

Operation Manual

Contract Functional Dates

03.50.30

15.07.2020

Contents

1 FUNCTIONAL DATES	3
2 CONFIGURING RULES FOR CALCULATING FUNCTIONAL DATES	5
2.1 General Setup Scheme and Main Parameters for Calculating Dates	5
2.2 Additional Parameters for Calculating Dates	11
2.3 Specifics of Setup for Calculating Billing Date	14
2.3.1 Rules for Calculating Billing Date	15
2.3.2 Defining the Billing Day Parameter with Contract Parameters/Classifiers	17
2.3.3 Defining the Billing Day Parameter with an Event	18
2.3.4 Settings for Limiting Billing Cycle Length	18
2.3.5 Recommended Settings for Billing Date	20
3 USING FUNCTIONAL DATES	23
3.1 Processing Due Normalisation	24
3.2 Processing Standing Payment Orders	25
3.3 Charging Misc Fees	25
4 RECALCULATING FUNCTIONAL DATES	26
4.1 Entering a New Date or Recalculating a Date in Customer Service Form	26
4.2 Recalculating Billing Date	27
4.2.1 Recalculation according to an Event	27
4.2.2 Recalculation when a Classifier Value Changes	27
4.2.3 Recalculation when a Contract Parameter Changes	28
4.2.4 Recalculation through the "Recalc Billing Cycle" Form	28

This document is intended for bank or processing centre employees responsible for configuring Way4 and contains information about setup of rules for calculating contract functional dates.

While working with this document, it is recommended that users refer to the following reference material from Way4 documentation series:

- Products and Contract Subtypes
- Way4 Accounting Schemes
- Way4 Global Parameters
- Way4 Instalments
- Daily Procedures

The following conventions are used throughout this document:

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



Warnings about potentially hazardous situations or actions.



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

1 Functional Dates

Contract functional dates are dates used to work with credit cards (work with contract loan debt).



Main functional dates are calculated when opening a new billing cycle during execution of the "Contracts Daily Update" procedure. Dates are calculated for one (opening) billing cycle. Calculated dates are saved. If calculation parameters change, these changes generally will be applied in calculating dates when opening the next billing cycle.

Dates (except Due Date dates) can be changed in the "Customer Service" form, see the section "Entering a New Date or Recalculating a Date in Customer Service Form".

Contract functional dates include the following main dates:

- "Due Date" (DUE_DATE) – date shown to the client in a statement as the scheduled date a loan payment is due. This date can be the basis for calculating other contract dates.
- "Full Payment Date" (FP_DATE) – date of full payment to meet grace period conditions (Grace Date). The date by which the full amount for a billing cycle must actually be paid (with consideration of the permissible discrepancy – Variance), specified in the statement, so that interest is not charged for this period. Usually the "Full Payment Date" corresponds to the "Due Date" or exceeds the "Due Date" by several days.



Note that to use this date, an Event must be set up with the predefined code FULL_PAYMENT (i.e. interest will not be charged only if an Event with this code opens). In addition, accounting scheme account templates must be set up. For more information, contact the Way4 vendor.

Conditions are checked and a FULL_PAYMENT Event is opened during the "Contracts – Daily Update" procedure when opening the day corresponding to the calculated "Full Payment Date".

If the IN_THE_MORNING=N; tag is used, conditions are checked and a FULL_PAYMENT Event is opened during the "Contracts – Daily Update" procedure when closing the day preceding the calculated "Full Payment Date".

- "Late Payment Date" (LP_DATE) – if the minimum amount of a loan payment was not made by this date, a fee is charged for the minimum payment being overdue.



To use this date (i.e. to charge a fee when this date arrives), an Event must be set up with the predefined code LATE_PAYMENT. For more information, contact the Way4 vendor.

Conditions are checked and a LATE_PAYMENT Event is opened during the "Contracts – Daily Update" procedure when opening the day corresponding to the calculated "Late Payment Date".

If the IN_THE_MORNING=N; tag is used, conditions are checked and a LATE_PAYMENT Event is opened during the "Contracts – Daily Update" procedure when closing the day preceding the calculated "Full Payment Date".

- "Delinquency Date" (DLQ_DATE) – date the debt becomes delinquent (funds are transferred to a delinquent debt account). This date works for account templates with the "Contract Due" value of the *Due Type* parameter. Several accounts may be set up for different terms of delinquency. When a DLQ_DATE arrives, funds are moved to the corresponding accounts, depending on the term of delinquency.
- "Billing Date" (BILL_DATE) – billing cycle end date.
- "Direct Debit Date" (DD_DATE) – this value is used to set up payment of loan debt from an account in another bank. For example, for SEPA, the DD_DATE date is used as the date a Collection is generated for a client. See the document "SEPA Interface Setup".

When working with instalment loans, the following functional dates can be used in the Way4 Instalments module ("Instalment Dates" group).

- "Instalment Effective Date" (INSTL_EFF_DATE) – date an instalment becomes effective.
- "Instalment Due Date" (INSTL_DUE_DATE) – scheduled due date of instalment (date by which the instalment must actually be paid).
- "Instalment Report Date" (INSTL_REP_DATE) – scheduled date of payment shown in a report for the client. Generally corresponds to "Instalment Due Date".

For more information, see the document "Instalment Loans in Way4™". The Way4 Instalments module is supplied according to a separate agreement with the Way4 vendor.

2 Configuring Rules for Calculating Functional Dates



When configuring parameters for calculating a contract's main functional dates (DUE_DATE, FP_DATE, LP_DATE, DD_DATE, DLQ_DATE), note that the calculated date must be less than or equal to the date of the next's billing cycle's last day.

Example.

The billing cycle 01.06.2020 – 30.06.2020 is being opened.

A contract's main functional dates calculated for the cycle (DUE_DATE, FP_DATE, LP_DATE, DD_DATE, DLQ_DATE) must be less than or equal to 31.07.2020.

A calculated date that is greater than the last day of the next billing cycle (for example, after 31.07.2020) may lead to incorrect system behavior. In particular, interest will not be waived in "Waive After Full Payment" interest accrual mode, FULL_PAYMENT or LATE_PAYMENT Events will not be generated, orders and fees that are related to these contract dates (with "Use Contract Date" in the Date Event and Charge Event fields, respectively) will not be processed.

