

Events

Contents

INTRODUCTION	1
CHAPTER 1. OVERVIEW OF SYSTEM EVENTS	3
Definition and Basic Properties of Events	3
Event Types	3
Generating Event Chains	8
CHAPTER 2. OPENING EVENTS	11
CHAPTER 3. PROCESSING EVENTS	25
Standard System Actions	25
Changing Contract Status	25
Generating Documents for Standing Payment Orders	26
Activating Additional Service Packages	26
Changing Contract Behaviour Types	27
Changing Contract Account Schemes	28
Generating Event Messages	28
Loan Loss Reserving	28
CHAPTER 4. CLOSING EVENTS	29
CHAPTER 5. EVENTS LOG	31
Description of the "Events for ... Contracts" Form	31
Starting Events by User Command	32
Contract Events Log	33
"Requested Event Uploading" Procedure	36
APPENDIX 1. EXAMPLES OF EVENT USE	39
Overdue Debt	39
No Contract Activity	40
APPENDIX 2. TAGS USED WHEN WORKING WITH EVENTS	42

Introduction


This document describes how WAY4 system Events are configured. Events automatically execute certain actions. For example, an Event can be configured to automatically change a contract status when certain conditions are met, or conversely, to take certain actions when a contract status changes.


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:

- Daily Procedures
- WAY4™ Service Packages
- WAY4™ Accounting Schemes
- Products and Contract Subtypes
- Configuration of Client Messages
- Standing Payment Orders
- WAY4™ Global Parameters
- Issuing Module
- Acquiring Module
- WAY4™ Dictionaries
- Documents
- Cardholder Statements
- WAY4™ Advanced Tariff Management
- Risk Monitoring
- Loan Loss Reserves
- "Notification Messaging"
- "WAY4 Accounting"

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. Overview of System Events

This section contains a description of the Event lifecycle in WAY4 and setup of Events.

Definition and Basic Properties of Events

Events are used for contracts to automatically execute various actions that must be executed when certain conditions are met in the system. For example, an Event can be configured to automatically notify a client of an overdue loan.

An Event life cycle consists of three steps: opening, processing and closing an Event. In the above mentioned example, an Event is opened when funds appear on an overdue account, it is closed when a delinquency is repaid, Event processing means placing a delinquency notification in each cardholder statement.

Events can be opened and closed both automatically and by a user command. For details on opening and closing Events, see "Opening Events" and "Closing Events".

Event processing means executing certain actions specified when configuring the corresponding Event type. By default, Events are processed at the moment of opening. At the same time, it is also possible to postpone Event processing until the next posting of macrotransactions generated for a contract for when the Event is configured (see the description of the *Post Immediate* field in the section "Event Types"). For more details on Event processing, see "Processing Events".

Event Types

Events can be created in the system according to the client and product category in these grids:

- For issuing contracts, use grid "Issuing Event Types" (Full → Configuration Setup → Products → Issuing Private Products (Issuing Corporate Products) → Issuing Event Types).
- For acquiring contracts, use grid "Acquiring Event Types" (Full → Configuration Setup → Products → Acquiring Products → Acquiring Event Types).

The fields of the indicated grids have the same structure. This document will now refer to the "Event Types" grid when referring to the above mentioned grids in general.

Before working with the grid, make sure that the correct financial institution is indicated in the status line. If users are allowed to work with several financial institutions, they should first select the desired institution in the "Get Financial Institution" form (see Fig. 1).

Fig. 1. Form for setting the financial institution when creating Event types

To enter information on Event types for private issuing contracts, select the "Full → Configuration Setup → Products → Issuing Private Products → Issuing Event Types" user menu item. The screen will display the "Issuing Event Types" form (see Fig. 2).



Contract	Name	Code	Duration Type	Duration	Due To Work Day	Next Event	Event Custom Code	Group Code	Post Immediate	Special Params
Card	Card Event D	CD	Single	0						CLOSE_EVENT=CA,CB,CC;
Card	Card Event C	CC	Single	0						
Card	Card Event B	CB	Single	0						ACC_SCHEME;
Card	Card Event A	CA	Single	0						N,ACC_SCHEME;

Fig. 2. Form for entering data on Event types

This form contains fields referring to the main parameters of Event types. To access full information about an Event type, click the [Full Info] button, which will bring the "Full Info for <Event type name>" form to the screen (see Fig. 3).

Fig. 3. Form for entering full information about Event types

The fields of the "Full Info for <Event type name>" form are described below:

- **Name** – name of the Event type
 Events with the same code must have the same name.
- **Code** – code that is used to identify this Event type in the database. In some cases, the field is used for other purposes (see "Opening Events")
 The code of an Event must be unique for each combination of the following parameters: financial institution, contract category ("Card", "Account", "Device") and Product category ("Issuing", "Acquiring", etc).
- **Group Code** – code of the Event type group used to filter Event types by certain characteristics

- *Post Immediate* – determines when Event processing is started and when an Event is closed; this field can take on one of the following values:

- "Post Immediate" – an Event will be processed immediately after it is opened.
- "Post Later" – an Event will be processed after the next posting of macrotransactions generated for the contract for which the Event is opened; if no macrotransactions are posted, it will be processed under the "Contracts Daily Update" procedure.



It is recommended to use Events with the "Post Later" value in setting up Events that open when a limiter is activated. See the description of the *Event Type* field in the section "Additional Parameters of Limiters (Details)" of the document "Usage Limiters".

- "Post and Close Later" – an Event will be processed after the next posting of macrotransactions generated for the contract for which the Event is opened; if no macrotransactions are posted, it will be processed under the "Contracts Daily Update" procedure; the Event will be closed immediately regardless of whether macrotransactions generated by its processing have been posted.
- "Post on CDU" – an Event will be processed at the end of the next Contracts – Daily Update procedure.



If an Event opens during execution of the "Contracts – Daily Update" procedure, this Event will be processed at the end of this (current) procedure.

Note that an Event is processed at the end of the "Contracts – Daily Update" procedure, not at the end of all procedures performed when opening/closing the business day. This means that if closing and opening a business day is split, an Event can be processed at the end of the evening "Contracts – Daily Update" procedure before the day has been fully closed.

The "Post on CDU" value makes it possible to send a client notification at the end of the "Contracts – Daily Update" procedure (when the value is "Post Later", for example, notification is sent following execution of the first macrotransaction for the contract).



Configuration of the *Post Immediate* parameter in a child Event from an Event Chain does not affect its processing. Child Events from an Event Chain are always processed together with the main Event.

- *Duration Type* – unit of time measuring a time interval at the end of which the Event will close or another Event from the Event chain will open (see the *Next Event* field); the field can take on one of the following values:

- "Day" – the time period is measured in calendar days
- "Month" – the time period is measured in months
- "Billing" – the time period is measured in billing cycles




It is recommended to use Events with the "Billing" value in the *Duration Type* field on the account contract level. Events with the

"Billing" value can be used on the subordinate card contract level if the algorithm for calculating the billing cycle end date did not change. If the billing cycle end date changed, the corresponding correction of the Event closing date will not be made on the subordinate contract level.

- "Single" – the Event is closed in the system by other means. For example, this duration type is set when an Event is configured to open when the balance of a contract account changes. In this case, the Event closes when the account balance is 0. This duration type is also set for Events that do not need to be closed.
- "Unique" – the Event is closed immediately after it is opened
- *Duration* – number value of the time interval length in the units set in the *Duration Type* field.


Specifics of *Duration* parameter processing:


- When the *Duration Type* parameter value is "Billing", the length of the period for an Event is calculated as the "*Duration* field value + 1". For example, when the value of the *Duration* parameter is "0", the length of the period for an Event will be equal to one billing cycle. When the *Duration* parameter value is "1" – to 2 billing cycles.
- When the *Duration Type* parameter value is "Month", the length of the period for an Event is fully determined by the value of the *Duration* field. I.e. when the value of the *Duration* parameter is "1", the length of the period of an Event will be equal to one month. When the value is "2" – to 2 months, etc.
- *Due To Work Day* – used when the date of closing the Event specified with the *Duration Type* and *Duration* fields is a day off, but it is necessary to close the Event on the first working day after the weekend. The field can take on either "Yes" or "No"

 Fields of the "Duration" group are filled in the parent form "Issuing Event Types" (see Fig. 2). In the "Full Info for <name of Event type>" form, the fields are displayed in view mode.

- *Fee Type* – shows the type of the fee charged when the Event is opened; the value of this field is selected from the list of fee types registered in the system. To access the list of fee types, use the "Full → Configuration Setup → Transaction Types → Fee Types" menu item.

The appropriate miscellaneous service must be configured for this fee type. See the "Service Configuration" section of WAY4™ Service Packages Administrator Manual.

 If an Event opens as the result of posting a macrotransaction, the supplementary macrotransaction for charging a fee (if the *Fee Type* field is filled in for the Event) inherits the GL Date (and Posting Date) from the original macrotransaction. In other cases (for example, if an Event is open as a result of processing an application or opened manually), the current banking date is specified as the GL Date and Posting Date of the supplementary macrotransaction for charging a fee.

- *New Status* – status associated with the contract while the Event is open; the list of contract statuses registered in the system can be viewed in the "Contract Statuses" grid (Full → Configuration Setup → Contract Types → Contract Statuses).
- *To Client Stop List* – determines whether the card should be placed on a stop list while the Event is open. This field can be set to either "Yes" or "No"
- *Credit Limit Action* – additional actions concerning the credit limit that may be executed when the Event is opened; the field can take on one of the following values:
 - "Set to Zero" – the contract's credit limit will be set to zero
 - "Suspend" – the contract will be given an additional credit limit equal to the value of the base credit limit, but with the opposite sign. Thus, the credit limit will be completely eliminated from the Amount Available
 - "Set To Event Amount" – the "Set To Event Amount" value makes it possible to set a credit limit for a contract. The credit limit amount and currency are specified in the *Comment Details* field using the AMOUNT and CURR tags, respectively. The currency code set in the CURR tag must correspond to the contract currency.
 - "None" (the default value) – no additional actions concerning the credit limit will be executed
- *Next Event* – Event that should open when the current Event closes or at the end of the time interval set in the *Duration* field. This configuration allows Event sequences to be created, where each Event can trigger the next one. For example, if a client loan becomes overdue, two different messages may be created for the client, where the second message is created 15 days after the first one. If the client repays the loan within 15 days from the moment the first message is created, then the second message is not created (for message configuration, see the section "Generating Event Messages").
 If the same Event should open with a given regularity (this configuration can be used, for example, to charge a monthly fee), specify the same Event Type in the *Next Event* field. In this setup, a new instance of the Event automatically opens when the old one closes and this process will continue infinitely. These cyclic Events always open on the same day – on the day the first Event in the chain opens.
- The *Custom Event Code* field is used to specify a special code. If the field is filled in, when posting the Event, the custom procedure CUST_EVENT_POST is called, to which this code is passed as the Event code.
- *Special Parameters* – the *Special Parameters* field is used to define special parameters as tags. For more details, see the section "Tags Used when Working with Events".

The "Issuing Event Types" form (see Fig. 2) contains the following buttons:

- The [Action] button contains the following context menu commands:

- "Check" – when the "Check" command is executed, WAY4 checks that the fields of the specified Event Type have been filled in correctly and informs users of the check's results.
- "Delete" – delete an Event Type record. When "Delete" is selected, a check is made for links in WAY4 to this Event Type:
 - ◆ If Events of this type are set up in Service Packages, Accounting Schemes, account templates, standing payment orders, usage limiters, in an Event Chain (*Next Event Type* field in the "Event Chain" form), in the *Next Event* field of another Event in the "Event Types" form, an error message is generated, and the record is not deleted.
 - ◆ If the code of an Event Type is set for the Product ("Start Events" form), for classifier values (*Result Event Code* field) in settings allowing the value of a user classifier to be changed when an Event opens (Classifier Actions), a warning is generated and the record is deleted.
- The [Event Chain] button opens the "Event Chain for <Event type name>" form used to generate an Event chain (see "Generating Event Chains").
- The [Messages] button is used to invoke the message template through which the database creates client messages (see the section "Generating Event Messages").
- Clicking on the [Used By] button will open the "Used By for <Event type name>" form. It lists States whose activity depends on the status of the current Event (see the "State Parameters" section in the States Administrator Manual).
- The [Mapping] button is used to mark Events using "Configuration Groups" classifiers. The procedure for marking Events is similar to that for marking Products. For more detailed information about configuring "Classification Groups" classifiers and about marking procedures, see the section ""Configuration Groups" Classifiers" of the Products and Contract Subtypes Administrator Manual.

The "Full Info for <name of Event type>" form contains the [Tagged Data] button, used to optimise work with the *Special Parameters* field. Clicking this button opens a grid form for entering and editing tags in the *Special Parameters* field.

Generating Event Chains

An Event chain is a set of Events that will be automatically opened/closed when another Event called the main Event of the chain is opened/closed.

To generate an Event chain, select the main Event in the "Issuing Event Types" form (see Fig. 2 in section "Event Types") and click the [Event Chain] button. This will open the "Event Chain for <Event type name>" grid form (see Fig. 4).

Next Event Type	For Contract	Active If State	Active If State Status	Chain Details
Event Chain 1	Contract Itself			

Ins Del Query

Fig. 4. Form for generating an Event chain

Each record in the table corresponds to an Event in the Event chain and contains the following fields:

- *Next Event Type* – Event type
- *For Contract* – determines for what contracts (for the contract for which the main Event is opened (hereinafter, current contract) or for contracts linked to this contract) this Event will be opened. The field can take on one of the following values:
 - "For Self" – the Event will be opened for the current contract only
 - "Sub Contracts" – the Event will be opened for every subcontract linked to the current contract with the "Main/Sub" link type
 - "Main Contract" – the Event will be opened for the main contract to which the current contract is linked with the "Main/Sub" link type
 - "Liab Subs" – the Event will be opened for every subcontract linked to the current contract with the "Liability" link type
 - "Liab Main" – the Event will be opened for the main contract to which the current contract is linked with the "Liability" link type
 - "For Client" – the Event will be opened for all contracts of the client to whom the current contract belongs
 - "Service Contract" – the Event will be opened for the service contract set up for the current contract's service group. In this case, leave the *Next Event Type* field blank, and specify the code of the opened Event in the *Chain Details* field
 - "Client Service Contract" – the Event will be opened for the service contract set up for the service group of the client to whom the current contract belongs
 - "Device Service Contract" – the Event will be opened for the service contract set up for the service group of the current contract's device
 - "Base Contract" – the Event will be opened for the contract for which the current contract is a related one
 - "Related Contract" – the Event will be opened for every contract related to the current contract through a certain relation type; the relation type is specified through the "RELATION_TAG=<relation type code>," tag in the *Chain Details* field. Relation type codes are specified when relation types are registered (Full → Configuration Setup → Accounting Setup → Contract Relations); a relation type is determined on the related contract level.

- "Related Bank Contract" – the Event will be opened for the fee contract (bank contract whose accounts are used to transfer fees for the current contract)
- "Custom" – this value is not used in the current version
- *Active if State* – specifies the State (see the States Administrator Manual) whose status determines whether the Event specified in the *Next Event Type* will be opened for contracts specified through the *For Contract* field
- *Active if State Status* – determines the status of the State specified in the *Active if State* field. If the field reads "Yes", the Event will be opened for contracts specified through the *For Contract* field only if the specified State is active. If the field reads "No", the Event will be opened only if the State is inactive.
- *Chain Details* – this field is used to define special parameters as tags. For more details, see the section "Tags Used when Working with Events".
- *Step N* – determines the order for opening Events in a chain. I.e. an event with a *Step N* field value of "1" will be processed earlier than an Event with a value of "2".



Note that there are two stages of processing an Event chain: loading and posting Events in chains. Both in loading and in posting, Events are checked (including verification of conditions for opening Events, for example, depending on the status of another Event, etc.). The *Step N* field determines the order for loading and for posting Events in a chain.

- If the *Step N* field is not filled in, the main Event is loaded first, and then linked Events are loaded in the order in which they are specified in the chain. After all Events have been loaded, they are posted in the same order in which they are specified in the chain.
- If the *Step N* field is filled in, the main Event is loaded first, and then Events from the chain in the order specified in the *Step N* field. After all Events have been loaded, they are posted according to the *Step N* field.
- If an Event from the chain depends on the main Event or previous linked Events in the chain, set the *DEPENDENT_CHAIN* tag in the *Special Parameters* field of the Event on which loading of other Events depends. In this case, the main Event is loaded and posted immediately, and then Events from the chain are loaded one by one and posted immediately (in the order specified by the *Step N* field or in the order in which they are specified in the chain, if the field is not filled in)

If the tag is not set, a dependent Event may not be posted since at the loading stage there still aren't any results for posting the main Event and previous Events in the chain, and accordingly, conditions for opening the dependent Event may not be met (the dependent Event will be declined).

Chapter 2. Opening Events

Events can be opened both automatically and by a user command. For more details on opening Events by a user command, see section "Starting Events by User Command".

In its turn, automatic Event opening can be executed in three ways: direct Event opening, code-based Event opening and opening an Event as part of an Event chain (for information on Event chains, see section "Event Types").

Furthermore, WAY4 allows opening and closing Events for a contract through a user command. For more details, see subheading "Closing Events".

The table below lists situations where Events are used and how they are applied during direct Event opening (see Table 1).

