

# Configuration of Client Messages

# Contents

INTRODUCTION	1
CHAPTER 1. CLIENT MESSAGE TYPES	2
CHAPTER 2. CONFIGURING MESSAGE TEMPLATES	3
Template Configuration Form	3
Translating Messages into Local Languages	6
CHAPTER 3. USE OF VARIABLES	8
List of Variables	8
Characteristics of variable use	18
%ADDR...%	19
%ACCOUNT...%	19
%ACCOUNT_CAT...%	22
%ACCOUNT_TAG%	22
%ACC_TEMPL...%	22
%BEHAVIOR_TYPE%	23
%CLNT_ADD_INFO...%	23
%CONTRACT_BALANCE%	23
%USG_LIM...%	23
%PCNT%	24
%SUM%	24
%APPL_INFO...%	24
%PARTY...%	25
%TRANS_TAGS%	25
%TRANS_ACC...%	26
Use of Masks	26
Variable Prefixes	27
Variable Formatting	29
Shift in Date Variable Values	31
Using Modifiers for String Values of Variables	32
Using Variable Modifiers for Sending Messages in XML Format	32
Using Conditional Operators	33
CHAPTER 4. TAGS IN THE <i>MESSAGE DETAILS</i> FIELD OF A MESSAGE TEMPLATE	35



# Introduction

This document is intended for bank or processing centre employees responsible for configuring WAY4 and describes configuration of templates for creating client messages in WAY4.

When working with this document, it is recommended that users refer to the following resources from the WAY4 documentation series:

- "Events"
- "WAY4™ Accounting Schemes"
- "WAY4™ Service Packages"
- "WAY4™ Dictionaries"
- "WAY4 Client and Contract Classifiers"
- "Case Configuration" (the document is supplied according to an additional agreement with the WAY4 vendor).
- "Contract Functional Dates".

The document uses the following conventions:

- Field names in screen forms are displayed in *italics*.
- Names of screen form buttons are encased in square brackets, such as [Approve].
- Sequences for selecting user menu items are shown with arrows, as in "Configuration Setup" → Contract Types".
- Warnings about potentially hazardous situations or actions are marked with the  sign.
- Messages marked with the  sign contain information about important features, additional facilities, or the optimal use of certain system functions.

# Chapter 1. Client Message Types

WAY4 supports two kinds of messages:

- Messages generated when an Event opens (Event Messages).


One of the standard system actions when an Event opens and for the duration of the Event is the generation of template-based messages and their delivery in statements, reports, e-mail, SMS messages, etc.

Message templates can be configured using the subordinate form "Messages for <Event type name>" (see Fig. 1 in the section "Template Configuration Form"), opened by clicking the [Messages] button in one of the following forms:


- For issuing contracts, use the "Issuing Event Types" form (Full → Configuration Setup → Products → Issuing Private Products (Issuing Corporate Products) → Issuing Event Types).
  - For acquiring contracts, use the "Acquiring Event Types" form (Full → Configuration Setup → Products → Acquiring Products → Acquiring Event Types).
- Marketing and informational messages.
  - These messages are not linked to Events, and are linked to an Accounting Scheme, Service Package or Product. For example, a client statement may include a message with information on new bank services.

Marketing and informational messages can be configured in one of the following ways:

- Using the "Group Msg for <name of Accounting Scheme>", opened by clicking the [Group Msg] button in the form for configuring Accounting Schemes (see the section "Configuring Message Templates (Group Msg)" of the document "WAY4™ Accounting Schemes").
- Using the "Group Msg for <name of Service Package>", opened by clicking the [Group Msg] button in the form for configuring Service Packages (see the section "Configuring Message Templates (Group Msg)" of the document "WAY4™ Service Packages").
- Using the "Group Msg for <Product name>" form opened by clicking the [Group Msg] button in the form for configuring a Product (see the section "Configuring Message Templates (Group Msg)" of the document "Products and Contract Sub-Types").

 Templates for both message types are configured in the same way. For more information on how message templates are set up, see the section "Configuring Message Templates".

## Chapter 2. Configuring Message Templates

 The "Messages for..." and "Group Msg for..." forms are used to configure client message templates and contain the same set of fields and buttons. The configuration procedure is shown using the "Messages for <Event type name>" form as an example.

### Template Configuration Form

The "Messages for <Event type name>" form, used to configure message templates is shown in Fig. 1.

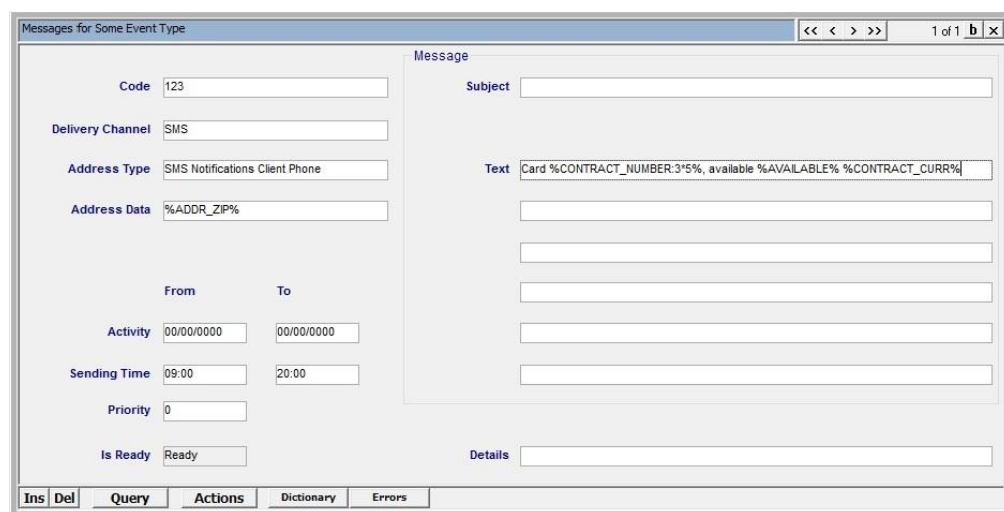


Fig. 1. Form for configuring the template of a message generated when an Event is opened.

The form contains the following fields:

- *Delivery Channel* – channel for sending messages to the client. The field can have one of the following values:
  - "To Statement" – for statements.
  - "To Ministatement" – for mini-statements.
  - "To Letter" – for exporting the message to a text file that will be sent by mail.
  - "SMS" – for sending an SMS message to a mobile phone.
  - "E-mail" – for sending a message by e-mail.
  - "Memo" – for saving messages in the record log kept for each contract by Customer Support specialists.
  - "To Auth Response" – to send an information message together with an authorization response. This information message is displayed on the ATM (kiosk) screen or printed on a receipt.
  - "To Device" – to show an informational message on the ATM screen or print on a receipt.

- "Display Card" – to show an informational message on the card screen.
- "None" – other messages.
- *Address Type* – the type of address to which the message will be sent. As a result of a card transaction, it is possible to create several messages to be sent to different addresses, and their text may vary. For example, as a result of a retail card transaction, the message to one mobile phone may contain information only about the transaction amount, to another mobile phone, about the amount and the merchant on whose device the transaction was executed, and a message to an e-mail address may provide a more detailed description of the transaction. New address types are entered and already existing ones are edited in the "Address Types" grid form (Full → Configuration Setup → Client Classifiers → Address Types).



Note that a message will only be generated if the given address type is specified for a client's contract (see the section "Client and Contract Address Support" in the document "Issuing Module User Manual").

- *Code* – the code that is used to configure additional parameters of messages. For example, to create a marketing message and add it to the client's statement, the code specified in this field should be set as the "P\_FINMESSAGECODE" parameter value of the "Cardholder\_Contract\_Statement.RDF" statement (see the section "Monthly Contract Statements" of the document "Cardholder Statements User Manual").



It is recommended to set unique codes for message templates in one object (for example, a Service Package or Accounting Scheme).




The template of a marketing message with the same value of the *Delivery Channel* field can be set in different levels of the Product (in the Product, Service Package, Accounting Scheme). In this case, the template for generating a message is determined according to the template's *Code* field (the template with the "smallest" code will be selected).


- *Address Data* – address template for message sending. The value is a variable that is replaced in the message with the corresponding information from the client address. For the following address types, the template is specified by default when approving the corresponding object (Service Package, Accounting Scheme, Product), or as the result of checking parameters when registering an Event type:
  - For the "SMS" address type, the variable %ADDR\_ZIP% is specified. In the message, this variable is filled in with data from the *ZIP* field of the address.
  - For the "E-mail" address type with the variable %E-MAIL%. In the message, this variable is filled in with e-mail data from the *E-mail* field of the address.
  - For the "Letter" address type, a check is made of whether the *Address Data* field is filled in. If the field is not filled in, a warning message is generated, the status of the message template and

corresponding object (Service Package, Accounting Scheme, Product) remains "Not Ready".

If the *Address Data* field of the message is not filled in (if no address of this type is found for the client), no message will be generated (stored in the *evnt\_msg* table) and will not be sent to the client. For a message to be generated in this case, specify the tag *SKIP\_ADDR\_CHCK*; in the *Details* field of the message template. This setting is used to send messages for a contract to linked clients.


- *Active From, Active To* – field used to limit the period during which the client will receive this message. The format of both fields is: "<dd>/<mm>/<yyyy>", where <dd> is day, <mm> is month, <yyyy> is year.
- *Sending Time From, Sending Time To* – fields specifying the hours during which the client will receive this message. The format of both fields is: "<hh>:<mm>", where <hh> is hours, <mm> is minutes.

 In general, a message will be sent to a client only when an Event opens in the specified time period. If an Event opens at a time not falling in the interval between *Sending Time From* and *Sending Time To*, no message will be generated. For a message to be generated by an Event that opens outside the specified time interval and sent to the client, specify the *USE\_DELAY\_TIME* tag in the *Message Details* field.

 When calculating the time for sending messages based on parameters set in the *Sending Time From* and *Sending Time To* fields, the time zone of the current branch is considered. The shift from the main financial institution's time zone is determined in the *Time Zone* field of the "Branches" form (Full → Configuration Setup → Main Tables → Branches).

- *Message Text* – block of six fields for entering the message text.

WAY4 allows variables to be used in message templates. When a message is generated, variables are replaced by the values of database table fields. The list of these variables and their use are given in the section "Use of Variables".

 When generating a marketing message to be included in a client statement, the *LABEL* and *VALUE* tags must be used in the *Message Text* field. For example, "LABEL=debt ;VALUE=%ACCOUNT\_DUE\_AMOUNT:\*::DATE\_TO%";.

- *Message Details* – field used to specify special message parameters as tags. See the section "Tags in the *Message Details* Field of a Message Template".
- *Is Ready* – field shows whether changes made to this template have been approved. A template is approved when approving the corresponding object (Service Package, Accounting Scheme, Product) or as the result of checking parameters when registering an Event type.
- *Priority* – determines the priority for sending a message; rules for interpreting this field's values are determined by the service that sorts and sends messages generated on the basis of the corresponding templates.

The form contains the following buttons:

- The [Actions] → [To Dictionary], [Dictionary] buttons – used to translate messages (see the section "Translating Messages into Local Languages").
- The [Clear Time] button – used to delete the values in the *Sending Time From*, *Sending Time To* fields.