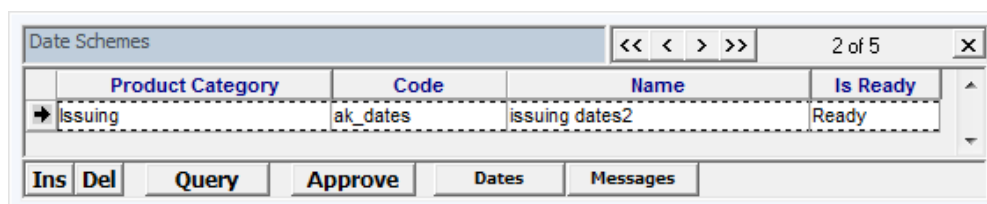
2.1 General Setup Scheme and Main Parameters for Calculating Dates



This section describes specifics of setting up the main contract functional dates (DUE_DATE, FP_DATE, LP_DATE, DD_DATE, and DLQ_DATE). For detailed information about functional dates used in the Way4 Instalments module, see the document "Instalment Loans in Way4".

Contract functional dates are set up as follows:

1. Configure a scheme for calculating functional dates in the "Date Schemes" form (Full → Configuration Setup → Products → Date Schemes), see Fig. 1.



Product Category	Code	Name	Is Ready
Issuing	ak_dates	issuing dates2	Ready

Fig. 1. "Date Schemes" form

The "Date Schemes" form contains the following fields:

- *Product Category* – Product category. A scheme is set up for a specific Product category (Issuing/Acquiring/Bank Accounting). For one Product category, several date schemes can be set up, that will be used in different Products.
- *Code* – date scheme code.
- *Name* – date scheme name.
- *Is Ready* – indicates whether the date scheme was approved:
 - "Ready" – date scheme has been approved.
 - "Not Ready" – date scheme has not been approved.



If the date scheme has not been approved, the corresponding Product for which the scheme was specified will have the "Not Ready" status.

2. Configure rules for calculating functional dates (or date templates). This is done in the "Dates for <name of date scheme>" form opened by clicking the [Dates] button in the "Date Schemes" form (see Fig. 2).

Dates for only due dates							
<< < > >> 1 of 5 b x							
Date Type	Base Date	Shift Base Date	Period	Period Unit	Period Tariff	Shift Result Date	Additional information
Due Date	First Day of Cycle	Always to the prev wrk day	0	Calendar Day	No	No	Ready
Full Payment Date	Contract Due Date	No	4	Calendar Day	No	No	Ready
Late Payment Date	Contract Due Date	No	5	Calendar Day	No	No	Ready
Delinquency Date	Contract Due Date	No	2	Calendar Day	No	No	Ready

Fig. 2. "Dates for <name of date scheme>" form

The "Dates for <name of date scheme>" form contains the following fields:

- *Date Type* – type of functional date (see the section "Functional Dates").
- *Base Date* – start date of calculation when calculating a functional date:
 - "First Day of Month" – the first day of the month in the billing cycle being opened. For example, when opening the billing cycle 25.11 – 24.12, calculation will start from 01.12.



If a functional date calculated according to the "First Day of Month" scheme corresponds to the first date of the current billing cycle, a shift to the next month is not made. FP_DATE and LP_DATE are exceptions. If they match, they will be shifted to the next day since these dates do not have to correspond to the first day of a billing cycle.

- "First Day of Cycle" – the first day of the billing cycle being opened.

- "First Day of Next Cycle" – the first day of the next billing cycle (billing cycle following the one being opened).
- "Last Day of Cycle" – the last day of the billing cycle being opened.
- "Contract Due Date" – the DUE_DATE date is the start of calculation.

The DUE_DATE date is calculated first. If the value of the *Base Date* parameter is "Contract Due Date", this date can be used to calculate the other types of functional date.

- "Last Day of Month" – Calculation begins on the last day of the previous billing cycle.



Only "Last Day of Month" can be used for BILL_DATE (Billing Date).

- "Instalment Trans Date" – the start of calculation is the date of the transaction for which the instalment plan is being calculated. This value is used to calculate dates for the "Instalment Dates" group when working with instalment loans. For more information, see the document "Instalment Loans in Way4™". The Way4 Instalments module is provided according to a separate agreement with the Way4 vendor.



Dates are calculated when opening a new billing cycle for a contract.

- *Period* – the period added to the *Base Date* parameter, for calculating a functional date.
- *Period Unit* – unit for calculating the period specified in the *Period* field:
 - "Working Day" – in working days.
 - "Calendar Day" – in calendar days (this is the default value used if the field is not filled in).
 - "Month" – in months.

The period for calculating a date can be set using a tariff with the "Service Value Days" role and with a code corresponding to the type of functional date (DUE_DATE, FP_DATE, LP_DATE, DLQ_DATE, DD_DATE). Settings for a period to calculate a date on the tariff level have a higher priority than settings in the "Dates for..." form. If no tariff is set, or the tariff has the "0" value, the value of the *Period* parameter from the "Dates for..." form will be used.

- *Period Tariff* – selection of a tariff with the "Service Value Days" role. This field makes it possible to redefine tariffs with preset codes (DUE_DATE, FP_DATE, LP_DATE, DLQ_DATE, DD_DATE) with the "Service Value Days" role (see above).

- *Shift Base Date* – shift the *Base Date* date to a working day, if this date falls on a non-working day.
 - "No" – do not shift the date to a working day.
 - "Holiday to the next wrk day" – if the *Base Date* date falls on a non-working day, shift the date to the next working day after the non-working days.
 - "Holiday to the prev wrk day" – if the *Base Date* date falls on a non-working day, shift the date to the last working day before the non-working days.
 - "Always to the next wrk day" – shift the date to the next working day (regardless of whether *Base Date* falls on a non-working or working day).
 - "Always to the prev wrk day" – shift the date to the previous working day (regardless of whether *Base Date* falls on a non-working or working day).
 - "Before the wrk day" – this value is used to shift BILL_DATE dates (see the section "Specifics of Setup for Calculating Billing Date").

