Operation Manual

Events

03.49.30

15.07.2020

Contents

This document describes how Way4 system Events are configured. Events automatically execute certain actions. For example, an Event can be configured to automatically change a contract status when certain conditions are met, or conversely, to take certain actions when a contract status changes.

This document is intended for bank or processing centre employees responsible for configuring Way4 and describes product creation and configuration.

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

- Daily Procedures
- Way4 Service Packages
- Way4 Accounting Schemes
- Products and Contract Subtypes
- Configuration of Client Messages
- Standing Payment Orders
- Way4 Global Parameters
- Issuing Module
- Acquiring Module
- Way4 Dictionaries
- Documents
- Cardholder Statements
- Way4 Advanced Tariff Management
- Risk Monitoring
- Loan Loss Reserves
- "Notification Messaging"
- "Way4 Accounting"

The following notation is used in this document:

- Field labels in screen forms are typed in *italics*.
- Button labels used in screen forms are placed in square brackets, such as [Approve].
- Menu selection sequences are shown with the use of arrows, such as Configuration Setup →Contract Types.

⚠ Warnings about potentially hazardous situations or actions.

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

1 Overview of System Events

This section contains a description of the Event lifecycle in Way4 and setup of Events.

1.1 Definition and Basic Properties of Events

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

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

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

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

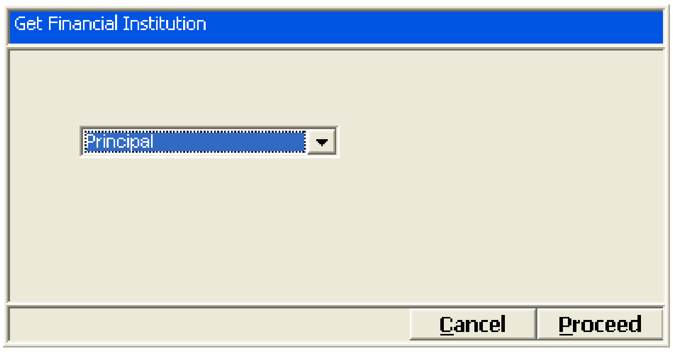
1.2 Event Types

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

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

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

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

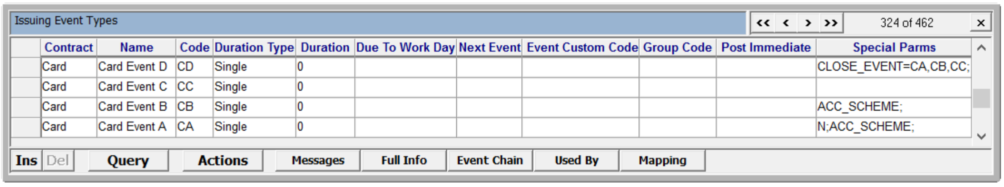


Get Financial Institution

Principal

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

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



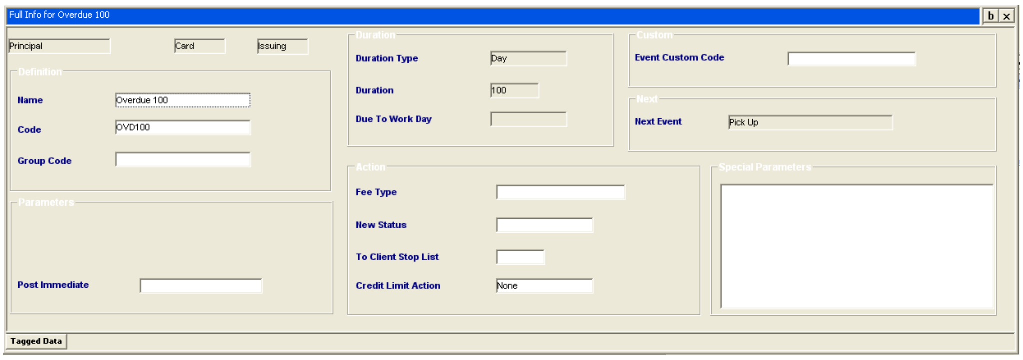
Issuing Event Types

324 of 462

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

Fig. 2. Form for entering data on Event types

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



Full Info for Overdue 100

Principal Card Issuing

Definition

Name: Overdue 100

Code: OVD100

Group Code:

Parameters

Post Immediate:

Duration

Duration Type: Day

Duration: 100

Due To Work Day:

Action

Fee Type:

New Status:

To Client Stop List:

Credit Limit Action: None

Custom

Event Custom Code:

Next

Next Event: Pick Up

Special Parameters

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

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

- *Name* – name of the Event type. Events with the same code must have the same name.

- *Code* – code that is used to identify this Event type in the database. In some cases, the field is used for other purposes (see "Opening Events")

> The code of an Event must be unique for each combination of the following parameters: financial institution, contract category ("Card", "Account", "Device") and Product category ("Issuing", Acquiring", etc).

- *Group Code* – code of the Event type group used to filter Event types by certain characteristics
- *Post Immediate* – determines when Event processing is started and when an Event is closed; this field can take on one of the following values:
  - "Post Immediate" – an Event will be processed immediately after it is opened.
  - "Post Later" – an Event will be processed after the next posting of macrotransactions generated for the contract for which the Event is opened; if no macrotransactions are posted, it will be processed under the "Contracts Daily Update" procedure.

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

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

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

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

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

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

- *Duration Type* – unit of time measuring a time interval at the end of which the Event will close or another Event from the Event chain will open (see the *Next Event* field); the field can take on one of the following values:
  - "Day" – the time period is measured in calendar days
  - "Month" – the time period is measured in months
  - "Billing" – the time period is measured in billing cycles

    It is recommended to use Events with the "Billing" value in the *Duration Type* field on the account contract level. Events with the "Billing" value can be used on the subordinate card contract level if the algorithm for calculating the billing cycle end date did not change. If the billing cycle end date changed, the corresponding correction of the Event closing date will not be made on the subordinate contract level.
  - "Single" – the Event is closed in the system by other means. For example, this duration type is set when an Event is configured to open when the balance of a contract account changes. In this case, the Event closes when the account balance is 0. This duration type is also set for Events that do not need to be closed.
  - "Unique" – the Event is closed immediately after it is opened.

> Note that for an Event type with the "Unique" value in the *Duration Type* field, an Event chain cannot be configured. This means that the *Next Event* field cannot be filled in for this Event type. The next Event will not be opened when a Unique Event closes.

- *Duration* – number value of the time interval length in the units set in the *Duration Type* field.

  Specifics of *Duration* parameter processing:
  - When the *Duration Type* parameter value is "Billing", the length of the period for an Event is calculated as the "*Duration* field value + 1". For example, when the value of the *Duration* parameter is "0", the length of

the period for an Event will be equal to one billing cycle. When the *Duration* parameter value is "1" – to 2 billing cycles.

- When the *Duration Type* parameter value is "Month", the length of the period for an Event is fully determined by the value of the *Duration* field. I.e. when the value of the *Duration* parameter is "1", the length of the period of an Event will be equal to one month. When the value is "2" – to 2 months, etc.

- *Due To Work Day* – used when the date of closing the Event specified with the *Duration Type* and *Duration* fields is a day off, but it is necessary to close the Event on the first working day after the weekend. The field can take on either "Yes" or "No"

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

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

  The appropriate miscellaneous service must be configured for this fee type. See the "Service Configuration" section of Way4 Service Packages Administrator Manual.