Table 1 Situations where direct Event opening is used


Situation	Event Characteristics
A contract account balance changes	An Event type must be specified in the account template configurations (see the "General" section in the WAY4 Accounting Schemes Administrator Manual). When funds are transferred to a contract's account that uses this template, the Event is opened; when the balance returns to zero, the Event is closed (see an example in the "Examples of Event Use").
A limiter is activated	An Event type must be specified in the limiter configuration. When the limiter is activated, the Event is opened; when the limiter's counter is reset, the Event is closed (see the "Additional Parameters of Limiters (Details)" section in the Usage Limiters Administrator Manual). It is recommended to set the <i>Duration Type</i> parameter of such an Event to "Unique" for the Event to close immediately after opening.
A document for to a standing payment order is generated	An Event type must be specified in the standing payment order configurations. The fields defining the payment order's activation frequency must be filled in as indicated (see the "Standing Payment Orders" Administrator Manual).
Another Event is closed	For the first Event in the chain, the next Event in the <i>Next Event</i> field of the "Event Types" form needs to be specified along with the time period after which the system should start processing the next Event in the <i>Duration</i> field.
An "external" event occurs	An "external" event means receiving a status message from an ATM, change of channel status, receiving a negative response code, etc. To open an Event in such a case, it is necessary to configure the "Usage Operation" table as indicated (see the Notification Messaging Administrator Manual).

A contract opens/closes	An Event opens automatically when a contract using the given Product opens/closes. Such Events are configured in the "Start Events for <name of Product>" form (see the section "Event Setup" in the "Products and Contract Subtypes" document).
-------------------------	--

To understand how Events are generated when the balances of accounts linked by normalisation rules change, let us consider the following example. Say that an Accounting Scheme contains two account templates, A and B with amount normalisation configured between them. Event 1 is specified in the *Open/Close Event* field for account A, Event 2 is specified for account B. Let us assume that the flow of funds occurred in the contract's accounts in the following sequence. First, funds appear in account A. Then, according to the normalisation rules, they are transferred from account A to account B. Then, a reverse entry from account B back to account A takes place. Finally, the funds are transferred from account A to some other account. This sequence of funds activity results in the following nested Event chain: Open Event 1 → Open Event 2 → Close Event 2 → Close Event 1.

To open an Event with a predefined code, the *Code* field of the "Event Types" form (see Fig. 2 in section "Event Types") is used. The table below (see Table 2) describes situations where these Events are used and their codes, which must be specified in the *Code* field of the Event Type.

Table 2 Situations where Events with predefined codes are used


Situation	Code	Comments
A contract status changes	<p>There are two options for setting an Event code:</p> <ul style="list-style-type: none"> • <code><Prefix>_<Status Code></code>, where <code><Prefix></code> is the value of the "CHANGE_STATUS_STR" additional global parameter. By default, it is "Change Status" (see the WAY4™ Global Parameters Administrator Manual), <code><Status Code></code> is the code of the new contract status (value of the <i>Code</i> field in the "Contract Statuses" form). • It is possible to set the code in the format <code><Prefix>_<Status External Code1>_<Status External Code2></code>, where <code><Status Code1></code> – <i>External Code</i> code for the old status of the contract (value of the <i>External Code</i> field in the "Contract Statuses" form) <code><Status Code2></code> – <i>External Code</i> code for the new status of the contract (value of the <i>External Code</i> field in the "Contract Statuses" form) In this case, an Event opens when the contract status changes from <code><Status External Code1></code> to <code><Status External Code2></code>. For this case, it is possible to set the code in the format <code><Prefix>EXT<Status External Code1>_<Status External Code2></code>. Note that an Event with this predefined code only opens when status is changed manually in Customer Service. When status changes according to an Event, through an application or when the contract form is edited and approved, an Event with this predefined code will only be opened if it is added to an Event chain. For more information, see the sections "Changing Contract Status" and "Tags Used when Working with Events"). <p> Note that when a contract status changes, only one Event can be open. The setting <code><Prefix>_<Status External Code1>_<Status External Code2></code> has priority. This means that if two Events are set up: <code><Prefix>_<Status Code2></code> and <code><Prefix>_<Status External Code1>_<Status External Code2></code>, the Event <code><Prefix>_<Status External Code1>_<Status External Code2></code> will open. In this case, configure an Event chain to open the Event <code><Prefix>_<Status Code2></code>.</p>	<p>To access the list of contract statuses registered in the system, use the "Contract Statuses" grid form (Full → Configuration Setup → Contract Types → Contract Statuses).</p>

Situation	Code	Comments
Locking a card until it is received by the cardholder	CP_<Production Code>, where <Production Code> is the value of the <i>Code</i> field in the "Production Events" form (Full → Configuration Setup → Transaction Types → Production Events)	
A credit limit changes	CREDIT_LIMIT_DOC	
An additional authorisation limit changes	ADD_CREDIT_LIMIT_DOC	
A contract is created	START_CONTRACT	An Event will be opened under the "Contracts Daily Update" procedure only if it is registered in the contract's Service Package
A contract is closed	The value of the "CLOSE_CONTRACT_STR" additional global parameter, which is "Close Contract" by default (see the WAY4™ Global Parameters Administrator Manual)	An Event will be opened only if it is registered in the contract's Service Package
Due normalisation between accounts is executed	DUE_FOR_<Code>, where <Code> is the code of the account type for which normalisation is executed	
Limit normalisation between accounts is executed	LIM_NORM_<Debit Acnt Type Code><Credit Acnt Type Code>, where <Debit Acnt Type Code> is the code of the debited account type and <Credit Acnt Type Code> is the code of the credited account type	An Event will be opened under the "Contracts Daily Update" procedure only if it is registered in the contract's Service Package
The interest accrual procedure is executed	INT_I<Account Type Code>, where <Account Type Code> is the code of the account type	An Event with the "INT_IP" code will be opened when interest is accrued for a deposit account
A financial document without an authorisation is processed for a card listed in the stop list	STOP_LIST	

Situation	Code	Comments
An incorrect PIN attempts counter is cleared forcibly	CLEAR_PIN_ATT	This Event opens if a counter reset is forced (for example, when executing the menu item "Clear PIN Attempts" in customer service workbench). The Event does not open when the counter is automatically reset (for example, when the client incorrectly enters a PIN during purchase (authorisation) and subsequent correct entry of the PIN).
A financial reversal document is processed if limit normalisation was executed between the registration of the initial authorisation and its reversal	INTEREST_WARNING	An Event only opens when the value of the global parameter PATCH_REVERSE_INTEREST is "Y".
A card is activated (the first PIN-based transaction is executed for a card; a card is unlocked manually, etc.)	CARD_UNLOCK	
A processed document is reversed after reconciliation on a POS terminal	POS_RECON_REVERSE	For more details about reconciliation on a POS terminal, see the POS Management Administrator Manual

Situation	Code	Comments
A financial document of the "When available" category is processed if the interval specified in the <i>Expiry Day</i> field of the service is exceeded (see the "Transaction Parameters" section in the WAY4™ Service Packages Administrator Manual)	DOC_EXPIRED	For more details on processing financial documents of this category, see the "Posting "When Available" Type Documents" section in the Documents Administrator Manual
A custom procedure is executed		In the description of the Event, specify a special code in the <i>Custom Event Code</i> field. If this field is filled in, when posting the Event, the custom procedure CUST_EVENT_POST is called, to which this code is passed as the Event code.
A dispute case is opened	DSP_CASE_OPENED	
A dispute case is closed	DSP_CASE_CLOSED	
A debt is written off	DSP_CASE_WRITTENOFF	
An incoming dispute document is received	DSP_ITEM_INWARD_CREATED	
An overdraft appears (at the end of the billing cycle)	OVL_BILLING	The Event is opened at the end of a billing cycle. The Event will only be opened if the Event is registered in the contract's Service Package.

Situation	Code	Comments
An overdraft appears (at the end of the banking day)	OVL_DAILY	The Event is opened when the banking day closes. The Event is called from a custom procedure executed during "Contracts – Daily Update" procedures. The Event will only be opened if the Event is registered in the contract's Service Package.

Unblocking authorisation funds	AUTH_UNBLOCK	<p>The Event is opened when unblocking authorisations using the "Clear Old Pending" procedure (Full → Daily Procedures → Document Processing Step by Step → Clear Old Pending). The Event is only opened if this Event is registered in the contract's Service Package.</p> <p>This Event remains for backward compatibility.</p> <p>If several blocks were made for a document (for example, the main transaction fee was blocked and a custom fee was blocked), funds are unblocked in arbitrary order.</p> <p>An Event with the AUTH_UNBLOCK code opens after each block for the document is released, but if message sending is set up in the Event, a message is sent once when the first block is released. I.e. if the custom fee is unblocked first, the message will contain the amount available after unblocking the fee.</p> <p>A message about the amount available after the transaction amount has been unblocked will not be sent.</p> <p> If configurations are used in which fees are blocked separately (for example, in a custom fee setup), use an Event with the CLOSED_AUTH_DOC code.</p>
--------------------------------	--------------	---

Situation	Code	Comments
Transfer of interest accrued for a month to a special card account.	END_MONTH	<p>An Event opens at the end of the month (in fact, the Event opens during the execution of daily procedures when the first banking day of the new month opens; the Start Date is the last day of the past month). This Event must be registered in the contract's Service Package.</p> <p>If the <i>Next Event</i> field of an Event with the END_MONTH code contains the same Event Type:</p> <p>If the first Event opens, for example on September 30, all subsequent Events will also open on the 30th of each following month, regardless of the actual number of days in the month. (In February an Event will open on the last day, for example, the 28th).</p>

Situation	Code	Comments
<p>An Event with the CONTRACT_DELAY code opens if the date of the contract's last financial activity is less than the current banking date by the number of days set with the ACTIVITY_DELAY tag in the Service Package. In doing so, for example, the contract may be assigned another status. The date of the contract's last financial activity is taken from the automatically inserted ACTIVITY_DATE tag in the ext_data field of the contract.</p> <p>The ACTIVITY_DATE tag is filled in if the ACTIVITY_FOR tag is set in the Service or transaction subtype. If the ACTIVITY_FOR tag is not set, the contract opening date is considered the date of the last activity on the contract.</p> <p>.</p>	CONTRACT_DELAY%	<p>An Event only opens if this Event is registered in the contract's Service Package.</p> <p>Several Events with the CONTRACT_DELAY% code can be configured for various periods when there is no contract activity (to charge various fees, send notifications). The CONTRACT_DELAY prefix is mandatory; the postfix may be arbitrary. See the section "No Contract Activity"</p>
<p>An Event with this code (if such Event is configured) will open for a contract before the billing period closes, after calculating interest and executing due normalization. Allows an action to be executed at the end of a billing period (for example, charging a fee).</p>	END_BILLING	<p>An Event only opens if this Event is registered in the contract's Service Package.</p>

Situation	Code	Comments
Authorisation reversal	AUTH_REVERSE	An Event opens when processing an authorization reversal. Can be used to generate messages for the cardholder. The Event must be registered in the contract's Service Package.
Creation of an authentication scheme	<Authentication scheme type code>_ENROLL	An Event opens when an authentication method is enabled (authentication scheme is created). See the section "Setting Parameters in WAY4 for Interaction with the WAY4 Authentication Server" of the document "Administering WAY4™ Authentication Server" in the authentication server distribution. This functionality is provided according to a separate agreement with the WAY4 vendor.
Closing of an authentication scheme	<Authentication scheme type code>_CLOSE	An Event opens when an authentication scheme is closed. See the section "Setting Parameters in WAY4 for Interaction with the WAY4 Authentication Server" of the document "Administering WAY4™ Authentication Server" in the authentication server distribution. This functionality is provided according to a separate agreement with the WAY4 vendor.


Situation	Code	Comments
SMS notification when the debited amount exceeds the authorisation amount	AMOUNT_ALERT	<p>An Event opens if the amount of the financial document differs from the amount blocked by authorisation.</p> <p>An Event is used together with a tariff with the "Threshold" role and the predefined code AMOUNT_ALERT.</p> <p>The possible difference between the amount of the financial document and the authorisation amount at which no notification is sent (i.e. an Event with the AMOUNT_ALERT code does not open) is specified in the tariff.</p>
Unblocking authorisation funds	CLOSED_AUTH_DOC	<p>An Event opens after all blocking for a document has been removed. If sending messages is set up in an Event, when a message to a client is created, the amount available after all unblocking is sent.</p> <p>An Event is opened when unblocking authorisations with the "Clear Old Pending" (Full → Daily Procedures → Document Processing Step by Step → Clear Old Pending) procedure. An Event is only opened if this Event is registered in the contract's Service Package.</p>

Situation	Code	Comments
Setting a contract or client classifier value on a date in the future, specifying the expiration date of the value. The Event Type should be given the "Single" value in the <i>Duration Type</i> field	CLS	<p>An event is opened on the Date From date specified for the classifier value. The Event closes automatically on the <i>Date To</i> date specified for the classifier value. See the section "Manually Changing Classifier Values" of the document "WAY4™ Client and Contract Classifiers".</p> <p>An Event is only opened if this Event is registered in the contract's Service Package.</p>
Setting a contract or client classifier value on a date in the future, without an expiration date. The Event Type should be given the "Unique" value in the <i>Duration Type</i> field	CLS_UNIQUE	<p>An event is opened on the Date From date specified for the classifier value. The Event closes immediately after the corresponding classifier's value has been set. See the section "Manually Changing Classifier Values" of the document "WAY4™ Client and Contract Classifiers".</p> <p>An Event is only opened if this Event is registered in the contract's Service Package.</p>
Number of attempts to enter a PIN has been exceeded	PIN_ATT_EXC	<p>An Event is opened when the permitted number of attempts to enter a PIN has been exceeded.</p>

Situation	Code	Comments
Close plastic on reissue. "Unique" should be set in the <i>Duration Type</i> field for the Event type.	CLOSE_PLASTIC_TRANS_STATUS	When plastic is reissued, the old plastic is closed. When closing the old plastic an Event with this code opens. The time when the old plastic is blocked depends on the value of the global parameter CLOSE_PREV_PLASTIC (see the document "WAY4 Global Parameters").

Let us consider an example of how an Event with a predefined code is used when a contract status changes. Say that we need to configure an Event that is opened when a card is lost. Let us assume that the "CS" value is assigned to the CHANGE_STATUS_STR parameter, and the "PickUp L 41" status must be assigned to the card contract (the status code is "41"). To configure the required Event type, we must specify the "CS 41" value in the *Code* field.

Let us consider an example of an Event opened when a custom procedure is executed. So that every ATM operation followed by a message with the "117" response code (Suspicious Transaction) appears in the cardholder statement, we must register an Event with "EPRO117" assigned to the *Custom Event Code* parameter for acquiring Products. It should be noted that a special fixed custom code and not a code format is used for the procedure in the above mentioned example. Currently, the system cannot open an Event upon receiving other response codes. For example, if *Custom Event Code* = "EPRO110" or some other is specified for an Event, the Event will not be opened when a message with the "110" response code is received.

 In standard configurations of Products, it is not recommended to use Events with an *Event Start Date* that differs from the current banking date. For example, when a date in the future is specified in the *Event Start Date* field, the Event will be checked daily when executing "Contracts – Daily Update", creating an additional loan on the system. *Event Start Date* can be set using applications (see the document "Advanced Applications R2") or when manually opening Events (see the section "Starting Events by User Command").

Chapter 3. Processing Events

Events can be used to perform certain actions set up in a corresponding Event type and executed while the Event is open.

Standard System Actions

Standard system actions executed when processing Events are directly indicated in the Event registration grid "Event Types". This is used for the following system operations:

- Charging a fee; for this, fill in the *Fee Type* field of the "Event Types" grid
- Changing the contract status; for this, indicate the new contract status in the *New Status* field. For more information, see section "Changing Contract Status".
- Putting the card on a stop-list; for this, indicate "Yes" in the *To Client Stop List* field
- To start the custom procedure; to do so, specify a special code in the *Custom Event Code* field. If this field is filled in, when posting the Event, the custom procedure CUST_EVENT_POST is called, to which this code is passed as the Event code.

Changing Contract Status

To use an Event to change the status of a contract, specify the contract's new status in the Event type's *New Status* field. Note that if an Event changed a contract's status, an Event with the predefined code <Prefix>_<Status Code> (see the section "Opening Events") will only open if it was added to the Event chain.

The presence of the NOT_BETTER; tag in the Event type that changed the contract's status should be considered (see the section "Tags Used when Working with Events"). If the tag is present, the contract status may not change and an Event with a predefined code will be opened.

Example.

- Contract statuses are configured:
 - "Card Do not honor" with "10" in the *PR* field.
 - "PickUP L 41" with "100" in the *PR* field.
- The corresponding Events to change the contract's status are configured"
 - "Event Status to PickUP L 41".
 - "Event Status to do no Honor 05" with the NOT_BETTER; tag. For this Event, an Event with a predefined code that must open when the status

changes is specified in the "Event Chain" form (for example, attaching an Additional Service Package when contract status changes).

When contract status is changed through the "Event Pickup 41" event, the contract status will become "PickUp L 41".

When contract status is changed through the "Event Do not honor 05" event, the contract status does not change (since the priority of the status being assigned is lower than that of the current status). An Event with the predefined code from the Event chain will open.

Generating Documents for Standing Payment Orders

The system can create documents for standing payment orders when an Event is opened or closed. For details on configuring standing payment orders, see the Standing Payment Orders Administrator Manual.

Activating Additional Service Packages

The system can automatically activate/deactivate additional Service Packages for a contract's Service Package when Events are opened/closed for this contract (see the "Configuring Additional Service Packages" section in the WAY4™ Service Packages Administrator Manual).

To do this, click on the [Events] button in a grid for configuring Service Packages, for example, in the "Private Issuing Service Packs" grid. As a result, the screen will display the "Events for <name of Service Package>" form (see Fig. 5).

