Installation and Configuration Manual

Configuration of Client Messages

03.50.30

12.11.2019

Contents

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"
- "Setup Tags".

The following notation is used in this document:

- 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.

ⓘ Information about important features, additional options or the best use of certain system functions.

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").

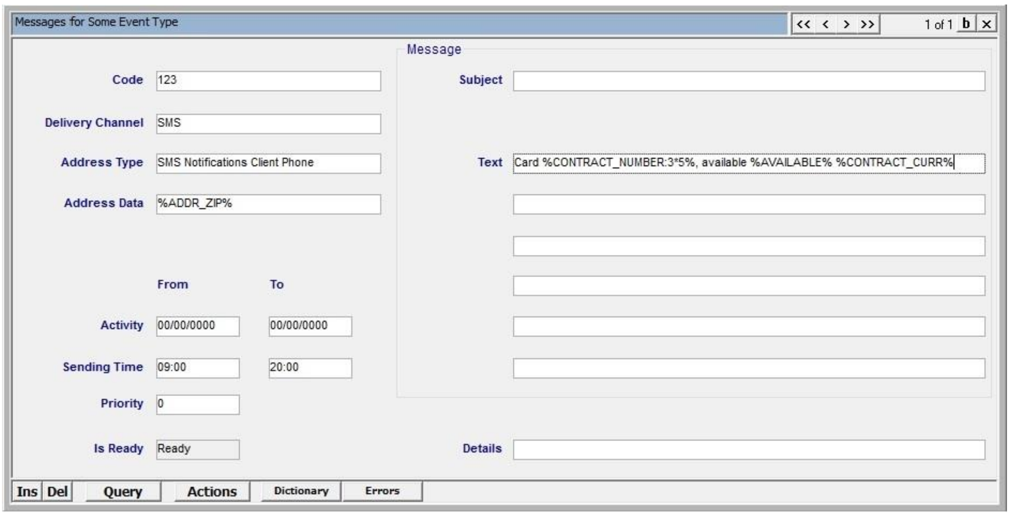
ⓘ Templates for both message types are configured in the same way. For more information on how message templates are set up, see the section "Configuring Message Templates".

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.

2.1 Template Configuration Form

The "Messages for ‹Event type name›" form, used to configure message templates is shownin Fig. 1.



Messages for Some Event Type

1 of 1

Message

Code	123
Delivery Channel	SMS
Address Type	SMS Notifications Client Phone
Address Data	%ADDR_ZIP%

	From	To
Activity	00/00/0000	00/00/0000
Sending Time	09:00	20:00
Priority	0	
Is Ready	Ready	

Subject	
Text	Card %CONTRACT_NUMBER:3*5%, available %AVAILABLE% %CONTRACT_CURR%
Details	

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 as follows:

  - By the *Time Zone* field’s value of the "Branches" form (Full → Configuration Setup → Main Tables → Branches) for the branch that is specified in the client contract or (if data is absent) in client details.
  - If the time zone of the branch was not found, it is determined for a financial institution that is specfiied in the client contract according to the MSG_TIME_ZONE tag’s value in the *Special Parms* field of the "Financial Institutions" form (Full → Configuration Setup → Main Tables → Financial Institutions) or (if the tag is absent) by the *Time Zone* field’s value of the correspondng financial institution.

> ⓘ A shift from the time zone to the west is specified with a positive value, and a shift to the east is specified with a negative value.

- *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 used when processing client messages" of the document "Setup tags".
- *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.

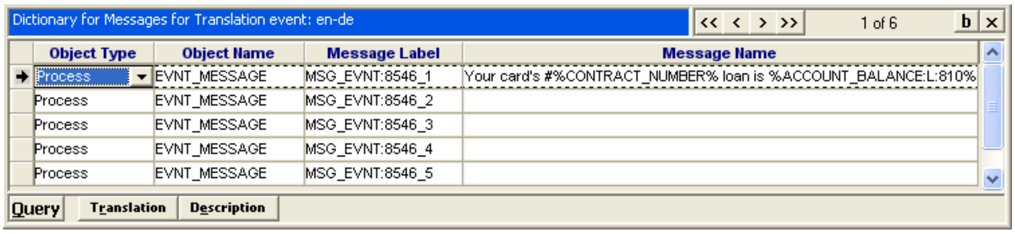
It is necessary to use [Actions] →[To Dictionary] for correct saving results of editing template with message translation. Otherwise, the changes will not be saved in the system list Message Dictionary (see the section "Translating Messages into Local Languages").