## Translating Messages into Local Languages

WAY4 allows users to translate message text contained in the *Message Text* field into local languages registered in the system language dictionary (Full → Configuration Setup → Client Classifiers → Languages).

To translate a row in the *Message Text* block, proceed as follows:

- In the "Messages for <Event type name>" form, select the desired row and click the [Actions] → [To Dictionary] button. As a result, the current message will be registered in the Message Dictionary, and the user will be informed of this by the message "Message Put into Dictionary". To access the Message Dictionary, select the "Full → Configuration Setup → Main Tables → Message Dictionary" menu item. For instructions on working with the Message Dictionary, see the section "Message Dictionary" of the document "WAY4™ Dictionaries".
- In the "Messages for <Event type name>" form, click the [Dictionary] button. The "Dictionary for Messages for <Event type name>" form will be displayed (see Fig. 2).

Object Type	Object Name	Message Label	Message Name
Process	EVNT_MESSAGE	MSG_EVNT:8546_1	Your card's #%CONTRACT_NUMBER% loan is %ACCOUNT_BALANCE:L:810%
Process	EVNT_MESSAGE	MSG_EVNT:8546_2	
Process	EVNT_MESSAGE	MSG_EVNT:8546_3	
Process	EVNT_MESSAGE	MSG_EVNT:8546_4	
Process	EVNT_MESSAGE	MSG_EVNT:8546_5	

Fig. 2. Messages registered in the Message Dictionary

- In the "Dictionary for Messages for <Event type name>" form, select the desired row and click the [Translation] button to display the "Translation for <message>" form (see Fig. 3).

Language	Message Name
GERMAN	Ihre Kreditkarte N#%CONTRACT_NUMBER%64% debt ist %ACCOUNT_BALANCE:L:810% %CONTRACT_CURR%

Fig. 3. Translation form

To add a translated message, click the [Ins] button to add a row to the table, select the desired language from the *Language* drop-down list and enter the required text in the selected language in the *Message Name* field.

WAY4 allows the same message to be translated into many languages depending on the address to which the messages are being sent. For example, a message can be sent as a statement in the client's language, but the client will receive it on his mobile phone in the base language (English).



To do so, WAY4 has a global parameter `EVNT_MSG_STORE_AS_TEMPLATE` that determines the storage format of messages that are created when an Event opens (see the document "WAY4™" Global Parameters). This parameter can have one of the following values:

- "N" (default value); in this case when a message is created, the system checks whether a translation exists for it in the language specified in the client record:
  - If no translation into the client's language exists, the message is created in the message template language, that is, in English.
  - If a translation into the client's language exists, the message is created in the client's language.
- "Y"; in this case, a message created when an Event opens is stored as a template with calculated variable values. The process that sends this message to the target will translate the text according to configured parameters when the message is sent.

For example, a cardholder is enrolled in a service that notifies him of fund activity on his card account. The cardholder's language is Russian, and the bank employee's language is German. In this case it is appropriate to have the following types of translations of the notification text:

- ◆ Russian – this translation is used when generating a statement for the cardholder.
- ◆ German – this translation is used when generating a statement for bank employees.

The message text, compiled using transliteration (letter-by-letter conversion of a text from one alphabet into another alphabet; for example, if a Russian text is written in Latin letters) – this translation is used to create messages sent to a cardholder's mobile phone.

## Chapter 3. Use of Variables

Variables in message template text are indicated using the % (percent) symbol. For example, if the variable %CONTRACT\_NUMBER% is used in a template, the contract number will replace it when a message is generated.

### List of Variables

The table below shows the list of variables used in message templates with a description of their values. This list can be expanded as needed to carry out various tasks related to message creation.

*Table 1. List of variables for configuring message templates*

№	Name	Value
Contract information		
1.	%CONTRACT_NUMBER%	Contract number
2.	%CONTRACT_NAME%	Contract name
3.	%MAIN_CONTRACT_NUMBER%	Parent contract number. If the current contract is not a subcontract of another contract, this is an empty value.
4.	%MAIN_MAIN_CONTRACT_NUMBER%	Number of the highest-ranking contract in the contract tree. If the current contract is not a subcontract of another contract, this is the number of the current contract.
5.	%TR_FIRST_NAM%	Data for embossing
6.	%TR_LAST_NAM%	
7.	%TR_COMPANY%	
8.	%CONTRACT_F_I%	Contract financial institution
9.	%RBS_NUMBER%	Contract RBS number
10.	%RBS_MEMBER_ID%	Payment system member ID
11.	%CON_CAT%	Contract category
12.	%PCAT%	Product category
13.	%CONTRACT_TYPE%	Contract type
14.	%CONTRACT_SUBTYPE%	Contract subtype
15.	%ACCOUNT_SCHEME%	Name of Accounting Scheme
16.	%SERVICE_PACK%	Name of Service Package
17.	%BEHAVIOR_TYPE%	Contract behavior type identifier (see the section "%BEHAVIOR_TYPE%")
18.	%BEHAVIOR_TYPE_CODE%	Contract behavior type code (see the section "%BEHAVIOR_TYPE%")
19.	%CONTRACT_CURR%	Contract currency
20.	%AVAILABLE%	Amount available to the contract. If special text is used in an Event-generated message, the variable will reflect the amount available before executing the transaction that opened the Event
21.	%CREDIT_LIMIT%	Credit limit value
22.	%AUTH_LIMIT_AMOUNT%	Same as %CREDIT_LIMIT%
23.	%CONTRACT_DATE_OPEN%	Contract opening date
24.	%CONTRACT_STATUS%	Contract status

№	Name	Value
25.	%CARD_EXPIRE%	Month and year of card expiry
26.	%ACNT_ADD_INFO_01%	Additional contract information (see the section "%CLNT_ADD_INFO...%").
27.	%ACNT_ADD_INFO_02%	
28.	%ACNT_ADD_INFO_03%	
29.	%ACNT_ADD_INFO_04%	
30.	%SIC%	Contract SIC (for device contracts). Selected from the TR_SIC field (ACNT_CONTRACT table). If this field is not filled in, the SIC will be taken from the DEVICE_PARMS table.
31.	%RISK_SCHEME%	Contract risk scheme
32.	%IPS_PRODUCT%	Name of the card product in the payment system (value of the <i>IPS Product</i> field of the contract subtype)
33.	%PRODUCT_CODE%	Contract Product code ( <i>Code</i> field of the "Products" form)
34.	%PRODUCT_CODE_2%	Additional contract Product code ( <i>Code 2</i> field of the "Products" form).
35.	%PRODUCT_CODE_3%	Additional contract Product code ( <i>Code 3</i> field of the "Products" form).
36.	%CLASSIFIER%	Value of contract or client classifiers, depending on a specific code (defined by the mandatory parameter <Classifier Code>)
37.	%DECISION%	Value of a calculated classifier, depending on a specific code (defined by the mandatory parameter <DecisionCode>)
38.	%FDT_<DT_CODE>:[date format]%	Specific type of functional date. A contract date type code is set as the value of <DT_CODE> (see the document "Contract Functional Dates"). One of the following templates can be used as the date format [date format]: YYMMDD YYMMDDHHMISS YYMMDDHH24MISS HHMMSS MMDD YDDD YYYY-MM-DD
39.	%PAST_DUE_DATE:[delinquency group]:[date format]%	Date on which delinquency arose. The delinquency group code and date format can be used as the values of the optional parameters [delinquency group] and [date format], respectively (see the section "Configuring Display of Past Due Date and Past Due Days in Customer Service Workbench" of the document "WAY4™ Accounting Schemes").
Contract balance information		
40.	%TOTAL_BLOCKED%	Total amount of blocked funds
41.	%OWN_BLOCKED%	Amount of funds blocked in the current contract
42.	%SUB_BLOCKED%	Amount of funds blocked in subcontracts
43.	%TOTAL_BALANCE%	Total balance
44.	%OWN_BALANCE%	Total balance of the current contract
45.	%SUB_BALANCE%	Balance of subordinate contracts
46.	%BALANCE%	Contract balance (same as %TOTAL_BALANCE%)(

№	Name	Value
47.	%CONTRACT_BALANCE%	Allows the value of a specified contract balance type in a specified currency to be shown (for details on use, see the section "%CONTRACT_BALANCE%").
48.	%ACNT_BALANCE%	Same as %CONTRACT_BALANCE%
49.	%CONTRACT_DUE_DATE%	Date until which an amount that has been billed must be paid. The date is calculated according parameters of the contract's Accounting Scheme.
50.	%BILLING_BALANCE: <BALANCE_CODE>%	Balance on the end date of the previous filling cycle, corresponding to the balance type code set in the {BALANCE_CODE> parameter.
Information about contract accounts		
51.	%ACC_TEMPL_GL_TYPE%	Value of the <i>GL Type</i> field of the "Template for <account type name>" form (template of the account used in the contract's Accounting Scheme, see the section "%ACC_TEMPL...%")
52.	%ACC_TEMPL_GL_NUMBER%	Value of the <i>GL #</i> field of the "Template for <account type name>" form (template of the account used in the contract's Accounting Scheme, see the section "%ACC_TEMPL...%")
53.	%ACC_TEMPL_HD_GL_NUMBER%	Value of the <i>HeadOffice GL Number</i> field of the "Template for <account type name>" form (template of the account used in the contract's Accounting Scheme, see the section "%ACC_TEMPL...%")
54.	%ACC_TEMPL_ACCOUNT_NAME%	Value of the <i>Account Name</i> field of the "Template for <account type name>" form (template of the account used in the contract's Accounting Scheme, see the section "%ACC_TEMPL...%")
55.	%ACC_TEMPL_INTEREST_ALGORITHM%	Value of the <i>Interest Algorithm</i> field of the "Template for <account type name>" form (template of the account used in the contract's Accounting Scheme, see the section "%ACC_TEMPL...%")
56.	%ACC_TEMPL_USE_GL%	Value of the <i>Aggregate GL For</i> field of the "Template for <account type name>" form (template of the account used in the contract's Accounting Scheme, see the section "%ACC_TEMPL...%")
57.	%ACCOUNT_NUMBER%	Number of the account whose type is set as the value of an additional parameter (see the section "%ACC_TEMPL...%")
58.	%ACCOUNT_TYPE	Name of the type of account whose code is set as the value of an additional parameter (see the section "%ACC_TEMPL...%")
59.	%ACCOUNT_DUE_TYPE%	Type of due normalization for the account whose type is set as the value of an additional parameter (see the section "%ACC_TEMPL...%")
60.	%ACCOUNT_BALANCE%	Makes it possible to show the contract balance considering only balances on accounts of a specific type, in a selected currency or on a selected date (see the section "%ACCOUNT_BALANCE%")
61.	%ACCOUNT_DUE_DATE%	Payment due date (see the section %ACCOUNT_DUE_DATE%, %ACCOUNT_DUE_AMOUNT%)