Event Type	New Serv Pack Type	New Pack	Additional Pack	Switch Tag	Tariff Domain	Report Type	Parameters	Is Ready
Change SP		555-SP-NORM-Our Priv VISA		Yes				Ready
Additional SP		555-SP-NORM-Our Priv VISA-Add		Yes				Ready
Change SP for N days		555-SP-NORM-Our Priv VISA		Yes				Ready

Fig. 5. Form for configuring automatic activation/deactivation of Additional Service Packages

In the grid, click on the [Ins] button to add an empty row and fill in the following fields:

- *Event Type* – name of the Event type, which by being opened will activate/deactivate an Additional Service Package
- *New Serv Pack Type* and *New Pack* – these field are left for backward compatibility
- *Tariff Domain* – used to select one of tariff domains registered in the system; the field can be used to assign tariff domains to contracts when Events are opened/closed for the contracts
- *Additional Pack* – name of the Additional Service Package that will be active/inactive while the Event is open
- *Switch Tag* – determines whether the Additional Service Package is active (the "Yes" value) or inactive (the "No" value) while the Event is open

- *Report Type* – used to redefine the contract's set of reports when an Event is opened (a set of reports is specified in the field of the same name on the Product or contract level)
- *Is Ready* – shows whether changes made to the current Service Package's Event configuration have been approved

Changing Contract Behaviour Types

The system allows users to change a contract's behaviour type when an Event is opened.

To do this, click on the [Event] button in a grid for entering and editing Accounting Scheme information, for example, in the "Private Issuing Account Scheme" grid (see the "Configuring Events" section in the WAY4™ Accounting Schemes Administrator Manual). The screen will display the "Events for <name of Accounting Scheme>" grid (see Fig. 6).

Event Type	New Scheme	Replen Tag	Old Beh Type	New Beh Type	Parameters	Is Ready
Change Bhv Type		Yes		Events test behavior		Ready

Fig. 6. Grid for configuring Events to change contract behaviour types

The "Events for <name of Accounting Scheme>" grid contains the following fields:

- *Event Type* – name of the Event type
- *New Scheme* – this field is used to select a new Accounting Scheme that will be assigned to the contract when this Event is activated (when the contract's behaviour group changes). For example, when an overdue payment appears, the contract can be assigned an Accounting Scheme with a higher interest rate in the loan account.
- *Replen Tag* – this parameter affects interest accrual in contract accounts. The field can take on one of the following values:
 - "Yes" – When an Event changing a behaviour type is opened, the contract accounts will accrue interest for the period from the last accrual date until the date the contract behaviour type is changed
 - "No" – When an Event changing a behaviour type is opened, interest is not accrued for the contract accounts
- *Old Beh Type* – this field indicates the current behaviour type; if this field is left empty when configuring an Event, it will change the behaviour type regardless of the current behaviour type. If this field contains a value, the Event will only change the behaviour type if the current type corresponds to the one set in this field
- *New Beh Type* – this field indicates the new behaviour type value
- *Parameters* – additional system parameterisation while the Event is open.

- This field can indicate the code of the type of the account to which interest will be accrued while an Event is executed. Account types registered in the database can be found in the "Account Types" grid (Full → Configuration Setup → Accounting Setup → Account Types), the account type code is the symbol in the *Code* field
- This field is used to define special parameters as tags. For more details, see the section "Tags Used when Working with Events".
- *Is Ready* – shows whether changes made to the Accounting Scheme have been approved

Changing Contract Account Schemes

It is possible to change a contract Account Scheme when an Event opens.

To do so, in the form for entering and editing Account Scheme data, for example, in the "Private Issuing Account Schemes" form, click the [Events] button (see the section "Configuring Events" in the document "WAY4™ Accounting Schemes"). The "Events for <Account Scheme name>" form will open (see Fig. 6 in the section "Changing Contract Behaviour Types").

The Event type name (*Event Type* field) and new Account Scheme that will be applied to the contract when this Event activates (*New Scheme* field) are set in this form. The original Accounting Scheme will be returned to when this Event closes.



When assigning a new Account Scheme, note that this scheme must be compatible with the original Account Scheme (account templates opened for the contract must be registered in this scheme). Assignment of an incompatible Account Scheme may result in system operation errors.

Generating Event Messages

WAY4 allows users to create messages according to a template when an Event is opened and to send them periodically while the Event is open.

To enter a message template, click on the [Messages] button in the "Event Types" grid (see Fig. 2 in section "Event Types"). As a result, the screen will display the "Messages for <Event type name>" grid, which is described in the "Configuration of Client Messages" document.

Loan Loss Reserving


The system allows reserves to be accrued on credit card contracts regardless of their behaviour types and based instead on opened Events. Detailed information on the system configuration used for this purpose may be found in the Loan Loss Reserves Administrator Manual.


Chapter 4. Closing Events

Events can be closed both automatically and by a user command.

Events are closed automatically when a specified date comes or when certain conditions are met (see "Opening Events"). When an Event is closed automatically, contract parameters are re-assigned the values they had before the Event was opened (for example, the status is restored, the card is excluded from the stop list, additional Service Packages are switched off). The exception is fee documents generated while an Event was open, no reversal documents are created for them.


When an Event is closed by a user command (see "Events Log"), contract parameters are re-assigned the values they had before the Event was opened, and reversal documents are created for fee documents generated while the Event was open.

 Note that if a contract status was changed manually while an Event that had changed its status was still open, the Event closing will not restore its initial status.

 Also, note that before closing an Event, the system waits until all macrotransactions generated as a result of its processing are posted. At the same time, an Event can be closed regardless of macrotransactions posting (see description of field *Post Immediate* in the "Event Types" section).

If the Event closing date (see the description of the *End Date* field in the section "Contract Events Log") falls on a weekend/holiday, the following settings affect calculation (shift) of the End Date:

- The global parameter DUE_TO_WRK_DAY:
 - DUE_TO_WRK_DAY="Y" – if the Event closing date falls on a weekend/holiday, the nearest working day is specified as the Event closing date.
 - DUE_TO_WRK_DAY="N" – the Event closing date is set regardless of weekends/holidays; this is the default value.
- The CLOSE_ON_EVENING tag set in the Event type (see "Tags Used when Working with Events").
- Mode for opening the business day – the mode for combining operations for opening and closing the day, or the mode for separating these operations.

 When the value of the global parameter DUE_TO_WRK_DAY is "N", the date of the weekend/holiday is specified in the *End Date* field if the Event's closing date falls on a weekend/holiday. This date will be changed when the Event closes on the next working day after the weekend/holiday (or on the last working day before the weekend/holiday if the CLOSE_ON_EVENING tag and separate procedures for opening/closing the day are used, see Fig. 7). The date in the *End Date* field will be changed to the date of the corresponding working day.

For more information about how these settings affect the shift in an Event's closing date (End Date), see Fig. 7.

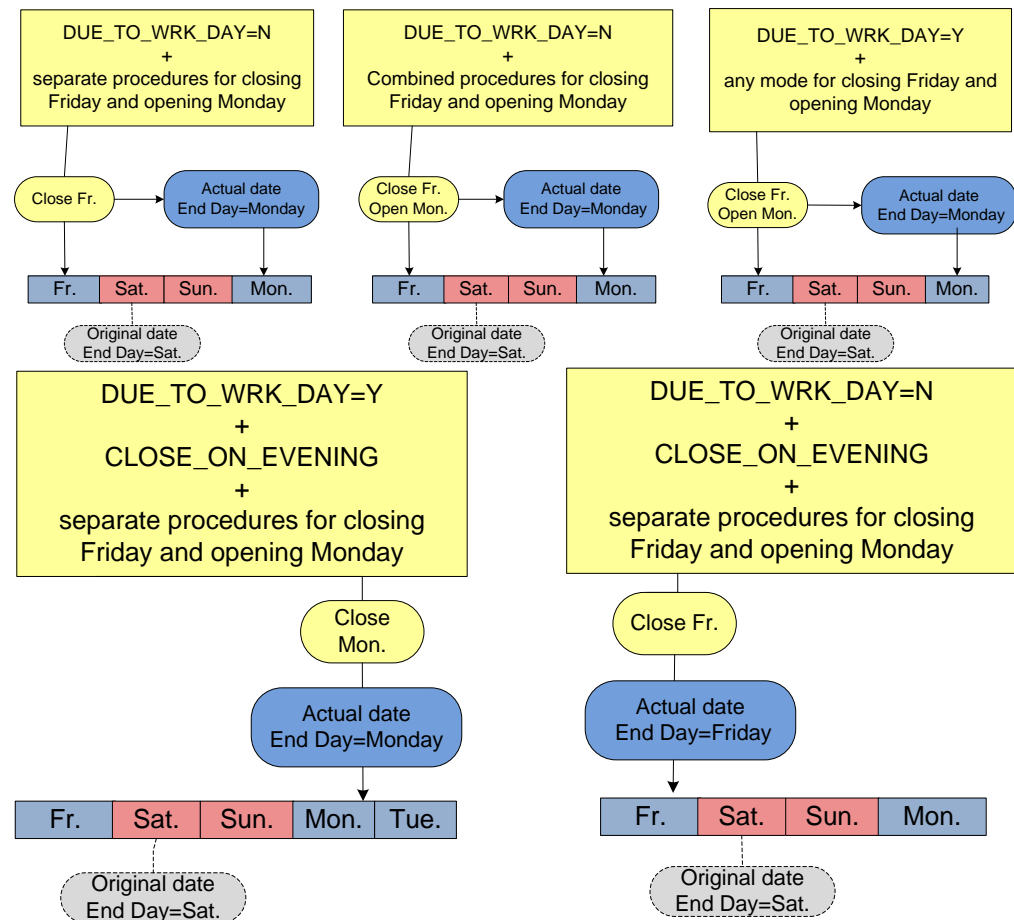


Fig. 7. Influence of system settings on the shift in an Event's End Date

If the *Next Event* field is filled in for an Event, its length will be counted down from the actual End Day of the original Event (see Fig. 7).

Chapter 5. Events Log

All Events processed by the system are registered in the Events Log, which records the state of the contract at the time before the Event is processed. When an Event is closed, the contract parameters must return to the status they had before the Event was opened. Therefore, the Events Log allows Events to be closed both automatically after an indicated period or by a user command.

Description of the "Events for ... Contracts" Form

The Events Log can be accessed in various ways depending on the contract category:

- For issuing contracts, use the "Events for Issuing Contracts" grid (Issuing → Contracts Input & Update → Events for Issuing Contracts)
- For acquiring contracts, use the "Events for Acquiring Contracts" grid (Acquiring → Acquiring Contracts → Events for Acquiring Contracts)

All of these grids contain identical fields.

For example, to view the Events log for issuing contracts, select the "Issuing → Contracts Input & Update → Events for Issuing Contracts" menu path. As a result, the screen will display the "Events for Issuing Contracts" grid (see Fig. 8).

Contract #	Base Relation	Contract Name	Contract Type	Open	Status	Curr	Available
6799990131156972		TEST CARDHOLDE	Our Cirrus/Maestro	08/05/2007	Card OK	EUR	-29,00
001-P-042281		WAY4 TEST	Client Account	31/07/2009	Account OK	USD	0,00

Fig. 8. Form "Events for Issuing Contracts"

The "Events for Issuing Contracts" grid contains a list of all issuing contracts registered in the system and has the following fields:

- *Contract #* – number of the issuing contract
- *Base Relation* – type of same-tier relationship between contracts. Detailed information on same-tier relations may be found in the Issuing Module User Manual and Acquiring Module User Manual.
- *Contract Name* – name of the issuing contract
- *Contract Type* – type of the contract; the list of registered issuing contract types is displayed according to the contract category in the following forms:
 - For card contracts – "Card Contract Types" (Full → Configuration Setup → Contract Types → Card Contract Types)
 - For accounting contracts – "Accounting Contract Types" (Full → Configuration Setup → Contract Types → Accounting Contract Types)
 - For device contracts – "Device Contract Types" (Full → Configuration Setup → Contract Types → Device Contract Types)

- *Open* – date when the contract was opened
- *Status* – contract status; the list of contract statuses registered in the system is presented in grid "Contract Statuses" (Full → Configuration Setup → Contract Types → Contract Statuses)
- *Curr* – contract currency
- *Available* – amount available in the contract accounts

The [Full Info] button in the "Events for <contract category>" form (see Fig. 8) is used to access full information about a contract.

The [Event] button is used to open an Event by a user command (see Starting Events by User Command).

The [Events Log] button is used to work with the Events Log (see Contract Events Log).

The [States] button is used to access active States and Events.

Starting Events by User Command


To start an Event for a contract, go to the "Events for <contract category>" grid form (see Fig. 8 in section "Description of the "Events for ... Contracts" Form"), select a contract, and click on the [Event] button.

The screenshot shows a window titled "Event For Contract". Inside, there are several labeled input fields. "Institution" contains the text "Principal". "Product Category" contains "Issuing". "Event Type" is an empty text box. "Event Start Date" contains "00/00/0000" with the text "(or today)" to its right. "Event Finish Date" contains "00/00/0000" with the text "(or by event rules)" to its right. "Comment Details" is an empty text box. At the bottom right of the window are two buttons: "Cancel" and "Proceed".

Fig. 9. Form for opening Events by a user command

In the "Event for Contract" form that comes up on the screen (see Fig. 9), do the following:

- In the *Event Type* field, select an Event from the drop-down list of registered Events.
- In the *Event Start Date* field, enter the date when the Event should open. The current banking date is the default date.

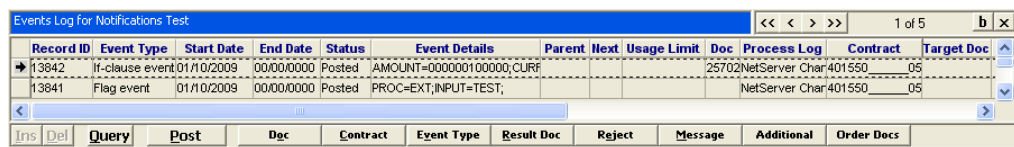
 In standard configurations of Products, it is not recommended to use Events with an *Event Start Date* that differs from the current banking date. For example, when a date in the future is specified in the *Event Start Date* field, the Event will be checked daily when executing "Contracts – Daily Update", creating an additional loan on the system.

- If necessary, enter the date when the Event should into the *Event Finish Date* field. If this field is left blank, the closing date will be selected automatically according to the configuration of the Event type.

After that, click on the [Proceed] button. If the Event opens successfully, the system will inform the user of that fact by the "Event Queued" message. In this case, the Event assumes the "Waiting" or "Posted"/"Suspended" status depending on the parameters of the Event type and the current banking date.

Contract Events Log

To view the Events Log, select the desired contract and click on the [Events Log] button. The screen will display the "Events Log for <contract name>" grid (see Fig. 10).



Record ID	Event Type	Start Date	End Date	Status	Event Details	Parent	Next	Usage Limit	Doc	Process Log	Contract	Target Doc
13842	If clause event	01/10/2009	00/00/0000	Posted	AMOUNT=000000100000;CLRF						25702NetServer Char401550	05
13841	Flag event	01/10/2009	00/00/0000	Posted	PROC=EXT;INPUT=TEST;						NetServer Char401550	05

Fig. 10. Contract Events Log

The "Events Log for <contract name>" grid contains the following fields:

- *Record ID* – unique identification number of the Event record in the Events Log
- *Event Type* – name of the Event type; to view more information about this Event type, click on the [Event Type] button
- *Start Date* – calendar date when the Event opens; if the date indicated in this field falls on a weekend, the Event will open at the start of the first working day after the indicated calendar date. The system distinguishes between weekends and working days through the business calendar (see the "Business Calendar" section)
- *End Date* – calendar date when the Event closes; if the date indicated in the field falls on a weekend day, the Event will close at the start of the first working day after the indicated calendar date. The system distinguishes between weekends and working days through the business calendar
- *Status* – status of the Event; Event posting in the system consists of several stages, and each completed stage in the process changes an Event status:
 - The waiting stage. At this time, the Event has status "Waiting".
 - The Event posting stage. As a result, one of the following statuses may be assigned to an Event:
 - ◆ "Posted" – the Event has been opened successfully. Events take on this status if the close date is not known when the Event is opened (that is,

the *Duration Type* and *Duration* fields in the "Issuing Event Types" form are undefined if the Event is opened automatically or the date is not specified if the Event is opened by a user command)

- ◆ "Suspended" – the Event has been successfully opened; Events take on this status when the close date has been determined through the *Duration Type* and *Duration* fields in the "Issuing Event Types" form if the Event is opened automatically or the date is specified if the Event is opened by a user command
- The Event is closed. Depending on how the Event is closed, it can take on one of the following final statuses:
 - ◆ "Inactive" – the Event is closed because it expires at the time determined by the values of the *Duration Type* and *Duration* fields in the "Issuing Event Types" form if the Event is opened automatically or by the value of the *Event Finish Date* field if the Event is opened by a user command
 - ◆ "Closed" – the Event is closed when the conditions that have opened it change; for example, the Event is opened when the account balance changes. Then, clearing the balance will close the Event with this status
 - ◆ "Rejected" – the Event is closed either manually (see the description of the [\[Reject\]](#) button below) or because conditions that have opened it change; for example, if the Event is opened because of an authorisation request, then its reversal will make the Event status "Rejected"
 - ◆ "Decline" – the Event is closed by other reasons, for example, it is manually closed during the waiting stage



When reissuing a lost card, Events with the "Closed" and "Rejected" statuses are not copied to the new card contract.

