("Y") and "Affiliated" ("N")

Name	Value	Description
		categories. If a contract has no higher-ranking contracts, it will be used. If a higher-ranking contract belongs to another LIAB_CATEGORY, for example "Reporting" ("R")), no move up the hierarchy will be made. I.e. a contract from the middle of the Liability hierarchy will be used, the top one in in the "Full Liability" ("Y") and "Affiliated" ("N") categories.
MAX_BAL_TYPE	<balance type code>	The MAX_BAL_TYPE tag sets the code of the balance type being checked. This balance is compared with the amount set in a tariff with the Threshold role (the checked tariff is set with the LIMIT_AMOUNT=<tariff type code>; tag). If the balance type is greater than the amount set in the tariff, an order is not activated. Tariff setup is described in the document "WAY4 Advanced Tariff Management".
MIN_BAL_TYPE_SIGN	"+" "-"	The tags must be used when checking minimum/maximum balances that can have positive or negative values. For example, Service balances are kept as positive, but when posting reversals, the sign may change. The tags are used together with the MIN_BAL_TYPE and MAX_BAL_TYPE tags. If MIN_BAL_TYPE_SIGN=+; is set, a negative value for a balance will be considered a zero balance. If MIN_BAL_TYPE_SIGN=-; is set, a positive balance will be considered a zero balance.
MAX_BAL_TYPE_SIGN	"+" "-"	
IF_DATE	"CLIENT.BIRTH_DATE" "CLIENT.DATE_EXPIRE" "CLIENT.DATE_OPEN" "CLIENT.ADD_DATE_01" "CLIENT.ADD_DATE_02" "CONTRACT.FIRST_ACTIVITY_DATE" "CONTRACT.DATE_OPEN" "CONTRACT.LAST_BILLING" "CONTRACT.NEXT_BILLING" "CONTRACT.DATE_EXPIRE" "CONTRACT.CARD_EXPIRE"	IF_DATE group tags make it possible to configure actions (in this case, activation of a standing payment order) depending on contract/client dates. Contract/client dates can be compared with other dates (contract, client dates, system date, current banking date). Periods between dates can be checked according to various conditions. The IF_DATE tag sets the code of the date being checked, or the code of the date being compared with another date. Date codes can be contract/client dates specified in the following contract table (CONTRACT) or client table (CLIENT) fields: • "CLIENT.BIRTH_DATE" – a check is made according to the BIRTH_DATE field of the client's record • "CLIENT.DATE_EXPIRE" – a check is made according to the DATE_EXPIRE field of the client's record • "CLIENT.DATE_OPEN" – a check is made according to the DATE_OPEN field of the client's record (date on which the client was registered in the DB)

Name	Value	Description
		• "CLIENT.ADD_DATE_01" – a check is made according to the ADD_DATE_01 field of the client's record • "CLIENT.ADD_DATE_02" – a check is made according to the ADD_DATE_02 field of the client's record • "CONTRACT.FIRST_ACTIVITY_DATE" – a check is made according to the FIRST_ACTIVITY_DATE field of the contract's record • "CONTRACT.DATE_OPEN" – a check is made according to the DATE_OPEN field of the contract's record • "CONTRACT.LAST_BILLING" – a check is made according to the LAST_BILLING field of the contract's record • "CONTRACT.NEXT_BILLING"– a check is made according to the NEXT_BILLING field of the contract's record • "CONTRACT.DATE_EXPIRE" – a check is made according to the DATE_EXPIRE field of the contract's record • "CONTRACT.CARD_EXPIRE" – a check is made according to the CARD_EXPIRE field of the contract's record The tag is used together with the IF_DATE_VALUE and IF_DATE_FOR tags. A check of several dates for a client/contract can be set using the IF_DATE_NUMB, IF_DATE<N>, IF_DATE_VALUE<N> and IF_DATE_FOR<N> tags. See the description of the IF_DATE_NUMB tag.
IF_DATE_VALUE	"EMPTY" "NOT_EMPTY" <Rule><Base Date Code><Period Rule>	The IF_DATE_VALUE tag sets rules for checking the date specified with the IF_DATE tag. Tag values: • "EMPTY" – the condition is met if the field for the date set with the IF_DATE field is not filled in. • "NOT_EMPTY" – the condition is met if the field for the date set with the IF_DATE field is filled in. The field value is not checked. • The tag value can be composite and set in the following format (components are listed without delimiters): IF_DATE_VALUE=<Rule><Base Date Code><Period Rule>;, where: ▪ <Rule> – comparison rule. Possible values: ♦ "<" – less than ♦ ">" – greater than ♦ "<=" – less than or equal to ♦ ">=" – greater than or equal to ♦ If a comparison rule (<Rule>) is not set, the date is checked for matching (for the "=" value).