If the *Shift Base Date* field is not filled in, the value of the global parameter SHIFT_DATE_FROM_TO_WRK_DAY will be used when calculating dates.

- *Shift Result Date* – shift a calculated functional date to a working day, if this date falls on a non-working day (shift the date obtained after adding the period from the *Period* field to the *Base Date* date).
 - "No" – do not shift the date to a working day.
 - "Holiday to the next wrk day" – if the date falls on a non-working day, shift the date to the next working day after the non-working days.
 - "Holiday to the prev wrk day" – if the date falls on a non-working day, shift the date to the last working day before the non-working days.
 - "Always to the next wrk day" – shift the date to the next working day (regardless of whether the date falls on a non-working or working day).
 - "Always to the prev wrk day" – shift the date to the previous working day (regardless of whether the date falls on a non-working or working day).

If the *Shift Result Date* field is not filled in, the value of the global parameter SHIFT_TO_WRK_DAY will be used when calculating dates.

The *Shift Base Date* and *Shift Result Date* fields can be set using a tariff with the "Service Value Days" role and the code corresponding to the functional date type (DUE_DATE, FP_DATE, LP_DATE, DLQ_DATE, DD_DATE). To do so, the SHIFT_DATE_FROM_TO_WRK_DAY and SHIFT_TO_WRK_DAY tags (with the values "Y", "N", "P", "+", "-") are used in the tariff's *Apply Rules* field (for

more information, see the section "Tags used when working with tariffs" of the document "Setup Tags"). Tariff settings have a higher priority than the settings in the "Dates for..." form.

- *Additional Information* – field for entering additional tagged parameters, for example:
 - `IN_THE_MORNING=N`; – this tag (with the "N" value) can be used when setting up "Full Payment Date" or "Late Payment Date" dates only when separate procedures for opening and closing a business day are used. The tag can be set in a date template or in a tariff used in the corresponding date template. The `IN_THE_MORNING` tag with the "N" value makes it possible to process `FULL_PAYMENT` or `LATE_PAYMENT` Events during the evening procedure for processing contracts ("Contracts – Daily Update") when closing the day preceding the calculated date for "Full Payment Date" or "Late Payment Date", respectively. By default, Events are processed in the morning procedure for processing contracts, after entering the new banking date corresponding to the "Full Payment Date" or "Late Payment Date" date.
 - The `IN_THE_MORNING` tag does not affect the calculation of a functional date. The tag only affects when the corresponding Events are processed.
 - `PAYMENT_DUE_ADVANCE`, `DUE_TO_WRK_DAY`, `CALENDAR_TYPE`. See the section "Additional Parameters for Calculating Dates".

Note that the `PAYMENT_DUE_ADVANCE` and `DUE_TO_WRK_DAY` tags affect definition (calculation) of a functional date, whereas the `IN_THE_MORNING` tag defines when `FULL_PAYMENT` and `LATE_PAYMENT` Events are processed.

Examples of how different combinations of these parameters work.

Calculation of `FP_DATE`. The value Contract Due Date is set in the Base Date field; the period added to the Base Date is 2 days. The following parameters are set:
`IN_THE_MORNING=N`; `PAYMENT_DUE_ADVANCE=N`; `DUE_TO_WRK_DAY=Y`;

Example 1. `DUE_DATE` falls on Monday 20/02, and the calculated `FP_DATE` falls on 22/02 (Wednesday). If a payment is made on 21/02, it is recorded in evening CDU when "Full Payment" conditions are checked. I.e. the `FULL_PAYMENT` event will be processed when Tuesday is closed (the date of processing the Event will be 22/02).

Example 2. `DUE_DATE` falls on Saturday 20/05, and the calculated `FP_DATE` falls on 23/05 (Tuesday), since Sunday 21/05 is a weekend according to the calendar. If a payment is made on 22/05 (Monday), it is

recorded in evening CDU when "Full Payment" conditions are checked. I.e. the FULL_PAYMENT event will be processed when Monday is closed (the date of processing the Event will be 22/05).

If the IN_THE_MORNING tag is not set (in Example 2) or its value is "Y", FP_DATE is calculated the same way (FP_DATE falls on Tuesday 23/05) but the FULL_PAYMENT Event will be processed in morning CDU on Tuesday, not in evening CDU on Monday (the date of processing the Event will be 23/05).

- CALENDAR_TYPE=<name of business calendar type> - if the CALENDAR_TYPE tag is specified in the date template, the date is calculated according to this calendar's rules (i.e. when calculating dates, separate rules can be used for calculating non-working days/working days).

The tag DUE_TO_WRK_DAY=Y; is always used in combination with the tag PAYMENT_DUE_ADVANCE="N";. If the value of the tag PAYMENT_DUE_ADVANCE is "Y", this setting will duplicate the date scheme parameter *Shift Result Date To Working Date* with the "Y" value.

If the tags are not set, the tags of the same name set on the Accounting Scheme level are used for calculation, or the values of the global parameters of the same name.

- *Is Ready* - indicates whether changes in rules for determining the base date have been approved:
 - "Ready" - changes have been approved.
 - "Not Ready" - changes have not been approved.



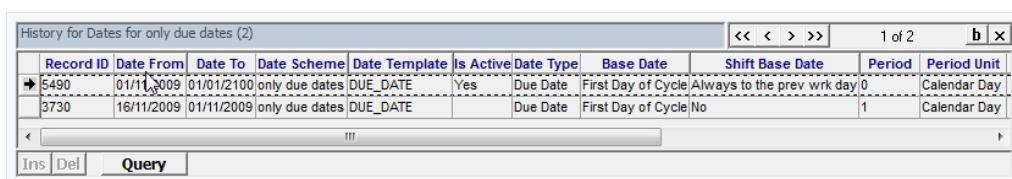
If rules for calculating a date have not been approved, the corresponding date scheme will have the "Not Ready" status and the corresponding Product for which this scheme is specified will have the "Not Ready" status.

To approve a date scheme, click [Approve] in the "Date Schemes" form. A date scheme can also be approved using the "Renew All For Institution" procedure (Full → Configuration Setup → Main Tables → Renew All For Institution).