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

  If an event with a configured fee opens during an authorisation, for the authorisation to be posted correctly, specify the SKIP_CHARGE tag for the Event. In this case, the fee will be posted separately from the authorisation (after the next posting of macrotransactions).

- *New Status* – status associated with the contract while the Event is open; the list of contract statuses registered in the system can be viewed in the "Contract Statuses" grid (Full → Configuration Setup → Contract Types → Contract Statuses).

- *To Client Stop List* – determines whether the card should be placed on a stop list while the Event is open. This field can be set to either "Yes" or "No"
- *Credit Limit Action* – additional actions concerning the credit limit that may be executed when the Event is opened; the field can take on one of the following values:
  - "Set to Zero" – the contract's credit limit will be set to zero
  - "Suspend" – the contract will be given an additional credit limit equal to the value of the base credit limit, but with the opposite sign. Thus, the credit limit will be completely eliminated from the Amount Available
  - "Set To Event Amount" – the "Set To Event Amount" value makes it possible to set a credit limit for a contract. The credit limit amount and currency are specified in the *Comment Details* field using the AMOUNT and CURR tags, respectively. The currency code set in the CURR tag must correspond to the contract currency.
  - "None" (the default value) – no additional actions concerning the credit limit will be executed
- *Next Event* – Event that should open when the current Event closes or at the end of the time interval set in the *Duration* field. This configuration allows Event sequences to be created, where each Event can trigger the next one. For example, if a client loan becomes overdue, two different messages may be created for the client, where the second message is created 15 days after the first one. If the client repays the loan within 15 days from the moment the first message is created, then the second message is not created (for message configuration, see the section "Generating Event Messages").

> Note that for an Event type with the "Unique" value in the *Duration Type* field, an Event chain cannot be configured. This means that the *Next Event* field cannot be filled in for this Event type. The next Event will not be opened when a Unique Event closes.

If the same Event should open with a given regularity (this configuration can be used, for example, to charge a monthly fee), specify the same Event Type in the *Next Event* field. In this setup, a new instance of the Event automatically opens when the old one closes and this process will continue infinitely. These cyclic Events always open on the same day – on the day the first Event in the chain opens.

- The *Custom Event Code* field is used to specify a special code. If the field is filled in, when posting the Event, the custom procedure CUST_EVNT_POST is called, to which this code is passed as the Event code.
- *Special Parameters* – the *Special Parameters* field is used to define special parameters as tags. For more details, see the section "Tags used when working with Events" of the "Setup Tags" document.

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

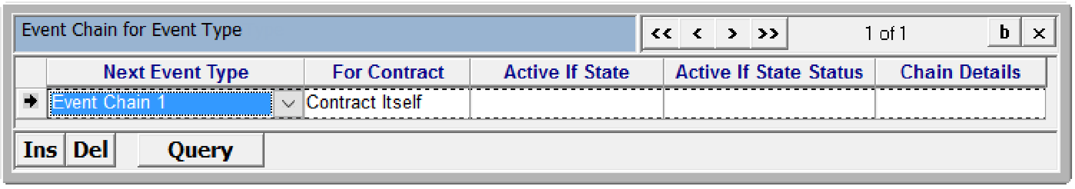
- The [Action] button contains the following context menu commands:
  - "Check" – when the "Check" command is executed, Way4 checks that the fields of the specified Event Type have been filled in correctly and informs users of the check's results.
  - "Delete" – delete an Event Type record. When "Delete" is selected, a check is made for links in Way4 to this Event Type:
    - If Events of this type are set up in Service Packages, Accounting Schemes, account templates, standing payment orders, usage limiters, in an Event Chain (*Next Event Type* field in the "Event Chain" form), in the *Next Event* field of another Event in the "Event Types" form, an Event of this type is referenced in the Event log (USAGE_ACTION), an error message is generated, and the record is not deleted.
    - If the code of an Event Type is set for the Product ("Start Events" form), for classifier values (*Result Event Code* field) in settings allowing the value of a user classifier to be changed when an Event opens (Classifier Actions), a warning is generated and the record is deleted.
- The [Event Chain] button opens the "Event Chain for ‹Event type name›" form used to generate an Event chain (see "Generating Event Chains").
- The [Messages] button is used to invoke the message template through which the database creates client messages (see the section "Generating Event Messages").
- Clicking on the [Used By] button will open the "Used By for ‹Event type name›" form. It lists States whose activity depends on the status of the current Event (see the "State Parameters" section in the States Administrator Manual).
- The [Mapping] button is used to mark Events using "Configuration Groups" classifiers. The procedure for marking Events is similar to that for marking Products. For more detailed information about configuring "Classification Groups" classifiers and about marking procedures, see the section ""Configuration Groups" Classifiers" of the Products and Contract Subtypes Administrator Manual.

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

1.3 Generating Event Chains

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

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



Event Chain for Event Type

1 of 1

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

Fig. 4. Form for generating an Event chain

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

- Next Event Type – Event type
- *For Contract* – determines for what contracts (for the contract for which the main Event is opened (hereinafter, current contract) or for contracts linked to this contract) this Event will be opened. The field can take on one of the following values:
  - "For Self" – the Event will be opened for the current contract only
  - "Sub Contracts" – the Event will be opened for every subcontract linked to the current contract with the "Main/Sub" link type
  - "Main Contract" – the Event will be opened for the main contract to which the current contract is linked with the "Main/Sub" link type
  - "Liab Subs" – the Event will be opened for every subcontract linked to the current contract with the "Liability" link type
  - "Liab Main" – the Event will be opened for the main contract to which the current contract is linked with the "Liability" link type
  - "For Client" – the Event will be opened for all contracts of the client to whom the current contract belongs
  - "Service Contract" – the Event will be opened for the service contract set up for the current contract's service group. In this case, leave the

Next Event Type field blank, and specify the code of the opened Event in the *Chain Details* field

  - "Client Service Contract" – the Event will be opened for the service contract set up for the service group of the client to whom the current contract belongs
  - "Device Service Contract" – the Event will be opened for the service contract set up for the service group of the current contract's device
  - "Base Contract" – the Event will be opened for the contract for which the current contract is a related one
  - "Related Contract" – An Event will be opened for all contracts related to the current contract with the specified relation type, with the exception of contracts with the "FREE RELATION" type (for more information, see the section "Related Cards" of the document "Issuing Module. User Manual"); the relation type is specified through the REL_TYPE=<relation type code>; tag in the *Chain Details* field. Relation type codes are specified when relation types are registered (Full →Configuration Setup → Accounting Setup →Contract Relations); a relation type is determined on the related contract level.
  - "Related Bank Contract" – the Event will be opened for the fee contract (bank contract whose accounts are used to transfer fees for the current contract)
  - "Related Applet" – an Event will be opened for Applet subcontracts.
  - "Custom" – this value is not used in the current version

- *Active if State* – specifies the State (see the States Administrator Manual) whose status determines whether the Event specified in the *Next Event Type* will be opened for contracts specified through the *For Contract* field
- *Active if State Status* – determines the status of the State specified in the *Active if State* field. If the field reads "Yes", the Event will be opened for contracts specified through the *For Contract* field only if the specified State is active. If the field reads "No", the Event will be opened only if the State is inactive.
- *Chain Details* – this field is used to define special parameters as tags. For more details, see the section "Tags used when working with Events" of the "Setup Tags" document.
- *Step N* – determines the order for opening Events in a chain. I.e. an event with a *Step N* field value of "1" will be processed earlier than an Event with a value of "2".