№	Name	Value
62.	%ACCOUNT_DUE_AMOUNT%	Amount of payment due (see the section %ACCOUNT_DUE_DATE%, %ACCOUNT_DUE_AMOUNT%)
63.	%ACCOUNT_INTEREST%	Allows information on contract account interest to be shown. If the billing cycle is still open, the amount of interest not yet accrued is reflected, if the billing cycle is closed, the amount of interest accrued for the cycle is reflected (see the section "%ACCOUNT_BALANCE%")
64.	%ACCOUNT_INTEREST_RATE%	Value of annual interest rate (see the section "%ACCOUNT_BALANCE%")
65.	%ACCOUNT_INTEREST_FEE%	Allows information on contract account interest fees to be shown. If the billing cycle is still open, the amount of the interest fee not yet collected is reflected, if the billing cycle is closed, the amount of the collected fee is reflected (see the section "%ACCOUNT_BALANCE%")
66.	%ACCOUNT_TURNOVER%	Contract account turnover (see the section "%ACCOUNT_BALANCE%")
67.	%ACCOUNT_TAG...%	%ACCOUNT_TAG...% variables make it possible to obtain data in the same way as the respective %ACCOUNT...% variables with consideration of arbitrary tags set in account schemes (see the section "%ACCOUNT_TAG%")
68.	%ACCOUNT_CAT...%	%ACCOUNT_CAT...% variables make it possible to obtain data in the same way as the respective %ACCOUNT...% variables with consideration of account categories (see the section "%ACCOUNT_CAT...%")
69.	%ACCOUNT_CURRENT_INTEREST%	Allows information about interest to be accrued at a particular point in time to be shown. If the billing period is still open, the amount to be accrued is reflected, if the billing cycle is closed, an empty value is shown (see the section "%ACCOUNT_BALANCE%")
70.	%ACCOUNT_CURRENT_INTEREST_FEE%	Allows information about contract account interest fees at a particular point in time to be shown. If the billing cycle is still open, the amount to be collected is reflected, if the billing cycle is closed, an empty value is shown (see the section "%ACCOUNT_BALANCE%")
71.	%DUE_AMOUNT%	Amount of funds on "Payment Due" category accounts at the closing of the previous billing cycle.
72.	%IMMEDIATE_AMOUNT%	Amount of funds on "Pay Immediate" category accounts at the closing of the previous billing cycle
73.	%BILLING_DATE%	Closing date of the current billing cycle
74.	%NEXT_BILLING_DATE%	Closing date of the next billing cycle. This variable can be used as an additional date parameter in %ACCOUNT_...% variables (see the section "Characteristics of variable use")
75.	%PREV_BILLING_DATE%	Closing date of the previous billing cycle. This variable can be used as an additional date parameter in %ACCOUNT_...% variables (see the section "Characteristics of variable use")

№	Name	Value
76.	%DUE_DATE%	Date of the next due normalisation calculated at the closing of the previous billing cycle
Information about the balances of contracts for a multi-currency product		
77.	%MULTI_CURRENCY_BALANCE:<Currency>%	Total balance amount (Total Balance) for account contracts in a specific currency (mandatory <Currency> parameter). If a contract with this currency type is not found, the variable is assigned the value of "0.00". If a top-level contract has several accounts in the same currency, the balance value according to this parameter will be shown as the total balance for all account contracts in this currency. The variable's value is generated in the context of client data (for cases when a top-level contract has account contracts for different clients).
78.	%MULTI_CURRENCY_CURR:<Currency>%	Letter currency code, corresponding to the numeric code set as the value of the <Currency> parameter.
79.	%MULTI_CURRENCY_BALANCE_ALL%	List of all currencies and total balances from a client's account contracts.
80.	%MULTI_CURRENCY_BALANCE_NONZERO%	List of all currencies and non-zero balance amounts from a client's account contracts.
Service Information. Data are taken from the Service or tariff (if a tariff is set in the Service)		
81.	%SERVICE\$TRANS_TYPE%	Name of transaction type
82.	%SERVICE\$FEE_MIN_AMOUNT%	Minimum size of the fee that will be charged regardless of the transaction amount.
83.	%SERVICE\$FEE_MAX_AMOUNT%	Maximum amount of the fee that can be charged for a transaction.
84.	%SERVICE\$FEE_BASE_AMOUNT%	Additional mandatory fee. Does not depend on interest rate.
85.	%SERVICE\$FEE_PERCENT%	Fee percent.
Tariff information		
86.	%TARIFF_TYPE%	Tariff form's <i>Tariff Type</i> field value.
87.	%TARIFF_DOMAIN%	Tariff form's <i>Tariff Domain</i> field value.
88.	%TARIFF_CODE_EXT%	Tariff form's <i>Tariff Code Ext</i> (TABLE_CODE_FROM) field value.
Client information (values of the "Client – Edit for <name of client>" form fields)		
89.	%CCAT%	<i>Client Category</i> field value
90.	%CLIENT_TYPE%	<i>Client Type</i> field value
91.	%CLIENT_SHORT_NAME%	<i>Short Name</i> field value
92.	%CLIENT_NUMBER%	<i>Reference #</i> field value
93.	%CLIENT_REG_NUMBER%	<i>Registration #</i> field value
94.	%CLIENT_REG_DETAILS%	<i>Registration Dtls</i> field value
95.	%SALUTATION%	Value of the first <i>Salutation</i> field
96.	%SALUTATION_SUFFIX%	Value of the second <i>Salutation</i> field
97.	%TITLE%	Same as %SALUTATION%
98.	%CLIENT_LAST_NAME%	<i>Last Name</i> field value
99.	%CLIENT_FIRST_NAME%	<i>First Name</i> field value
100.	%CLIENT_BIRTH_NAME%	<i>Birth Name</i> field value
101.	%CLIENT_FATHER_S_NAME%	<i>Middle Name</i> field value
102.	%CLIENT_BIRTH_DATE%	<i>Date of Birth</i> field value

<b>Nº</b>	<b>Name</b>	<b>Value</b>
103.	%CLIENT_BIRTH_PLACE%	<i>Place of Birth</i> field value
104.	%CLIENT_LANGUAGE%	<i>Language</i> field value
105.	%COMPANY%	<i>Company Name</i> field value
106.	%ITN%	<i>Individual Number</i> field value
107.	%CLIENT_SALUTATION%	Same as %SALUTATION%
108.	%CLIENT_F_I%	<i>Institution</i> field value
109.	%CLIENT_BRANCH%	<i>Branch</i> field value
110.	%GENDER%	<i>Gender</i> field value
111.	%CLNT_ADD_INFO_01%	Additional client information (see the section "%CLNT_ADD_INFO...%")
112.	%CLNT_ADD_INFO_02%	
113.	%CLNT_ADD_INFO_03%	
114.	%CLNT_ADD_INFO_04%	
115.	%PHONE%	<i>Phone</i> field value
116.	%PHONE_H%	<i>Phone (home)</i> field value
117.	%PHONE_M%	<i>Mobile</i> field value
Client/contract address information (field values from the "Addresses for <client name/contract number>" form), see "%ADDR...%"		
118.	%ADDR_TYPE%	<i>Type</i> field value
119.	%ADDR_NAME%	<i>Name</i> field value
120.	%ADDR_DELIVERY_TYPE%	<i>Delivery</i> field value
121.	%ADDR_LANGUAGE%	<i>Language</i> field value
122.	%ADDR_SALUTATION%	<i>Salutation</i> field value
123.	%ADDR_SALUTATION_SUFFIX%	<i>Suffix</i> field value
124.	%ADDR_FIRST_NAME%	<i>First Name</i> field value
125.	%ADDR_LAST_NAME%	<i>Last Name</i> field value
126.	%ADDR_BIRTH_NAME%	<i>Birth Name</i> field value
127.	%COUNTRY%	<i>Country</i> field value
128.	%COUNTRY_POST%	<i>Postal Code</i> field value in the "Full → Configuration Setup → Main Tables → Country Table" table for the country corresponding to the document's <i>Country</i> field
129.	%ADDR_STATE%	<i>State</i> field value
130.	%CITY%	<i>City</i> field value
131.	%ADDR_ZIP%	<i>ZIP</i> field value
132.	%ADDR_MUNICIPALITY_CODE%	Value of the <i>Municipality</i> field
133.	%ADDR_PHONE%	<i>Phone</i> field value
134.	%ADDR_PHONE_H%	<i>Phone (h)</i> field value
135.	%ADDR_PHONE_M%	<i>Phone (mob)</i> field value
136.	%ADDR_FAX%	<i>Fax</i> field value
137.	%ADDR_FAX_H%	<i>Fax (h)</i> field value
138.	%ADDR_LINE_1%	Client address information. Address information may differ depending on system variable values. To set the address type, execute the rpr.GET_ADDRESS function
139.	%ADDR_LINE_2%	
140.	%ADDR_LINE_3%	
141.	%ADDR_LINE_4%	
142.	%ADDR_ADD_INFO%	<i>Add Info</i> field value
143.	%E-MAIL%	<i>E-mail</i> field value
144.	%ADDR_URL%	<i>URL</i> field value
145.	%ADDR_TITLE%	Same as %ADDR_SALUTATION%

№	Name	Value
Information on Events and usage limiters		
146.	%IS_EVENT_OPENED%	<p>The variable returns one of the following values:</p> <p>'Y' – if an Event with a specified code has one of the following statuses on a specified date: "Posted", "Suspended", "Closed", "Inactive";</p> <p>'N' – in all other cases.</p> <p>The following are specified as mandatory parameters for the variable:</p> <p>&lt;Event Code&gt; – Event code;</p> <p>&lt;on Date&gt; – date on which the Event status is determined</p>
147.	%ACTIVE_EVENT_START_DATE%	Opening date of an Event with a specified code. Additional parameters are the same as %IS_EVENT_OPENED%
148.	%ACTIVE_EVENT_END_DATE%	Closing date of an Event with a specified code. Additional parameters are the same as %IS_EVENT_OPENED%
149.	%USG_LIM...%	%USG_LIM...% variables allow information on contract usage limiters to be shown (see the section "%USG_LIM...%")
Payment party requisites (field values from the "Parties for Doc General" form, additionally see the section "%PARTY...%")		
150.	%PARTY_AFFILIATION_IDT%	Affiliation Source Number field value
151.	%PARTY_AFFILIATION_DATE_FROM%	Affiliation Date From field value
152.	%PARTY_AFFILIATION_DATE_TO%	Affiliation Date To field value
153.	%PARTY_COUNTRY%	Country field value
154.	%PARTY_COUNTRY_NAME%	Country Name field value
155.	%PARTY_POSTAL_CODE%	Postal Code field value
156.	%PARTY_REGION%	Region field value
157.	%PARTY_CITY%	City field value
158.	%PARTY_CITY_EN%	City English field value
159.	%PARTY_STREET%	Street field value
160.	%PARTY_ADDRESS_LOCATION%	Location field value
161.	%PARTY_ADDRESS_LINE_1%	Address Line 1 field value
162.	%PARTY_ADDRESS_LINE_2%	Address Line 2 field value
163.	%PARTY_ADDRESS_LINE_3%	Address Line 3 field value
164.	%PARTY_ADDRESS_LINE_4%	Address Line 4 field value
165.	%PARTY_ADDRESS_LINE_EN_1%	Address Line En 1 field value
166.	%PARTY_ADDRESS_LINE_EN_2%	Address Line En 1 field value
167.	%PARTY_ADDRESS_LINE_EN_3%	Address Line En 1 field value
168.	%PARTY_ADDRESS_LINE_EN_4%	Address Line En 1 field value
169.	%PARTY_MUNICIPALITY_CODE%	Municipality Code field value
170.	%PARTY_ADDRESS_ADD_INFO%	Add Info field value
171.	%PARTY_CONTACT_URL%	URL field value
172.	%PARTY_CONTACT_E_MAIL%	E-Mail field value
173.	%PARTY_CONTACT_PHONE%	Phone field value
174.	%PARTY_CONTACT_FUNCTION%	Function field value
175.	%PARTY_ACC_NUMBER%	Account Number field value
176.	%PARTY_ACC_MEMBER_ID%	Account Member Id field value
177.	%PARTY_ACC_CURR%	Account Currency field value