2.2 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).



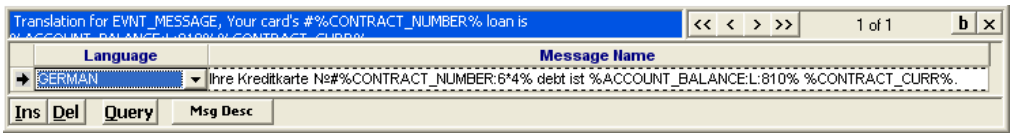
Dictionary for Messages for Translation event: en-de

1 of 6

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).



Translation for EVNT_MESSAGE, Your card's #%CONTRACT_NUMBER% loan is

1 of 1

Language	Message Name
GERMAN	Ihre Kreditkarte №#%CONTRACT_NUMBER:6*4% 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.

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.

3.1 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	
%CONTRACT_NUMBER%	Contract number
%CONTRACT_NAME%	Contract name
%MAIN_CONTRACT_NUMBER%	Parent contract number. If the current contract is not a subcontract of another contract, this in an empty value.
%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.
%TR_FIRST_NAM%	Data for embossing
%TR_LAST_NAM%	
%TR_COMPANY%	
%CONTRACT_F_I%	Contract financial institution
%RBS_NUMBER%	Contract RBS number
%RBS_MEMBER_ID%	Payment system member ID
%CON_CAT%	Contract category
%PCAT%	Product category
%CONTRACT_TYPE%	Contract type
%CONTRACT_SUBTYPE%	Contract subtype
%ACCOUNT_SCHEME%	Name of Accounting Scheme
%SERVICE_PACK%	Name of Service Package
%BEHAVIOR_TYPE%	Contract behavior type identifier (see the section "%BEHAVIOR_TYPE%")
%BEHAVIOR_TYPE_CODE%	Contract behavior type code (see the section "%BEHAVIOR_TYPE%")
%CONTRACT_CURR%	Contract currency

Name	Value
%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
%CREDIT_LIMIT%	Credit limit value
%AUTH_LIMIT_AMOUNT%	Same as %CREDIT_LIMIT%
%CONTRACT_DATE_OPEN%	Contract opening date
%CONTRACT_STATUS%	Contract status
%CARD_EXPIRE%	Month and year of card expiry
%ACNT_ADD_INFO_01%	Additional contract information (see the section "%CLNT_ADD_INFO...%").
%ACNT_ADD_INFO_02%	
%ACNT_ADD_INFO_03%	
%ACNT_ADD_INFO_04%	
%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.
%RISK_SCHEME%	Contract risk scheme
%IPS_PRODUCT%	Name of the card product in the payment system (value of the *IPS Product* field of the contract subtype)
%PRODUCT_CODE%	Contract Product code (*Code* field of the "Products' form)
%PRODUCT_CODE_2%	Additional contract Product code (*Code 2* field of the "Products" form).
%PRODUCT_CODE_3%	Additional contract Product code (*Code 3* field of the "Products" form).
%CLASSIFIER%	Value of contract or client classifiers, depending on a specific code (defined by the mandatory parameter <Classifier Code>)
%DECISION%	Value of a calculated classifier, depending on a specific code (defined by the mandatory parameter <DecisionCode>)
%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

Name	Value
%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	
%TOTAL_BLOCKED%	Total amount of blocked funds
%OWN_BLOCKED%	Amount of funds blocked in the current contract
%SUB_BLOCKED%	Amount of funds blocked in subcontracts
%TOTAL_BALANCE%	Total balance
%OWN_BALANCE%	Total balance of the current contract
%SUB_BALANCE%	Balance of subordinate contracts
%BALANCE%	Contract balance (same as %TOTAL_BALANCE%)(
%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%").
%ACNT_BALANCE%	Same as %CONTRACT_BALANCE%
%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.
%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.
%ACC_TEMPL_GL_TYPE%	Value of the *GL Type* 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...%")
%ACC_TEMPL_GL_NUMBER%	Value of the *GL #* 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...%")
%ACC_TEMPL_HD_GL_NUMBER%	Value of the *HeadOffice GL Number* 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...%")
%ACC_TEMPL_ACCOUNT_NAME%	Value of the *Account Name* 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...%")
%ACC_TEMPL_INTEREST_ALGORITHM%	Value of the *Interest Algorithm* 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...%")
%ACC_TEMPL_USE_GL%	Value of the *Aggregate GL For* 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...%")
%ACCOUNT_NUMBER%	Number of the account whose type is set as the value of an additional parameter (see the section "%ACC_TEMPL...%")