- *Event Details* – this field contains additional technical information about the Event, e.g. "INPUT=<Comments>," means that the Event was opened manually and the <Comments> value was specified in the *Comment Details* field when opening the Event; "END=<Reason>," means that the Event was closed, and the <Reason> value indicates the reason of its closing
- *Parent* – identifier of the Event that opened the current Event as part of an Event chain
- *Next* – Event that will be opened when the current Event closes
- *Usage Limit* – name of the usage limiter that opened the Event
- *Doc* – identification code of the document whose creation opened the Event; to access information on this document, use the [\[Doc\]](#) button
- *Process Log* – reference to the record in the Process Log describing the process that opened the Event
- *Contract* – number of the contract for which the Events Log is opened; to view more information on this contract, click on the [\[Contract\]](#) button

- *Target Doc* – this field contains the identification code of the document created as a result of the Event activation, to view information on this document, use the [Result Doc] button
- *New Scheme* – this field is left for backward compatibility
- *New Pack* – this field is left for backward compatibility
- *New Status* – status assigned to the contract as a result of Event activation
- *Std Order* – identification code of the standing order for which the document that opened the Event was generated; the [Order Docs] button is used to display information about this document
- *Custom Event Code* – a special code passed to the CUST_EVENT_POST procedure as the Event code.
- *Fee Type* – type of the fee that was charged when the Event was opened
- *New Beh Type* – behaviour type assigned to the contract after the Event was opened

The [Post] button in the "Events Log for <contract name>" grid (see Fig. 10) invokes the Event posting procedure.

The [Message] button is used to access messages created for the current Event.

To close an Event, select it from the Events list and click on the [Reject] button. The screen will display the "Reject for Events Log for <contract name>" form (see Fig. 11).

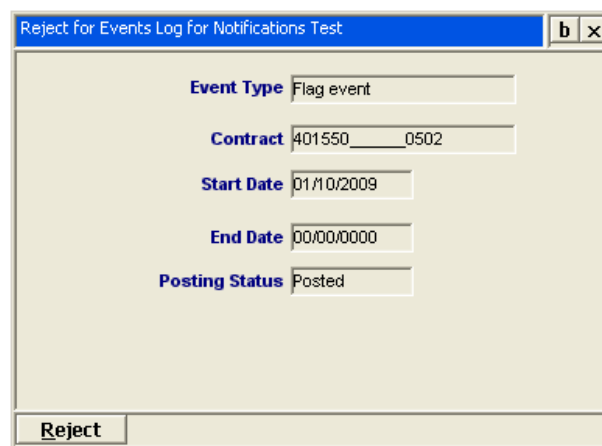


Fig. 11. Form for manually closing Events

The "Reject for Events Log for <contract name>" form displays Event parameters. To close the Event, click on the [Reject] button. If the Event is successfully closed, the "Event Reversed" message will be brought to the screen.

The [Additional] button in the "Events Log for <name of contract>" form (see Fig. 10) is used to open the "Additional for Events Log for <name of contract>" form (see Fig. 13 in the Examples of Event Use section). It contains information on operations executed by the system as a result of Events processing.

Action Code	Old ID	New ID	Event Details	Status
ORDER	1514	120544		P

Ins Del Query

Fig. 12. Form containing information on operations executed as a result of Events processing

This form contains the following fields:

- *Action Code* – operation code; the field can take on one of the following values:
 - ORDER – a document is generated for a standing payment order
 - TOGGLE_PACK – an Additional Service Package is activated/deactivated
- *Old ID* – field is reserved for future use
- *New ID* – identifier of a document created for a standing payment order
- *Event Details* – additional Event data (see the description of the [field of the same name](#) in the "Events Log for <name of contract>" form)
- *Status* – operation status

"Requested Event Uploading" Procedure

"Requested Events" are Events on which system business logic depends and whose activity for contracts is analysed regularly in large-scale procedures ("Document Processing", "Contracts – Daily Update", "Reserves Accrual"). For example, Events related to balance types, Events setting a reserve portfolio for a contract, Events with the predefined codes FULL_PAYMENT, LATE_PAYMENT, etc.

All Events processed by WAY4 are registered in the Event log in the USAGE_ACTION table. Only a small number of Events are "Requested Events".


WAY4 provides for a mechanism to synchronise and quickly search for "Requested Events" by contracts. Using this mechanism, the time taken for a response to an inquiry on the activity or inactivity of an Event for a contract is significantly shortened.

WAY4 itself marks the required Event types as "Requested Events". These Event types initially fall into a special "Not Ready" collection. Immediately after marking, all newly opening and closing Events of this type for contracts begin synchronisation to the special table ADD_PACK_INC. Not all information from USAGE_ACTION is synchronised, only a part of it, which significantly saves on data volume.

Since synchronisation is started after marking by WAY4, all Events of this type opened earlier remain unsynchronised (i.e. they are contained in the USAGE_ACTION table and absent from the ADD_PACK_INC). In the case of Event types recently registered in WAY4, this may be just a few unsynchronised records. In the case of Event types that have long been in existence in WAY4, but that were only recently marked by the system, there may be a significant

number of unsynchronised records. This makes it impossible to use the quick search mechanism, since it is unknown for which contracts which Event types are synchronised. As a result, instead of limiting the search by the small ADD_PACK_INC table, WAY4 continues to search in the large USAGE_ACTION table.


The "Requested Events Uploading" procedure fully synchronises these Event types. After full synchronisation, the Event type goes from the "Not Ready" collection to the "Ready" collection. For WAY4, this move is a sign that synchronisation has been executed and in the ADD_PACK_INC table data for this Event type are full and reliable. This allows optimisation of these Event searches, both by contract and by Event type. And, correspondingly, to decrease the time it takes to execute large-scale procedures. Data for each Event type is fully synchronised once.

 The Event search process is optimised (system performance improved) only after the "Requested Events Uploading" procedure is executed.

The "Requested Events Uploading" procedure can be started during daily procedures or separately (Full → Daily Procedures → Start of Day Step By Step → Requested Events Uploading).

Recommendations for starting the "Requested Event Uploading" procedure:

- **Initial start** of the procedure is recommended to be done separately from daily procedures since it may take a significant amount of time (due to a large number of records in the "Not Ready" collection).
- **Daily repeat start** after initial execution of the "Requested Event Uploading" procedure is recommended to be done every day. The frequency and means of running the procedure depends on the number of records in the "Not Ready" collection of the DM_COLLECTION table. Before executing the procedure, it is recommended to estimate the number of "Not Ready" records (estimate the number of Event types for subsequently starting the "Requested Events Uploading" procedure) and, respectively, the time it will take to process them, using the menu item "Full → Daily Procedures → Start of Day Step By Step → Requested Event Types Check":
 - If there are no records in the "Not Ready" collection, it is not necessary to run the procedure. If there are no records in the "Not Ready" collection, the procedure will take a few seconds.
 - If the number of records in "Not Ready" > 0, the menu item should be started:
 - ◆ In daily procedures, if there is a small number of records.
 - ◆ Separately from daily procedures using the menu item "Full → Daily Procedures → Start of Day Step By Step → Requested Events Uploading", if the number of records in the "Not Ready" is such that it is estimated their processing will be lengthy.

 The "Requested Event Uploading" procedure should not be started when running the processes "Document Processing", "Contracts – Daily Update", and "Reserves Accrual".

After running the "Requested Events Uploading" for the first time, the appearance of new records in the "Not Ready" collection may be caused by the following actions in WAY4:

- Creation of a new financial institution and copying settings to this institution.
- Creating new Event types invoked unambiguously in large-scale processes. For example, in the "Contracts – Daily Update" procedure, Events with the predefined codes FULL-PAYMENT and LATE_PAYMENT are searched for.

When the "Requested Events Uploading" is started, this action is registered in the process log.

Appendix 1. Examples of Event Use

Overdue Debt

The following example will illustrate the use of Events in the system. Say that a bank is set to follow this sequence of actions when a client's loan becomes overdue:

- A message is sent to the client when the loan reaches the overdue state.
- If the loan is not repaid within 25 days, the card is blocked, and a notification to that effect is generated.
- After 20 more days, the card must be added to the stop list and picked up if the user attempts to authorise it

For this sequence to be realised, it is necessary to take the following actions:

- Configure three Events as shown in Fig. 13.
- Create a message template. An example of a message template for the "Overdue" Event is presented below (see Fig. 13). A template for the "Block Card" Event should contain information on the bank card, for example, "Be informed that your card %CONTRACT_NUMBER:4X5% is blocked."
- In the *Open/Close Event* field of the overdue account template (for example, "CI OVD"), indicate "Overdue"
- For the account template to which funds will be transferred after 20 days (for example, "CI OVD Long"), indicate "Pick Up" in the *Open/Close Event* field.

Contract	Name	Code	Duration Type	Duration	Due To Work Day	Next Event	Event Custom Code	Group Code	Post Immediate	Special Params
Card	OVERDUE	OVD	Day	25		Block Card				
Card	Pick Up	STOP	Single	0						
Card	Block Card	BLOCK	Single	0						

Messages for OVERDUE

Delivery Channel: To Statement

Address: Statement

Message Text: Dear %TITLE% %FIRST NAME% %LAST NAME%,
Be informed that you have Bad Debts of card %CONTRACT_NUMBER:4x5%

Active From: 18/01/2011

Active To: 00/00/0000

Sending Time From: 00:00

Sending Time To: 00:00

Fig. 13. Example of system Event setup

When the system is set up as described above, the following processes will be activated. If the loan is not repaid on time, funds will be transferred to the "CI OVD" account according to due normalisation. The "Overdue" Event indicated in the *Open/Close Event* field of the "CI OVD" template will open, and

the database will create a message according to the message template (see Fig. 13). The client will receive this message in the next statement.

If the loan is repaid within 25 days, the amount on the "CI OVD" account will be annulled according to limit normalisation and the "Overdue" will be closed.

If the loan is not repaid, the "Overdue" Event will close and the "Block Card" Event will open. The card contract will take on the "Do not Honour" status, and the database will create a corresponding message.

If the loan is repaid within 20 days, the amount on the "CI OVD" account will be annulled according to limit normalisation, the "Block Card" Event will close, and the card contract will return to the status it had had before it was blocked.

If the loan continues to be unpaid after 20 days, the amount on the "CI OVD" account will be transferred to the "CI OVD Long" account. As a result, the "Pick Up" Event will open, which will assign the "Pick Up 04" status to the card contract, and the card will be put into the stop list. Since the "Block Card" Event has not closed, the client will receive a notification of the card being blocked along with every account statement.

No Contract Activity

Several Events can be set up for various periods when there is no contract activity (to charge fees, send notifications). To do so, an Event with the predefined code CONTRACT_DELAY% is used (the CONTRACT_DELAY prefix is mandatory in this Event's code, the postfix is optional or may be arbitrary).

Setup for these Events is illustrated in the following example: an Event with the CONTRACT_DELAY20 code will activate if there were no financial transactions with the contract for 20 days and an Event with the code CONTRACT_DELAY60 – if there were no financial transactions with the contract for 60 days.

Setup procedure:

- The period of no contract activity is set up using the ACTIVITY_DELAY tag in the Service Package (see the description of the tag in the section "Tags Used when Working with Events").

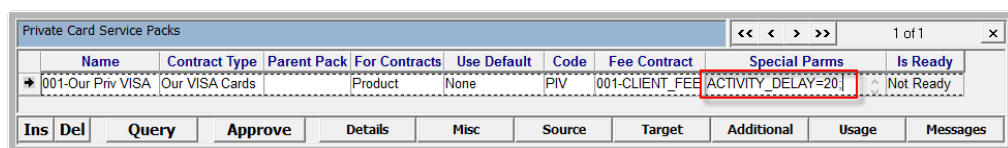


Fig. 14. Setting up the period of no contract activity in a Service Package

- In the transaction subtype, set up the ACTIVITY_FOR tag so the Posting Date of the next transaction for this subtype is recorded in the contract's ext_data field using the ACTIVITY_DATE tag (see Fig. 15 and the description of the ACTIVITY_FOR tag in the section "Tags used when Posting Documents" of the document "Documents").

SubTypes for Retail									
Source Cat	Target Cat	Source Type	Target Type	Source Acc Type	Target Acc Type	Triggered Event	Fee Algorithm	Options	Name
Device	Card	Our POS	Our VISA Cards	Merchant Receivable	CI Deposit		ACTIVITY_FOR=T		Retail
<input type="button" value="Ins"/> <input type="button" value="Del"/> <input type="button" value="Query"/>									

Fig. 15. Setting up a transaction subtype

- Event type setup (see Fig. 16):

- An Event type with the predefined code CONTRACT_DELAY is set up. The Event opens if there were no transactions with the contract for 20 days (since the ACTIVITY_DELAY=20; tag is set in the Service Package, see Fig. 14). Set the tag ONE_AT_TIME; in this Event type and the "Single" type so the Event doesn't open each time daily procedures are run (Contracts – Daily update).

i If the period without contract activity exceeds the number of days specified with the ACTIVITY_DELAY tag, the CONTRACT_DELAY=Y tag is automatically set in the contract.

- For each period (20, 60 days without contract activity) a separate Event type is set up. In the example, these are Event types with the CONTRACT_DELAY20 and CONTRACT_DELAY60 codes. An Event chain is used to set up opening these Events as follows:
 - The CONTRACT_DELAY20 Event type is set up in an Event chain for the CONTRACT_DELAY main Event type. When a CONTRACT_DELAY Event opens, a CONTRACT_DELAY20 Event opens automatically (and the corresponding fee is charged).
 - If no financial transactions were made with a contract in 60 days, the CONTRACT_DELAY20 Event closes (goes to the "InActive" status) and the CONTRACT_DELAY60 Event specified in the *Next Event* field of the CONTRACT_DELAY20 Event opens, see Fig. 16.

See the description of the Event with the predefined tag CONTRACT_DELAY% in the section "Opening Events".

Event Types										<< < > >>				1 of 3		x	
	Product	Contract	Institution	Name	Code	Group Code	Duration Type	Duration	Next Event	Custom Code	Special	Parms					
★	Issuing	Card	Principal	CONTRACT_DELAY	CONTRACT_DELAY		Single	0				ONE_AT_TIME					
	Issuing	Card	Principal	CONTRACT_DELAY20	CONTRACT_DELAY20		Day	20	CONTRACT_DELAY60								
	Issuing	Card	Principal	CONTRACT_DELAY60	CONTRACT_DELAY60		Day	40									
Ins Del Query Check Messages Full Info Event Chain Used By Mapping																	
Event Chain for CONTRACT_DELAY										<< < > >>				1 of 1		b	x
	Next Event Type			For Contract		Active If State		Active If State Status		Chain Details		Step N					
★	CONTRACT_DELAY20			Contract Itself								0					
Ins Del Query																	

Fig. 16. Example of Event setup

i If transactions with the contract resume, all open Events with the predefined code CONTRACT_DELAY% close (and the CONTRACT_DELAY=Y tag in the contract is automatically removed).

Appendix 2. Tags Used when Working with Events

Name	Value	Description
Tags in the <i>Special Parameters</i> field of an Event type		
CLOSE_ON_EVENING		An Event will be closed during CDU, before the banking date is entered.
NOT_CLOSE_AFTER_EXPIRE		This tag is only used if the <i>Next Event</i> field is filled in for the Event and a time period (Duration) is set. When this tag is set, after expiration of the effective period, the Event set in the <i>Next Event</i> field will open. The original Event will not close (remains in an unclosed status). This tag, for example, is used for Events that change Behaviour Type.
ACC_SCHEME		When processing an Event, a contract's accounts will be brought into accordance with the Accounting Scheme (if the Accounting Scheme was changed).
CLOSE_FOR_DUPLICATE		When copying a contract under which there is an Event (in the "Posted" or "Suspended" status) with this tag, the Event is copied under the new contract and closed. The Event remains in the earlier state under the original contract. Events are copied, for example, when copying a card contract in the case of a lost card - the tag allows a processed Event that is executed when the lost card is reissued to be closed.
MOVE_FOR_DUPLICATE		This tag makes it possible when copying a card contract to transfer an earlier posted Event that is executed when reissuing a lost card. The event is closed under the old contract, under the new contract the Event remains - meaning that the Event is transferred from the old contract to the new one.
LOCAL_DATE		This parameter allows the current banking date of the financial institution to be used as an Event opening date (<i>not</i> the date specified in the <i>Start Date</i> field). Used for "Post Later" and "Post and Close Later" Event types.

Name	Value	Description
START_ON_EVENING		An Event will be posted during evening Contract Daily Update (before entering the banking date following the Event opening calendar date in the <i>Start Date</i> field). The Event is posted by the process "Full → DB Administrator Utilities → Special OpenWay Utilities → Parallel Run → Posting Events" started after execution of the "Evening Contract Daily Update" procedure. By default (if the tag is not set), the Event is posted during the daily opening procedure of the banking day corresponding to the calendar date of the Event's opening in the <i>Start Date</i> field.
FROM_VALID		Contract status is changed according to the value in the <i>New Status</i> field only for contracts with the "Valid" status type.
NOT_BETTER		Contract status according to the value in the <i>New Status</i> field is only changes if the priority of the new status is higher than the priority of the old status. Status priority is specified using the "PR" value in the <i>Code Parms</i> field in the "Full → Configuration Setup → Contract Types → Contract Statuses" table. For example, for the status "PickUp S 43" the <i>Code Parms</i> field contains the variable "PR=43" and for the status "Card Do not honour", the <i>Code Parms</i> field contains the variable "PR=5". Then if the card has already been stolen, it is not assigned the status "Card Do not honour" if an Event opens that is posted when delinquent debt on a loan occurs.
CLOSE_EVENT	<EventCode1>, <EventCode2>, ..., <EventCodeN>	When this Event opens, an Event (Events) with the specified code (codes) will be closed. Codes are comma-delimited. The Event code is the <i>Code</i> field value of the "Issuing Event Types" form.
UNLOCK_PLASTIC		When posting this Event, the plastic for this contract will be unlocked.
ADD_MSG	<message code>	When processing an Event, a message with a certain code, configured for the Accounting Scheme and/or Service Package will be generated for a contract. Note: a message code mask can be set as the tag value (using the service character "%").
SKIP_CHARGE		A fee set in the <i>Fee Type</i> field will not be charged at the time the Event opens, but after the next posting of macrotransactions generated for this contract; if no macrotransactions are processed, the fee will be withheld within the CDU procedure.