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

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

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

2 Opening Events

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

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

Furthermore, Way4 allows opening and closing Events for a contract through a user command. For more details, see subheading "Closing Events".

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

Table 1 Situations where direct Event opening is used

Situation	Event Characteristics
A contract account balance changes	An Event type must be specified in the account template configurations (see the "General" section in the Way4 Accounting Schemes Administrator Manual). When funds are transferred to a contract's account that uses this template, the Event is opened; when the balance returns to zero, the Event is closed (see an example in the "Examples of Event Use").
A limiter is activated	An Event type must be specified in the limiter configuration. When the limiter is activated, the Event is opened; when the limiter's counter is reset, the Event is closed (see the "Additional Parameters of Limiters (Details)" section in the Usage Limiters Administrator Manual). It is recommended to set the *Duration Type* parameter of such an Event to "Unique" for the Event to close immediately after opening.
A document for to a standing payment order is generated	An Event type must be specified in the standing payment order configurations. The fields defining the payment order's activation frequency must be filled in as indicated (see the "Standing Payment Orders" Administrator Manual).

Another Event is closed	For the first Event in the chain, the next Event in the *Next Event* field of the "Event Types" form needs to be specified along with the time period after which the system should start processing the next Event in the *Duration* field.
An "external" event occurs	An "external" event means receiving a status message from an ATM, change of channel status, receiving a negative response code, etc. To open an Event in such a case, it is necessary to configure the "Usage Operation" table as indicated (see the Notification Messaging Administrator Manual).
A contract opens/closes	An Event opens automatically when a contract using the given Product opens/closes. Such Events are configured in the "Start Events for <name of Product>" form (see the section "Event Setup" in the "Products and Contract Subtypes" document).

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

For Event 1 to close when funds go from one account to another (to account "B") in which Event 2 opens, specify the NOT_USED_IN_CHAIN; tag for Event 1.

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

Table 2 Situations where Events with predefined codes are used

Situation	Code	Comments
A contract status changes	There are two options for setting an Event code: • <Prefix>_<Status Code>, where <Prefix> is the value of the "CHANGE_STATUS_STR" additional global parameter. By default, it is "Change Status" (see the Way4 Global Parameters Administrator Manual), <Status Code> is the code of the new contract status (value of the *Code* field in the "Contract Statuses" form). • It is possible to set the code in the format <Prefix>_<Status External Code1>_<Status External Code2>, where <Status Code1> – *External Code* code for the old status of the contract (value of the *External Code* field in the "Contract Statuses" form) <Status Code2> – *External Code* code for the new status of the contract (value of the *External Code* field in the "Contract Statuses" form) In this case, an Event opens when the contract status changes from <Status External Code1> to <Status External Code2>. For this case, it is possible to set the code in the format <Prefix>EXT<Status External Code1>_<Status External Code2>. Note that an Event with this predefined code opens by default (i.e. without addition to the Event chain) only when the status is manually changed in Customer Service Workbench. When status changes according to an Event, through an application or when the contract form is edited and approved, an Event with this predefined code will only be opened if it is added to an Event chain. For more information, see the section "Changing Contract Status"	To access the list of contract statuses registered in the system, use the "Contract Statuses" grid form (Full → Configuration Setup →Contract Types →Contract Statuses).

Situation	Code	Comments
	and the section "Tags used when working with Events" of the "Setup Tags" document). Note that when a contract status changes, only one Event can be open. The setting <Prefix>_<Status External Code1>_<Status External Code2> has priority. This means that if two Events are set up: <Prefix>_<StatusCode2> and <Prefix>_<Status External Code1>_<Status External Code2>, the Event <Prefix>_<Status External Code1>_<Status External Code2> will open. In this case, configure an Event chain to open the Event <Prefix>_<StatusCode2>.	
Locking a card until it is received by the cardholder	CP_<Production Code>, where <Production Code> is the value of the *Code* field in the "Production Events" form (Full → Configuration Setup → Transaction Types → Production Events)	
A credit limit changes	CREDIT_LIMIT_DOC	
An additional authorisation limit changes	ADD_CREDIT_LIMIT_DOC	
A contract is created	START_CONTRACT	An Event will be opened under the "Contracts Daily Update" procedure only if it is registered in the contract's Service Package
A contract is closed	The value of the "CLOSE_CONTRACT_STR" additional global parameter, which is "Close Contract" by default (see the Way4 Global Parameters Administrator Manual)	An Event will be opened only if it is registered in the contract's Service Package
Due normalisation between accounts is executed	DUE_FOR_<Code>, where <Code> is the code of the account type for which normalisation is executed	

Situation	Code	Comments
Limit normalisation between accounts is executed	LIM_NORM_<Debit Acnt Type Code><Credit Acnt Type Code>, where <Debit Acnt Type Code> is the code of the debited account type and <Credit Acnt Type Code> is the code of the credited account type	An Event will be opened under the "Contracts Daily Update" procedure only if it is registered in the contract's Service Package
The interest accrual procedure is executed	INT_I<Account Type Code>, where <Account Type Code> is the code of the account type	An Event with the "INT_IP" code will be opened when interest is accrued for a deposit account
A financial document without an authorisation is processed for a card listed in the stop list	STOP_LIST	
An incorrect PIN attempts counter is cleared forcibly	CLEAR_PIN_ATT	This Event opens if a counter reset is forced (for example, when executing the menu item "Clear PIN Attempts" in customer service workbench). The Event does not open when the counter is automatically reset (for example, when the client incorrectly enters a PIN during purchase (authorisation) and subsequent correct entry of the PIN).
A financial reversal document is processed if limit normalisation was executed between the	INTEREST_WARNING	An Event only opens when the value of the global parameter

Situation	Code	Comments
registration of the initial authorisation and its reversal		PATCH_REVERSE_INTEREST is "Y".
A card is activated (the first PIN-based transaction is executed for a card; a card is unlocked manually, etc.)	CARD_UNLOCK	
A processed document is reversed after reconciliation on a POS terminal	POS_RECON_REVERSE	For more details about reconciliation on a POS terminal, see the POS Management Administrator Manual
A financial document of the "When available" category is processed if the interval specified in the *Expiry Day* field of the service is exceeded (see the "Transaction Parameters" section in the Way4 Service Packages Administrator Manual)	DOC_EXPIRED	For more details on processing financial documents of this category, see the "Posting "When Available" Type Documents" section in the Documents Administrator Manual
A custom procedure is executed		In the description of the Event, specify a special code in the *Custom Event Code* field. If this field is filled in, when posting the Event, the custom procedure CUST_EVNT_POST is