The "Due Date" date (DUE_DATE) is the main date for calculation. If rules for calculating the "Due Date" date (DUE_DATE) have not been configured for a date scheme, this scheme cannot be approved.

- When approving a date scheme, if a date from the "Due Dates" group was configured for the scheme, a check is made for a configured "Due Date" date.
- If a date scheme is specified in the Product level, when approving the Product, a search is made for a configured "Due Date" in all cases (including if not a single date was configured for the scheme).
- When approving a date scheme, a check is made whether date templates from the "Due Dates" group were deleted. If date templates were deleted, a check is made whether these templates are used in Products with this scheme (for account templates in Accounting Schemes specified in these Products). If a date used in a Product was deleted, an error message will be displayed and the date scheme will not be approved.

The [History] button of the "Dates for..." form (see Fig. 2) opens the "History for Dates for..." form that shows the history of changes to functional date parameters (see Fig. 3).



Record ID	Date From	Date To	Date Scheme	Date Template	Is Active	Date Type	Base Date	Shift Base Date	Period	Period Unit
5490	01/11/2009	01/01/2100	only due dates	DUE_DATE	Yes	Due Date	First Day of Cycle	Always to the prev wrk day	0	Calendar Day
3730	16/11/2009	01/11/2009	only due dates	DUE_DATE		Due Date	First Day of Cycle	No	1	Calendar Day

Fig. 3. Viewing the history of changes to date parameters

The form contains a record with the functional date's current approved parameters and the history of changes to this date's parameters (historic records).

The *Is Active* field value of the record with the functional date's current approved parameters is "Yes". These parameters are used when calculating a functional date for a contract. When a date scheme is approved, a new record with the "Yes" value of the *Is Active* field is added and the previous record is deactivated.



An expanded set of contract functional dates can be configured in Way4. To do so, custom functional dates are used. These settings can only be used after agreement with the Way4 vendor.

2.2 Additional Parameters for Calculating Dates

This section describes parameters that affect the calculation of dates. These parameters can be set in date templates, in tariffs with the "Service Value Days"

role that correspond to date templates, in Accounting Schemes, financial institutions, in global parameters, etc.

List of parameters:

- Calendar type – the parameter affects how holidays and working days are determined when calculating a contract functional date. The parameter is set in a financial institution's *Calendar Type* field and can be redefined with the `CALENDAR_TYPE=<name of business calendar type>` tag. The value of the parameter is determined according to priority (in descending order):
 - Tag in the date template.
 - Tag in the tariff linked with the date template.
 - Tag in the Accounting Scheme.
 - If the tag is not specified, the calendar type specified for the financial institution is used (*Calendar Type* field).



It is not recommended to use the calendar setting in date templates and in their tariffs, except when different calendars must be used to calculate different contract dates.

- The `DUE_TO_WRK_DAY` and `PAYMENT_DUE_ADVANCE` parameters affect the date calculation algorithm. **In general, these parameters are set globally or in an Accounting Scheme and affect not only the calculation of a date, but also due normalization. If necessary, these parameters can be defined in date templates or in the corresponding tariffs to achieve specific objectives. In this case, parameter values only affect date calculation without affecting the mechanism of due normalization, etc.** When calculating dates, the values of the `DUE_TO_WRK_DAY` and `PAYMENT_DUE_ADVANCE` parameters are determined according to priority:
 - Tag in the date template.
 - Tag in the tariff linked with the date template.
 - Tag in the Accounting Scheme.
 - Global parameter. The default value is "N".
 - For date templates with the "Billing Date" and "Due Date" types, the value of these parameters is always equal to "N", regardless of the configuration. In this case, the *Shift Result Date* parameter can be used to shift to a working day.

For all date types, when the value of the `DUE_TO_WRK_DAY` parameter is "N", no additional date shifts (except for a shift using *Shift Result Date*.) are

made. The value of the PAYMENT_DUE_ADVANCE parameter is not analysed.

When the value of the DUE_TO_WRK_DAY parameter is "Y", a calculated date is shifted to a working day if according to the calendar the date falls on a weekend/holiday.

When the value of the DUE_TO_WRK_DAY parameter is "Y", the value of the PAYMENT_DUE_ADVANCE parameter is also analysed:

- Combination of DUE_TO_WRK_DAY="Y" and PAYMENT_DUE_ADVANCE="N":

With regard to calculation of dates, this combination of parameters affects all date types.

For this combination of parameters, a calculated date is additionally shifted to the second working day after weekends/holidays. I.e. if Saturday and Sunday are weekends, the calculated date will be shifted to Tuesday if it originally falls on Saturday, Sunday, or Monday. This makes it possible to perform due normalization or other actions related to a functional date when Tuesday's business day is opened, not when Monday's business day is opened. Since according to regulations, Monday's business day may be opened on Friday evening, it is too early to perform due normalization when the procedure for opening Monday's business day is performed.

If the calculated date falls on a working day, there will be no shift. I.e., if a date falls on Friday, for example, normalization is performed when Friday's business day is opened.

- Combination of DUE_TO_WRK_DAY="Y" and PAYMENT_DUE_ADVANCE="Y":

With regard to calculation of dates, this combination of parameters only affects calculation of "Full Payment Date" and "Late Payment Date" dates. For this combination of parameters, a date is always additionally shifted up by one calendar day. This is related to the specifics of processing FULL_PAYMENT and LATE_PAYMENT Events. See the examples below:

Example 1.

To calculate "Late Payment Date", the "Contract Due Date" base date is used (see the *Base Date* field). "0" is specified in the *Period* field. The DUE_TO_WRK_DAY="Y" and PAYMENT_DUE_ADVANCE="Y" parameters are set.

If the base date "Due Date" falls on a working day, "Late Payment Date" will be equal to "Due Date" + 1 day and the LATE_PAYMENT Event will be

processed when the next business day with the date after "Due Date" is opened. I.e. if, for example, "Due Date" falls on a Friday, "Late Payment Date" will be equal to Saturday and a LATE_PAYMENT Event will be processed when Monday's business day is opened (i.e. on the next business day that opens).

If the IN_THE_MORNING=N tag is specified in a "Late Payment Date" template or in the corresponding tariff, a LATE_PAYMENT Event will be processed when closing "Due Date" business day.