Name	Value	Description
MTR_FEE		When charging a fee according to an Event, allows the macrotransaction from the fee to be "appended" to the original document, which results in reversal of the fee when the original transaction is reversed.
PREDEF_AMNT		When calculating a fee charged according to an Event, allows the fee amount and currency to be taken from Event tags: from the AMOUNT and CURR tags set in the <i>Event Details</i> (EVENT_DETAILS) field of the "Event for <contract name>" form (USAGE_ACTION table).
REVERSED_BY_DOC;		<p>If this tag is set in an Event:</p> <ul style="list-style-type: none"> • When the document for which this Event opened is reversed, all actions caused by the Event will be reversed. Example. In a Service Package, a limiter is set that calls an Event when the card status changes. When a transaction subject to the conditions of this limiter is made, the contract status changes. If this transaction is reversed, by default, the Event is not reversed. When the tag is set, all consequences of the Event will be reversed. • When recalculating a contract's lifecycle with Reversal Management module tools, this Event is not reopened. I.e., the fee set up in the Event (Event type's <i>Fee Type</i> field) will not be charged again. <p>Example</p> <p>A document is generated for an Event. This document is reversed in the usual way (when a reversal document is posted). If this Event falls in the Reversal Management correction plan's effective period, the Event is reopened by default, i.e. the fee is charged again. When the REVERSED_BY_DOC; tag is set, when the document is reversed, the Event will be assigned the "Rejected" status, and will not be opened after recalculation (repeat charging of the fee is disabled).</p>
RENEW_TYPE_TAG	<tag name>	<p>Allows the value of the <i>Duration Type</i> parameter in an Event to be replaced in the following way:</p> <p>If the RENEW_TYPE_TAG=<tag name>; is specified in the <i>Special Params</i> field in the Event, when this Event opens for the contract, the system receives the value of this <i>tag</i> from the ext_data field of the contract (from the acnt_contract table).</p> <p>If <i>Duration Type</i> must be replaced for an Event type under a Product (Start Events) with the "Lifecycle" type, the system receives the value of this tag from the custom_data field of the Product (from the</p>

Name	Value	Description
		appl_product table). If a non-null value is received, it will replace the <i>Duration Type</i> parameter in this Event. See also RENEW_PERIOD_TAG.
RENEW_PERIOD_TAG	<tag name>	Allows the value of the <i>Duration</i> parameter in an Event to be replaced in the following way: If the RENEW_PERIOD_TAG=<tag name>; is specified in the <i>Special Params</i> field in the Event, when this Event opens for the contract, the system receives the value of this tag from the ext_data field of the contract (from the acnt_contract table). If <i>Duration</i> must be replaced for an Event type under a Product (Start Events) with the "Lifecycle" type, the system receives the value of this tag from the custom_data field of the Product (from the appl_product table). If a non-null value is received, it will replace the <i>Duration</i> parameter in this Event. See also RENEW_TYPE_TAG.
AVAILABILITY	"G" "R" "D"	Makes it possible to open an Event only for those contract for which this Event is registered in the Service Package. Parameter values: <ul style="list-style-type: none"> • "R" – An Event must always be registered in the Service Package for it to open. • "G" – It is not necessary for an Event to be registered in the Service Package for it to open. • "D" – default value, corresponds to standard system behaviour; Events opened manually don't require registration in a Service Package; when an Event opens automatically (i.e. an Event with predefined codes opens), it must be registered.
ONE_AT_TIME		Only one active Event with this tag (with one code) can be open any time for this contract.
IF_CS_TYPE	<CS_STATUS_TYPE.CODE>	This tag sets the type of checked classifier. 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

Name	Value	Description
		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>	An Event will be opened only if clients or their contracts correspond to the set classifier value. 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>	An Event will not be opened if the value specified in the tag corresponds with the value of the classifier in the contract or client. 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.
CALENDAR_TYPE	<name of business calendar type>	The tag makes it possible to close/reopen an Event according to a specific calendar differing from the financial institution's calendar.
IF_INACTIVE_EVNT	<EventCode>	An Event will be opened only if the Event specified in the tag (with the specified code) is closed.
IF_ACTIVE_EVNT	<EventCode>	An Event will be opened only if the Event specified in the tag (with the specified code) is opened.

Name	Value	Description
STATUS_CATEGORY	"V" "I" "D"	An Event opens and is processed (actions specified when the corresponding Event type was configured are performed) only if the contract is in the status with the set category (the value of the <i>Is Valid</i> field of the "Contract Statuses" form "Full → Configuration Setup → Contract Types → Contract Statuses"): <ul style="list-style-type: none"> • "V" – Valid • "D" – Decline • "I" – Invalid
PROD_CHANGE_RULE	"CLOSE" "OPEN" "OPEN,CHANGE_ONLY"	This parameter allows the state of an Event to be changed when the Product changes. <ul style="list-style-type: none"> • When the value is "CLOSE", when a Product changes, the Event in the contract closes (when the Product is still old) • When the value is "OPEN" – this Event opens when a contract is created and when the Product changes (when the Product has already changed). • "OPEN,CHANGE_ONLY" - this Event only opens when the Product changes. The Event does not open when the Product opens. Default behaviour (if the tag is not set) – when the Product changes, the Event under the contract does not change its state. In this case, the Event only opens when the contract is created.
ONE_FOR_DOC		The tag ONE_FOR_DOC allows one notification message to be sent to a client for one transaction document. For example, if one transaction violates several risk rules.
TILL_FIRST_DAY		This parameter is used for Events with the "Month" value in the <i>Duration Type</i> field and makes it possible to close an Event at the end of a calendar month (regardless of when it was opened). This tag is used for recurring <i>Events</i> (Event type with the same Event type specified in the <i>Next Event</i> field).
CLOSE_IF_EMPTY	<balance type code1>, <balance type code2>, OPEN_CASE_LIST	Used in the contract closing procedure. Set up in the Event used for closing contracts (see the section "Opening Events"). Balance type codes and/or the OPEN_CASE_LIST value, separated by commas are specified as the tag value. The following check is made: <ul style="list-style-type: none"> • If balance type codes are specified as the tag value, a check is made of whether under the contract there are non-null balances with these codes. IMPORTANT! Both standard balance types (AVAILABLE, TOTAL_BALANCE), and those configured in an Accounting Scheme

Name	Value	Description
		<p>can be used.</p> <p>If there are non-null balances, the Event does not close (i.e. the contract is not closed). The Event remains in the Suspended status and will be checked during the next daily closing.</p> <p>The CLOSE_IF_EMPTY_LIMIT tag and tariff with the "Threshold" role can be used to set up the maximum balance amount at which the contract will be closed. If the balance amount is less than the amount set in the tariff, the Event will close.</p> <ul style="list-style-type: none"> • When the OPEN_CASE_LIST value is present, a check is made to see if the contract and its subcontracts have open cases (in particular, "Dispute" category). • When the global parameter CHECK_OPEN_APPL_CASE="Y", a check is made for open cases for all domain categories. • When CHECK_OPEN_APPL_CASE="N" - for all domain categories except application cases. <p>If open cases (in the "Active" status) are found, the Event is not closed.</p>
IF_PARM	<tagged parameter name> CLIENT.SHORT_NAME CLIENT.FIRST_NAME CLIENT.LAST_NAME CLIENT.COMPANY_NAME CLIENT.AGE	<p>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".</p> <p>A number of client parameters can be checked. To do so, the following values can be specified as the tag's value:</p> <p>CLIENT.SHORT_NAME – check the <i>Short Name</i> field in the client form.</p> <p>CLIENT.FIRST_NAME – check the <i>First Name</i> field in the client form.</p> <p>CLIENT.LAST_NAME – check the <i>Last Name</i> field in the client form.</p> <p>CLIENT.COMPANY_NAME – check the <i>Company Name</i> field in the client form (place of work)</p> <p>CLIENT.AGE – client age. Determined according to date of birth (<i>Date of Birth</i> field) and the system date (i.e. age at the time of the request is determined).</p> <p>These client attributes do not have to be registered as contract custom parameters.</p> <p>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.</p>

Name	Value	Description
IF_PARM_VALUE	"<tagged parameter value>" "NOT_EMPTY" "EMPTY" "LIST_WITH:<list of values separated by commas>" "BETWEEN<value> AND <value>"	<p>An Event is opened and processed (actions specified when setting up the corresponding Event type are executed) 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.</p> <p>When the IF_PARM_VALUE=EMPTY; tag is set, the Event opens if the specified parameter doesn't have a value. Note that with this value, the Event will also open if the parameter is not present in the contract.</p> <p>When "IF_PARM_VALUE=LIST_WITH:<list of values separated by commas>" is set, the Event opens if even one value from the list (LIST_WITH) matches that set in the contract.</p> <p>When "IF_PARM_VALUE=BETWEEN<value> AND <value>" the Event opens 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.</p> <p>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.</p>
ZERO_BAL_TYPE	<balance type code>	An Event will only open 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.
NON_ZERO_BAL_TYPE	<balance type code>	An Event will only open 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.
POST_DUE	"Y" "P" "N"	<p>This parameter makes it possible to shift the dates of an entry made as the result of activating an Event (activation of a payment order configured in the Event) in the Instalment Management module.</p> <ul style="list-style-type: none"> • When the value is "Y" – Posting Date and GL Date are set to the previous calendar day, except for cases when the entry is posted on the first calendar day of a new billing cycle. In this case, Posting Date and GL Date are set to the current calendar day. • When the value is "P" – Posting Date and GL Date are set to the previous working day.

Name	Value	Description
		<ul style="list-style-type: none"> When the value is "N" – Posting Date and GL Date are set to the current banking date.
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: <ul style="list-style-type: none"> 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 codeN> code2>,...,<currency <currency name1>, <currency nameN> name2>,...,<currency <tag name>	An Event is only opened and posted (actions specified when setting up the corresponding Event type) if the document currency specified with the IF_CURRENCY tag corresponds to parameters set in the IF_CURRENCY_VALUE tag. Possible values: <ul style="list-style-type: none"> EMPTY – this currency is not specified. NOT_EMPTY – this currency is specified. USE_IN_BANK – this currency has the <i>Use in Bank</i> 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 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.

Name	Value	Description
IF_CURRENCY_RULE	ACC_SCHEME EQUAL NOT_EQUAL NOT_IN_LIST	<p>To set additional conditions for checking a currency, use the IF_CURRENCY_VALUE tag together with the IF_CURRENCY_RULE tag:</p> <ul style="list-style-type: none"> • 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.
BILLING_EVNT	"E" "S"	<p>For Events with the "Billing" value in the <i>Duration Type</i> field, makes it possible to close an Event at the end of a billing cycle:</p> <ul style="list-style-type: none"> • "E" – when closing the last banking day of the billing cycle. • "S" (default value) – when opening the first day of the new billing cycle.
CLOSE_IF_EMPTY_LIMIT	<tariff type code>	<p>The tag is used together with the CLOSE_IF_EMPTY tag when closing contracts. Set in an Event used to close contracts (see the section "Opening Events"). The tag sets the code for a tariff type with the "Threshold" role that determines the balance amount (if the contract has non-empty balances) at which an Event is closed (i.e. the contract is closed). I.e. if the balance amount is less than the amount set in the tariff, the contract is closed.</p>
MAX_SETTL_AMOUNT	<amount> <custom parameter>	<p>The MIN_SETTL_AMOUNT and MAX_SETTL_AMOUNT tags allow document amount limits to be set for which this Event will be activated</p>

Name	Value	Description
MIN_SETTL_AMOUNT	<amount> <custom parameter>	(for example, to send a notification by an Event) when a transaction is made with the corresponding Service (Service with the SERV_EVNT=<this Event type's code>; set; see the description of the SERV_EVNT tag in the transaction subtype). The value is specified as a fixed amount in the contract currency. The minimum and/or maximum amount that is specified is compared with the document's Settlement Amount. An Event is processed if the document amount is more than MIN_SETTL_AMOUNT and/or less than MAX_SETTL_AMOUNT. Contract and client custom parameters can be used as tag values (see the section "Contract and Client Custom Parameters" of the document "WAY4™ Client and Contract Classifiers") .
CLOSE_INST_PLANS	"Y"	The CLOSE_INST_PLANS=Y; tag is used to automatically close a contract's instalment plans when an Event activates. If only some instalment plans should be closed for a contract, the codes of these plans (<i>Service Code</i> field value of the corresponding instalment schemes) should be listed, separated by commas, as the value of the CLOSE_BY_SERV_CODE tag. If the CLOSE_BY_SERV_CODE tag is not set, all instalment plans for the contract will be closed. This tag is used when working with the WAY4 Instalments module. The module is delivered according to a separate agreement with the WAY4 vendor.
CLOSE_BY_SERV_CODE	<Service Code1, ServiceCode2,...,ServiceCodeN>	The tag makes it possible to define a contract's instalment plans that will be closed when an Event with the CLOSE_INST_PLANS=Y tag opens. The codes of these plans (<i>Service Code</i> field value of the of the corresponding instalment schemes) should be listed, separated by commas, as the value of the CLOSE_BY_SERV_CODE tag. If the CLOSE_BY_SERV_CODE tag is not set, all instalment plans for the contract will be closed. This tag is used when working with the WAY4 Instalments module. The module is delivered according to a separate agreement with the WAY4 vendor.
IF_PARM_FOR	"BILLING" "LIABILITY" "TOP" "BASE"	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.

Name	Value	Description
	"DOC_SOURCE" "DOC_TARGET" "FROM_DOC" "CONTRACT_ROLE" CONTRACT_ROLE_PARM=<tag name> "LIAB_CATEGORY"	<ul style="list-style-type: none"> • "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 <i>Source Contract</i> (source_contract) field (see the "Doc-Brief" form). • "DOC_TARGET" – for the contract specified in the document's <i>Target Contract</i> (target_contract) field (see the "Doc-Brief" form). • "FROM_DOC" – the contract is taken from the document's <i>Add Data</i> (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: 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

Name	Value	Description
		<p>"R" – "Reporting" category "A" – "Only Check Balance" category.</p> <p>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.</p> <p>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.</p>
IF_CS_TYPE_FOR	<p>"BILLING" "LIABILITY" "TOP" "BASE" "DOC_SOURCE" "DOC_TARGET" "FROM_DOC" "CONTRACT_ROLE" CONTRACT_ROLE_PARM=<tag name> "LIAB_CATEGORY"</p>	<p>The IF_CS_TYPE_FOR tag is used together with IF_CS group tags to redefine the contract for which these checks are made.</p> <ul style="list-style-type: none"> • "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. • "DOC_SOURCE" – for the contract specified in the document's <i>Source Contract</i> (source_contract) field (see the "Doc-Brief" form). • "DOC_TARGET" – for the contract specified in the document's <i>Target Contract</i> (target_contract) field (see the "Doc-Brief" form). • "FROM_DOC" – the contract is taken from the <i>Add Data</i> (add_info) field. • "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

Name	Value	Description
		<p>in the hierarchy will be selected.</p> <ul style="list-style-type: none"> 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 "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"	<p>Redefines the contract for which a check is made using the IF_CURRENCY tag. Possible values:</p> <ul style="list-style-type: none"> "BILLING" – for the account contract from which settlement is made.

Name	Value	Description
	"BASE" "DOC_SOURCE" "DOC_TARGET" "FROM_SERVICE" "FROM_DOC" "RELATED" "DOC_TARGET_NUMBER" "CONTRACT_ROLE" CONTRACT_ROLE_PARM=<tag name> "LIAB_CATEGORY"	<ul style="list-style-type: none"> • "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 in the <i>Source Contract</i> field of the document (source_contract) of the document (see the "Doc-Brief" form). • "DOC_TARGET" – for the contract in the <i>Target Contract</i> field (target_contract) of the document (see the "Doc-Brief" form). • "FROM_SERVICE" – for the contract specified in the Service (fee_contract, fee_account fields). • "FROM_DOC" – the contract is taken from the document's <i>Add Data</i> field (add_info) according to the tag specified using the CONTRACT_TAG tag in the same field) • "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

Name	Value	Description
		<p>in this field using the LIAB_CATEGORY=<category1>,<category2> tag.</p> <p>Liability category codes, separated by commas, are specified as the LIAB_CATEGORY tag value:</p> <p>"Y" – "Full Liability" category</p> <p>"N" – "Affiliated" category</p> <p>"R" – "Reporting" category</p> <p>"A" – "Only Check Balance" category.</p> <p>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.</p>
ZERO_BAL_TYPE_FOR	"BILLING" "LIABILITY" "TOP" "BASE" "DOC_SOURCE" "DOC_TARGET" "FROM_SERVICE" "FROM_DOC" "RELATED" "CONTRACT_ROLE" CONTRACT_ROLE_PARM=<tag name> "LIAB_CATEGORY"	<p>Redefines the contract for which a check is made using the ZERO_BAL_TYPE tag. Possible values:</p> <ul style="list-style-type: none"> • "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 in the <i>Source Contract</i> field of the document (source_contract) of the document (see the "Doc-Brief" form). • "DOC_TARGET" – for the contract in the <i>Target Contract</i> field (target_contract) of the document (see the "Doc-Brief" form). • "FROM_SERVICE" – for the contract specified in the Service (fee_contract, fee_account fields). • "FROM_DOC" – the contract is taken from the document's <i>Add Data</i> field (add_info) according to the tag specified using the CONTRACT_TAG tag in the same field) • "RELATED" – related contract, with the relation specified in the RELATION tag.