Situation	Code	Comments
		called, to which this code is passed as the Event code.
A dispute case is opened	DSP_CASE_OPENED	
A dispute case is closed	DSP_CASE_CLOSED	
A debt is written off	DSP_CASE_WRITTENOFF	
An incoming dispute document is received	DSP_ITEM_INWARD_CREATED	
An overdraft appears (at the end of the billing cycle)	OVL_BILLING	The Event is opened at the end of a billing cycle. The Event will only be opened if the Event is registered in the contract's Service Package.
An overdraft appears (at the end of the banking day)	OVL_DAILY	The Event is opened when the banking day closes. The Event is called from a custom procedure executed during "Contracts – Daily Update" procedures. The Event will only be opened if the Event is registered in the contract's Service Package.
Unblocking authorisation funds	AUTH_UNBLOCK	The Event is opened when unblocking authorisations using the "Clear Old Pendings" procedure (Full → Daily Procedures →Document Processing Step by Step →Clear Old Pendings). The Event is only opened if this Event is

		registered in the contract's Service Package. This Event remains for backward compatibility. If several blocks were made for a document (for example, the main transaction fee was blocked and a custom fee was blocked), funds are unblocked in arbitrary order. An Event with the AUTH_UNBLOCK code opens after each block for the document is released, but if message sending is set up in the Event, a message is sent once when the first block is released. I.e. if the custom fee is unblocked first, the message will contain the amount available after unblocking the fee. A message about the amount available after the transaction amount has been unblocked will not be sent. If configurations are used in which fees are blocked separately (for example, in a custom fee setup), use an Event

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

Situation	Code	Comments
An Event with the CONTRACT_DELAY code opens if the date of the contract's last financial activity is less than the current banking date by the number of days set with the ACTIVITY_DELAY tag in the Service Package. In doing so, for example, the contract may be assigned another status. The date of the contract's last financial activity is taken from the automatically inserted ACTIVITY_DATE tag in the ext_data field of the contract. The ACTIVITY_DATE tag is filled in if the ACTIVITY_FOR tag is set in the Service or transaction subtype. If the ACTIVITY_FOR tag is not set, the contract opening date is considered the date of the last activity on the contract. .	CONTRACT_DELAY%	An Event only opens if this Event is registered in the contract's Service Package. Several Events with the CONTRACT_DELAY% code can be configured for various periods when there is no contract activity (to charge various fees, send notifications). The CONTRACT_DELAY prefix is mandatory; the postfix may be arbitrary. See the section "No Contract Activity"

Situation	Code	Comments
An Event with this code (if such Event is configured) will open for a contract before the billing period closes, after calculating interest and executing due normalization. Allows an action to be executed at the end of a billing period (for example, charging a fee).	END_BILLING	An Event only opens if this Event is registered in the contract's Service Package.
Authorisation reversal	AUTH_REVERSE	An Event opens when processing an authorization reversal. Can be used to generate massages for the cardholder. The Event must be registered in the contract’s Service Package.
Creation of an authentication scheme	<Authentication scheme type code>_ENROLL	An Event opens when an authentication method is enabled (authentication scheme is created). See the section "Setting Parameters in Way4 for Interaction with the Way4 Authentication Server" of the document "Administering Way4 Authentication Server" in the authentication server distribution. This functionality is supplied according to an

Situation	Code	Comments
		additional agreement with OpenWay.
Closing of an authentication scheme	<Authentication scheme type code>_CLOSE	An Event opens when an authentication scheme is closed. See the section "Setting Parameters in Way4 for Interaction with the Way4 Authentication Server" of the document "Administering Way4 Authentication Server" in the authentication server distribution. This functionality is supplied according to an additional agreement with OpenWay.
SMS notification when the debited amount exceeds the authorisation amount	AMOUNT_ALERT	An Event opens if the amount of the financial document differs from the amount blocked by authorisation. An Event is used together with a tariff with the "Threshold" role and the predefined code AMOUNT_ALERT. The possible difference between the amount of the financial document and the authorisation amount at which no notification is sent (i.e. an Event with the AMOUNT_ALERT

Situation	Code	Comments
		code does not open) is specified in the tariff.
Unblocking authorisation funds	CLOSED_AUTH_DOC	An Event opens after all blocking for a document has been removed. If sending messages is set up in an Event, when a message to a client is created, the amount available after all unblocking is sent. An Event is opened when unblocking authorisations with the "Clear Old Pendings" (Full → Daily Procedures → Document Processing Step by Step →Clear Old Pendings) procedure. An Event is only opened if this Event is registered in the contract's Service Package.
Setting a contract or client classifier value on a date in the future, specifying the expiration date of the value. The Event Type should be given the "Single" value in the *Duration Type* field	CLS	An event is opened on the Date From date specified for the classifier value. The Event closes automatically on the *Date To* date specified for the classifier value. See the section "Manually Changing Classifier Values" of the document "Way4 Client and Contract Classifiers".

Situation	Code	Comments
		An Event is only opened if this Event is registered in the contract's Service Package.
Setting a contract or client classifier value on a date in the future, without an expiration date. The Event Type should be given the "Unique" value in the *Duration Type* field	CLS_UNIQUE	An event is opened on the Date From date specified for the classifier value. The Event closes immediately after the corresponding classifier's value has been set. See the section "Manually Changing Classifier Values" of the document "Way4 Client and Contract Classifiers". An Event is only opened if this Event is registered in the contract's Service Package.
Number of attempts to enter a PIN has been exceeded	PIN_ATT_EXC	An Event is opened when the permitted number of attempts to enter a PIN has been exceeded.
Close plastic on reissue. "Unique" should be set in the *Duration Type* field for the Event type.	CLOSE_PLASTIC_TRANS_STATUS	When plastic is reissued, the old plastic is closed. When closing the old plastic an Event with this code opens. The time when the old plastic is blocked depends on the value of the global parameter CLOSE_PREV_PLASTIC (see the document "Way4 Global Parameters").

Situation	Code	Comments
An Event with the predefined code NO_ACTIVE_CARDS is opened during the "Contracts – Daily Update" procedure if cards for this contract are expired or blocked. The Event remains open for the period specified in the Event type. If during this period cards are issued/reissued for the contract, the Event closes. To automatically close a contract when this Event expires (if for this Event's activity period a new card wasn't issued/reissued) it is recommended to configure the next Event in the chain, that will close the contract.	NO_ACTIVE_CARDS	An Event is only opened if this Event is registered in the contract's Service Package. A check of whether a contract has non-null balances with specific codes can be set up for an Event (see the description of the CLOSE_IF_EMPTY tag). Additional settings are required to use this functionality (for more information, contact OpenWay representatives).

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

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

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

3 Processing Events

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

3.1 Standard System Actions

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

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

3.2 Changing Contract Status

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

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

Example.

- Contract statuses are configured:
  - "Card Do not honor" with "10" as the value of the "PR" variable in the *Code Parms* field.

    - "PickUP L 41" with "100" as the value of the "PR" variable in the *Code Parms* field.

  The priority of statuses is defined using the "PR" variable in the *Code Parms* field of the table "Full →Configuration Setup →Contract Types →Contract Statuses". For example, for the status "PickUp S 43", the *Code Parms* field contains the variable "PR=43" and for the status "Card Do not honour", the *Code Parms* field contains the variable "PR=5". Therefore, if a card was stolen, it will not be assigned the status "Card Do not honour" if an Event opens that is processed when past due loan debt arises.

- The corresponding Events to change the contract's status are configured"
    - "Event Status to PickUP L 41".
    - "Event Status to do no Honor 05" with the NOT_BETTER; tag. For this Event, an Event with a predefined code that must open when the status changes is specified in the "Event Chain" form (for example, attaching an Additional Service Package when contract status changes).