In both cases, "Late Payment Date" is equal to "Due Date" + 1 day.

Example 2.

To calculate "Late Payment Date", the "Contract Due Date" base date is used (see the *Base Date* field). "0" is specified in the *Period* field. The DUE_TO_WRK_DAY="Y" and PAYMENT_DUE_ADVANCE="Y" parameters are set. In the calendar, Saturday and Sunday are weekends. "Due Date" falls on a Saturday.

In this case, "Late Payment Date" will be equal to Tuesday. A LATE_PAYMENT Event will be processed when Tuesday's business day is opened (i.e. system behaviour in this case is the same as when DUE_TO_WRK_DAY="Y", PAYMENT_DUE_ADVANCE="N").

If the IN_THE_MORNING=N tag is set in a "Late Payment Date" template or in the corresponding tariff, a LATE_PAYMENT Event will be processed when closing Monday's business day.

In both cases, "Late Payment Date" is equal to Tuesday; i.e. the second working day after the weekend.

For information about how the global parameter PAYMENT_DUE_ADVANCE or the PAYMENT_DUE_ADVANCE tag in an Accounting Scheme affects due normalisation, see the section "Processing Due Normalisation".

2.3 Specifics of Setup for Calculating Billing Date

The end date of a billing cycle, Billing Date (BILL_DATE), is calculated based on:

- The Billing Day parameter (scheduled day on which the billing cycle ends – a specific calendar day of the month set as a whole number from 1 to 31) and
- A possible shift in this date.

A possible date shift is set in the "Dates..." form, see the section "Rules for Calculating Billing Date".

The Billing Day parameter can be set:

- In the *Period* field of the "Dates..." form, see the section "Rules for Calculating Billing Date".
- Using a tariff set in a date scheme for Billing Date (tariff with the "Billing Scheme" role). See the description of the *Period* field in the section "General Setup Scheme and Main Parameters for Calculating Dates" and the section "Tariffs with the "Billing Scheme" Role" of the document "Way4™ Advanced Tariff Management".

Note that when the numeric value of a tariff with the "Billing Scheme" role are changed in the middle of a billing cycle, the new values will be used for calculation in the next billing cycle.

- With a contract parameter. See the section "Defining the Billing Day Parameter with Contract Parameters/Classifiers".
- With an Event. See the section "Defining the Billing Day Parameter with an Event".
- Using a tariff with the "Billing Date" role. See the section "Tariffs with the "Billing Scheme" Role" of the document "Way4™ Advanced Tariff Management" and the "Billing Tariff" section of the document "Way4™ Accounting Schemes".
- In the *Billing Day* field of an Accounting Scheme. See the section "Billing Day" of the document "Way4™ Accounting Schemes".



It is not recommended to set the Billing Day parameter in an Accounting Scheme, using Events, and using a tariff with the "Billing Date" role. It is also not recommended to set a date shift using the EOC_TO_WRK_DAY and TO_WRK_DAY tags in an Accounting Scheme. This functionality remains for backward compatibility.

Settings limiting the length of a billing cycle may affect calculation of Billing Date (see the section "Settings for Limiting Billing Cycle Length").

The Billing Date calculated for the current billing cycle is specified in a contract's *next_billing_date* field.

2.3.1 Rules for Calculating Billing Date

Fig. 4 shows a sample setup of rules for calculating Billing Date.

Date Type	Base Date	Shift Base Date	Period	Period Unit	Period Tariff	Shift Result Date	Additional information	Is Ready
Billing Date	Last Day of Month	No	7	Calendar Day	No		CALENDAR_TYPE=EXTRA	Ready

Ins Del Query History

Fig. 4. Rules for calculating Billing Date

The form's fields should be filled in as follows:

- Specify "Billing Date" in the *Date Type* field.
- Specify "Last Day of Month" in the *Base Date* field.
- Specify "No" in the *Shift Base Date* field. The start date of calculation cannot be shifted. This date is calculated from the last date of the previous billing cycle.
- Specify "Calendar Day" in the *Period Unit* field.
- The value of the Billing Day parameter is set in the *Period* field. This is a whole number from 1 to 31 that will be added to *Base Date* (i.e. to the last day of the previous billing cycle) to calculate the Billing Date functional date.

For the values 29, 30, 31 when calculating the actual date for the end of a billing cycle, the last day of the month will be used in cases when the number of days in the month is less than the number specified.



A specific day of the month is explicitly specified in the *Period* field. This day will be considered the last day of the billing cycle (if the *Shift Result Date* field is not filled in).

- The following values can be set in the *Shift Result Date* field:
 - "No" – no shift.
 - "Holiday to the next wrk day" – the billing cycle's end date is shifted to a working day if it falls on a weekend/holiday.
 - "Before the wrk day" – the billing cycle's end day is shifted so the start date of the next billing cycle falls on a working day.
- The *Period Tariff* field is not mandatory. As for other functional dates, the code of a tariff type with the "Service Value Days" role can be set in this field (by default, a search is made for a tariff with the BILL_DATE code). The Billing Day parameter can be redefined using a contract custom parameter (see the section below).
- Tags redefining where to search for the Billing Day parameter can be set in the *Add Info* field:

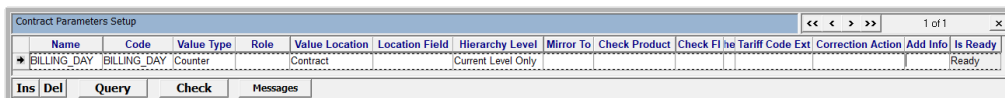
- Tags affecting limits on the length of a billing cycle (MIN_BILLING, MAX_BILLING, FIRST_BILLING). See the section "Maximum Length of a Billing Cycle when Recalculating the Current Date".
- Tags defining the minimum length of a billing cycle depending on other type of dates in a scheme. See the section "Minimum Length of the Current Billing Cycle Depending on other Contract Dates".

2.3.2 Defining the Billing Day Parameter with Contract Parameters/Classifiers

To use contract custom parameters to redefine the *Billing Day* parameter, register a contract parameter or classifier in the "Contract Parameters Setup" form (Full → Configuration Setup → Common Handbooks → Contract Parameters Setup), see Fig. 5.