<b>Nº</b>	<b>Name</b>	<b>Value</b>
178.	%PARTY_ACC_DATE_OPEN%	<i>Account Date Open</i> field value
179.	%PARTY_ACC_BRANCH%	<i>Account Branch Code</i> field value
180.	%PARTY_TYPE%	<i>Party Type</i> field value
181.	%PARTY_CLIENT%	<i>Party Client</i> field value
182.	%PARTY_CCAT%	<i>Client Category</i> field value
183.	%PARTY_RESIDENCE%	<i>Residence</i> field value
184.	%PARTY_NAME%	<i>Party Name</i> field value
185.	%PARTY_NAME_EN%	<i>Party Name English</i> field value
186.	%PARTY_BIRTH_DATE%	<i>Birth-Reg Date</i> field value
187.	%PARTY_REG_NUMBER_TYPE%	<i>Reg Number Type</i> field value
188.	%PARTY_REG_NUMBER%	<i>Reg Number</i> field value
189.	%PARTY_REG_DETAILS%	<i>Reg Details</i> field value
190.	%PARTY_ITN%	<i>Individual Tax Number</i> field value
191.	%PARTY_TAX_POSITION%	<i>Tax Position</i> field value
192.	%PARTY_AFFILIATION_CODE%	<i>Affiliation Code</i> field value
Information about a transaction (field values from the "All Docs" or "Doc-General" forms)		
193.	%TRANS_AMOUNT%	<i>Trans Amount</i> field value
194.	%TRANS_CURR%	<i>Trans Curr</i> field value
195.	%TRANS_DATE%	<i>Trans Date</i> field value
196.	%TRANS_DR_CR%	Transaction type <i>DRICR</i> field value. Defined by the <i>Transaction Type</i> field of the document.
197.	%TRANS_TYPE%	Value from the "Message Dictionary" dictionary that is localized for the client's language and corresponds to the <i>Trans Type</i> field value.
198.	%DFLT_TRANS_TYPE%	<i>Trans Type</i> field value. If a transaction type check must be included in a message generation condition (see "Using Conditional Operators"), it is recommended to use this variable. If a transaction type name must be included in message text in a client's language, it is recommended to use the %TRANS_TYPE% variable.
199.	%TRANS_SIC%	<i>SIC Code</i> field value
200.	%TRANS_COUNTRY%	<i>Trans Country</i> field value
201.	%TRANS_CITY%	<i>Trans City</i> field value
202.	%TRANS_DETAILS%	<i>Trans Details</i> field value
203.	%TRANS_SOURCE_CODE%	<i>Source Code</i> field value
204.	%TRANS_TARGET_CODE%	<i>Target Code</i> field value
205.	%TRANS_RETURN_CODE%	<i>Return Code</i> field value
206.	%TRANS_RC_DESCRIPTION%	Text of the message with the code from the <i>Return Code</i> field
207.	%TRANS_REASON_DETAILS%	<i>Reason Details</i> field value
208.	%TRANS_TAGS%	<i>Add Data</i> field value (see the section "%TRANS_TAGS%")
209.	%DOC_TAGS%	Same as %TRANS_TAGS%
210.	%TRANS_RET_REF_NUMBER%	<i>Ret Ref Number</i> field value
211.	%TRANS_ACQ_REF_NUMBER%	<i>Acq Ref Number</i> field value
212.	%TRANS_ISS_REF_NUMBER%	<i>Iss Ref Number</i> field value
213.	%TRANS_SOURCE_REG_NUM%	<i>S Reg Num</i> field value
214.	%TRANS_AUTH_CODE%	<i>Auth Code</i> field value

№	Name	Value
215.	%TRANS_DOC_AMND_DATE%	<i>Amendment Date</i> field value
216.	%TRANS_DOC_AMND_OFFICER%	<i>Amendment Officer</i> field value
217.	%TRANS_DOC_ID%	<i>Record Id</i> field value
218.	%TRANS_SOURCE_NUMBER%	<i>Source Number</i> field value
219.	%TRANS_TARGET_NUMBER%	<i>Target Number</i> field value
220.	%TRANS_FX_SETTL_DATE%	<i>FX Settl Date</i> field value
221.	%DOC_RECONS_AMOUNT%	<i>Amount Reconcil</i> field value
222.	%RECONS_CURR%	<u>Currency</u> field value for <i>Amount Reconcil</i>
223.	%ADD_SERVICE_INFO%	<i>Service Info</i> field value of the "Services for <name of additional online service>" form of the Service on which the transaction was made
224.	%TRANS_NAME%	Name of the transaction according to the "Message Dictionary"
225.	%TRANS_ACC_AMOUNT%	Amount debited from client account when document was posted (see the section "%TRANS_ACC...%").
226.	%TRANS_ACC_CURR%	Currency of the amount debited from the client account (see the section "%TRANS_ACC...%").
227.	%TRANS_ACC_FEE%	Fee charged to the client account (see the section "%TRANS_ACC...%").
228.	%TRANS_ACC_TOTAL%	Sum of the %TRANS_ACC_AMOUNT% and %TRANS_ACC_FEE% values (see the section "%TRANS_ACC...%").
229.	%TRANS_ACC_ADV_AMOUNT%	Amount of an entry for a macrotransaction with the "Advice" request category (see the section "%TRANS_ACC...%").
230.	%TRANS_ACC_ADV_FEE%	Fee for a macrotransaction with the "Advice" category (see the section "%TRANS_ACC...%").
231.	%TRANS_ACC_ADV_TOTAL%	Sum of the %TRANS_ACC_ADV_AMOUNT% and %TRANS_ACC_ADV_FEE% values (see the section "%TRANS_ACC...%").
232.	%TRANS_ACC_REV_AMOUNT%	Amount of an entry for a macrotransaction with the "Reversal" request category (see the section "%TRANS_ACC...%").
233.	%TRANS_ACC_REV_FEE%	Fee for a macrotransaction with the "Reversal" category (see the section "%TRANS_ACC...%").
234.	%TRANS_ACC_REV_TOTAL%	Sum of the %TRANS_ACC_REV_AMOUNT% and %TRANS_ACC_REV_FEE% values (see the section "%TRANS_ACC...%").
235.	%POSTING_DATE%	Value of the <i>Date</i> field in the "M-trans for..." form – banking date of posting the macrotransaction for the corresponding document.
236.	%TRANS_ACC_MTR_DETAILS%	<i>Mtr Details</i> field value of the form "M-trans for..." – macrotransaction details for the corresponding document.
237.	%ENTRY_FX_RATE_VALUE%	FX rate
238.	%TRANS_BLOCKED_AMOUNT%	Amount blocked as the result of the transaction
239.	%TRANS_BLOCKED_CURR%	Currency of funds blocked as the result of the transaction

№	Name	Value
240.	%TRANS_BLOCKED_FEE%	Fee amount blocked as the result of the transaction
241.	%TRANS_SERVICE_CLASS%	<i>Service Class</i> classifier transaction type.
242.	%TRANS_SOURCE_SPC%	Relation type ( <i>Relation Type</i> field value) of the source contract and the related contract.
243.	%TRANS_TARGET_SPC%	Relation type ( <i>Relation Type</i> field value) of the target contract and the related contract
244.	%DOC_ACTION%	Document's ACTION field value
245.	%TRANS_MISC_FEE:<UsageType>%	Fee amount and letter code of the currency for the fee charged if a <UsageType> limiter is activated.
246.	%TRANS_MISC_LIM_FEE:<UsageType>%	<p>The value by which the limit set by a &lt;UsageType&gt; limiter must be exceeded for a %TRANS_MISC_FEE:&lt;UsageType&gt;% to be charged. The amount and letter code of the currency is passed in the variable.</p> <p>Example of a template: 'Fee of %TRANS_MISC_FEE:LIM1% for exceeding the monthly cash withdrawal limit by an amount of %TRANS_MISC_LIM_FEE:LIM1%'</p> <p>Example of a message: 'A fee of 3 EUR for exceeding the monthly cash withdrawal limit by 3000 EUR'.</p>
247.	%SIBLING_MTR%	<p>Indicates ("Y"/"N") if a transaction was made for a payment order with the SIBLING_ORDER code (if funds are insufficient, the necessary amount is debited from the contract's other currency accounts). This variable can be used as a condition for getting data in statements for multi-currency Product contracts (P_EXT_TRANS_DETAILS parameter value).</p>
Information about amount of accrued bonus points (support of "Loyalty Programs", WAY4 Loyalty module, is optional functionality and is provided according to an additional agreement with the WAY4 vendor).		
248.	%TRANS_BONUS_ALL%	Amount of all bonuses points, regardless of the account type, with <i>Account Role</i> = "BONUS" (based on contract data).
249.	%TRANS_BONUS_AT:<AccountType>%	Amount of all bonus points, regardless of account type with <i>Account Role</i> = "BONUS" (based on contract data).
250.	%TRANS_BONUS_LIST%	Comma-separated list of bonus points (based on contract data)
251.	%TRANS_BONUS_DOC_ALL%	Amount of all bonus points, regardless of account type, with <i>Account Role</i> = "BONUS" (based on document data).
252.	%TRANS_BONUS_DOC_AT:<AccountType>%	Amount of all bonus points for the account type specified as the <AccountType> parameter value (based on document data).


№	Name	Value
253.	%TRANS_BONUS_DOC_LIST[:<Prefix>:<Expire>:<Days>]%	<p>Comma-separated list of bonus points with their effective periods (based on document data). &lt;Prefix&gt;, &lt;Expire&gt;, &lt;Days&gt; can be codes registered in the "Message Dictionary" handbook of text elements (Full\Configuration Setup\Main Tables\ Message Dictionary; <i>Object Type</i> = 'Transaction', <i>Object Name</i> = 'Bonus'). An effective period is determined based on the VALUE_DAYS tag value. If an effective period is not defined, only the amount of points will be included in the variable, without the &lt;Expire&gt; and &lt;Days&gt; text elements. The code of the corresponding account type is output together with the amount. Example of the variable's values:</p> <p>'Bonus List: 100 SPASIBO, 100 z expired After 21 days!'</p> <p>Here:</p> <p>'Bonus List:' is the &lt;Prefix&gt; parameter value.</p> <p>'100' is the amount of bonus points.</p> <p>'SPASIBO' and 'z' are account type codes.</p> <p>'expired After' – &lt;Expire&gt; parameter value.</p> <p>'21' – VALUE_DAYS value.</p> <p>'days!' – &lt;Days&gt; parameter value.</p>
Information on a payment order for which a document was generated ("Full Info For St. Orders for ..." form field values). Variables are used with the STANDING_ORDER\$ prefix		
254.	%PAYMENT_TYPE%	Payment type ( <i>Payment Type</i> field value)
255.	%PAYMENT_RECEIVER%	Payment recipient ( <i>Standard Payee</i> field value)
256.	%TARGET_DETAILS_1%	Additional information about a correspondent account ( <i>Target Details 1</i> field value)
257.	%TARGET_DETAILS_2	Additional information about a correspondent account <i>Target Details 2</i> field value)
258.	%TRANS_DETAILS%	Additional information about a payment ( <i>Payment Details</i> field value)
259.	%COMMENT_TEXT%	Comments for an order ( <i>Order Comment</i> field value)
260.	%REASON_DETAILS%	Tagged information from the <i>Posting Details</i> field
261.	%ORDER_CODE%	Standing Order code ( <i>Order Code</i> field value)
Other Variables		
262.	%PIN_ATTEMPTS%	Remaining number of PIN attempts.
263.	%MAX_PIN_ATTEMPTS%	Maximum number of PIN attempts set for the current contract.
264.	%OFFICER%	Current user's identifier
265.	%OFFICER_GROUP%	Current user's group

## Characteristics of variable use

Special formats for variables, allowing the use of additional parameters are provided for in order to perform various tasks in the system.