Name	Value	Description
		<ul style="list-style-type: none"> • "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: "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

Name	Value	Description
		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" "DOC_SOURCE" "DOC_TARGET" "FROM_SERVICE" "FROM_DOC" "RELATED" "CONTRACT_ROLE" CONTRACT_ROLE_PARM=<tag name> "LIAB_CATEGORY"	<p>Redefines the contract for which a check is made using the NON_ZERO_BAL_TYPE tag. Possible values:</p> <ul style="list-style-type: none"> • "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 in the <i>Source Contract</i> field of the document (source_contract) of the document (see the "Doc-Brief" form). • "DOC_TARGET" – for the contract in the <i>Target Contract</i> field (target_contract) of the document (see the "Doc-Brief" form). • "FROM_SERVICE" – for the contract specified in the Service (fee_contract, fee_account fields). • "FROM_DOC" – the contract is taken from the document's <i>Add Data</i> field (add_info) according to the tag specified using the CONTRACT_TAG tag in the same field) • "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: 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:

Name	Value	Description
		<p>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</p> <ul style="list-style-type: none"> • "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. <p>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.</p> <p>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.</p>
IF_CS_NUMB	<number of additional checks>	<p>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.</p>
IF_PARM_NUMB	<number of additional checks>	<p>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</p>

Name	Value	Description
		IF_PARM_NUMB=3; the IF_PARM1, IF_PARM2, IF_PARM3 tags are checked in addition to the IF_PARM tag.
CLOSE_BY_LIM_CODE	<codes of limits that must be closed>	<p>The tag is used to set up automatic "closing" of individual limits. See the section "Configuring Instalment Limits" of the document "Instalment Loans in WAY4".</p> <p>This tag is used when working with the WAY4 Instalments module. The module is delivered according to a separate agreement with the WAY4 vendor.</p>
CLOSE_INST_LIMITS		<p>The tag is used to set up automatic "closing" of individual limits. See the section "Configuring Instalment Limits" of the document "Instalment Loans in WAY4".</p> <p>This tag is used when working with the WAY4 Instalments module. The module is delivered according to a separate agreement with the WAY4 vendor.</p>
SKIP_ON_SECONDARY_DOC		<p>The tag makes it possible to open an Event only when the original document is posted and not open an Event when a secondary document is posted: when reposting a document, posting an adjustment or reversal. Used for example, in an Event type set in a usage limiter or in an Event type opened when a balance changes. Note that the Event will be skipped only if it is directly linked to the document (for example, posting the document caused the balance to change). In this case, there will be a link to the document in the usage_action table.</p> <p>An Event cannot be skipped if there is a chain of actions that led to the Event opening for the document. For example, if posting the original document caused a change in a balance for which a classifier value was set, and setting the classifier, led to an Event opening. In this case, when a secondary document is posted, the Event will be reopened, regardless of the tag, since when the classifier was set, information that this was for a document is not registered.</p>
DEPENDENT_CHAIN		<p>If an Event from a chain depends on the main Event or previous linked Events in a chain, set the DEPENDENT_CHAIN tag in the <i>Special Parameters</i> field of the Event depending on which other Events are loaded. In this case, the main Event is opened and processed first, and then Events from the chain are loaded one-by-one and processed (in</p>

Name	Value	Description
		<p>the order defined by the <i>Step N</i> field or in the order in which they are specified in the chain, if the field is not filled in).</p> <p>If the tag is not set, a dependent Event cannot be processed, since at the loading stage there are still no results for processing the main Event and previous Events in the chain and therefore the condition for opening the dependent Event cannot be met (the dependent Event will be declined).</p> <p>See the section "Generating Event Chains" of the document "Events".</p>
DAY	<Billing Day parameter value>	The tag sets the Billing Day parameter value that is used when calculating Billing Date
STORNO_FEE_ACTION	SAVE REAPPLY	<p>The tag affects rules for charging a fee according to an Event when correcting transactions for a contract. The tag is used in the "Reversal Management" module. The module requires an additional license.</p> <p>Tag values:</p> <ul style="list-style-type: none"> • "SAVE" – if an Event was not open before the correction plan was applied, and is opened at the same time the plan is applied due to a change in the contract's state, fee charging can be switched off by setting the STORNO_FEE_ACTION=SAVE; tag in the Event type. • "REAPPLY" – the STORNO_FEE_ACTION=REAPPLY; tag in the Event type makes it possible to save fee parameters when applying a correction plan: <ul style="list-style-type: none"> ♦ If an Event was open before the correction plan was applied and it was used to charge a fee, the fee will be saved when transactions are corrected, even if the Event will not be reopened (due to changes in the contract's state). The fee is not recalculated, and is applied without changes. ♦ if an Event was not open before the correction plan was applied (and, accordingly, no fee was charged according to it) and an Event opens when the correction plan is applied, a fee will not be charged.
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"	<p>IF_DATE group tags make it possible to configure actions (in this case, opening and posting an Event) 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.</p> <p>The IF_DATE tag sets the code of the date being checked, or the code of the date being compared with another date.</p>

Name	Value	Description
	"CONTRACT.LAST_BILLING" "CONTRACT.NEXT_BILLING" "CONTRACT.DATE_EXPIRE" "CONTRACT.CARD_EXPIRE"	<p>Date codes can be contract/client dates specified in the following contract table (CONTRACT) or client table (CLIENT) fields:</p> <ul style="list-style-type: none"> • "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) • "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 <p>The tag is used together with the IF_DATE_VALUE and IF_DATE_FOR tags.</p> <p>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.</p>
IF_DATE_VALUE	"EMPTY" "NOT_EMPTY" <Rule><Base Date Code><Period Rule>	<p>The IF_DATE_VALUE tag sets rules for checking the date specified with the IF_DATE tag.</p> <p>Tag values:</p> <ul style="list-style-type: none"> • "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

Name	Value	Description
		<p>the IF_DATE field is filled in. The field value is not checked.</p> <ul style="list-style-type: none"> 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: <ul style="list-style-type: none"> <Rule> – comparison rule. Possible values: <ul style="list-style-type: none"> ♦ "<" – 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). <Base Date Code> – code of the date with which the date from the IF_DATE tag is compared. Possible values: <ul style="list-style-type: none"> ♦ 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 <ul style="list-style-type: none"> ♦ <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>. <p>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.</p> <p>Examples of checks: Check of whether a client is already 40 years old:</p>

Name	Value	Description
		<p>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;</p>
IF_DATE_FOR	"BILLING" "LIABILITY" "TOP" "BASE" "DOC_SOURCE" "DOC_TARGET" "FROM_DOC" "CONTRACT_ROLE" CONTRACT_ROLE_PARM=<tag name> "LIAB_CATEGORY"	<p>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:</p> <ul style="list-style-type: none"> • "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, 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

Name	Value	Description
		<p>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.</p> <ul style="list-style-type: none"> 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 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.

Name	Value	Description
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.
Tags in the <i>Custom Event Code</i> field of an Event type		
ACNT_TAG	<tag/contract parameter name>	When an Event opens, a contract can be assigned the specific value of a specific tag. To do so, the ACNT_TAG tag is used to set an arbitrary tag. The tag's value is set in the same field in the format <tag name>=<tag value>. The tag and its value are recorded in the contract's ext_data field when the Event opens. When this Event closes, the tag is removed from the contract's ext_data field. If a contract parameter with the "Classifier" value of the <i>Mirror To</i> field corresponds to the tag, the parameter is set according to rules configured in the "Contract Classifier" form. When the event is Closed, this parameter is reset (or restored to the previous state set, for example, in the Product).
Tags in the <i>Chain Details</i> field of "Event Chain" form:		
UNLINKED		When opening an Event in a chain, allows the Event to be "detached" from the main Event. When the "UNLINKED" tag is specified, further actions with the main Event will not influence the current Event; in particular, closing the main Event does not lead to closing the current Event.
REL_TYPE	<relation type code>	This tag is used if the "Related Contract" value is selected in the <i>For Contract</i> field. The tag allows an Event to be opened for all contracts related to this contract with the relation type set with the REL_TYPE tag (the relation type code is taken from the form "Full → Configuration Setup → Accounting Setup → Contract Relations")
INHERIT_EVENT_DETAILS		When an Event included in an Event Chain opens, allows the parameters of the main Event's <i>Special Params</i> field to be inherited. To do so, select the main Event in the "Issuing Event Types" form, click

Name	Value	Description
		the [Event Chain] button and set this tag in the <i>Chain Details</i> field of those Events that will inherit the <i>Special Params</i> field parameters of the main Event.
IF_CS_TYPE	<CS_STATUS_TYPE.CODE>	<p>This tag sets the type of checked classifier. 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".</p> <p>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.</p>
IF_CS_VALUE	<CS_STATUS_VALUE.CODE 1>, <CS_STATUS_VALUE.CODE 2>,...,<CS_STATUS_VALUE.CODE N>	<p>An Event will be opened only if clients or their contracts correspond to the set classifier value. 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".</p> <p>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.</p>
IF_NOT_CS_VALUE	<CS_STATUS_VALUE.CODE 1>, <CS_STATUS_VALUE.CODE 2>,...,<CS_STATUS_VALUE.CODE N>	<p>An Event will not be opened if the value specified in the tag corresponds with the value of the classifier in the contract or client. 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".</p> <p>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.</p>
IF_INACTIVE_EVNT	<EventCode>	An Event will be opened only if the Event specified in the tag (with the specified code) is closed.

Name	Value	Description
IF_ACTIVE_EVNT	<EventCode>	An Event will be opened only if the Event specified in the tag (with the specified code) is opened.
IF_PARM	<tagged parameter name> CLIENT.SHORT_NAME CLIENT.FIRST_NAME CLIENT.LAST_NAME CLIENT.COMPANY_NAME CLIENT.AGE	<p>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".</p> <p>A number of client parameters can be checked. To do so, the following values can be specified as the tag's value:</p> <p>CLIENT.SHORT_NAME – check the <i>Short Name</i> field in the client form.</p> <p>CLIENT.FIRST_NAME – check the <i>First Name</i> field in the client form.</p> <p>CLIENT.LAST_NAME – check the <i>Last Name</i> field in the client form.</p> <p>CLIENT.COMPANY_NAME – check the <i>Company Name</i> field in the client form (place of work)</p> <p>CLIENT.AGE – client age. Determined according to date of birth (<i>Date of Birth</i> field) and the system date (i.e. age at the time of the request is determined).</p> <p>These client attributes do not have to be registered as contract custom parameters.</p> <p>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.</p>
IF_PARM_VALUE	"<tagged parameter value>" "NOT_EMPTY" "EMPTY" "LIST_WITH:<list of values separated by commas>" "BETWEEN<value> AND <value>"	<p>An Event is opened and processed (actions specified when setting up the corresponding Event type are executed) 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.</p> <p>When the IF_PARM_VALUE=EMPTY; tag is set, the Event opens if the specified parameter doesn't have a value. Note that with this value, the Event opens if the parameter is not present in the contract.</p> <p>When "IF_PARM_VALUE=LIST_WITH:<list of values separated by commas>" is set, the Event opens if even one value from the list (LIST_WITH) matches that set in the contract.</p> <p>When "IF_PARM_VALUE=BETWEEN<value> AND <value>" the</p>

Name	Value	Description
		<p>Event opens 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.</p> <p>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.</p>
ZERO_BAL_TYPE	<balance type code>	An Event will only open 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.
NON_ZERO_BAL_TYPE	<balance type code>	An Event will only open 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.
IF_CURRENCY	<balance type code>	<p>This tag sets the document currency that must be checked.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • 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. <p>A contract's custom parameter can be registered as a "Primary" classifier.</p> <ul style="list-style-type: none"> • 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. <p>Used together with the IF_CURRENCY_VALUE, IF_CURRENCY_RULE, IF_CURRENCY_FOR tags.</p>
IF_CURRENCY_VALUE	<balance type code>	<p>An Event is only opened and posted (actions specified when setting up the corresponding Event type) if the document currency specified with the IF_CURRENCY tag corresponds to parameters set in the IF_CURRENCY_VALUE tag. Possible values:</p> <ul style="list-style-type: none"> • EMPTY – this currency is not specified. • NOT_EMPTY – this currency is specified.

Name	Value	Description
		<ul style="list-style-type: none"> • USE_IN_BANK – this currency has the <i>Use in Bank</i> 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 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	<balance type code>	<p>To set additional conditions for checking a currency, use the IF_CURRENCY_VALUE tag together with the IF_CURRENCY_RULE tag:</p> <ul style="list-style-type: none"> • 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 accounts 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.
IF_PARM_FOR	<balance type code>	<p>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.</p> <ul style="list-style-type: none"> • "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.

Name	Value	Description
		<ul style="list-style-type: none"> • "DOC_SOURCE" – for the contract specified in the document's <i>Source Contract</i> (source_contract) field (see the "Doc-Brief" form). • "DOC_TARGET" – for the contract specified in the document's <i>Target Contract</i> (target_contract) field (see the "Doc-Brief" form). • "FROM_DOC" – the contract is taken from the document's <i>Add Data</i> (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: 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

Name	Value	Description
		<p>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.</p> <p>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.</p>
IF_CS_TYPE_FOR	<balance type code>	<p>The IF_CS_TYPE_FOR tag is used together with IF_CS group tags to redefine the contract for which these checks are made.</p> <ul style="list-style-type: none"> • "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. • "DOC_SOURCE" – for the contract specified in the document's <i>Source Contract</i> (source_contract) field (see the "Doc-Brief" form). • "DOC_TARGET" – for the contract specified in the document's <i>Target Contract</i> (target_contract) field (see the "Doc-Brief" form). • "FROM_DOC" – the contract is taken from the <i>Add Data</i> (add_info) field. • "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

Name	Value	Description
		<p>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</p> <ul style="list-style-type: none"> • "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. <p>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.</p> <p>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 "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.</p> <p>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.</p>
IF_CURRENCY_FOR	<balance type code>	<p>Redefines the contract for which a check is made using the IF_CURRENCY tag. Possible values:</p> <ul style="list-style-type: none"> • "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.

Name	Value	Description
		<ul style="list-style-type: none"> • "DOC_SOURCE" – for the contract in the <i>Source Contract</i> field of the document (source_contract) of the document (see the "Doc-Brief" form). • "DOC_TARGET" – for the contract in the <i>Target Contract</i> field (target_contract) of the document (see the "Doc-Brief" form). • "FROM_SERVICE" – for the contract specified in the Service (fee_contract, fee_account fields). • "FROM_DOC" – the contract is taken from the document's <i>Add Data</i> field (add_info) according to the tag specified using the CONTRACT_TAG tag in the same field) • "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
		<p>"Y" – "Full Liability" category "N" – "Affiliated" category "R" – "Reporting" category "A" – "Only Check Balance" category.</p> <p>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.</p>
ZERO_BAL_TYPE_FOR	<balance type code>	<p>Redefines the contract for which a check is made using the ZERO_BAL_TYPE tag. Possible values:</p> <ul style="list-style-type: none"> • "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 in the <i>Source Contract</i> field of the document (source_contract) of the document (see the "Doc-Brief" form). • "DOC_TARGET" – for the contract in the <i>Target Contract</i> field (target_contract) of the document (see the "Doc-Brief" form). • "FROM_SERVICE" – for the contract specified in the Service (fee_contract, fee_account fields). • "FROM_DOC" – the contract is taken from the document's <i>Add Data</i> field (add_info) according to the tag specified using the CONTRACT_TAG tag in the same field) • "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: ZERO_BAL_TYPE_FOR=CONTRACT_ROLE;CONTRACT_ROLE=PAYMENT_LEVEL;. In this case, a search will be made for a contract

Name	Value	Description
		<p>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.</p> <ul style="list-style-type: none"> • 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 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	<balance type code>	Redefines the contract for which a check is made using the NON_ZERO_BAL_TYPE tag. Possible values:

Name	Value	Description
		<ul style="list-style-type: none"> • "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 in the <i>Source Contract</i> field of the document (source_contract) of the document (see the "Doc-Brief" form). • "DOC_TARGET" – for the contract in the <i>Target Contract</i> field (target_contract) of the document (see the "Doc-Brief" form). • "FROM_SERVICE" – for the contract specified in the Service (fee_contract, fee_account fields). • "FROM_DOC" – the contract is taken from the document's <i>Add Data</i> field (add_info) according to the tag specified using the CONTRACT_TAG tag in the same field) • "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: 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 • "LIAB_CATEGORY" – in this case, a search for a contract will be

Name	Value	Description
		<p>made upward in a Liability hierarchy within the category/categories set in this field using the LIAB_CATEGORY=<category1>,<category2> tag.</p> <p>Liability category codes, separated by commas, are specified as the LIAB_CATEGORY tag value:</p> <p>"Y" – "Full Liability" category</p> <p>"N" – "Affiliated" category</p> <p>"R" – "Reporting" category</p> <p>"A" – "Only Check Balance" category.</p> <p>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.</p>
IF_CS_NUMB	<balance type code>	<p>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 are 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.</p>
IF_PARM_NUMB	<balance type code>	<p>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.</p>
IF_DATE	<balance type code>	<p>IF_DATE group tags make it possible to configure actions (in this case, opening and posting an Event) depending on contract/client dates.</p>

Name	Value	Description
		<p>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.</p> <p>The IF_DATE tag sets the code of the date being checked, or the code of the date being compared with another date.</p> <p>Date codes can be contract/client dates specified in the following contract table (CONTRACT) or client table (CLIENT) fields:</p> <ul style="list-style-type: none"> • "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) • "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 <p>The tag is used together with the IF_DATE_VALUE and IF_DATE_FOR tags.</p> <p>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.</p>

Name	Value	Description
IF_DATE_VALUE	<balance type code>	<p>The IF_DATE_VALUE tag sets rules for checking the date specified with the IF_DATE tag.</p> <p>Tag values:</p> <ul style="list-style-type: none"> • "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: <ul style="list-style-type: none"> ▪ <Rule> – comparison rule. Possible values: <ul style="list-style-type: none"> ♦ "<" – 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). ▪ <Base Date Code> – code of the date with which the date from the IF_DATE tag is compared. Possible values: <ul style="list-style-type: none"> ♦ 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 <ul style="list-style-type: none"> ♦ <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>. <p>The tag is used together with the IF_DATE and IF_DATE_FOR tags.</p>

Name	Value	Description
		<p>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.</p> <p>Examples of checks:</p> <p>Check of whether a client is already 40 years old: IF_DATE=CLIENT.BIRTH_DATE;IF_DATE_VALUE=<=DB_DATE-40Y;</p> <p>Check that a card has not expired: IF_DATE=CONTRACT.CARDS_EXPIRE;IF_DATE_VALUE=>=DB_DATE;</p> <p>Check that a card expires after a month: IF_DATE=CONTRACT.CARD_EXPIRE;IF_DATE_VALUE=<=DB_DATE+1M;</p>
IF_DATE_FOR	<balance type code>	<p>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.</p> <p>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.</p> <p>Possible values for the IF_DATE_FOR tag:</p> <ul style="list-style-type: none"> • "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, 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)

Name	Value	Description
		<ul style="list-style-type: none"> • "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 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")

Name	Value	Description
		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	<balance type code>	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.
Tags in the <i>Parameters</i> field of the "Events for <name of Accounting Scheme>" form:		
CALC_INT	<account code 1>, <account code 2>, ..., <account code N>	When this Event (registered for given account scheme) opens, additional interest is accrued on accounts with specified codes.
IF_CS_TYPE	<CS_STATUS_TYPE.CODE>	This tag sets the type of checked classifier. 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>	An Event will be opened only if clients or their contracts correspond to the set classifier value. 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.