If the corresponding parameter is registered in the "Contract Parameters Setup" form and specified in the contract, the contract's Billing Date parameter will be determined using this contract custom parameter (i.e. this setting of the Billing Day parameter has a higher priority).



Name	Code	Value Type	Role	Value Location	Location Field	Hierarchy Level	Mirror To	Check Product	Check File	Tariff Code Ext	Correction Action	Add Info	Is Ready
BILLING_DAY	BILLING_DAY	Counter		Contract		Current Level Only							Ready

Fig. 5 Defining the Billing Day parameter with a contract parameter

Fill in "Contract Parameters Setup" form fields as follows:

- Specify "BILLING_DAY" in the *Code* field.

By default, a parameter with the BILLING_DAY code is used when searching for a *Billing Day* parameter. When date schemes are used, any parameter code can be used to calculate Billing Date. To do so, specify the PERIOD_PARM=<contract parameter>; code in the *Additional Information* field in rules for calculating Billing Date (see Fig. 2 in the section "General Setup Scheme and Main Parameters for Calculating Dates").

- *Name* – custom name of parameter.
- Specify "Counter" in the *Value Type* field.
- *Value Location* – the following values can be set in this field:
 - "Contract" – a search for the parameter's value will be made in the contract.
 - "Classifier" – a search for the parameter's value will be made in the corresponding classifier.

- *Hierarchy Level* – defines the hierarchy level in which the parameter is set if "Contract" is specified in the *Location of Current Value* field. When "Classifier" is specified in the *Location of Current Value* field, the *Hierarchy Level*/field is not filled in.
- *Mirror To* – the following values can be set in this field:
 - If the field is not filled in changes are not logged.
 - "Classifier" – changes to the parameter's value are logged in the classifier's history. In this case, a classifier with the same code and list of values from 1 to 31 must be registered.

For more information, see the section "Contract and Client Custom Parameters" of the document "Way4™ Client and Contract Classifiers".

2.3.3 Defining the Billing Day Parameter with an Event



This functionality remains for backward compatibility. If the first time Way4 was installed, version 03.44.30 or higher was installed, it is not recommended to use this method. It is recommended to use a contract parameter to define the Billing Day parameter (see the section "Defining the Billing Day Parameter with Contract Parameters/Classifiers").

Register a hardcoded BILLING_DAY Event type. In the *Special Params* field of this Event type, set the DAY=<Billing Day parameter value>; tag.

The Event type code can be set as the value of the PERIOD_EVENT=BILLING_DAY; tag in the *Additional Information* field in rules for calculating Billing Date (see the section "General Setup Scheme and Main Parameters for Calculating Dates").

In this setup, when searching for a parameter value, this activity of this Event for the contract is analysed.



An Event can be used to change the value of the classifier in which the value of the Billing Day parameter for the contract is stored (see the section "Changing Classifier Values with an Event" of the document "Way4™ Client and Contract Classifiers"). In this case, a regular Event type is used, i.e. the Event isn't analysed when searching for a parameter value.

2.3.4 Settings for Limiting Billing Cycle Length

2.3.4.1 One Billing Date in a Calendar Month

The MIN_BILLING=C; tag can be specified in the *Additional Information* field in a date scheme template, in a Product's *Custom Data* field, in a tariff's *Apply Rules* field (tariff with the "Billing Tariff" role), in the *Special Parameters* field of an Accounting Scheme,.

The MIN_BILLING=C; tag specifies that there may be only one Billing Date during a calendar month. If this condition is not met, a billing cycle's end date is shifted to the end date of the next billing cycle.

The tag is checked when calculating Billing Date for the first billing cycle and for subsequent billing cycles, including when a date is recalculated in the middle of a cycle.

2.3.4.2 Specific Conditions for the First Billing Cycle

The FIRST_BILLING and FIRST_BILLING_UNIT tags can be specified in a date scheme template, in a Product's *Custom Data* field, in the *Special Parameters* field of an Accounting Scheme, in a tariff's *Apply Rules* field (tariff with the "Service Value Days" or "Billing Tariff" role).

The value of the FIRST_BILLING tag redefines the value of the MIN_BILLING tag when calculating the end date of the first billing cycle.

The FIRST_BILLING=ANY; tag makes it possible to set any length for the first billing cycle.

The FIRST_BILLING=<number of days>; tag makes it possible to set the minimum length in days for the first billing cycle.

The FIRST_BILLING=<number of months>;FIRST_BILLING_UNIT=M; tags makes it possible to set the minimum length in months for the first billing cycle.

2.3.4.3 Maximum Length of a Billing Cycle when Recalculating the Current Date

The MAX_BILLING and MAX_BILLING_UNIT tags can be specified in a date scheme template, in a Product's *Custom Data* field, in the *Special Parameters* field of an Accounting Scheme, in a tariff's *Apply Rules* field (tariff with the "Service Value Days" or "Billing Tariff" role).

The tag is only checked when recalculating the end date of the current billing cycle.

If the length of the current billing cycle must be extended when changing the billing cycle's end day, the MAX_BILLING tag is checked. A check is made that the length of the billing cycle does not exceed the length in days specified as the MAX_BILLING tag value. If it does, the current billing cycle remains unchanged and a new Billing Date is applied only in the next billing cycle.



Note that this tag is not used in scheduled calculation of the next Billing Date.

2.3.4.4 Minimum Length of the Current Billing Cycle Depending on other Contract Dates

If date schemes are used, rules can be defined that limit changes to the length of the current billing cycle depending on other contract dates.

The <date type code>_RULE=<number of days>; tag in the *Additional Information* field of the Billing Date template is used to do so.

When the tag is set, a new date for the end of the billing cycle cannot be less than the specified date, plus the number of day set in the tag.

For example, the LP_DATE_RULE=2; tag means that the new date for the end of the billing cycle cannot be less than Late Payment Date plus two days.

By default, DUE_DATE is analysed in Way4, on the principle of DUE_DATE_RULE=0;. I.e. if DUE_DATE is defined in a date scheme, the new end date for a billing cycle cannot be less than DUE_DATE.