The ":" (colon) sign is used to separate the names of variables and parameters, as well as in between parameters.

Additional parameters for several variables are described below. Parameters are shown in angular brackets "< >". Optional parameters are additionally shown in square brackets "[ ]".

 Optional parameters are used as additional conditions for filtering data when generating the value of a variable. In a number of cases, the absence of a parameter is interpreted as any of its possible values. For example, if the <AccCodeList> parameter (list of account types) is not specified for the %ACCOUNT\_BALANCE% variable, the variable's value will include information for all a contract's accounts.

## %ADDR...%

%ADDR...% variables (for example, %ADDR\_LINE\_1%, %ADDR\_ZIP%, %ADDR\_NAME%, as well as other variables for getting client/contract address data) can be used with the following additional parameters:


```
%ADDR_LINE_1[:<AddrTypeCode>[:<LanguageCode>]]%
```

where:

- <AddrTypeCode> is the address type code (from the list "Address Types" – "Full → Configuration Setup → Client Classifiers → Address Types").
- <LanguageCode> is the language code according to the system dictionary ("Full → Configuration Setup → Client Classifiers → Languages").

## %ACCOUNT...%

This section describes characteristics of %ACCOUNT...% variable use.

 One of the variables described in the subsection "Additional date parameters" can be used as an additional date parameter for %ACCOUNT...% variables. For example, the expression %ACCOUNT\_BALANCE:P::PREV\_BILLING\_DATE% will be replaced by the value of the balance of an account with the "P" code (Client Deposit) at the time of closing the previous billing cycle (see the section "%ACCOUNT\_BALANCE%").


### %ACCOUNT\_BALANCE%

The %ACCOUNT\_BALANCE% variable can be used with the following additional parameters:

```
%ACCOUNT_BALANCE[:<AccCodeList>[:<Curr>[:<Date>[:<DR>]]]]%
```

where

- <AccCodeList> – account type code (or a comma-delimited list of codes) corresponding to the *Code* field value in the "Account Types" form (Full → Configuration Setup → Accounting Setup → Account Types). Only account balances for this account type will be considered when calculating the value of the %ACCOUNT\_BALANCE% variable.


 An account type code can consist of one or several characters. The characters of an account type code consisting of several characters must always be set apart by commas. In the absence of commas, each symbol is understood to be a separate code (for example, a code specified in the

format %ACCOUNT\_BALANCE:ps1% is understood to be three separate codes: p, s, 1). Codes are delimited as follows:

- When codes are part of a list, they are delimited as usual (for example, %ACCOUNT\_BALANCE:p,ps1,ps2%).
- If a single code consisting of several characters is specified, it must be set apart by commas (for example, %ACCOUNT\_BALANCE:.,ps1,%).
- <Curr> – currency code. When calculating the value of the %ACCOUNT\_BALANCE% variable, only the balances of contract accounts in this currency will be considered. The result will also be given in this currency; if the currency is not specified, the balances of accounts in all currencies will be totaled in the contract currency.
- <Date> – the date in YYYY-MM-DD format. When calculating the value of the %ACCOUNT\_BALANCE% variable, account balances on this date will be considered; by default, balances on the current date are considered.
- DR – specified to transform a negative number into a positive one.

Example 1. The code of deposit accounts for physical persons is "P". Therefore, the following expression in the message template %ACCOUNT\_BALANCE:P% makes it possible to obtain the balance amount on all deposit accounts of the contract (there can be several accounts in different currencies) at the present time in the contract currency.

Example 2. The following expression in the message template: %ACCOUNT\_BALANCE::840:2009-13-03% makes it possible to obtain the balance on all USD contract accounts on March 13, 2009.

 The same configuration method is used for the variables: %ACCOUNT\_INTEREST%, %ACCOUNT\_INTEREST\_FEE%, %ACCOUNT\_CURRENT\_INTEREST%, %ACCOUNT\_CURRENT\_INTEREST\_FEE%, %ACCOUNT\_INTEREST\_RATE% (with the exception of the DR and <Curr> parameter).

### **%ACCOUNT\_DUE\_DATE%, %ACCOUNT\_DUE\_AMOUNT%**

%ACCOUNT\_DUE\_DATE% and %ACCOUNT\_DUE\_AMOUNT% variables can be used with the following additional parameters:

```
%ACCOUNT_DUE_DATE[:<AccCodeList>[:<Curr>[:<Date>[:<DR>[:<Number of repayment>]]]]]%
```

where

- <AccCodeList>, <Curr>, <Date>, <DR> – for a description of parameters, see the section "%ACCOUNT\_BALANCE%"
- <Number of repayment> – number of due payment on which information is shown.

### **%ACCOUNT\_TURNOVER%**

The %ACCOUNT\_TURNOVER% variable can be used with the following additional parameters:

```
%ACCOUNT_TURNOVER[:<AccCodeList>[:<Curr>[:<Date>[:<DR/CR>
[:<Number of repayment>[:<ServiceClassList>[:<AccCodeExcludeList>]]]]]]%]
```

where

- <AccCodeList>, <Curr>, <Date> – for a description of parameters, see the section "%ACCOUNT\_BALANCE%".
- <DR/CR> – debit or credit transactions are considered when calculating account turnover.
- <ServiceClassList> – list of transaction type codes for transactions that will be considered when calculating account turnover.
- <AccCodeExcludeList> – code of account type (or comma-delimited list of codes), fund flow to/from which will not be considered when calculating turnover on this account.

### **Additional date parameters**

Additional date parameters can be used for variables showing information on account balances (which, in particular are used for generating statements, SMS notifications). For example, the account balance at a particular point in time can be shown (at the time of message creation), or the account balance at the opening of a contract's billing cycle.

- %<variable name>::LOCAL\_DATE% shows the variable value on the date of message creation (the default value).
- %<variable name>::BEGIN\_DATE% – shows the variable value on the date of the beginning of this contract's billing cycle.
- %<variable name>::NEXT\_BILLING\_DATE% – shows the variable value on the date of this contract's next billing cycle.
- %<variable name>::PREV\_BILLING\_DATE% - shows the variable value on the closing date of this contract's previous billing cycle.
- %<variable name>::BALANCE\_DATE% – shows the variable value on the date specified by the *Date To* parameter of the run statement dialog.
- %<variable name>::DATE\_FROM% – shows the variable value on the date:
  - For monthly statements on a contract – the date of the beginning of the cycle, specified by the *DateTo* parameter of the run statement dialog.
  - For current statements on a contract – a date specified by the *Date From* parameter of the run statement dialog.
- %<name of variable>::DATE\_TO% – shows the variable value on the date:
  - For a statement – on the date specified by the *Date To* parameter of the run statement dialog.

For example, %ACCOUNT\_BALANCE:P:: BEGIN\_DATE%.



When delimiters are used (:) and the date parameter is absent, the default value (LOCAL\_DATE) is used.

## %ACCOUNT\_CAT...%

%ACCOUNT\_CAT...% variables (for example, %ACCOUNT\_CAT\_BALANCE%, %ACCOUNT\_CAT\_DUE\_DATE% and others) can be used with additional parameters in the same way as the respective variables %ACCOUNT...% (%ACCOUNT\_BALANCE%, %ACCOUNT\_DUE\_DATE% etc.).

Example:

```
%ACCOUNT_CAT_BALANCE[:<AccCatCodeList>[:<Curr>[:<Date>[:<DR>]]]]%
```

Where

- <AccCatCodeList> – account type category code (or comma-delimited list of codes) corresponding to the value in the *Category* field of the "Account Types" form (Full → Configuration Setup → Accounting Setup → Account Types).
- <Curr>, <Date>, <DR> – for a description of parameters, see the section "%ACCOUNT\_BALANCE%".

## %ACCOUNT\_TAG%

%ACCOUNT\_TAG% variables make it possible to get aggregated data for accounts whose templates contain the tag specified in the name of the variable, with a value that is the same as one of the values listed in this variable.

Names for this variable type are set as follows:

- After the "%ACCOUNT\_TAG\_" prefix, one of the following literals is specified: "TURNOVER", "BALANCE", "DUE\_AMOUNT", "INTEREST", "INTEREST\_FEE", "CURRENT\_INTEREST", "CURRENT\_INTEREST\_FEE"; these literals determine the type of data that are received.
- After the literal, an underline character "\_" is specified and the name of the tag set in the account template for which data must be obtained.

For example, the variable %ACCOUNT\_TAG\_BALANCE\_STMT:1,2,7% makes it possible to get aggregated values for the balances of accounts whose templates (in the *Template Details* field) contain the STMT tag with at least one of the following values: 1, 2, 7.

## %ACC\_TEMPL...%

%ACC\_TEMPL...% variables can be used with the following additional parameters:


```
%ACC_TEMPL_INTEREST_ALGORITHM:<Code>[:<Curr>]%
```

where

- <Code> – the account type code for which template parameters are specified. The code corresponds to the *Code* field value in the "Account Types" form (Full → Configuration Setup → Accounting Setup → Account Types).



- <Curr> – account currency (if the contract has accounts in different currencies).


 The same configuration method is used for the following variables:  
%ACCOUNT\_NUMBER%, %ACCOUNT\_DUE\_TYPE%,  
%ACCOUNT\_TYPE%.

## %BEHAVIOR\_TYPE%

The %BEHAVIOR\_TYPE% value can be used with the following additional parameter:

```
%BEHAVIOR_TYPE[:<on Date>]%
```

where <on Date> – the date in YYYY-MM-DD format. When calculating the value of the %BEHAVIOR\_TYPE% variable, the contract behavior type for the given date is specified.

 The <on Date> parameter is used to configure the variables %BEHAVIOR\_TYPE\_CODE%, %IS\_EVENT\_OPENED%, %ACTIVE\_EVENT\_START\_DATE%, %ACTIVE\_EVENT\_END\_DATE%.

## %CLNT\_ADD\_INFO...%

The variables %CLNT\_ADD\_INFO...% (%CLNT\_ADD\_INFO\_01%, %CLNT\_ADD\_INFO\_02% and others) can be used with the following additional parameter:

```
%CLNT_ADD_INFO_01[:<TagName>]%
```

where <TagName> – the name of the tag whose value must be obtained

 %ACNT\_ADD\_INFO...% variables are configured in the same way.


## %CONTRACT\_BALANCE%

The %CONTRACT\_BALANCE% variable can be used with the following additional parameters:

```
%CONTRACT_BALANCE[:<BalanceTypeCode>[:<Curr>[:<Date>[:DR]]]]%
```

where

- <BalanceTypeCode> – balance type code (the *Code* field value of the "Balance Types" dictionary).
- <Curr> – code of the currency in which the balance type value will be calculated.
- <Date> – the date in YYYY-MM-DD format, for which the balance type value will be calculated; by default, the current date is used.
- DR – specified to convert a negative result into a positive number.