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

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

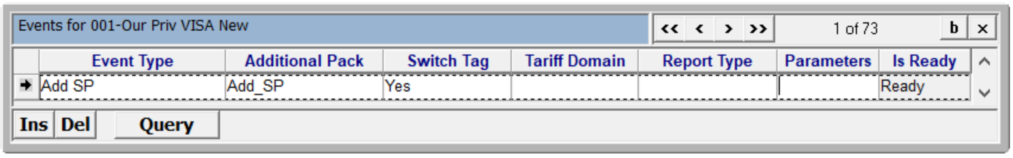
3.3 Generating Documents for Standing Payment Orders

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

3.4 Activating Additional Service Packages

The system can automatically activate/deactivate additional Service Packages for a contract's Service Package when Events are opened/closed for this contract (see the "Configuring Additional Service Packages" section in the Way4 Service Packages Administrator Manual).

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



Event Type	Additional Pack	Switch Tag	Tariff Domain	Report Type	Parameters	Is Ready
Add SP	Add_SP	Yes				Ready

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

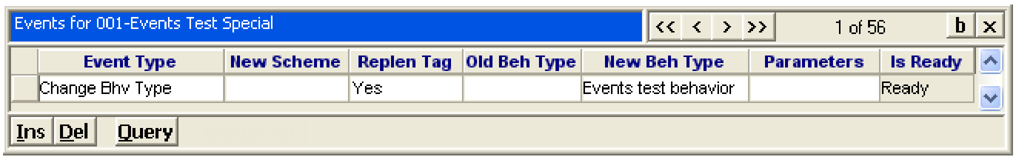
In the grid, click on the [Ins] button to add an empty row and fill in the following fields:

- *Event Type* – name of the Event type, which by being opened will activate/deactivate an Additional Service Package
- *Tariff Domain* – used to select one of tariff domains registered in the system; the field can be used to assign tariff domains to contracts when Events are opened/closed for the contracts
- *Additional Pack* – name of the Additional Service Package that will be active/inactive while the Event is open
- *Switch Tag* – determines whether the Additional Service Package is active (the "Yes" value) or inactive (the "No" value) while the Event is open
- *Report Type* – used to redefine the contract's set of reports when an Event is opened (a set of reports is specified in the field of the same name on the Product or contract level)
- *Is Ready* – shows whether changes made to the current Service Package's Event configuration have been approved

3.5 Changing Contract Behaviour Types

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

To do this, click on the [Event] button in a grid for entering and editing Accounting Scheme information, for example, in the "Private Issuing Account Scheme" grid (see the "Configuring Events" section in the Way4 Accounting Schemes Administrator Manual). The screen will display the "Events for ‹name of Accounting Scheme›" grid (see Fig. 6).



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

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

The "Events for ‹name of Accounting Scheme›" grid contains the following fields:

- *Event Type* – name of the Event type
- *New Scheme* – this field is used to select a new Accounting Scheme that will be assigned to the contract when this Event is activated (when the contract's behaviour group changes). For example, when an overdue payment appears, the contract can be assigned an Accounting Scheme with a higher interest rate in the loan account.
- *Replen Tag* – this parameter affects interest accrual in contract accounts. The field can take on one of the following values:
  - "Yes" – When an Event changing a behaviour type is opened, the contract accounts will accrue interest for the period from the last accrual date until the date the contract behaviour type is changed
  - "No" – When an Event changing a behaviour type is opened, interest is not accrued for the contract accounts
- *Old Beh Type* – this field indicates the current behaviour type; if this field is left empty when configuring an Event, it will change the behaviour type regardless of the current behaviour type. If this field contains a value, the Event will only change the behaviour type if the current type corresponds to the one set in this field
- *New Beh Type* – this field indicates the new behaviour type value
- *Parameters* – additional system parameterisation while the Event is open.
  - This field can indicate the code of the type of the account to which interest will be accrued while an Event is executed. Account types registered in the database can be found in the "Account Types" grid (Full → Configuration Setup → Accounting Setup → Account Types), the account type code is the symbol in the *Code* field
  - This field is used to define special parameters as tags. For more details, see the section "Tags used when working with Events" of the "Setup Tags" document.
- *Is Ready* – shows whether changes made to the Accounting Scheme have been approved

3.6 Changing Contract Account Schemes

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

To do so, in the form for entering and editing Account Scheme data, for example, in the "Private Issuing Account Schemes" form, click the [Events] button (see the section "Configuring Events" in the document "Way4 Accounting Schemes"). The

"Events for <Account Scheme name> form will open (see Fig. 6 in the section "Changing Contract Behaviour Types").

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

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

3.7 Generating Event Messages

Way4 allows users to create messages according to a template when an Event is opened and to send them periodically while the Event is open.

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

An example of setup for client notification if the available balance during authorisation is less than a specific threshold is shown in the section "Examples of Event Use".

3.8 Loan Loss Reserving

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

4 Closing Events

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

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

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

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

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

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

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

> ⓘ When the value of the global parameter DUE_TO_WRK_DAY is "N", the date of the weekend/holiday is specified in the *End Date* field if the Event's closing date