Name	Value	Description
		▪ <Base Date Code> – code of the date with which the date from the IF_DATE tag is compared. Possible values: ♦ Values that are described for the IF_DATE tag can be specified (i.e. dates from the contract or client table). ♦ "LDATE" – current banking date of the contract's financial institution ♦ "DB_DATE" – system (calendar) data (the system date is used without time) ▪ <Period Rule> – period that should be added or subtracted from the <Base Date Code> date. Set in the format <sign> <number of units ><period unit of measurement>, where ♦ <sign> – "+" or "-" (sign for adding or subtracting the specified period) ♦ <number of units> – number of days/months/years (see below) ♦ <period unit of measurement> – "D" (days), "M" (months), "Y" (years). Only one value can be set (i.e. one type of period is set in the tag). ♦ If the period <Period Rule> is not set, the two specified dates are compared according to the condition <Rule>. The tag is used together with the IF_DATE and IF_DATE_FOR tags. A check of several dates for a client/contract can be set using the IF_DATE_NUMB, IF_DATE<N>, IF_DATE_VALUE<N>, and IF_DATE_FOR<N> tags. See the description of the IF_DATE_NUMB tag. Examples of checks: Check of whether a client is already 40 years old: IF_DATE=CLIENT.BIRTH_DATE;IF_DATE_VALUE=<=DB_DATE-40Y; Check that a card has not expired: IF_DATE=CONTRACT.CARDS_EXPIRE;IF_DATE_VALUE=>=DB_DATE; Check that a card expires after a month: IF_DATE=CONTRACT.CARD_EXPIRE;IF_DATE_VALUE=<=DB_DATE+1M;
IF_DATE_FOR	"BILLING" "LIABILITY" "TOP" "BASE" "DOC_SOURCE" "DOC_TARGET" "FROM_DOC" "CONTRACT_ROLE" CONTRACT_ROLE_PARM=<tag name> "LIAB_CATEGORY"	The IF_DATE_FOR tag is used together with the IF_DATE/IF_DATE_VALUE tags to redefine the contract/client for which the check is being made. Note. If the IF_DATE tag is used to check a date from the client table, first a search is made for a contract according to the IF_DATE_FOR tag, and then this date is checked for the corresponding client. Possible values for the IF_DATE_FOR tag: • "BILLING" – the check is made for the account contract used for settlement. • "LIABILITY" – the check is made for the higher ranking contract in the Liability hierarchy. • "TOP" – the check is made for the top contract in the hierarchy. • "BASE" – the check is made for the main contract in a "Main/Sub" hierarchy,

Name	Value	Description
		with which this contract is related. • "DOC_SOURCE" – the check is made for the contract specified in the *Source Contract* (source_contract) field of the document (see the form "Doc-Brief") • "DOC_TARGET" – the check is made for the contract specified in the *Target Contract* (target_contract) field of the document (see the form "Doc-Brief") • "FROM_DOC" – the check is made for the contract from the *Add Data* (add_info) field of the document according to the tag specified with the CONTRACT_TAG tag (in the same field) • "CONTRACT_ROLE" – for this value, specify the CONTRACT_ROLE=<role code> tag in this field; for example: IF_PARM_FOR=CONTRACT_ROLE;CONTRACT_ROLE=PAYMENT_LEVEL;. In this case, a search will be made for a contract with the CONTRACT_ROLE=<role code>; tag upward in a Liability hierarchy (in our example, a search for the CONTRACT_ROLE=PAYMENT_LEVEL tag). The check is made for the contract with the specified tag. If no contract with this tag is found in the Liability contract, the top contract in the hierarchy will be selected. • When a search for a contract must be made in a Liability hierarchy according to the value of an arbitrary tag, the CONTRACT_ROLE_PARM=<tag name>; tag must be additionally used in the configuration: For example, when the following settings are specified in the configuration: CONTRACT_ROLE=LEVEL1;CONTRACT_ROLE_PARM=LEVEL; WAY4 will search upward in the Liability hierarchy for a contract with the LEVEL= LEVEL=LEVEL1; tag. The check is made for the contract with the specified tag • "LIAB_CATEGORY" – in this case, a search for a contract will be made upward in a Liability hierarchy within the category/categories set in this field using the LIAB_CATEGORY=<category1>,<category2> tag. Liability category codes, separated by commas, are specified as th LIAB_CATEGORY tag value: "Y" – "Full Liability" category "N" – "Affiliated" category "R" – "Reporting" category "A" - "Only Check Balance" category. For example, when the IF_PARM_FOR=LIAB_CATEGORY;LIAB_CATEGORY=Y,N; tags are specified, a search is made for the top contract in a "Liability" hierarchy, starting from the current contract within the "Full Liability" ("Y") and "Affiliated" ("N") categories. If a contract has no higher-ranking contracts, it will be used. If a higher-ranking

Name	Value	Description
		contract belongs to another LIAB_CATEGORY, for example "Reporting" ("R")), no move up the hierarchy will be made. I.e. a contract from the middle of the Liability hierarchy will be used, the top one in in the "Full Liability" ("Y") and "Affiliated" ("N") categories. A check of several dates for a client/contract can be specified using the IF_DATE_NUMB, IF_DATE<N>, IF_DATE_VALUE<N>, and IF_DATE_FOR<N> tags. See the description of the IF_DATE_NUMB tag.
IF_DATE_NUMB	<number of additional checks>	The tag sets the number of additional checks for the IF_DATE tag. The tag is used together with the IF_DATE<N>, IF_DATE_VALUE<N>, and IF_DATE_FOR<N> tags. I.e. by default, the IF_DATE tag is checked. If the IF_DATE_NUMB=<number of additional checks>; tag is set, the specified number of IF_DATE<N> tags is checked. For example, when IF_DATE_NUMB=3; in addition to the IF_DATE tag, the IF_DATE1, IF_DATE2, and IF_DATE3 tags are checked.