 The <BalanceTypeCode> parameter is used to configure the %ACNT\_BALANCE% variable.

## %USG\_LIM...%

%USG\_LIM...% variables:

- %USG\_LIM\_MAX\_NUMBER% -maximum number of transactions.
- %USG\_LIM\_MAX\_AMOUNT% - maximum total amount of transactions for a period.
- %USG\_LIM\_MAX\_SINGLE\_AMOUNT% - maximum amount of one transaction.
- %USG\_LIM\_CURRENT\_NUMBER% - current value of transaction counter.
- %USG\_LIM\_CURRENT\_AMOUNT% - current value of amount counter.
- %USG\_LIM\_UNSPENT\_AMOUNT% - unspent amount within the limit.
- %USG\_LIM\_UNSPENT\_NUMBER% - number of transactions that can still be made within the limit.
- %USG\_LIM\_CURR% - limiter currency.
- %USG\_LIM\_START\_DATE% - counter activity start date.
- %USG\_LIM\_END\_DATE% - counter activity end date.
- %USG\_LIM\_ACTIVE% - code of the limiter's current state (value of the Current Status field in the "Swch Usage for <client name>" form for the corresponding contract).

are used with the following parameter:

%USG_LIM_MAX_NUMBER:<UsageTemplateCode>
---

where <UsageTemplateCode> – limiter template code.

## %PCNT%

The %PCNT% variable is used when it is necessary to display a % (percent) symbol in a message.

## %SUM%

The %SUM% variable makes it possible to determine the sum of values for a list of variables (comma-delimited), for example:

%SUM(%OWN\_BALANCE%,%SUB\_BALANCE%)%

## %APPL\_INFO...%

%APPL\_INFO...% variables allow data from the APPL\_INFO table with additional information about applications (see the section "Application Additional Information Classifiers (Application Info Types)" of the document "Advanced Applications Module (workflow configuration)") to be used in messages. %APPL\_INFO...% variables can be used with the following additional parameters:

%APPL_INFO_<uppercase_field_code>[*<FormatCode>]:<info_type_code>%
--

where

- <uppercase\_field\_code> is the APPL\_INFO table field code in capital letters.

- <info\_type\_code> is the additional information classifier code (Application Info Type). The classifier code is specified in the *Code* field of the "Application Info Types" form (Advanced Applications R2 → Setup → Application Info Types).

## %PARTY...%

To include party payment requisites (based on client or contract data) in messages not related to transaction notifications (for example, to include in a client statement), %PARTY...% variables are used with the following additional parameters:

```
%PARTY_AFFILIATION_IDT:<PaymTypeCode>:<PartyTypeCode>%
```

where:

- <PaymTypeCode> is the payment type code (corresponding to the value of the *Code* field in the "Payment on Account Types" form).
- <PartyTypeCode> is the payment party role code (corresponding to the value of the *Party Type* field in the "Parties for ..." form):
  - IV – "Payer"
  - IB – "Payer Bank"
  - IC – "Payer Corresp. Bank"
  - PE – "Payee"
  - PB – "Payee Bank"
  - PC – "Payee Corresp. Bank"
  - BF – "Beneficiary"
  - UC – "Ultimate Creditor"
  - UD – Ultimate Debtor

These parameters are not mandatory in transaction notification message templates (in this case, payment party requisites are defined according to the document created as the result of the transaction). To specify a payment party role, use prefixes in the format INV\_PARTY\_<PartyCode>\$ (see "Variable Prefixes").

## %TRANS\_TAGS%

The %TRANS\_TAGS% variable can be used with the following parameters:

```
%TRANS_TAGS [ :<TAG_NAME>[:<Length>]]%
```

Where:

- <TAG\_NAME> is the name of the tag predefined for a document's *Add Data* field (see "Full → Documents Input & Update → Doc - General Form → Doc - General").
- The <Length> parameter is used if the received data needs to fit in a report; in this case, the width of the column where the data will be placed can be specified in this parameter.



The same configuration is used for the %DOC\_TAGS% variable.

## %TRANS\_ACC...%

%TRANS\_ACC...% variables (for example, %TRANS\_ACC\_CURR%, %TRANS\_ACC\_AMOUNT % and others) can be used with the following additional parameters:

```
%TRANS_ACC_CURR[:<ServiceClasses>[:<AccCodes>]]%
```

where

- <ServiceClasses> – transaction type code (or comma-delimited list of codes).
- <AccCodes> – account type code (or comma-delimited list of codes), corresponding to the value of the *Code* field in the "Account Types" form (Full → Configuration Setup → Accounting Setup → Account Types).

## Use of Masks

In the message template for the variables %CONTRACT\_NUMBER%, %TRANS\_SOURCE\_NUMBER%, %TRANS\_TARGET\_NUMBER% a "fill mask" can be specified in one of the following ways:

- %CONTRACT\_NUMBER:<N><X><M>[:K][:C]%, where:
  - <N> – number of visible digits at the beginning of the number.
  - <X> – filler symbol for the hidden part of the number.
  - <M> – number of visible digits at the end of the number.
  - <K> – number of filler symbols; this parameter is not mandatory.
  - <C> – contract category code; only contract numbers belonging to this category will be masked. By default, these are card contracts. The possible values are: "C" – card contracts; "B" – bank contracts; "A" – account contracts and "M" – device contracts; this is not a mandatory parameter, "C" is the default value.

For example, the actual card number is – 67999994599790985. If the variable format is %CONTRACT\_NUMBER:3\*5%, the resulting string will be 679\*\*\*\*\*90985. If the variable format is %CONTRACT\_NUMBER:4\*4:3%, the resulting string will be 6799\*\*\*0985.

- %CONTRACT\_NUMBER[:<MaskString>[:<X>[:<N>[:<M>]]]]%, where
  - <MaskString> is the template for a contract number. This is a string which may contain the <X> and <N> symbols, as well as symbols that divide the number into sections (any symbol can be used as a divider, except those indicated as <X> and <N>); the string may not contain a colon ":".
  - <X> is a filler symbol for the hidden part of the number; the symbols of the contract number that fall on positions marked with <X> will be substituted with the symbol <X>; this parameter is not mandatory and "\*" is used as the default symbol.

- `<N>` indicates positions where the number is open. This parameter is not mandatory and `"#"` is used as the default symbol.
- `<M>` is a contract category code; only contract numbers belonging to this category will be masked. By default, these are card contracts. The possible values are: `"C"` – card contracts; `"B"` – bank contracts; `"A"` – account contracts and `"M"` – device contracts; this is not a mandatory parameter, `"C"` is the default value.

Example 1. For instance, if the card number is 4015500184238081, and the message template indicates the following variable format: `CONTRACT_NUMBER:####-****-****-####%`, the string received after substitution is `4015-****-****-8081`.

Пример 2. For instance, if an account contract number is 40702810500013216954 and the variable format is specified as `%CONTRACT_NUMBER:????_$$$_??????_$$$$?:$.A%`, the string received after substitution is `????_810_??????_16954`.

## Variable Prefixes

To carry out various tasks, the system can work with special prefixes for some variables. The `"$"` (dollar) sign is used to separate the prefix and variable names.

To get information about a contract depending on an existing contract hierarchy, variables with the following prefixes can be used:

- `MAIN$` - makes it possible to determine the parameters of the main contract for this contract; if the current contract is not a subcontract, an empty value is returned.
- `MAIN_MAIN$` - makes it possible to determine the parameters of the top contract in the contract tree; if the current contract is not a subcontract, returns the current contract's parameters.
- `MAIN_SEE_MAIN$` - makes it possible to determine the parameters of the main account contract with consideration of the authorization scenario (set in the *Auth Scenario* field for the current contract).
- `LIAB` or `LIAB_MAIN$` - makes it possible to determine the parameters of the contract linked with the current contract in a Liability hierarchy.
- `LIAB_MAIN_MAIN$` - makes it possible to determine the parameters of the main contract in a Liability hierarchy.

The `INV_PARTY_<PartyCode>$` prefix determining the payment party role can be used to get payment party requisites.

- `INV_PARTY_IV$` – payer
- `INV_PARTY_IB$` – payer bank
- `INV_PARTY_IC$` – payer correspondent bank
- `NV_PARTY_PE$` – payment recipient
- `INV_PARTY_PB$` – payment recipient bank

- INV\_PARTY\_PC\$ – payment recipient correspondent bank
- INV\_PARTY\_BF\$ – beneficiary (if differs from payment recipient)
- INV\_PARTY\$ – "our" payment party
- INV\_PARTY\_BNK\$ – "our" bank
- INV\_PARTY\_CORR\$ – "our" correspondent bank
- OPPOSITE\_PARTY\$ – counterparty for "our" payment party
- OPPOSITE\_PARTY\_BNK\$ – counterparty bank
- OPPOSITE\_PARTY\_CORR\$ – counterparty correspondent bank.
- The DR\$ and CR\$ prefixes indicate debit and credit accounts respectively. They can be used for the following variables:
  - %ACC\_TEMPL\_GL\_NUMBER%
  - %ACC\_TEMPL\_HD\_GL\_NUMBER%
  - %ACC\_TEMPL\_ACCOUNT\_NAME%
  - %ACCOUNT\_NUMBER%
  - %ACCOUNT\_TYPE%
  - %ACCOUNT\_DUE\_TYPE%

For example, if the expression %DR\$ACC\_TEMPL\_GL\_NUMBER% is indicated in a template, the GL number of the debit account template will be received.

The following prefixes can be used for variables related to transaction information:

- The prefixes TRANS\_SRC\$ and TRANS\_TGT\$ are used to send information on the contract involved in a transaction (the source or the target respectively). Information about a client whose contract was involved in a transaction can be sent in the same way.

Using one of these prefixes will also help avoid the following situation. If the current Event opened as a result of a usage limiter, the variables associated with client and contract data (for example, %CONTRACT\_NUMBER%), are substituted with the field values of the contract for which the limiter was activated and not of the contract involved in the transaction. To include information in the message about the contract involved in the transaction and its client, use the prefixes TRANS\_SRC\$ and TRANS\_TGT\$.

For example, a limiter is set on an issuing account contract that has a card subcontract with Usage Scenario = "Main and Own". When this limiter is activated, an Event creating a message will open. To include the contract target number in the message, use the %TRANS\_TGT\$CONTRACT\_NUMBER% variable in the template.

- The TRANS\_SUMM\$ prefix is used if the current transaction is a separate batch message and the corresponding batch header information needs to be included in the message.

- The TRANS\_PREV\$ prefix is used to receive information on the previous document, for example, an original document for a reversal or a presentment for a chargeback.
- The TRANS\_ORIG\$ prefix is used to receive information on the current document or, if the current document is a reversal or a secondary document, information on the original document.
- The TRANS\_OPPOSITE\$ prefix is used to receive information about the counterparty contract based on document data.
- The MTR\_OPPOSITE\$ prefix is used to get information about the counterparty contract based on macrotransaction data.
- The STANDING\_ORDER\$ prefix is used to receive information about the payment order for which the document was created

## Variable Formatting

During message generation, values of numeric and text variables and date/time variables in a custom format can be used. To set up formats, use the "Formatting Styles" form (see Fig. 4), opened by selecting the "Full → Configuration Setup → Client Classifiers → Formatting Styles" menu item.