2.3.5 Recommended Settings for Billing Date

Recommended settings for Billing Date:

- To store the Billing Day parameter, it is recommended to use the BILLING_DAY contract parameter (or with any code that is set in the PERIOD_PARM=<parameter code>; tag).
- If the parameter's value must be checked from a number of allowed values – a link is configured between the parameter and a classifier (through the *Value Type, Mirror To* fields in which the "Classifier" value is set. Allowed values are listed in a classifier with the same code.
- If not all possible values (from 1 to 31) are used for the Billing Day parameter, it is recommended to use tariffs with the "Service Value Days" role to define (store) the Billing Day parameter.
- If a link between Billing Date and Due Date is required, set up through Tariff Books:
 - Set up a Book whose code corresponds to the code of the parameter responsible for the Billing Day parameter (ATTACH_BY_CODE=BILLING_DAY;).
 - Set up a domain with tariff plans so that the code of the domain with the tariff plan corresponds to a specific value of the Billing day parameter.
 - In the domain, set a tariff for calculating Due Date.
 - Selection of Due Date can be managed through the "IF_PARM_VALUE=BETWEEN...", depending on Billing Date.

- If a preconfigured Billing Date and Due Date pair of dates must be selected for contract, and one Billing Date can be combined with different Due Dates (for example, a choice of Billing Date=1, Due Date=5 and BillingDate=1, Due Date=15). In this case, it is also recommended to use setup through Tariff Books, but to set a different custom parameter, other than BILLING_DAY (another Book code) that will define the selected tariffing for dates. In this case, a tariff plan will contain a tariff type for setting Billing Day and Due Date specified in the date scheme.
- If the Billing Date for a top contract must be linked with a subordinate Liability contract, it is recommended to store Billing Date in a custom parameter at the "Top Level" and to specify possible values for the Billing Day parameter in the linked classifier. In the date template with the BILL_DATE type or in the Product, specify the RECALC_BILLING=COPY_TO_LIAB_SUBS; for synchronized copying of Billing Date to subcontracts when it is recalculated in the top contract.

3 Using Functional Dates

The scheme for calculating dates is set on the Product level in the *Date Scheme* field (see the section "Main Product Parameters" of the document "Products and Contract Subtypes").



If there is a date with the Billing Date type in a scheme, when approving a Product, a check is made that "Month" is set in the *Interval* field of the corresponding Accounting Scheme. Otherwise, an error message will be generated.

A date scheme may be inherited for child Products in a "Liability" hierarchy (when the [Approve] button's context menu command "Fill Down Date Scheme for Children" is executed in the form for the main Product) – in this case, the date scheme for the child Product is updated according to the main Product's settings. In a "Main/Sub" hierarchy, the date scheme set up in the main contract is by default used for a subcontract (i.e. there is no inheritance for sub-Products in a "Main/Sub" hierarchy).

Any of these dates can be used to process due normalisation, standing payment orders, and non-transaction fees (see the sections below). It is important to note the following:

- If an action takes place when opening a day, it will be performed when opening the day specified in the *USE_DUE_DATE* tag.
- If an action takes place when closing a day, it will be performed when closing the day specified in the *USE_DUE_DATE* tag.
- It is recommended to use a date that always falls (is shifted) on a working day (see the above description of the *Shift Result Date* parameter and the *DUE_TO_WRK_DAY* tag).
- If settings related to date shifts (*Shift Result Date* parameter, *DUE_TO_WRK_DAY* tag) were not made when configuring date templates, Accounting Scheme parameters or global parameters will be used.
- If a functional date is used that falls on a non-working day, orders, normalisations, and non-transaction fees are always processed when opening Monday's business day or when closing Monday's business day, depending on the following parameters:
 - Normalisation is performed depending on the global parameter *PAYMENT_DUE_ADVANCE* or the *PAYMENT_DUE_ADVANCE* tag in the

Accounting Scheme. If the value is "Y" – when opening Monday; if the value is "N" – when closing Monday.

- Orders are processed depending on the global parameter ORDER_IN_START_OF_DAY or the ORDER_IN_START_OF_DAY tab in the Accounting Scheme. If the value is "Y" – when opening Monday; if the value is "N" – when closing Monday.
- Non-transaction fees are processed depending on the IN_START_OF_DAY tag in the transaction subtype. If the tag is not set (or if its value is "Y") – when opening Monday; when the value is "N" – when closing Monday.

3.1 Processing Due Normalisation

Any of these dates can be used to process due normalisation. To do so, for account templates with the "Contract Due" value of the *Due Type* parameter, specify the contract's functional date that will be used for normalisation. The date is specified with the USE_DUE_DATE=<functional date code>; tag in the *Template Details* field (for example, USE_DUE_DATE=LP_DATE;). By default (if the tag is not set), to process due normalisation, the "Delinquency Date" date is used (DLQ_DATE) or "Due Date" (DUE_DATE), if DLQ_DATE is not set. For more information, see the section "Ageing" of the document "Way4™ Accounting Schemes".

If a calculated date falls on a weekend/holiday according to the financial institution's (or Accounting Scheme's) calendar, and for the financial institution every day, including weekends/holidays is opened when the "Contracts - Daily Update" procedure is performed, due normalisation will be performed when opening/closing (according to the PAYMENT_DUE_ADVANCE parameter) the first working day after the calculated date, according to the calendar of the financial institution/Accounting Scheme.

For due normalisation it is not recommended to use a functional date that can fall on a weekend/holiday. It is recommended to use setting for shifting the date to a working day (see the above description of the *Shift Result Date* parameter and DUE_TO_WRK_DAY tag).

If a functional date is used that falls on a weekend/holiday, normalisation is performed depending on the PAYMENT_DUE_ADVANCE global parameter or the PAYMENT_DUE_ADVANCE tag in the Accounting Scheme. If the value is "Y" – when opening Monday; if the value is "N" – when closing Monday.



When the value of an account template's *Due Type* field is "Contract Due", the value of the DUE_TO_WRK_DAY parameter is always interpreted in Way4 as "Y" (regardless of the global parameter's settings) and can be redefined only on the

date scheme level using the tag of the same name (see the description of the *Additional Information* field).