Name	Value	Description
IF_NOT_CS_VALUE	<CS_STATUS_VALUE.CODE 1>, <CS_STATUS_VALUE.CODE 2>,...,<CS_STATUS_VALUE.CODE N>	An Event will not be opened if the value specified in the tag corresponds with the value of the classifier in the contract or client. 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_INACTIVE_EVNT	<EventCode>	An Event will be opened only if the Event specified in the tag (with the specified code) is closed.
IF_ACTIVE_EVNT	<EventCode>	An Event will be opened only if the Event specified in the tag (with the specified code) is opened.
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 <i>Short Name</i> field in the client form. CLIENT.FIRST_NAME – check the <i>First Name</i> field in the client form. CLIENT.LAST_NAME – check the <i>Last Name</i> field in the client form. CLIENT.COMPANY_NAME – check the <i>Company Name</i> field in the client form (place of work) CLIENT.AGE – client age. Determined according to date of birth (<i>Date of Birth</i> 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>,

Name	Value	Description
		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>"	<p>An Event is opened and processed (actions specified when setting up the corresponding Event type are executed) 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.</p> <p>When the IF_PARM_VALUE=EMPTY; tag is set, the Event opens if the specified parameter doesn't have a value. Note that with this value, the Event opens if the parameter is not present in the contract.</p> <p>When "IF_PARM_VALUE=LIST_WITH:<list of values separated by commas>" is set, the Event opens if even one value from the list (LIST_WITH) matches that set in the contract.</p> <p>When "IF_PARM_VALUE=BETWEEN<value> AND <value>" the Event opens 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.</p> <p>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.</p>
ZERO_BAL_TYPE	<balance type code>	An Event will only open 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.
NON_ZERO_BAL_TYPE	<balance type code>	An Event will only open 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.
IF_CURRENCY	<balance type code>	<p>This tag sets the document currency that must be checked.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • 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).

Name	Value	Description
		<ul style="list-style-type: none"> • 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. <p>Used together with the IF_CURRENCY_VALUE, IF_CURRENCY_RULE, IF_CURRENCY_FOR tags.</p>
IF_CURRENCY_VALUE	<balance type code>	<p>An Event is only opened and posted (actions specified when setting up the corresponding Event type) if the document currency specified with the IF_CURRENCY tag corresponds to parameters set in the IF_CURRENCY_VALUE tag. Possible values:</p> <ul style="list-style-type: none"> • 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 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	<balance type code>	<p>To set additional conditions for checking a currency, use the IF_CURRENCY_VALUE tag together with the IF_CURRENCY_RULE tag:</p> <ul style="list-style-type: none"> • 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 accounts 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

Name	Value	Description
		<p>IF_CURRENCY_RULE=NOT_EQUAL; the value of IF_CURRENCY_VALUE is also set as the value of the IF_CURRENCY tag.</p> <ul style="list-style-type: none"> • 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.
IF_PARM_FOR	<balance type code>	<p>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.</p> <ul style="list-style-type: none"> • "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 <i>Source Contract</i> (source_contract) field (see the "Doc-Brief" form). • "DOC_TARGET" – for the contract specified in the document's <i>Target Contract</i> (target_contract) field (see the "Doc-Brief" form). • "FROM_DOC" – the contract is taken from the document's <i>Add Data</i> (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: 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;

Name	Value	Description
		<p>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</p> <ul style="list-style-type: none"> • "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. <p>Liability category codes, separated by commas, are specified as the LIAB_CATEGORY tag value:</p> <p>"Y" – "Full Liability" category "N" – "Affiliated" category "R" – "Reporting" category "A" – "Only Check Balance" category.</p> <p>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.</p> <p>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.</p>
IF_CS_TYPE_FOR	<balance type code>	<p>The IF_CS_TYPE_FOR tag is used together with IF_CS group tags to redefine the contract for which these checks are made.</p> <ul style="list-style-type: none"> • "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. • "DOC_SOURCE" – for the contract specified in the document's <i>Source Contract</i> (source_contract) field (see the "Doc-Brief" form). • "DOC_TARGET" – for the contract specified in the document's <i>Target</i>

Name	Value	Description
		<p><i>Contract</i> (target_contract) field (see the "Doc-Brief" form).</p> <ul style="list-style-type: none"> • "FROM_DOC" – the contract is taken from the <i>Add Data</i> (add_info) field). • "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

Name	Value	Description
		<p>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.</p> <p>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.</p>
IF_CURRENCY_FOR	<balance type code>	<p>Redefines the contract for which a check is made using the IF_CURRENCY tag. Possible values:</p> <ul style="list-style-type: none"> • "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 in the <i>Source Contract</i> field of the document (source_contract) of the document (see the "Doc-Brief" form). • "DOC_TARGET" – for the contract in the <i>Target Contract</i> field (target_contract) of the document (see the "Doc-Brief" form). • "FROM_SERVICE" – for the contract specified in the Service (fee_contract, fee_account fields). • "FROM_DOC" – the contract is taken from the document's <i>Add Data</i> field (add_info) according to the tag specified using the CONTRACT_TAG tag in the same field) • "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.

Name	Value	Description
		<ul style="list-style-type: none"> 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_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.
ZERO_BAL_TYPE_FOR	<balance type code>	<p>Redefines the contract for which a check is made using the ZERO_BAL_TYPE tag. Possible values:</p> <ul style="list-style-type: none"> "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 in the <i>Source Contract</i> field of the

Name	Value	Description
		<p>document (source_contract) of the document (see the "Doc-Brief" form).</p> <ul style="list-style-type: none"> • "DOC_TARGET" – for the contract in the <i>Target Contract</i> field (target_contract) of the document (see the "Doc-Brief" form). • "FROM_SERVICE" – for the contract specified in the Service (fee_contract, fee_account fields). • "FROM_DOC" – the contract is taken from the document's <i>Add Data</i> field (add_info) according to the tag specified using the CONTRACT_TAG tag in the same field) • "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: 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: "Y" – "Full Liability" category

Name	Value	Description
		<p>"N" – "Affiliated" category "R" – "Reporting" category "A" – "Only Check Balance" category.</p> <p>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.</p>
NON_ZERO_BAL_TYPE_FOR	<balance type code>	<p>Redefines the contract for which a check is made using the NON_ZERO_BAL_TYPE tag. Possible values:</p> <ul style="list-style-type: none"> • "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 in the <i>Source Contract</i> field of the document (source_contract) of the document (see the "Doc-Brief" form). • "DOC_TARGET" – for the contract in the <i>Target Contract</i> field (target_contract) of the document (see the "Doc-Brief" form). • "FROM_SERVICE" – for the contract specified in the Service (fee_contract, fee_account fields). • "FROM_DOC" – the contract is taken from the document's <i>Add Data</i> field (add_info) according to the tag specified using the CONTRACT_TAG tag in the same field) • "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: 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

Name	Value	Description
		<p>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.</p> <ul style="list-style-type: none"> 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. <p>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.</p> <p>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.</p>
IF_CS_NUMB	<balance type code>	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

Name	Value	Description
		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	<balance type code>	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.
IF_DATE	<balance type code>	<p>IF_DATE group tags make it possible to configure actions (in this case, opening and posting an Event) 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.</p> <p>The IF_DATE tag sets the code of the date being checked, or the code of the date being compared with another date.</p> <p>Date codes can be contract/client dates specified in the following contract table (CONTRACT) or client table (CLIENT) fields:</p> <ul style="list-style-type: none"> • "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) • "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

Name	Value	Description
		<p>LAST_BILLING field of the contract's record</p> <ul style="list-style-type: none"> • "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 <p>The tag is used together with the IF_DATE_VALUE and IF_DATE_FOR tags.</p> <p>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.</p>
IF_DATE_VALUE	<balance type code>	<p>The IF_DATE_VALUE tag sets rules for checking the date specified with the IF_DATE tag.</p> <p>Tag values:</p> <ul style="list-style-type: none"> • "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: <ul style="list-style-type: none"> ▪ <Rule> – comparison rule. Possible values: <ul style="list-style-type: none"> ♦ "<" – 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). ▪ <Base Date Code> – code of the date with which the date from the IF_DATE tag is compared. Possible values: <ul style="list-style-type: none"> ♦ 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

Name	Value	Description
		<p>without time)</p> <ul style="list-style-type: none"> ▪ <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 <ul style="list-style-type: none"> ♦ <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>. <p>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.</p> <p>Examples of checks:</p> <p>Check of whether a client is already 40 years old: IF_DATE=CLIENT.BIRTH_DATE;IF_DATE_VALUE<=DB_DATE-40Y;</p> <p>Check that a card has not expired: IF_DATE=CONTRACT.CARDS_EXPIRE;IF_DATE_VALUE<=>DB_DATE;</p> <p>Check that a card expires after a month: IF_DATE=CONTRACT.CARD_EXPIRE;IF_DATE_VALUE<=DB_DATE+1M;</p>
IF_DATE_FOR	<balance type code>	<p>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.</p> <p>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.</p> <p>Possible values for the IF_DATE_FOR tag:</p> <ul style="list-style-type: none"> • "BILLING" – the check is made for the account contract used for settlement. • "LIABILITY" – the check is made for the higher ranking contract in the

Name	Value	Description
		<p>Liability hierarchy.</p> <ul style="list-style-type: none"> • "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, 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. <p>Liability category codes, separated by commas, are specified as th LIAB_CATEGORY tag value:</p>

Name	Value	Description
		<p>"Y" – "Full Liability" category "N" – "Affiliated" category "R" – "Reporting" category "A" – "Only Check Balance" category.</p> <p>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.</p> <p>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.</p>
IF_DATE_NUMB	<balance type code>	<p>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.</p>
Tags in the <i>Parameters</i> field of the "Events for <name of Service Package>" form:		
RISK_FACTOR	from "0" to "9"	When this Event is opened the risk factor specified in the RISK_FACTOR=<value>; tag will be changed (or set) in the RISK_FACTOR field in a contract.
IF_INACTIVE_EVNT	<EventCode>	An Event will be opened only if the Event specified in the tag (with the specified code) is closed.
IF_ACTIVE_EVNT	<EventCode>	An Event will be opened only if the Event specified in the tag (with the specified code) is opened.
IF_CS_TYPE	<CS_STATUS_TYPE.CODE>	This tag sets the type of checked classifier. Used together with the IF_CS_VALUE, IF_NOT_CS_VALUE, IF_CS_TYPE_FOR tags. For

Name	Value	Description
		<p>more information, see the section "Executing Actions Depending on Classifier Values" of the document "WAY4™ Client and Contract Classifiers".</p> <p>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.</p>
IF_CS_VALUE	<CS_STATUS_VALUE.CODE 1>, <CS_STATUS_VALUE.CODE 2>,...,<CS_STATUS_VALUE.CODE N>	<p>An Event will be opened only if clients or their contracts correspond to the set classifier value. 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".</p> <p>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.</p>
IF_NOT_CS_VALUE	<CS_STATUS_VALUE.CODE 1>, <CS_STATUS_VALUE.CODE 2>,...,<CS_STATUS_VALUE.CODE N>	<p>An Event will not be opened if the value specified in the tag corresponds with the value of the classifier in the contract or client. 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".</p> <p>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.</p>
IF_PARM	<tagged parameter name> CLIENT.SHORT_NAME CLIENT.FIRST_NAME CLIENT.LAST_NAME CLIENT.COMPANY_NAME CLIENT.AGE	<p>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".</p> <p>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 <i>Short Name</i> field in the client form.</p>

Name	Value	Description
		<p>CLIENT.FIRST_NAME – check the <i>First Name</i> field in the client form.</p> <p>CLIENT.LAST_NAME – check the <i>Last Name</i> field in the client form.</p> <p>CLIENT.COMPANY_NAME – check the <i>Company Name</i> field in the client form (place of work)</p> <p>CLIENT.AGE – client age. Determined according to date of birth (<i>Date of Birth</i> field) and the system date (i.e. age at the time of the request is determined).</p> <p>These client attributes do not have to be registered as contract custom parameters.</p> <p>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.</p>
IF_PARM_VALUE	"<tagged parameter value>" "NOT_EMPTY" "EMPTY" "LIST_WITH:<list of values separated by commas>" "BETWEEN<value> AND <value>"	<p>An Event is opened and processed (actions specified when setting up the corresponding Event type are executed) 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.</p> <p>When the IF_PARM_VALUE=EMPTY; tag is set, the Event opens if the specified parameter doesn't have a value. Note that with this value, the Event opens if the parameter is not present in the contract.</p> <p>When "IF_PARM_VALUE=LIST_WITH:<list of values separated by commas>" is set, the Event opens if even one value from the list (LIST_WITH) matches that set in the contract.</p> <p>When "IF_PARM_VALUE=BETWEEN<value> AND <value>" the Event opens 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.</p> <p>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.</p>
ZERO_BAL_TYPE	<balance type code>	<p>An Event will only open 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.</p>

Name	Value	Description
NON_ZERO_BAL_TYPE	<balance type code>	An Event will only open 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.
IF_CURRENCY	<balance type code>	<p>This tag sets the document currency that must be checked.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • 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. <p>A contract's custom parameter can be registered as a "Primary" classifier.</p> <ul style="list-style-type: none"> • 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. <p>Used together with the IF_CURRENCY_VALUE, IF_CURRENCY_RULE, IF_CURRENCY_FOR tags.</p>
IF_CURRENCY_VALUE	<balance type code>	<p>An Event is only opened and posted (actions specified when setting up the corresponding Event type) if the document currency specified with the IF_CURRENCY tag corresponds to parameters set in the IF_CURRENCY_VALUE tag. Possible values:</p> <ul style="list-style-type: none"> • 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 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.

Name	Value	Description
IF_CURRENCY_RULE	<balance type code>	<p>To set additional conditions for checking a currency, use the IF_CURRENCY_VALUE tag together with the IF_CURRENCY_RULE tag:</p> <ul style="list-style-type: none"> • 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 accounts 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.
IF_PARM_FOR	<balance type code>	<p>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.</p> <ul style="list-style-type: none"> • "BILLING" – for the account contract from which settlement is made. • "LIABILITY" – for a higher-ranking contract in a "Liability" hierarchy. • "TOP" – from 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 <i>Source Contract</i> (source_contract) field (see the "Doc-Brief" form). • "DOC_TARGET" – for the contract specified in the document's <i>Target Contract</i> (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). • "DOC_SOURCE" – for the contract specified in the document's <i>Source Contract</i> (source_contract) field (see the "Doc-Brief" form).