Name	Value
%ACCOUNT_TYPE	Name of the type of account whose code is set as the value of an additional parameter (see the section "%ACC_TEMPL...%")
%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...%")
%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%")
%ACCOUNT_DUE_DATE%	Payment due date (see the section %ACCOUNT_DUE_DATE%, %ACCOUNT_DUE_AMOUNT%)
%ACCOUNT_DUE_AMOUNT%	Amount of payment due (see the section %ACCOUNT_DUE_DATE%, %ACCOUNT_DUE_AMOUNT%)
%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%")
%ACCOUNT_INTEREST_RATE%	Value of annual interest rate (see the section "%ACCOUNT_BALANCE%")
%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%")
%ACCOUNT_TURNOVER%	Contract account turnover (see the section "%ACCOUNT_BALANCE%")
%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%")
%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...%")
%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%")
%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%")
%DUE_AMOUNT%	Amount of funds on "Payment Due" category accounts at the closing of the previous billing cycle.

Name	**Value**
%IMMEDIATE_AMOUNT%	Amount of funds on "Pay Immediate" category accounts at the closing of the previous billing cycle
%BILLING_DATE%	Closing date of the current billing cycle
%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")
%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")
%DUE_DATE%	Date of the next due normalisation calculated at the closing of the previous billing cycle
Information about contract balances opened for the multi-currency Product	
%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).
%MULTI_CURRENCY_CURR:<Currency>%	Letter currency code, corresponding to the numeric code set as the value of the <Currency> parameter.
%MULTI_CURRENCY_BALANCE_ALL%	List of all currencies and total balances from a client's account contracts.
%MULTI_CURRENCY_BALANCE_ NONZERO%	List of all currencies and non-zero balance amounts from a client's account contracts.
%MULTI_CURRENCY_AVAILABLE: <Currency>%	Amount available for a contract in the specified currency (<Currency> parameter). If a contract in this currency is not found, the amount available is determined for the original contract with conversion at the middle rate (FX Middle).
%MULTI_CURRENCY_AVAILABLE_ALL%	List of currency code-amount available pairs from the client's account contracts. Values within pairs are separated by spaces, and pairs in the list are separated by commas; for example: RUR 3567, USD 567, EUR 1673.
%MULTI_CURRENCY_AVAILABLE_ NONZERO%	List of currency code-non-zero amount available pairs from the client's account contracts. Values within pairs are separated by spaces, and pairs in the list are separated by commas; for example: RUR 3567, USD 567, EUR 1673.
Information about Services. Data is taken from the Service or tariff (if a tariff is set in the Service)	
%SERVICE$TRANS_TYPE%	Name of transaction type
%SERVICE$FEE_MIN_AMOUNT%	Minimum size of the fee that will be charged regardless of the transaction amount.
%SERVICE$FEE_MAX_AMOUNT%	Maximum amount of the fee that can be charged for a transaction.