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

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

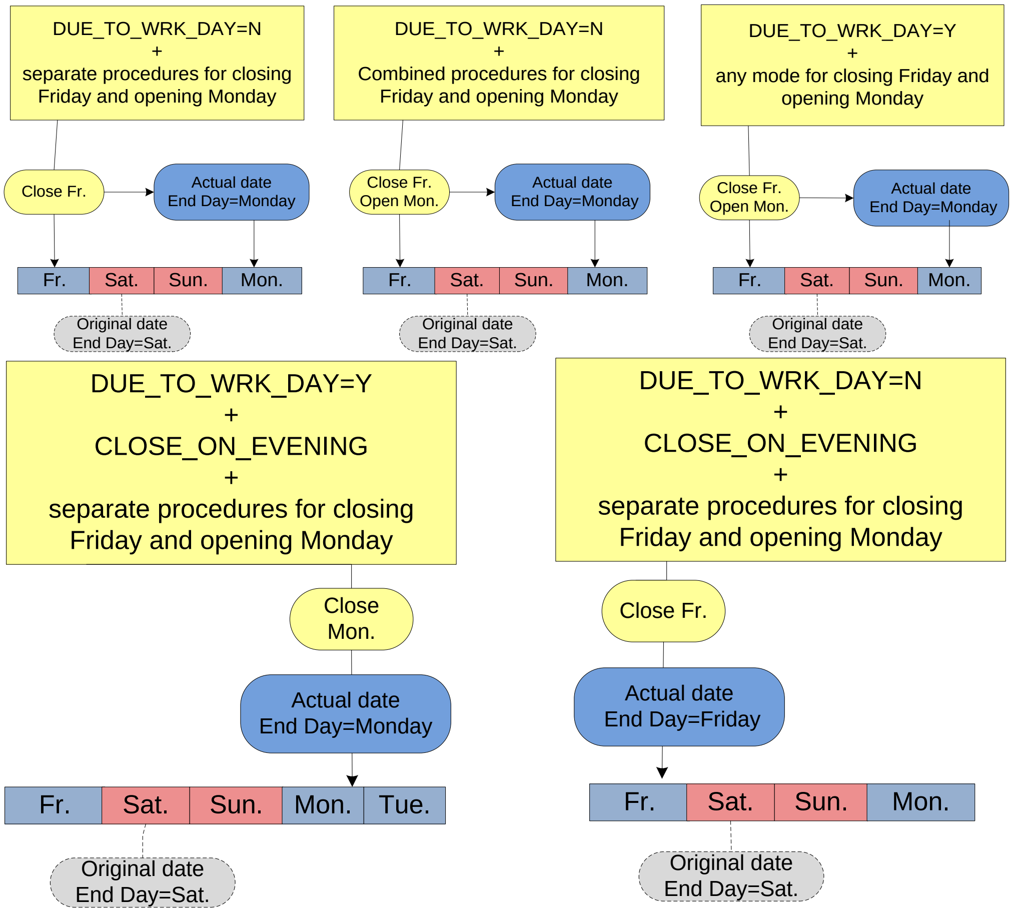

DUE_TO_WRK_DAY=N + separate procedures for closing Friday and opening Monday
Close Fr.
Actual date End Day=Monday
Fr. Sat. Sun. Mon.
Original date End Day=Sat.
DUE_TO_WRK_DAY=N + Combined procedures for closing Friday and opening Monday
Close Fr. Open Mon.
Actual date End Day=Monday
Fr. Sat. Sun. Mon.
Original date End Day=Sat.
DUE_TO_WRK_DAY=Y + any mode for closing Friday and opening Monday
Close Fr. Open Mon.
Actual date End Day=Monday
Fr. Sat. Sun. Mon.
Original date End Day=Sat.
DUE_TO_WRK_DAY=Y + CLOSE_ON_EVENING + separate procedures for closing Friday and opening Monday
Close Mon.
Actual date End Day=Monday
Fr. Sat. Sun. Mon. Tue.
Original date End Day=Sat.
DUE_TO_WRK_DAY=N + CLOSE_ON_EVENING + separate procedures for closing Friday and opening Monday
Close Fr.
Actual date End Day=Friday
Fr. Sat. Sun. Mon.
Original date End Day=Sat.

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

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

5 Events Log

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

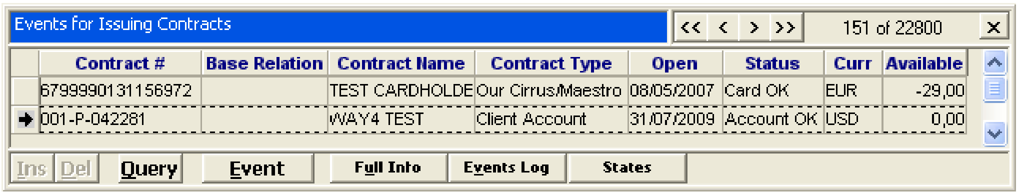
5.1 Description of the "Events for … Contracts" Form

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

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

All of these grids contain identical fields.

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



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

Fig. 8. Form "Events for Issuing Contracts"

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

- *Contract #* – number of the issuing contract
- *Base Relation* – type of same-tier relationship between contracts. Detailed information on same-tier relations may be found in the Issuing Module User Manual and Acquiring Module User Manual.
- *Contract Name* – name of the issuing contract
- *Contract Type* – type of the contract; the list of registered issuing contract types is displayed according to the contract category in the following forms:

- For card contracts – "Card Contract Types" (Full →Configuration Setup → Contract Types →Card Contract Types)
- For accounting contracts – "Accounting Contract Types" (Full → Configuration Setup →Contract Types →Accounting Contract Types)
- For device contracts – "Device Contract Types" (Full → Configuration Setup →Contract Types →Device Contract Types)

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

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

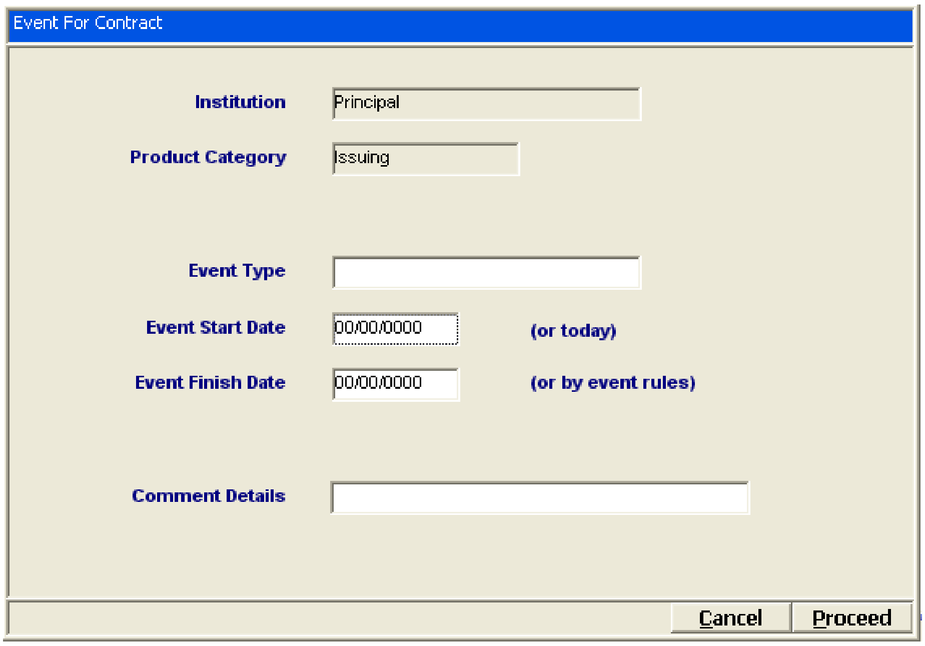
The [Event] button is used to open an Event by a user command (see Starting Events by User Command).

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

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

5.2 Starting Events by User Command

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



Event For Contract

Institution	Principal	
Product Category	Issuing	
Event Type		
Event Start Date	00/00/0000	(or today)
Event Finish Date	00/00/0000	(or by event rules)
Comment Details		

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

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

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

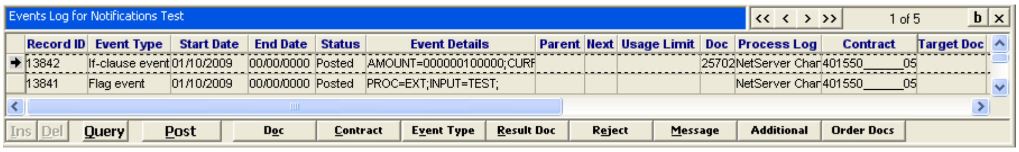
> In standard configurations of Products, it is not recommended to use Events with an *Event Start Date* that differs from the current banking date. For example, when a date in the future is specified in the *Event Start Date* field, the Event will be checked daily when executing "Contracts – Daily Update",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

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

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

5.3 Contract Events Log

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



Events Log for Notifications Test

	Record ID	Event Type	Start Date	End Date	Status	Event Details	Parent	Next	Usage Limit	Doc	Process Log	Contract	Target Doc
➔	13842	If-clause event	01/10/2009	00/00/0000	Posted	AMOUNT=000000100000;CURF				25702	NetServer Char	401550______05	
	13841	Flag event	01/10/2009	00/00/0000	Posted	PROC=EXT;INPUT=TEST;					NetServer Char	401550______05	