Formatting Type	Style Code	Language	Language Code2	Country Code2	Pattern	Replacement	Decimal Separator	Grouping Separator
Text								
Date/Time	YY				yy			
Number	My_Num1				#####0.00			
Date/Time	My_DT2				dd-MM-yyyy HH-mm-ss			
Date/Time	My_D1				dd-MM-yyyy			

Fig. 4. Formatting styles form

This form contains the following fields:

- *Formatting Type* – determines the variables for which this format is used. The field can have one of the following values:
  - "Number" – for numeric variables.
  - "Date/Time" – for date/time variables.
  - "Text" – used to transform the display format of text variable values (variables returning a text value) using regular expressions.
- *Style Code* – user-specified format code. If the same code is used for several formats, the system searches for the format necessary to generate client messages by the values of the *Language*, *Language Code2*, and *Country Code2* fields.
- *Language* – client language.
- *Language Code2* – two-symbol client language code used to additionally separate clients by language (e.g. to separate clients using American and British English).
- *Country Code2* – two-symbol client country code used to additionally separate clients by region (e.g. to separate clients in multilingual countries).

- *Pattern* – template for displaying variable values. Table 2 below shows examples of templates:



 When the value of the *Formatting Type* field is "Text", a regular expression is specified in this field that makes it possible to associate text with the specified template (for more information, see the description of the *Replacement* field).

Table 2. Examples of templates

Template used	Result
Numeric variable templates	
#####0.00	1.23
#####0.000	1.234
#####00.00	01.23
Numeric variable template when thousands must be delimited (the "thousand" delimiter position is marked by commas, the decimal delimiter by a decimal point).	
###,###,###,##0.00#	100,250.23
Date/time variable templates	
yyyy.MM.dd 'at' HH:mm:ss	2009.02.25 at 12:08:56
EEE, MMM dd, "yy, HH:mm a	Wed, Feb 25, '09, 12:08 PM
EEE, dd MMM yyyy hh 'o'clock' a	Wed, 25 Feb 2009 12 o'clock PM

 When thousands are delimited: if a space is used as the delimiter, it must be set in the *Grouping Separator* field as the combination "Alt+0160".

- *Replacement* – replacement expression (regular expression) for transforming the display of a variable value. The field is only accessible for records with the "Text" value in the *Formatting Type* field.

Table 3. Examples of variable transformation using regular expressions


No	Regular expression ( <i>Pattern</i> )	Replacement expression ( <i>Replacement</i> )	Result
1	(...)(.*)	xxx\$2	xxx34586975478
Example 1. The regular expression (...)(.*) indicates that the original string is regarded as two groups of characters. There are 3 characters in the first group (indicated as three periods) and the second group contains all remaining characters (indicated as a period and asterisk). The replacement expression xxx\$2 indicates that the first group of characters must be replaced by xxx characters and the second group (\$2 – pointer to the group number) remains without changes.			
2	(....)(.....)(.*)	\$1zzz\$3	3480zzz9368845





Example 2. The regular expression (....)(.....)(.\*) indicates that the original string is regarded as three groups of characters. The first group contains 4 characters (four periods), the second - 8 characters (eight periods), and the third – all remaining characters (indicated as a period and asterisk). The replacement expression \$1zzz\$3 indicates that the first and third groups of characters remain unchanged, and all characters in the second group must be replaced with zzz characters.

- *Decimal Separator* – symbol used as a decimal separator. This field is only used for numerical variable formats.
- *Grouping Separator* – symbol used to separate three-digit groups. This field is only used for numerical variable formats.

For variable values to be displayed in the necessary format, message templates must contain an expression %<Variable>\*<Format\_Code>% instead of a variable name %<Variable>%, where <Format\_Code> is the value of the *Style Code* field in the "Formatting Styles" form. For example, the expression "%TRANS\_AMOUNT\*NUMBER\_STYLE\_CODE%" can be used for transaction amounts, and the expression "%TRANS\_DATE\*DATE\_STYLE\_CODE%" can be used for transaction dates.

 If other additional parameters are used, the formatting style code must be specified immediately after the name of the variable before the first colon separating the parameters. For example, "%ACCOUNT\_BALANCE[\*<FormatCode>][:<AccCodeList>].

 If the "default" value is specified in the *Style Code* field of the "Formatting Styles" form, this style is used as the default style if the formatting style is not specified in the message template. The default style is set for each value of the *Formatting Type* field.

 If a format code (<FormatCode>) is not specified for a variable date, or is not specified explicitly (for example %TRANS\_DATE\*YYYY-MM-DD%), a template with the "default" value in the *Style Code* field of the "Formatting Styles" form is used. If a default date template is not specified, the format "YYYY-MM-DD HH24:MI:SS" is used for the parameter value.

## Shift in Date Variable Values

When a message is created it is possible to shift the values of date variables. To specify a date with a "shift", use the expression

%<DATE\_TAG\_NAME>[\*<+/-><number><D/M>]][\*<FormatCode>]%,  
where:

- <DATE\_TAG\_NAME> is the name of the date variable.
- <+/-> when the "+" (" -") sign is specified, the given date will be increased (decreased).
- <number> is the value by which the date will be increased (decreased).
- <D/M> – unit of measurement, specified by the parameter < number > (days / months).

- <FormatCode> date display format.

Examples:

1. The expression `%SYSDATE*+30*YYYY-MM-DD%` inserts the date in the message (system date + 30 days) in the format YYYY-MM-DD.
2. The expression `%SYSDATE*+30D*YYYY-MM-DD%` inserts the date in the message (system date + 30 days) in the format YYYY-MM-DD.
3. The expression `%SYSDATE*+30M*YYYY-MM-DD%` inserts the date in the message (system date + 30 months) in the format YYYY-MM-DD.
4. The expression `%SYSDATE*-30M*YYYY-MM-DD%` inserts the date in the message (system date - 30 months) in the format YYYY-MM-DD.



As the date format, any format can be used configured in formatting styles according to general rules (see the section "Variable Formatting").

## Using Modifiers for String Values of Variables

Additional modifiers may be used to control the length of a variable's string value:

- `SUBSTR*<FirstPos>[*<Length>]` – extracts a substring beginning from <FirstPos> position with <Length> characters.
- `FIRST*<Length>` – extracts a substring from the beginning of a string with <Length> characters.
- `LAST*<Length>` – extracts <Length> last characters.
- `RPAD*<Length>[*<PadChar>]` – pads a string to the right with <PadChar> characters up to <Length>.
- `LPAD*<Length>[*<PadChar>]` – pads a string to the left with <PadChar> characters up to <Length>.

For example the variable:

```
%ADDR_ADD_INFO*FIRST*10%
```

returns the first 10 characters from the ADDR\_ADD\_INFO value.

## Using Variable Modifiers for Sending Messages in XML Format

In a number of cases (for example to send a client SMS/e-mail notifications), generated messages are converted to XML format. Therefore, when using variables in the templates of such messages, the specifics of XML syntax must be considered.

It is recommended to use a special XMLESC modifier to escape XML control characters in a variable value.

```
%ADDR_LINE_1*XMLESC%
```

XMLESC escapes the following characters:

- & (ampersand) - &amp
- " (straight quotation mark) - &quot

- ' (apostrophe) - &apos
- < ("less than" sign) - &lt
- > ("greater than" sign) - &gt

An XMLESC modifier can be used together with modifiers controlling the length of a variable's string value. For example, a variable in the form

```
%ADDR_LINE_1*SUBSTR*0*10*XMLESC%
```

returns 10 characters from the value of ADDR\_LINE\_1 with the correct XML character escape.

## Using Conditional Operators

When generating a client message template, operators such as %IF%, %THEN%, %ELSE%, etc. can be used. Operators can be embedded.

Variables can be used to set conditions.

Example.

```
%IF%<Condition 1>
%THEN%<Message №1>
%ELSIF%<Condition 2>
%THEN%<Message №2>
%ELSE%<Message №3>%END%
```

The following operators can be used to compare variable values with values set in a condition:

- "=" – equal to
- "!=" – not equal to
- ">" – greater than
- "<" – less than
- ">=" – greater than or equal to
- "<=" – less than or equal to

Logical variables returning Boolean values ("Y" – True; "N" – False) can also be used as conditions:

- %IS\_EVENT\_OPENED:<Event Code>:<on Date>% – determines whether an Event with the specific code is opened.
- %IS\_ADD\_PACK\_ACTIVE:<Serv\_Pack\_Code>% – determines whether an additional Service Package with the specific code is attached.

For example:

```
%IF%%CONTRACT_BALANCE%>=0 AND %IS_EVENT_OPENED:BAL_EV%
%THEN%Your event_is opened. Current balance is %CONTRACT_BALANCE%
%CONTRACT_CURR%
%ELSE%Your current loan is %CONTRACT_BALANCE% %CONTRACT_CURR%%END%
```

Message text can be generated depending on document tag values (the document's *Add Data* field), for example, as follows:

```
%IF%%DOC_TAGS:PTID%=C55
%THEN%Text about C55
%ELSIF%%DOC_TAGS:PTID%=C56
%THEN%Text about C56
%ELSE%Other text%END%
```

In some cases, it may be necessary to limit sending a message according to a specific condition. In some system settings, it is possible to generate an empty message (no text for a certain %IF% condition) that will not be sent. For example:

```
%IF%%CONTRACT_BALANCE%>=0
%THEN%
%END%
```



Care must be taken when using conditional structures in message templates, weighing necessity against the complexity of their use due to possible additional computational load on WAY4.

## Chapter 4. Tags in the *Message Details* Field of a Message Template

Name	Value	Description
Tags in the <i>Message Details</i> field of a message template:		
IF_INACTIVE_EVNT	<EventCode>	A message will only be generated if the Event specified in the tag (with the specified code) is closed.
IF_ACTIVE_EVNT	<EventCode>	A message will only be generated if the Event specified in the tag (with the specified code) is opened.
REGISTERED_ONLY		A message will only be generated if the Event type for which this message is generated is registered in the contract's Service Package.
USE_DELAY_TIME		The tag makes it possible to generate a message according to an Event that occurred outside the interval set in the message template's <i>Sending Time From</i> and <i>Sending Time To</i> fields. The generated message will be sent to the client in the next <i>Sending Time From</i> and <i>Sending Time To</i> interval.
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.C ODE N>	This tag allows a message to be sent 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>	This tag makes it possible to not send (and not generate) a message 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.
AFF_TYPE	<relation type code>	This tag is used to send copies of messages to related clients. The relation type code from the handbook "Full\Configuration Setup\Client Classifiers\Linked Client Types" is used as the AFF_TYPE=<relation type code>; tag value. If this tag is set, when generating a message a search will be made for clients related in this way with the contract or client for which the message is being generated. If for the related clients found addresses with the type specified in the message template are configured, they will be sent a copy of the generated message.
ADD_ADDR_TYPES	<address type code1>,<address type code2>,...,<address type codeN>	Allows a list of additional address types to be specified for sending a copy of the message ADD_ADDR_TYPES=<AddrTypeCode1>,<AddrTypeCode2>,...,<AddrTypeCodeN>;
MAX_EVNT_MESSAGE_DELAY	<hh>	Sets the maximum time (in hours) for attempts to send a message. After this time expires, no more attempts are made to send the message.
MAX_SEND_TRY	<number of tries>	Sets the maximum number of attempts to send a message.
ADDRESS_DATA	<variable>	The value of this tag is used to generate message target address data. For example, in a message that will be sent to a mobile phone as an SMS, specify ADDRESS_DATA=%ADDR_PHONE_M%. In this case, the phone number is taken from an additional address registered for this contract or client.
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.