Name	**Value**
%SERVICE$FEE_BASE_AMOUNT%	Additional mandatory fee. Does not depend on interest rate.
%SERVICE$FEE_PERCENT%	Fee percent.
Information about tariffs	
%TARIFF_TYPE%	Tariff form's *Tariff Type* field value.
%TARIFF_DOMAIN%	Tariff form's *Tariff Domain* field value.
%TARIFF_CODE_EXT%	Tariff form's *Tariff Code Ext* (TABLE_CODE_FROM) field value.
Client information (values of fields from the "Client – Edit for <client name>" form)	
%CCAT%	*Client Category* field value
%CLIENT_TYPE%	*Client Type* field value
%CLIENT_SHORT_NAME%	*Short Name* field value
%CLIENT_NUMBER%	*Reference #* field value
%CLIENT_REG_NUMBER%	*Registration #* field value
%CLIENT_REG_DETAILS%	*Registration Dtls* field value
%SALUTATION%	Value of the first Salutation field
%SALUTATION_SUFFIX%	Value of the second *Salutation* field
%TITLE%	Same as %SALUTATION%
%CLIENT_LAST_NAME%	*Last Name* field value
%CLIENT_FIRST_NAME%	*First Name* field value
%CLIENT_BIRTH_NAME%	*Birth Name* field value
%CLIENT_FATHER_S_NAME%	*Middle Name* field value
%CLIENT_BIRTH_DATE%	*Date of Birth* field value
%CLIENT_BIRTH_PLACE%	*Place of Birth* field value
%CLIENT_LANGUAGE%	*Language* field value
%COMPANY%	*Company Name* field value
%ITN%	*Individual Number* field value
%CLIENT_SALUTATION%	Same as %SALUTATION%
%CLIENT_F_I%	*Institution* field value
%CLIENT_BRANCH%	*Branch* field value
%GENDER%	*Gender* field value
%CLNT_ADD_INFO_01%	Additional client information (see the section "%CLNT_ADD_INFO...%")
%CLNT_ADD_INFO_02%	
%CLNT_ADD_INFO_03%	
%CLNT_ADD_INFO_04%	
%PHONE%	*Phone* field value
%PHONE_H%	*Phone (home)* field value
%PHONE_M%	*Mobile* field value
Contract/client address information (values of fields from the "Addresses for <client name/contract number>" form), see "%ADDR...%"	
%ADDR_TYPE%	*Type* field value
%ADDR_NAME%	*Name* field value
%ADDR_DELIVERY_TYPE%	*Delivery* field value

Name	Value
%ADDR_LANGUAGE%	*Language* field value
%ADDR_SALUTATION%	*Salutation* field value
%ADDR_SALUTATION_SUFFIX%	*Suffix* field value
%ADDR_FIRST_NAME%	*First Name* field value
%ADDR_LAST_NAME%	*Last Name* field value
%ADDR_BIRTH_NAME%	*Birth Name* field value
%COUNTRY%	*Country* field value
%COUNTRY_POST%	*Postal Code* field value in the "Full → Configuration Setup → Main Tables → Country Table" table for the country corresponding to the document's *Country* field
%ADDR_STATE%	*State* field value
%CITY%	*City* field value
%ADDR_ZIP%	*ZIP* field value
%ADDR_MUNICIPALITY_CODE%	Value of the *Municipality* field
%ADDR_PHONE%	*Phone* field value
%ADDR_PHONE_H%	*Phone (h) field value*
%ADDR_PHONE_M%	*Phone (mob)* field value
%ADDR_FAX%	*Fax* field value
%ADDR_FAX_H%	*Fax (h)* field value
%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
%ADDR_LINE_2%	
%ADDR_LINE_3%	
%ADDR_LINE_4%	
%ADDR_ADD_INFO%	*Add Info* field value
%E-MAIL%	*E-mail* field value
%ADDR_URL%	*URL* field value
%ADDR_TITLE%	Same as %ADDR_SALUTATION%
%TELEPHONE_LIST[:<AddrTypeCode>[:<LanguageCode>]]%	List of client phone numbers (main and additional) for the specified address type (parameters are described in the section "%ADDR...%").
%EMAIL_LIST[:<AddrTypeCode>[:<LanguageCode>]]%	List of client e-mail addresses (main and additional) for the specified address type (parameters are described in the section "%ADDR...%").
Information about Events and usage limiters	
%IS_EVENT_OPENED%	The variable returns one of the following values: 'Y' – if an Event with a specified code has one of the following statuses on a specified date: "Posted", "Suspended", "Closed", "Inactive"; 'N' – in all other cases. The following are specified as mandatory parameters for the variable: <Event Code> – Event code; <on Date> – date on which the Event status is determined