Fig. 10. Contract Events Log

The "Events Log for ‹contract name›" grid contains the following fields:

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

the Event is opened automatically or the date is specified if the Event is opened by a user command

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

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

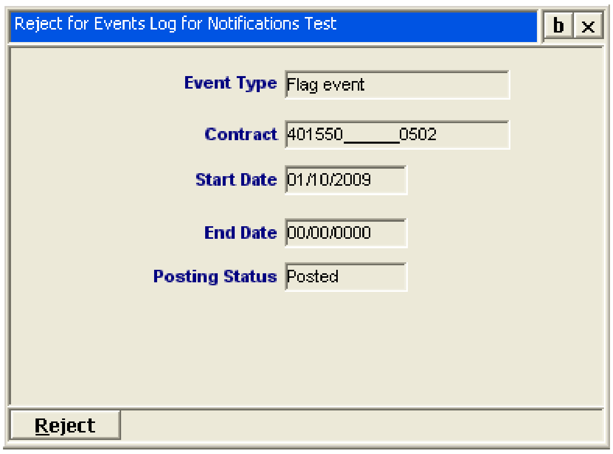
- *Event Details* – this field contains additional technical information about the Event, e.g. "INPUT=<Comments>;" means that the Event was opened manually and the <Comments> value was specified in the *Comment Details* field when opening the Event; "END=<Reason>;" means that the Event was closed, and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

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

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

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

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



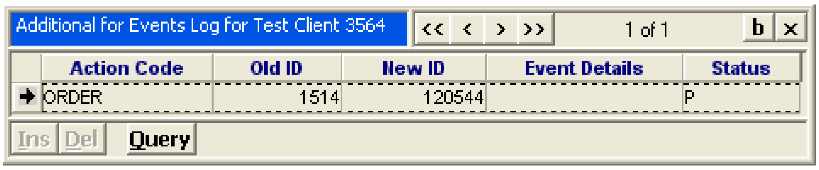
Reject for Events Log for Notifications Test

Event Type	Flag event
Contract	401550______0502
Start Date	01/10/2009
End Date	00/00/0000
Posting Status	Posted

Fig. 11. Form for manually closing Events

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

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



Additional for Events Log for Test Client 3564

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

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

This form contains the following fields:

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

5.4 "Requested Event Uploading" Procedure

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

All Events processed by Way4 are registered in the Event log in the USAGE_ACTION table. Only a small number of Events are "Requested Events".

Way4 provides for a mechanism to synchronise and quickly search for "Requested Events" by contracts. Using this mechanism, the time taken for a response to an inquiry on the activity or inactivity of an Event for a contract is significantly shortened.

Way4 itself marks the required Event types as "Requested Events". These Event types initially fall into a special "Not Ready" collection. Immediately after marking, all newly opening and closing Events of this type for contracts begin synchronisation to the special table ADD_PACK_INC. Not all information from

USAGE_ACTION is synchronised, only a part of it, which significantly saves on data volume.

Since synchronisation is started after marking by Way4, all Events of this type opened earlier remain unsynchronised (i.e. they are contained in the USAGE_ACTION table and absent from the ADD_PACK_INC). In the case of Event types recently registered in Way4, this may be just a few unsynchronised records. In the case of Event types that have long been in existence in Way4, but that were only recently marked by the system, there may be a significant number of unsynchronised records. This makes it impossible to use the quick search mechanism, since it is unknown for which contracts which Event types are synchronised. As a result, instead of limiting the search by the small ADD_PACK_INC table, Way4 continues to search in the large USAGE_ACTION table.

The "Requested Events Uploading" procedure fully synchronises these Event types. After full synchronisation, the Event type goes from the "Not Ready" collection to the "Ready" collection. For Way4, this move is a sign that synchronisation has been executed and in the ADD_PACK_INC table data for this Event type are full and reliable. This allows optimisation of these Event searches, both by contract and by Event type. And, correspondingly, to decrease the time it takes to execute large-scale procedures. Data for each Event type is fully synchronised once.

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

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

Recommendations for starting the "Requested Event Uploading" procedure:

- **Initial start** of the procedure is recommended to be done separately from daily procedures since it may take a significant amount of time (due to a large number of records in the "Not Ready" collection).
- **Daily repeat start** after initial execution of the "Requested Event Uploading" procedure is recommended to be done every day. The frequency and means of running the procedure depends on the number of records in the "Not Ready" collection of the DM_COLLECTION table. Before executing the procedure, it is recommended to estimate the number of "Not Ready" records (estimate the number of Event types for subsequently starting the "Requested Events Uploading" procedure) and, respectively, the time it will take to process them, using the menu item "Full →Daily Procedures →Start of Day Step By Step →Requested Event Types Check":

- If there are no records in the "Not Ready" collection, it is not necessary to run the procedure. If there are no records in the "Not Ready" collection, the procedure will take a few seconds.
- If the number of records in "Not Ready" > 0, the menu item should be started:
    - ♦ In daily procedures, if there is a small number of records.
    - ♦ Separately from daily procedures using the menu item "Full →Daily Procedures → Start of Day Step By Step → Requested Events Uploading", if the number of records in the "Not Ready" is such that it is estimated their processing will be lengthy.

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

After running the "Requested Events Uploading" for the first time, the appearance of new records in the "Not Ready" collection may be cause by the following actions in Way4:

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

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

6 Examples of Event Use

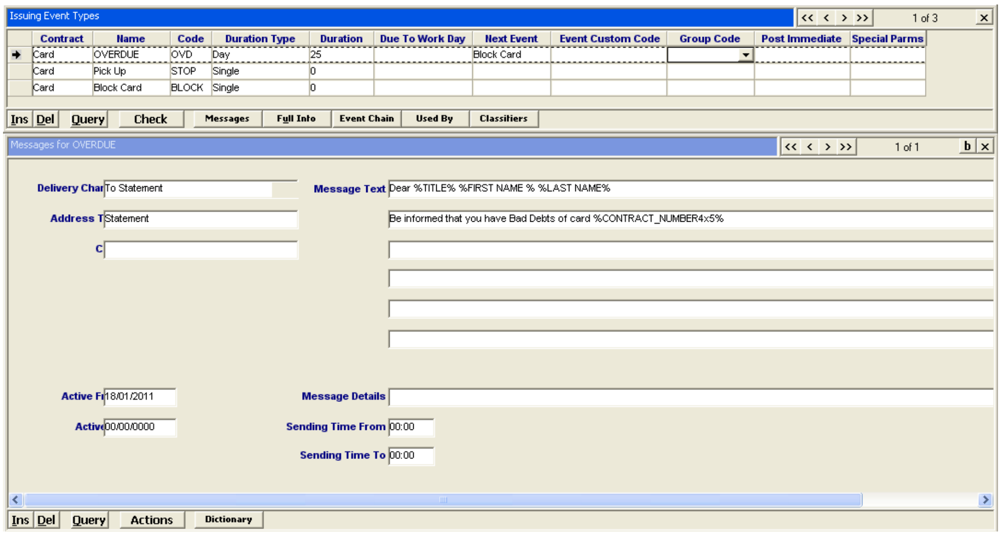
6.1 Overdue Debt

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

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

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

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



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

Delivery Char: To Statement

Address T: Statement

C:

Message Text: Dear %TITLE% %FIRST NAME % %LAST NAME%
Be informed that you have Bad Debts of card %CONTRACT_NUMBER4x5%

Active Fr: 18/01/2011

Active: 00/00/0000

Message Details:

Sending Time From: 00:00

Sending Time To: 00:00

Fig. 13. Example of system Event setup

When the system is set up as described above, the following processes will be activated. If the loan is not repaid on time, funds will be transferred to the "Cl OVD" account according to due normalisation. The "Overdue" Event indicated in the *Open/Close Event* field of the "Cl OVD" template will open, and the database will create a message according to the message template (see Fig. 13). The client will receive this message in the next statement.