Name	Value	Description
		<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). 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&lt;N&gt;, IF_PARM_VALUE&lt;N&gt;, IF_PARM_FOR&lt;N&gt; 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>A message will be sent only if clients and their contracts correspond to the set classifier value. Used together with the IF_PARM tag. The IF_PARM_VALUE=NOT_EMPTY; tag is used to check for the existence of a parameter (the tag set using the IF_PARM tag) without checking its value. When the IF_PARM_VALUE=EMPTY; tag is set, the message is sent if the specified parameter doesn't have a value. Note that with this value, a message will also be sent if the parameter is not present in the contract.</p> <p>When "IF_PARM_VALUE=LIST_WITH:&lt;list of values separated by commas&gt;" is set, the message is sent if even one value from the list (LIST_WITH) matches that set in the contract.</p> <p>When "IF_PARM_VALUE=BETWEEN&lt;value&gt; AND &lt;value&gt;" the message is sent if the value in the contract falls in the range set by the tag (for example, IF_PARM_VALUE=BETWEEN40 AND 50;). The value must be set with spaces, as shown in the example. It is possible to set a check of several parameters for a contract by using the IF_PARM_NUMB, IF_PARM&lt;N&gt;, IF_PARM_VALUE&lt;N&gt;, IF_PARM_FOR&lt;N&gt; tags. See the description of the IF_PARM_NUMB tag.</p>
ZERO_BAL_TYPE	<balance type code>	A message is only generated 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>	A message is only generated 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.
SKIP_ADDR_CHCK		The tag disables the necessity to check address parameters set in a message template. A message will be generated even if the value of the <i>Address Data</i> field is not set or no address of this type (Address Type) field is found for the client. Usually, this setting is used to send messages to linked clients.

Name	Value	Description
SENSITIVE	%<Variable1>%, %<Variable2>%, ..., %<VariableN>%	This tag determines the list of variables whose values must be hidden (replaced with the "*" character (asterisk)) in the text of a message stored in the database after: <ul style="list-style-type: none"> <li>the message has been sent successfully.</li> <li>the maximum number of attempts to send the message has been reached.</li> <li>the time permitted for sending the message has expired.</li> </ul>
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"> <li>Document tag containing the currency code.</li> <li>TRANS_CURR – transaction currency from the document.</li> <li>SETTL_CURR – settlement currency from the document.</li> <li>RECONS_CURR – reconciliation currency from the document (in the currency in which the source bank provides transaction information to the payment system).</li> <li>CONTRACT – contract currency.</li> <li>Tag or contract custom parameter with a currency code.</li> </ul> A contract's custom parameter can be registered as a "Primary" classifier. <ul style="list-style-type: none"> <li>Code of the classifier with a currency code.</li> <li>Explicit specification of currency – the tag value may be a code or the name of a certain currency.</li> </ul> Used together with the IF_CURRENCY_VALUE, IF_CURRENCY_RULE, IF_CURRENCY_FOR tags.
IF_CURRENCY_VALUE	EMPTY NOT_EMPTY USE_IN_BANK <currency code1>, <currency code2>, ..., <currency codeN> <currency name1>, <currency name2>, ..., <currency nameN> <tag name>	A message will only be sent if the document currency specified using the IF_CURRENCY tag corresponds to parameters set in the IF_CURRENCY_VALUE tag. Possible tag values: <ul style="list-style-type: none"> <li>EMPTY – this currency is not specified.</li> <li>NOT_EMPTY – this currency is specified</li> <li>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).</li> <li>A code or name of a certain currency or several currencies, separated by commas can be used as the tag value.</li> <li>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.</li> </ul>



Name	Value	Description
IF_CURRENCY_RULE		<p>Used to set up additional conditions for checking currency together with the IF_CURRENCY_VALUE tag:</p> <ul style="list-style-type: none"> <li>• 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=&lt;list of account codes separated by commas&gt;. 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.</li> <li>• 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.</li> <li>• 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.</li> </ul>
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"> <li>• "BILLING" – for the account contract from which settlement is made.</li> <li>• "LIABILITY" – for a higher-ranking contract in a "Liability" hierarchy.</li> <li>• "TOP" – for the top contract in a hierarchy.</li> <li>• "BASE" – for the main contract in a "Main/Sub" hierarchy, with which this contract is related.</li> <li>• "DOC_SOURCE" – for the contract specified in the document's Source Contract (source_contract) field (see the "Doc-Brief" form).</li> <li>• "DOC_TARGET" – for the contract specified in the document's Target Contract (target_contract) field (see the "Doc-Brief" form).</li> <li>• "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).</li> <li>• "CONTRACT_ROLE" – for this value, specify the CONTRACT_ROLE=&lt;role code&gt; 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=&lt;role code&gt;; 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</li> </ul>

Name	Value	Description
		<p>top contract in the hierarchy will be selected.</p> <ul style="list-style-type: none"> <li>• 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=&lt;tag name&gt;; 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</li> <li>• "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=&lt;category1&gt;,&lt;category2&gt; 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.</li> </ul> <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&lt;N&gt;, IF_PARM_VALUE&lt;N&gt;, IF_PARM_FOR&lt;N&gt; tags. See the description of the IF_PARM_NUMB tag.</p>
IF_CS_TYPE_FOR	"BILLING" "LIABILITY" "TOP" "BASE" "DOC_SOURCE" "DOC_TARGET" "FROM_DOC" "CONTRACT_ROLE" CONTRACT_ROLE_PARM=<ta	<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"> <li>• "BILLING" – for the account contract from which settlement is made.</li> <li>• "LIABILITY" – from a higher-ranking contract in a "Liability" hierarchy.</li> <li>• "TOP" – from the top contract in a hierarchy.</li> <li>• "BASE" – from the main contract in a "Main/Sub" hierarchy, with which this contract is related.</li> <li>• "DOC_SOURCE" – for the contract specified in the document's <i>Source Contract</i> (source_contract) field (see the "Doc-Brief" form).</li> </ul>

Name	Value	Description
	g "LIAB_CATEGORY" name>	<ul style="list-style-type: none"> <li>• "DOC_TARGET" – for the contract specified in the document's <i>Target Contract</i> (target_contract) field (see the "Doc-Brief" form).</li> <li>• "FROM_DOC" – the contract is taken from the <i>Add Data</i> (add_info) field.</li> <li>• "CONTRACT_ROLE" – for this value, specify the CONTRACT_ROLE=&lt;role code&gt; 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=&lt;role code&gt;; 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.</li> <li>• 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=&lt;tag name&gt;; 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</li> <li>• "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=&lt;category1&gt;,&lt;category2&gt; 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&lt;N&gt;, IF_CS_VALUE&lt;N&gt;, IF_NOT_CS_VALUE&lt;N&gt;,</li> </ul>

Name	Value	Description
		IF_CS_TYPE_FOR<N>. See the description of the IF_CS_NUMB tag.
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.</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>• "BILLING" – for the account contract from which settlement is made.</li> <li>• "LIABILITY" – for a higher-ranking contract in a "Liability" hierarchy.</li> <li>• "TOP" – for the top contract in a hierarchy.</li> <li>• "BASE" – for the main contract in a "Main/Sub" hierarchy, with which this contract is related.</li> <li>• "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).</li> <li>• "DOC_TARGET" – for the contract in the <i>Target Contract</i> field (target_contract) of the document (see the "Doc-Brief" form).</li> <li>• "FROM_SERVICE" – for the contract specified in the Service (fee_contract, fee_account fields).</li> <li>• "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)</li> <li>• "RELATED" – related contract, with the relation specified in the RELATION tag.</li> <li>• "CONTRACT_ROLE" – for this value, specify the CONTRACT_ROLE=&lt;role code&gt; 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=&lt;role code&gt;; 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.</li> <li>• 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=&lt;tag name&gt;; 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</li> </ul>

Name	Value	Description
		<p>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"> <li>• "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=&lt;category1&gt;,&lt;category2&gt; tag.</li> </ul> <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_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"> <li>• "BILLING" – for the account contract from which settlement is made.</li> <li>• "LIABILITY" – for a higher-ranking contract in a "Liability" hierarchy.</li> <li>• "TOP" – for the top contract in a hierarchy.</li> <li>• "BASE" – for the main contract in a "Main/Sub" hierarchy, with which this contract is related.</li> <li>• "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).</li> <li>• "DOC_TARGET" – for the contract in the <i>Target Contract</i> field (target_contract) of the document (see the "Doc-Brief" form).</li> <li>• "FROM_SERVICE" – for the contract specified in the Service (fee_contract, fee_account fields).</li> <li>• "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)</li> <li>• "RELATED" – related contract, with the relation specified in the RELATION tag.</li> <li>• "CONTRACT_ROLE" – for this value, specify the CONTRACT_ROLE=&lt;role code&gt; tag in this field; for example:</li> </ul>

Name	Value	Description
		<p>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=&lt;role code&gt;; 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"> <li>• 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=&lt;tag name&gt;; 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</li> <li>• "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=&lt;category1&gt;,&lt;category2&gt; 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.</li> </ul> <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>

Name	Value	Description
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"> <li>• "BILLING" – for the account contract from which settlement is made.</li> <li>• "LIABILITY" – for a higher-ranking contract in a "Liability" hierarchy.</li> <li>• "TOP" – for the top contract in a hierarchy.</li> <li>• "BASE" – for the main contract in a "Main/Sub" hierarchy, with which this contract is related.</li> <li>• "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).</li> <li>• "DOC_TARGET" – for the contract in the <i>Target Contract</i> field (target_contract) of the document (see the "Doc-Brief" form).</li> <li>• "FROM_SERVICE" – for the contract specified in the Service (fee_contract, fee_account fields).</li> <li>• "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)</li> <li>• "RELATED" – related contract, with the relation specified in the RELATION tag.</li> <li>• "CONTRACT_ROLE" – for this value, specify the CONTRACT_ROLE=&lt;role code&gt; 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=&lt;role code&gt;; 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.</li> <li>• 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=&lt;tag name&gt;; 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</li> <li>• "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=&lt;category1&gt;,&lt;category2&gt; tag. Liability category codes, separated by commas, are specified as the LIAB_CATEGORY tag value: "Y" – "Full Liability" category</li> </ul>

Name	Value	Description
		<p>"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&lt;N&gt;, IF_CS_VALUE&lt;N&gt;, IF_NOT_CS_VALUE&lt;N&gt;,            IF_CS_TYPE_FOR&lt;N&gt; tags. I.e. by default the IF_CS_TYPE tag is checked. If the            IF_CS_NUMB=&lt;number of additional checks&gt;; tag is set, the specified number of            IF_CS_TYPE&lt;N&gt; 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&lt;N&gt;, IF_PARM_VALUE&lt;N&gt;, IF_PARM_FOR&lt;N&gt; tags. I.e. by            default the IF_PARM tag is checked. If the IF_PARM_NUMB=&lt;number of additional            checks&gt;; tag is set, the specified number of IF_PARM&lt;N&gt; 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>