Name	Value
%ACTIVE_EVENT_START_DATE%	Opening date of an Event with a specified code Additional parameters are the same as %IS_EVENT_OPENED%
%ACTIVE_EVENT_END_DATE%	Closing date of an Event with a specified code. Additional parameters are the same as %IS_EVENT_OPENED%
%USG_LIM...%	%USG_LIM...% variables allow information on contract usage limiters to be shown (see the section "%USG_LIM...%")
Payment party requisites (values of fields from the "Parties for Doc- General" form, see the section "%PARTY...%".	
%PARTY_AFFILIATION_IDT%	*Affiliation Source Number* field value
%PARTY_AFFILIATION_DATE_FROM%	*Affiliation Date From* field value
%PARTY_AFFILIATION_DATE_TO%	*Affiliation Date To* field value
%PARTY_COUNTRY%	*Country* field value
%PARTY_COUNTRY_NAME%	*Country Name* field value
%PARTY_POSTAL_CODE%	*Postal Code* field value
%PARTY_REGION%	*Region* field value
%PARTY_CITY%	*City* field value
%PARTY_CITY_EN%	*City English* field value
%PARTY_STREET%	*Street* field value
%PARTY_ADDRESS_LOCATION%	*Location* field value
%PARTY_ADDRESS_LINE_1%	*Address Line 1* field value
%PARTY_ADDRESS_LINE_2%	*Address Line 2* field value
%PARTY_ADDRESS_LINE_3%	*Address Line 3* field value
%PARTY_ADDRESS_LINE_4%	*Address Line 4* field value
%PARTY_ADDRESS_LINE_EN_1%	*Address Line En 1* field value
%PARTY_ADDRESS_LINE_EN_2%	*Address Line En 1* field value
%PARTY_ADDRESS_LINE_EN_3%	*Address Line En 1* field value
%PARTY_ADDRESS_LINE_EN_4%	*Address Line En 1* field value
%PARTY_MUNICIPALITY_CODE%	*Municipality Code* field value
%PARTY_ADDRESS_ADD_INFO%	*Add Info* field value
%PARTY_CONTACT_URL%	*URL* field value
%PARTY_CONTACT_E_MAIL%	*E-Mail* field value
%PARTY_CONTACT_PHONE%	*Phone* field value
%PARTY_CONTACT_FUNCTION%	*Function* field value
%PARTY_ACC_NUMBER%	*Account Number* field value
%PARTY_ACC_MEMBER_ID%	*Account Member Id* field value
%PARTY_ACC_CURR%	*Account Currency* field value
%PARTY_ACC_DATE_OPEN%	*Account Date Open* field value
%PARTY_ACC_BRANCH%	*Account Branch Code* field value
%PARTY_TYPE%	*Party Type* field value
%PARTY_CLIENT%	*Party Client* field value
%PARTY_CCAT%	*Client Category* field value
%PARTY_RESIDENCE%	*Residence* field value

Name	**Value**
%PARTY_NAME%	*Party Name* field value
%PARTY_NAME_EN%	*Party Name English* field value
%PARTY_BIRTH_DATE%	*Birth-Reg Date* field value
%PARTY_REG_NUMBER_TYPE%	*Reg Number Type* field value
%PARTY_REG_NUMBER%	*Reg Number* field value
%PARTY_REG_DETAILS%	*Reg Details* field value
%PARTY_ITN%	*Individual Tax Number* field value
%PARTY_TAX_POSITION%	*Tax Position* field value
%PARTY_AFFILIATION_CODE%	*Affiliation Code* field value
Information about transactions (values of fields from the "All Docs" or "Doc-General" form)	
%TRANS_AMOUNT%	*Trans Amount* field value
%TRANS_CURR%	*Trans Curr* field value
%TRANS_DATE%	*Trans Date* field value
%TRANS_DR_CR%	Transaction type *DR\CR* field value. Defined by the *Transaction Type* field of the document.
%TRANS_TYPE%	Value from the "Message Dictionary" dictionary that is localized for the client's language and corresponds to the *Trans Type* field value.
%DFLT_TRANS_TYPE%	*Trans Type* 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.
%TRANS_SIC%	*SIC Code* field value
%TRANS_COUNTRY%	*Trans Country* field value
%TRANS_CITY%	*Trans City* field value
%TRANS_DETAILS%	*Trans Details* field value
%TRANS_SOURCE_CODE%	*Source Code* field value
%TRANS_TARGET_CODE%	*Target Code* field value
%TRANS_RETURN_CODE%	*Return Code* field value
%TRANS_RC_DESCRIPTION%	Text of the message with the code from the *Return Code* field
%TRANS_REASON_DETAILS%	*Reason Details* field value
%TRANS_TAGS%	*Add Data* field value (see the section "%TRANS_TAGS%")
%DOC_TAGS%	Same as %TRANS_TAGS%
%TRANS_RET_REF_NUMBER%	*Ret Ref Number* field value
%TRANS_ACQ_REF_NUMBER%	*Acq Ref Number* field value
%TRANS_ISS_REF_NUMBER%	*Iss Ref Number* field value
%TRANS_SOURCE_REG_NUM%	*S Reg Num* field value
%TRANS_AUTH_CODE%	*Auth Code* field value
%TRANS_DOC_AMND_DATE%	*Amendment Date* field value
%TRANS_DOC_AMND_OFFICER%	*Amendment Officer* field value