Name	Value	Description
		<ul style="list-style-type: none"> • "DOC_TARGET" – for the contract specified in the document's <i>Target Contract</i> (target_contract) field (see the "Doc-Brief" form). • "FROM_DOC" – the contract is taken from the document's <i>Add Data</i> (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: 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"

Name	Value	Description
		<p>("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.</p> <p>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.</p>
IF_CS_TYPE_FOR	<balance type code>	<p>The IF_CS_TYPE_FOR tag is used together with IF_CS group tags to redefine the contract for which these checks are made.</p> <ul style="list-style-type: none"> • "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. • "DOC_SOURCE" – for the contract specified in the document's <i>Source Contract</i> (source_contract) field (see the "Doc-Brief" form). • "DOC_TARGET" – for the contract specified in the document's <i>Target Contract</i> (target_contract) field (see the "Doc-Brief" form). • "FROM_DOC" – the contract is taken from the <i>Add Data</i> (add_info) field. • "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:

Name	Value	Description
		<p>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</p> <ul style="list-style-type: none"> • "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. <p>Liability category codes, separated by commas, are specified as the LIAB_CATEGORY tag value:</p> <p>"Y" – "Full Liability" category "N" – "Affiliated" category "R" – "Reporting" category "A" – "Only Check Balance" category.</p> <p>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 "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.</p> <p>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.</p>
IF_CURRENCY_FOR	<balance type code>	<p>Redefines the contract for which a check is made using the IF_CURRENCY tag. Possible values:</p> <ul style="list-style-type: none"> • "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 in the <i>Source Contract</i> field of the document (source_contract) of the document (see the "Doc-Brief"

Name	Value	Description
		<p>form).</p> <ul style="list-style-type: none"> • "DOC_TARGET" – for the contract in the <i>Target Contract</i> field (target_contract) of the document (see the "Doc-Brief" form). • "FROM_SERVICE" – for the contract specified in the Service (fee_contract, fee_account fields). • "FROM_DOC" – the contract is taken from the document's <i>Add Data</i> field (add_info) according to the tag specified using the CONTRACT_TAG tag in the same field) • "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: "Y" – "Full Liability" category "N" – "Affiliated" category

Name	Value	Description
		<p>"R" – "Reporting" category "A" – "Only Check Balance" category.</p> <p>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.</p>
ZERO_BAL_TYPE_FOR	<balance type code>	<p>Redefines the contract for which a check is made using the ZERO_BAL_TYPE tag. Possible values:</p> <ul style="list-style-type: none"> • "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 in the <i>Source Contract</i> field of the document (source_contract) of the document (see the "Doc-Brief" form). • "DOC_TARGET" – for the contract in the <i>Target Contract</i> field (target_contract) of the document (see the "Doc-Brief" form). • "FROM_SERVICE" – for the contract specified in the Service (fee_contract, fee_account fields). • "FROM_DOC" – the contract is taken from the document's <i>Add Data</i> field (add_info) according to the tag specified using the CONTRACT_TAG tag in the same field) • "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: 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

Name	Value	Description
		<p>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.</p> <ul style="list-style-type: none"> 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. <p>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.</p> <p>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.</p>
NON_ZERO_BAL_TYPE_FOR	<balance type code>	<p>Redefines the contract for which a check is made using the NON_ZERO_BAL_TYP tag. Possible values:</p> <ul style="list-style-type: none"> "BILLING" – for the account contract from which settlement is made. "LIABILITY" – for a higher-ranking contract in a "Liability" hierarchy.

Name	Value	Description
		<ul style="list-style-type: none"> • "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 in the <i>Source Contract</i> field of the document (source_contract) of the document (see the "Doc-Brief" form). • "DOC_TARGET" – for the contract in the <i>Target Contract</i> field (target_contract) of the document (see the "Doc-Brief" form). • "FROM_SERVICE" – for the contract specified in the Service (fee_contract, fee_account fields). • "FROM_DOC" – the contract is taken from the document's <i>Add Data</i> field (add_info) according to the tag specified using the CONTRACT_TAG tag in the same field) • "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: 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 • "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>

Name	Value	Description
		<p>tag.</p> <p>Liability category codes, separated by commas, are specified as the LIAB_CATEGORY tag value:</p> <p>"Y" – "Full Liability" category</p> <p>"N" – "Affiliated" category</p> <p>"R" – "Reporting" category</p> <p>"A" – "Only Check Balance" category.</p> <p>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.</p>
IF_CS_NUMB	<balance type code>	<p>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.</p>
IF_PARM_NUMB	<balance type code>	<p>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.</p>
IF_DATE	<balance type code>	<p>IF_DATE group tags make it possible to configure actions (in this case, opening and posting an Event) depending on contract/client dates. Contract/client dates can be compared with other dates (contract, client dates, system date, current banking date). Periods between</p>

Name	Value	Description
		<p>dates can be checked according to various conditions.</p> <p>The IF_DATE tag sets the code of the date being checked, or the code of the date being compared with another date.</p> <p>Date codes can be contract/client dates specified in the following contract table (CONTRACT) or client table (CLIENT) fields:</p> <ul style="list-style-type: none"> • "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) • "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 <p>The tag is used together with the IF_DATE_VALUE and IF_DATE_FOR tags.</p> <p>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.</p>
IF_DATE_VALUE	<balance type code>	<p>The IF_DATE_VALUE tag sets rules for checking the date specified with the IF_DATE tag.</p> <p>Tag values:</p>

Name	Value	Description
		<ul style="list-style-type: none"> • "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: <ul style="list-style-type: none"> ▪ <Rule> – comparison rule. Possible values: <ul style="list-style-type: none"> ♦ "<" – 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). ▪ <Base Date Code> – code of the date with which the date from the IF_DATE tag is compared. Possible values: <ul style="list-style-type: none"> ♦ 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 <ul style="list-style-type: none"> ♦ <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>. <p>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</p>

Name	Value	Description
		<p>tag.</p> <p>Examples of checks:</p> <p>Check of whether a client is already 40 years old: IF_DATE=CLIENT.BIRTH_DATE;IF_DATE_VALUE=<=DB_DATE-40Y;</p> <p>Check that a card has not expired: IF_DATE=CONTRACT.CARDS_EXPIRE;IF_DATE_VALUE=>=DB_DATE;</p> <p>Check that a card expires after a month: IF_DATE=CONTRACT.CARD_EXPIRE;IF_DATE_VALUE=<=DB_DATE+1M;</p>
IF_DATE_FOR	<balance type code>	<p>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.</p> <p>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.</p> <p>Possible values for the IF_DATE_FOR tag:</p> <ul style="list-style-type: none"> • "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, 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

Name	Value	Description
		<p>_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.</p> <ul style="list-style-type: none"> • 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 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. <p>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</p>

Name	Value	Description
		IF_DATE_FOR<N> tags. See the description of the IF_DATE_NUMB tag.
IF_DATE_NUMB	<balance type code>	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.
Tags in the <i>Custom Rules</i> field of the "Start Events for <name of Product>" form:		
PROD_CHANGE_RULE	"CLOSE" "OPEN" "OPEN,CHANGE_ONLY"	The parameter allows the state of an Event to be changed when the Product changes: <ul style="list-style-type: none"> • When the value is "CLOSE", when the Product changes, the Event in the contract closes (when it is still the old Product) • When the value is "OPEN" – this Event opens when the contract is created and when the Product changes (when the Product has already changed). • "OPEN,CHANGE_ONLY" - this Event only opens when the Product changes. The Event does not open when the Product opens. Default behaviour (if the tag is not set) – when the Product changes, an Event under the contract does not change its state. In this case, the Event only opens when the contract is created.
IF_CS_TYPE	<CS_STATUS_TYPE.CODE>	This tag sets the type of checked classifier. 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>	An Event will be opened only if clients or their contracts correspond to the set classifier value. 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

Name	Value	Description
		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>	An Event will not be opened if the value specified in the tag corresponds with the value of the classifier in the contract or client. 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_INACTIVE_EVNT	<EventCode>	An Event will be opened only if the Event specified in the tag (with the specified code) is closed.
IF_ACTIVE_EVNT	<EventCode>	An Event will be opened only if the Event specified in the tag (with the specified code) is opened.
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 <i>Short Name</i> field in the client form. CLIENT.FIRST_NAME – check the <i>First Name</i> field in the client form. CLIENT.LAST_NAME – check the <i>Last Name</i> field in the client form. CLIENT.COMPANY_NAME – check the <i>Company Name</i> 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).

Name	Value	Description
		<p>These client attributes do not have to be registered as contract custom parameters.</p> <p>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.</p>
IF_PARM_VALUE	"<tagged parameter value>" "NOT_EMPTY" "EMPTY" "LIST_WITH:<list of values separated by commas>" "BETWEEN<value> AND <value>"	<p>An Event is opened and processed (actions specified when setting up the corresponding Event type are executed) 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.</p> <p>When the IF_PARM_VALUE=EMPTY; tag is set, the Event opens if the specified parameter doesn't have a value. Note that with this value, the Event opens if the parameter is not present in the contract.</p> <p>When "IF_PARM_VALUE=LIST_WITH:<list of values separated by commas>" is set, the Event opens if even one value from the list (LIST_WITH) matches that set in the contract.</p> <p>When "IF_PARM_VALUE=BETWEEN<value> AND <value>" the Event opens 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.</p> <p>It is possible to set a check of several parameters for a contract or of one parameter for different contracts 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.</p>
ZERO_BAL_TYPE	<balance type code>	An Event will only open 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.
NON_ZERO_BAL_TYPE	<balance type code>	An Event will only open 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.
IF_CURRENCY	<tag name> TRANS_CURR SETTL_CURR RECONS_CURR	<p>This tag sets the document currency that must be checked.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • Document tag containing the currency code. • TRANS_CURR – transaction currency from the document.

Name	Value	Description
	CONTRACT <tag or contract custom parameter> <classifier code> <three-digit numeric currency code or currency name>	<ul style="list-style-type: none"> • 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. <p>A contract's custom parameter can be registered as a "Primary" classifier.</p> <ul style="list-style-type: none"> • 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. <p>Used together with the IF_CURRENCY_VALUE, IF_CURRENCY_RULE, IF_CURRENCY_FOR tags.</p>
IF_CURRENCY_VALUE	EMPTY NOT_EMPTY USE_IN_BANK <currency code1>, <currency code2>, ..., <currency codeN> <currency name1>, <currency name2>, ..., <currency nameN> <tag name>	<p>An Event is only opened and posted (actions specified when setting up the corresponding Event type) if the document currency specified with the IF_CURRENCY tag corresponds to parameters set in the IF_CURRENCY_VALUE tag. Possible values:</p> <ul style="list-style-type: none"> • EMPTY – this currency is not specified. • NOT_EMPTY – this currency is specified. • USE_IN_BANK – this currency has the <i>Use in Bank</i> 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 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	<p>To set additional conditions for checking a currency, use the IF_CURRENCY_VALUE tag together with the IF_CURRENCY_RULE tag:</p> <ul style="list-style-type: none"> • 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 accounts is not set, the currency

Name	Value	Description
		<p>is searched for among all Accounting Scheme account templates.</p> <ul style="list-style-type: none"> • 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.
IF_PARM_FOR	"BILLING" "LIABILITY" "TOP" "BASE" "DOC_SOURCE" "DOC_TARGET" "FROM_DOC" "CONTRACT_ROLE" CONTRACT_ROLE_PARM=<tag name> "LIAB_CATEGORY"	<p>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.</p> <ul style="list-style-type: none"> • "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 <i>Source Contract</i> (source_contract) field (see the "Doc-Brief" form). • "DOC_TARGET" – for the contract specified in the document's <i>Target Contract</i> (target_contract) field (see the "Doc-Brief" form). • "FROM_DOC" – the contract is taken from the document's <i>Add Data</i> (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: 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

Name	Value	Description
		<p>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</p> <ul style="list-style-type: none"> • "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. <p>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.</p> <p>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.</p> <p>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.</p>
IF_CS_TYPE_FOR	"BILLING" "LIABILITY" "TOP" "BASE" "DOC_SOURCE" "DOC_TARGET"	<p>The IF_CS_TYPE_FOR tag is used together with IF_CS group tags to redefine the contract for which these checks are made.</p> <ul style="list-style-type: none"> • "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

Name	Value	Description
	"FROM_DOC" "CONTRACT_ROLE" CONTRACT_ROLE_PARM=<tag name> "LIAB_CATEGORY"	<p>which this contract is related.</p> <ul style="list-style-type: none"> • "DOC_SOURCE" – for the contract specified in the document's <i>Source Contract</i> (source_contract) field (see the "Doc-Brief" form). • "DOC_TARGET" – for the contract specified in the document's <i>Target Contract</i> (target_contract) field (see the "Doc-Brief" form). • "FROM_DOC" – the contract is taken from the <i>Add Data</i> (add_info) field. • "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

Name	Value	Description
		<p>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.</p> <p>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.</p>
IF_CURRENCY_FOR	"BILLING" "LIABILITY" "TOP" "BASE" "DOC_SOURCE" "DOC_TARGET" "FROM_SERVICE" "FROM_DOC" "RELATED" "DOC_TARGET_NUMBER" "CONTRACT_ROLE" CONTRACT_ROLE_PARM=<tag name> "LIAB_CATEGORY"	<p>Redefines the contract for which a check is made using the IF_CURRENCY tag. Possible values:</p> <ul style="list-style-type: none"> • "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 in the <i>Source Contract</i> field of the document (source_contract) of the document (see the "Doc-Brief" form). • "DOC_TARGET" – for the contract in the <i>Target Contract</i> field (target_contract) of the document (see the "Doc-Brief" form). • "FROM_SERVICE" – for the contract specified in the Service (fee_contract, fee_account fields). • "FROM_DOC" – the contract is taken from the document's <i>Add Data</i> field (add_info) according to the tag specified using the CONTRACT_TAG tag in the same field) • "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

Name	Value	Description
		<p>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.</p> <ul style="list-style-type: none"> 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. <p>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.</p> <p>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.</p>
ZERO_BAL_TYPE_FOR	"BILLING" "LIABILITY" "TOP" "BASE"	<p>Redefines the contract for which a check is made using the ZERO_BAL_TYPE tag. Possible values:</p> <ul style="list-style-type: none"> "BILLING" – for the account contract from which settlement is made. "LIABILITY" – for a higher-ranking contract in a "Liability" hierarchy.

Name	Value	Description
	"DOC_SOURCE" "DOC_TARGET" "FROM_SERVICE" "FROM_DOC" "RELATED" "CONTRACT_ROLE" CONTRACT_ROLE_PARM=<tag name> "LIAB_CATEGORY"	<ul style="list-style-type: none"> • "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 in the <i>Source Contract</i> field of the document (source_contract) of the document (see the "Doc-Brief" form). • "DOC_TARGET" – for the contract in the <i>Target Contract</i> field (target_contract) of the document (see the "Doc-Brief" form). • "FROM_SERVICE" – for the contract specified in the Service (fee_contract, fee_account fields). • "FROM_DOC" – the contract is taken from the document's <i>Add Data</i> field (add_info) according to the tag specified using the CONTRACT_TAG tag in the same field) • "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: 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>

Name	Value	Description
		<p>tag.</p> <p>Liability category codes, separated by commas, are specified as the LIAB_CATEGORY tag value:</p> <p>"Y" – "Full Liability" category</p> <p>"N" – "Affiliated" category</p> <p>"R" – "Reporting" category</p> <p>"A" – "Only Check Balance" category.</p> <p>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.</p>
NON_ZERO_BAL_TYPE_FOR	"BILLING" "LIABILITY" "TOP" "BASE" "DOC_SOURCE" "DOC_TARGET" "FROM_SERVICE" "FROM_DOC" "RELATED" "CONTRACT_ROLE" CONTRACT_ROLE_PARM=<tag name> "LIAB_CATEGORY"	<p>Redefines the contract for which a check is made using the NON_ZERO_BAL_TYPE tag. Possible values:</p> <ul style="list-style-type: none"> • "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 in the <i>Source Contract</i> field of the document (source_contract) of the document (see the "Doc-Brief" form). • "DOC_TARGET" – for the contract in the <i>Target Contract</i> field (target_contract) of the document (see the "Doc-Brief" form). • "FROM_SERVICE" – for the contract specified in the Service (fee_contract, fee_account fields). • "FROM_DOC" – the contract is taken from the document's <i>Add Data</i> field (add_info) according to the tag specified using the CONTRACT_TAG tag in the same field) • "RELATED" – related contract, with the relation specified in the RELATION tag. • "CONTRACT_ROLE" – for this value, specify the

Name	Value	Description
		<p>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.</p> <ul style="list-style-type: none"> • 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. <p>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.</p> <p>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</p>

Name	Value	Description
		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 are 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.
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, opening and posting an Event) 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: <ul style="list-style-type: none"> • "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) • "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

Name	Value	Description
		<p>ADD_DATE_02 field of the client's record</p> <ul style="list-style-type: none"> • "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 <p>The tag is used together with the IF_DATE_VALUE and IF_DATE_FOR tags.</p> <p>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.</p>
IF_DATE_VALUE	"EMPTY" "NOT_EMPTY" <Rule><Base Date Code><Period Rule>	<p>The IF_DATE_VALUE tag sets rules for checking the date specified with the IF_DATE tag.</p> <p>Tag values:</p> <ul style="list-style-type: none"> • "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: <ul style="list-style-type: none"> ▪ <Rule> – comparison rule. Possible values: <ul style="list-style-type: none"> ♦ "<" – 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). ▪ <Base Date Code> – code of the date with which the date from the

Name	Value	Description
		<p>IF_DATE tag is compared. Possible values:</p> <ul style="list-style-type: none"> ◆ 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 <ul style="list-style-type: none"> ◆ <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>. <p>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.</p> <p>Examples of checks:</p> <p>Check of whether a client is already 40 years old: IF_DATE=CLIENT.BIRTH_DATE;IF_DATE_VALUE<=DB_DATE-40Y;</p> <p>Check that a card has not expired: IF_DATE=CONTRACT.CARDS_EXPIRE;IF_DATE_VALUE<=>DB_DATE;</p> <p>Check that a card expires after a month: IF_DATE=CONTRACT.CARD_EXPIRE;IF_DATE_VALUE<=DB_DATE+1M;</p>
IF_DATE_FOR	"BILLING" "LIABILITY" "TOP" "BASE"	<p>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.</p> <p>Note. If the IF_DATE tag is used to check a date from the client table,</p>

Name	Value	Description
	"DOC_SOURCE" "DOC_TARGET" "FROM_DOC" "CONTRACT_ROLE" CONTRACT_ROLE_PARM=<tag name> "LIAB_CATEGORY"	<p>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.</p> <p>Possible values for the IF_DATE_FOR tag:</p> <ul style="list-style-type: none"> • "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, 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 names>; 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
		<ul style="list-style-type: none"> • "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. <p>Liability category codes, separated by commas, are specified as the LIAB_CATEGORY tag value:</p> <p>"Y" – "Full Liability" category "N" – "Affiliated" category "R" – "Reporting" category "A" – "Only Check Balance" category.</p> <p>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 the "Full Liability" ("Y") and "Affiliated" ("N") categories.</p> <p>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.</p>
IF_DATE_NUMB	<number of additional checks>	<p>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.</p>