3.2 Processing Standing Payment Orders

Any of these dates can be used to process standing payment orders. To do so, specify "Use Contract Date" as the value of the account template's *Date Event* parameter. The functional date that will be used is specified with the `USE_DUE_DATE=<date code>` tag in the order's *Posting Date* field, for example, `USE_DUE_DATE=DUE_DATE;`. By default (if the tag is not set), "DueDate" (DUE_DATE) is used to process orders. See the section "Determining Document Frequency and Amount" of the document "Standing Payment Orders".

If a functional date is used that falls on a weekend/holiday, orders are processed depending on the `ORDER_IN_START_OF_DAY` global parameter or the `ORDER_IN_START_OF_DAY` tag in the Accounting Scheme. If the value is "Y" – when opening Monday; if the value is "N" – when closing Monday.

3.3 Charging Misc Fees

Functional dates can be used to charge non-transaction (Misc) fees. To do so, the following setup is required:

- Specify the value "Use Contract Date" in the *Charge Event* field of the transaction type (Full → Configuration Setup → Transaction Types → Fee Types).
- Specify the tag `USE_DUE_DATE=<functional date code>;` in the *Fee Algorithm Options* field of the transaction subtype. By default (if the tag is not set), "DueDate" (DUE_DATE) is used to charge fees.

The tag `IN_START_OF_DAY=N;` set in the transaction subtype's *Fee Algorithm Options* field makes it possible to charge a fee during the evening Contracts – Daily Update procedure. When processing a fee in the evening Contracts – Daily Update procedure, the fee's posting date will be the date being closed.

If a functional date is used that falls on a weekend/holiday, misc fees are charged depending on the `IN_START_OF_DAY` tag in the transaction subtype. If the tag is not set (or its value is "Y") – when opening Monday; if the value is "N" – when closing Monday.

4 Recalculating Functional Dates

4.1 Entering a New Date or Recalculating a Date in Customer Service Form

Main functional dates are calculated when a new billing cycle is opened for a contract during the "Contracts Daily Update" procedure. Dates are calculated for one billing cycle (the one being opened). Calculated dates are saved. If calculation parameters change, in general these changes will be applied for calculating dates when opening the next billing cycle.

If calculation parameters change in the middle of a billing cycle, a functional date for the current billing cycle can be explicitly changed or recalculated in "Customer Service Workbench" (Customer Service → [Contract Dates]), with consideration of the following:

- Recalculation or explicit entry of a new date is not possible for Due Date.
- Billing Date (i.e. a billing cycle's end date) can be changed as follows:
 - Use the [Recalc Date] button to recalculate a date according to parameters that were changed earlier.
 - A new "Billing Day" parameter value can be set using the [Set Billing Day] button, with subsequent recalculation of Billing Date.

It is only possible to specify a "Billing Day" parameter with the [Set Billing Day] button if there is a registered contract parameter responsible for defining the Billing Day parameter.

- For other date types, it is possible to do the following:
 - Enter a new date using the [Change] button.
 - Use the [Recalc Date] button to recalculate a date according to parameters that were changed earlier.

When recalculating or setting a date, a check is made:

- For "Full Payment" and "Late Payment" dates, if for the previous date a shift was made to next_billing_date (billing cycle end date) + 1 day (see the description of the CURRENT_CYCLE tag in the section "Tags used when working with tariffs" of the document "Setup Tags"). In this case, the new date must be less than or equal to next_billing_date + 1 day.

The corresponding hardcoded Events linked with these dates will be activated before settlement transactions related to the end of the billing cycle.

- For other date types (except Due Date), a check is made that the new date is less than or equal to `next_billing_date` (billing cycle end date).



Note that it is possible a date will not be changed or recalculated. For example, if the new Billing Date does not meet the conditions set by the `MAX_BILLING` tag (see the section "Maximum Length of a Billing Cycle when Recalculating the Current Date").

Recalculation may result in a functional date that differs from the date specified by the user. For example, with consideration of a date shift according to the *Shift Result Date* parameter. In this case, the corresponding message will be generated.

4.2 Recalculating Billing Date

In addition to the ability to recalculate Billing Date or set the Billing Day parameter in the "Customer Service" form (see the section "Entering a New Date or Recalculating a Date in Customer Service "), Way4 supports the following ways for recalculating Billing Date:

- According to an Event.
- When a classifier value changes.
- When a contract parameter value changes.
- In the "Recalc Billing Cycle" form.

4.2.1 Recalculation according to an Event

To do so, specify the `RECALC_BILLING;` tag in the Event type.

An Event may simultaneously set the value of the *Billing Day* parameter through the `DAY` tag (see the section "Defining the Billing Day Parameter with an Event") or by setting a parameter/classifier (see the section "Defining the Billing Day Parameter with Contract Parameters/Classifiers") and then recalculate Billing Date.

4.2.2 Recalculation when a Classifier Value Changes

If a change in a classifier value must cause automatic recalculation of the current Billing Date, set the `RECALC_BILLING;` tag in the classifier's *Add Info* field. Any change in the value of this classifier will call the procedure for recalculation. If it is not possible to change the current Billing Date, the corresponding Warning will be registered in the process log.

It's possible that a classifier with the RECALC_BILLING; tag is not responsible for defining the *Billing Day* parameter (i.e. this classifier may have a code other than "BILLING_DAY"). However, a change in the value of this classifier may lead to a change in the tariff responsible for defining the Billing Day parameter, or to a change in rules for selecting a tariff.

4.2.3 Recalculation when a Contract Parameter Changes

Contract parameters are not used to call the procedure for recalculating a billing cycle's end date. However, if a contract parameter is directly linked with a classifier (through the *Location of Current Value*=Classifier field) or through the *Mirror To*=Classifier field, changes in the parameter's value will lead to a change in the classifier's value. The classifier may be marked with the RECALC_BILLING; tag (see the previous section) and call the procedure for recalculating the date.

.

4.2.4 Recalculation through the "Recalc Billing Cycle" Form

The current billing cycle's Billing Date can be recalculated for all contracts using the [Recalc ALL] button in the "Recalc Billing Cycle" form (Full →DB Administrator Utilities →Special Contract Utilities →Recalc Billing Cycle. For example, after tariffs have changed.