Name	Value
%TRANS_DOC_ID%	*Record Id* field value
%TRANS_SOURCE_NUMBER%	*Source Number* field value
%TRANS_TARGET_NUMBER%	*Target Number* field value
%TRANS_FX_SETTL_DATE%	*FX Settl Date* field value
%DOC_RECONS_AMOUNT%	*Amount Reconcil* field value
%RECONS_CURR%	*Currency* field value for *Amount Reconcil*
%ADD_SERVICE_INFO%	*Service Info* field value of the "Services for <name of additional online service>" form of the Service on which the transaction was made.
%TRANS_NAME%	Name of the transaction according to the "Message Dictionary". Determined based on data of posted financial documents. Used only for generating statements.
%TRANS_ACC_AMOUNT%	Amount debited from client account when document was posted (see the section "%TRANS_ACC...%"). Determined based on data of posted financial documents.
%TRANS_ACC_CURR%	Currency of the amount debited from the client account (see the section "%TRANS_ACC...%"). Determined based on data of posted financial documents.
%TRANS_ACC_FEE%	Fee charged to the client account (see the section "%TRANS_ACC...%"). Determined based on data of posted financial documents.
%TRANS_ACC_TOTAL%	Sum of the %TRANS_ACC_AMOUNT% and %TRANS_ACC_FEE% values (see the section "%TRANS_ACC...%"). Determined based on data of posted financial documents.
%TRANS_ACC_ADV_AMOUNT%	Amount of an entry for a macrotransaction with the "Advice" request category (see the section "%TRANS_ACC...%"). Determined based on data of posted financial documents.
%TRANS_ACC_ADV_FEE%	Fee for a macrotransaction with the "Advice" category (see the section "%TRANS_ACC...%"). Determined based on data of posted financial documents.
%TRANS_ACC_ADV_TOTAL%	Sum of the %TRANS_ACC_ADV_AMOUNT% and %TRANS_ACC_ADV_FEE% values (see the section "%TRANS_ACC...%"). Determined based on data of posted financial documents.
%TRANS_ACC_REV_AMOUNT%	Amount of an entry for a macrotransaction with the "Reversal" request category (see the section "%TRANS_ACC...%"). Determined based on data of posted financial documents.

Name	**Value**
%TRANS_ACC_REV_FEE%	Fee for a macrotransaction with the "Reversal" category (see the section "%TRANS_ACC...%"). Determined based on data of posted financial documents.
%TRANS_ACC_REV_TOTAL%	Sum of the %TRANS_ACC_REV_AMOUNT% and %TRANS_ACC_REV_FEE% values (see the section "%TRANS_ACC...%"). Determined based on data of posted financial documents.
%POSTING_DATE%	Value of the *Date* field in the "M-trans for..." form – banking date of posting the macrotransaction for the corresponding document. Determined based on data of posted financial documents.
%TRANS_ACC_MTR_DETAILS%	*Mtr Details* field value of the form "M-trans for..." – macrotransaction details for the corresponding document. Determined based on data of posted financial documents.
%ENTRY_FX_RATE_VALUE%	FX rate. Determined based on data of posted financial documents.
%TRANS_BLOCKED_AMOUNT%	Amount blocked as the result of the transaction. Determined based on posted authorisation and other financial documents that do not require authorization for the set amount limit (hereinafter the “sublimit financial documents”).
%TRANS_BLOCKED_CURR%	Currency of funds blocked as the result of the transaction. Determined based on posted authorisation and sublimit financial documents.
%TRANS_BLOCKED_FEE%	Fee amount blocked as the result of the transaction. Determined based on posted authorisation and sublimit financial documents.
%TRANS_SERVICE_CLASS%	*Service Class* classifier transaction type.
%TRANS_SOURCE_SPC%	Relation type (*Relation Type* field value) of the source contract and the related contract.
%TRANS_TARGET_SPC%	Relation type (*Relation Type* field value) of the target contract and the related contract
%DOC_ACTION%	Document's ACTION field value
%TRANS_MISC_FEE:<UsageType>%	Fee amount and letter code of the currency for the fee charged if a <UsageType> limiter is activated.