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

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

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

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

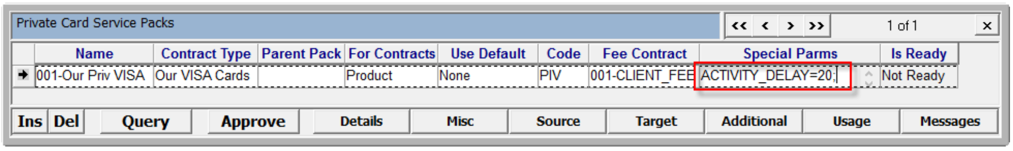
6.2 No Contract Activity

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

Setup for these Events is illustrated in the following example: an Event with the CONTRACT_DELAY20 code will activate if there were no financial transactions with the contract for 20 days and an Event with the code CONTRACT_DELAY40 – if there were no financial transactions with the contract for 40 days.

Setup procedure:

- The period of no contract activity is set up using the ACTIVITY_DELAY tag in the Service Package (see the description of the tag in the section "Tags used when configuring Service Packages and Services" of the document "Setup Tags").

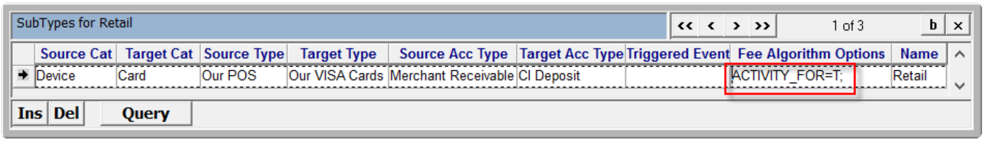


Private Card Service Packs

Name	Contract Type	Parent Pack	For Contracts	Use Default	Code	Fee Contract	Special Parms	Is Ready
001-Our Priv VISA	Our VISA Cards		Product	None	PIV	001-CLIENT_FEE	ACTIVITY_DELAY=20;	Not Ready

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

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



SubTypes for Retail

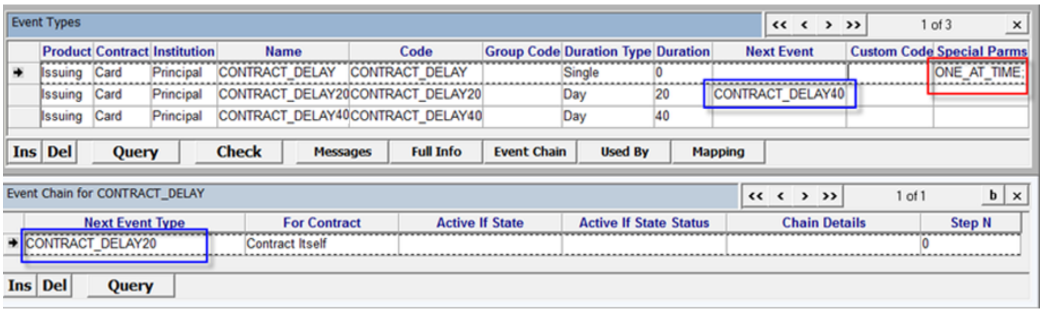
Source Cat	Target Cat	Source Type	Target Type	Source Acc Type	Target Acc Type	Triggered Event	Fee Algorithm Options	Name
Device	Card	Our POS	Our VISA Cards	Merchant Receivable	CI Deposit		ACTIVITY_FOR=T;	Retail

Fig. 15. Setting up a transaction subtype

- Event type setup (see Fig. 16):
  - An Event type with the predefined code CONTRACT_DELAY is set up. The Event opens if there were no transactions with the contract for 20 days (since the ACTIVITY_DELAY=20; tag is set in the Service Package, see Fig. 14). Set the tag ONE_AT_TIME; in this Event type and the "Single" type so the Event doesn't open each time daily procedures are run (Contracts – Daily update).

    If the period without contract activity exceeds the number of days specified with the ACTIVITY_DELAY tag, the CONTRACT_DELAY=Y tag is automatically set in the contract.
  - For each period (20, 40 days without contract activity) a separate Event type is set up. In the example, these are Event types with the CONTRACT_DELAY20 and CONTRACT_DELAY40 codes. An Event chain is used to set up opening these Events as follows:
    - ♦ The CONTRACT_DELAY20 Event type is set up in an Event chain for the CONTRACT_DELAY main Event type. When a CONTRACT_DELAY Event opens, a CONTRACT_DELAY20 Event opens automatically (and the corresponding fee is charged).
    - ♦ If no financial transactions were made with a contract in 40 days, the CONTRACT_DELAY20 Event closes (goes to the "InActive" status) and the CONTRACT_DELAY40 Event specified in the *Next Event* field of the CONTRACT_DELAY20 Event opens, see Fig. 16.

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



Event Types

	Product	Contract	Institution	Name	Code	Group Code	Duration Type	Duration	Next Event	Custom Code	Special Parms
➔	Issuing	Card	Principal	CONTRACT_DELAY	CONTRACT_DELAY		Single	0			ONE_AT_TIME;
	Issuing	Card	Principal	CONTRACT_DELAY20	CONTRACT_DELAY20		Day	20	CONTRACT_DELAY40		
	Issuing	Card	Principal	CONTRACT_DELAY40	CONTRACT_DELAY40		Day	40			

Event Chain for CONTRACT_DELAY

	Next Event Type	For Contract	Active If State	Active If State Status	Chain Details	Step N
➔	CONTRACT_DELAY20	Contract Itself				0

Fig. 16. Example of Event setup

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

6.3 Notification that an Amount Available's is under the Minimum Threshold

If during a transaction, a contract's amount available goes under a certain threshold, a notification can be sent to the client; to do so:

- Configure a separate Event with the "Unique" type:
  - In the Event's *Special Parms* field, specify the tags MAX_BAL_TYPE=AVAILABLE;LIMIT_AMOUNT=<code of a tariff type with the Threshold role>; to check the corresponding balance type and compare it with the threshold set using a tariff. Tariff setup is described in the document "Way4 Advanced Tariff Management".
  - Set up notification sending when this Event opens. I.e. a message will be generated if the amount available goes under the set threshold.
- Add the configured Event to the event chain for hardcoded Events that are used for standard notifications about the corresponding transaction. For example, to check the threshold and send a message during authorisation – to the package for the Event with the code AUTH_T_Q (for more information, see the section "Event Setup" of the document "Transaction Notifications". I.e. a notification that the specified threshold has been exceeded is sent in addition to the standard notification about the transaction.
- If the amount available is less than the specified threshold, a notification will be generated for each transaction (in our example, for each authorisation) until the balance is topped up.