Name	Value
%TRANS_MISC_LIM_FEE:<UsageType>%	The value by which the limit set by a <UsageType> limiter must be exceeded for a %TRANS_MISC_FEE:<UsageType>% to be charged. The amount and letter code of the currency is passed in the variable. 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%' Example of a message: 'A fee of 3 EUR for exceeding the monthly cash withdrawal limit by 3000 EUR'.
%SIBLING_MTR%	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). Determined based on data of posted financial documents.
%TRANS_BONUS_ALL%	Amount of all bonuses points, regardless of the account type, with *Account Role* = "BONUS" (based on contract data).
%TRANS_BONUS_AT:<AccountType>%	Amount of all bonus points, regardless of account type with *Account Role* = "BONUS" (based on contract data).
%TRANS_BONUS_LIST%	Comma-separated list of bonus points (based on contract data)
%TRANS_BONUS_DOC_ALL%	Amount of all bonus points, regardless of account type, with *Account Role* = "BONUS" (based on document data).
%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**
%TRANS_BONUS_DOC_LIST[:<Prefix>:<Expire>:<Days>]%	Comma-separated list of bonus points with their effective periods (based on document data). <Prefix>, <Expire>, <Days> can be codes registered in the "Message Dictionary" handbook of text elements (Full\Configuration Setup\Main Tables\Message Dictionary; *Object Type* = 'Transaction', *Object Name* = '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 <Expire> and <Days> text elements. The code of the corresponding account type is output together with the amount. Example of the variable's values: 'Bonus List: 100 SPASIBO, 100 z expired After 21 days!' Here: 'Bonus List:' is the <Prefix> parameter value. '100' is the amount of bonus points. 'SPASIBO' and 'z' are account type codes. 'expired After' – <Expire> parameter value. '21' – VALUE_DAYS value. 'days!'– <Days> parameter value.
Information about a payment order for which a document was generated (values of fields from the "Full Info For St. Orders for ..." form). The variables listed below are used with the prefix STANDING_ORDER$.	
%PAYMENT_TYPE%	Payment type (*Payment Type* field value)
%PAYMENT_RECEIVER%	Payment recipient (*Standard Payee* field value)
%TARGET_DETAILS_1%	Additional information about a correspondent account (*Target Details 1* field value)
%TARGET_DETAILS_2	Additional information about a correspondent account *Target Details 2* field value)
%TRANS_DETAILS%	Additional information about a payment (*Payment Details* field value)
%COMMENT_TEXT%	Comments for an order (*Order Comment* field value)
%REASON_DETAILS%	Tagged information from the *Posting Details* field
%ORDER_CODE%	Standing Order code (*Order Code* field value)
Other variables	
%PIN_ATTEMPTS%	Remaining number of PIN attempts.
%MAX_PIN_ATTEMPTS%	Maximum number of PIN attempts set for the current contract.
%OFFICER%	Current user's identifier
%OFFICER_GROUP%	Current user's group

3.2 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.

3.2.1 Additional date parameters

For variables showing information about account balances (which in particular are used when generating statements and SMS notifications), additional date parameters can be used. For example, an account's current balance (at the time the message was generated) can be displayed, or an account's balance at the start of a contract's billing cycle.

- %<variable name>:::LOCAL_DATE% – shows the variable value on the message generation date (default value).
- %<variable name>:::BEGIN_DATE% – shows the variable value on the start date of this contract's billing cycle.
- %<variable name>:::NEXT_BILLING_DATE% – shows the variable value on the start 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 when generating the statement.
- %<variable name>:::DATE_FROM% – shows the variable value on the following dates:
    - For a monthly contract statement – start date of the period specified by the *Date To* parameter when generating the statement.
    - For a current contract statement – the date specified by the *Date From* parameter when generating the statement.
- %<variable name>:::DATE_TO% – shows the variable value for the following date:
    - For a statement – the date specified by the *Date To* parameter when generating the statement.

For example, %ACCOUNT_BALANCE:P::BEGIN_DATE%.

When separators are used (:) and the date parameter is absent, the default value is used (LOCAL_DATE).

3.2.2 %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").

3.2.3 %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%").

3.2.3.1 %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).

3.2.3.2 %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.

3.2.3.3 %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 – parameters are described in the section "%ACCOUNT_BALANCE%".
- <Date> - the end date (in YYYY–MM–DD format) of the interval (the start date of the interval matches the start date of the billing cycle to which <Date> belongs) for generating information about account turnover. For example, when PREV_BILLING_DATE is specified (end date of the previous billing cycle), information about account turnover for the previous billing cycle will be provided. If no value is set for the parameter, information will be provided about account turnover for the period from the start of the current billing cycle to the date on which the Event that generated this message opened, inclusively.
- <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.

3.2.4 %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%".

> The procedure for using the <Date> parameter for the %ACCOUNT_CAT_TURNOVER% is the same as that described in the section "%ACCOUNT_TURNOVER%".

3.2.5 %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 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.

%ACCOUNT_TAG...% variables can be used with additional parameters, the set of which is determined by the corresponding literal. I.e. %ACCOUNT_TAG_BALANCE...% parameters correspond to parameters described for %ACCOUNT_BALANCE%, %ACCOUNT_TAG_TURNOVER...% parameters – to those described for %ACCOUNT_TURNOVER%, etc.

3.2.6 %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%.

3.2.7 %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%.

3.2.8 %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.

3.2.9 %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.

3.2.10 %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.

3.2.11 %PCNT%

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

3.2.12 %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%)%

3.2.13 %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).

3.2.14 %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").

3.2.15 %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.

3.2.16 %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).

3.2.17 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 – 6799994599790985. 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.

3.2.18 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.

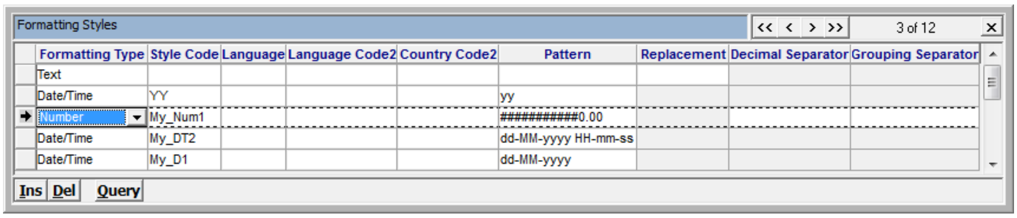
  Using one of these prefixes will also help avoid the following situation. If the current Event opened as a result of a usage limiter, the variables associated with client and contract data (for example, %CONTRACT_NUMBER%), are substituted with the field values of the contract for which the limiter was 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

3.2.19 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

<table>
<tr><th>№</th><th>Regular expression (<i>Pattern</i>)</th><th>Replacement expression (<i>Replacement</i>)</th><th>Result</th></tr>
<tr><td>1</td><td>(...)(.*)</td><td>xxx$2</td><td>xxx34586975478</td></tr>
<tr><td colspan="4">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 charaters (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.</td></tr>
<tr><td>2</td><td>(....)(........)(.*)</td><td>$1zzz$3</td><td>3480zzz9368845</td></tr>
<tr><td colspan="4">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.</td></tr>
</table>

- *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.

3.2.20 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").

3.2.21 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.

3.2.22 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.

3.2.23 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%
```

To manage showing of additional information when generating a client statement (for example, depending on the current contract status), the FRAME_ATTR=SHOWIF (to show information according to condition) or FRAME_ATTR=HIDEIF (to hide block of information according to condition) can be used. For example:

```
FRAME_ATTR=HIDEIF=%IS_EVENT_OPENED:NO_GRC_STATE:END$DATE_TO%;
```

or

```
FRAME_ATTR=SHOWIF=%ACCOUNT_TAG_BALANCE_STMT:PD_PREV::END$DATE_TO:DR%>0.00;
```

The conditional operators described above can be used. It is necessary to take into account that the "SHOWIF" and "HIDEIF" values apply to entire message block, i.e. all strings in the current template will be hidden or shown.

⚠ 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.