ORTEC Workforce Scheduling 7

User Manual

Payroll

March 2025

PUBLIC

ORTEC

Published: March 21, 2025

Version: 2025.1637531

Table of Contents

1 Payroll | Compensation rules

Compensation rules are used to determine pay and benefits, such as leave entitlements, and allowances like unsocial hours allowance.

Each activated setting represents a compensation rule that can be defined via **Employment conditions** in **OWS**. Compensation rules are used to control how data is recorded in **ORTEC WS** accounts and allow for the retrieval of a large amount of data from **ORTEC WS**. This section provides a complete overview of the compensation rules and is meant for reference purposes.

The most used compensation rules are:

- "Transferbooking based upon an expression" on page 44
- "Allowances per time interval" on page 17
- "Fixed entry for an activity type" on page 55
- "Fixed entries for activity class in a period" on page 51
- "Variable entry for an activity class" on page 71
- "Variable entry for leave request per calender day" on page 132
- "Fixed entry for leave request per calender day" on page 61
- "Fixed entries for public holidays " on page 8
- "Carry over balance" on page 46

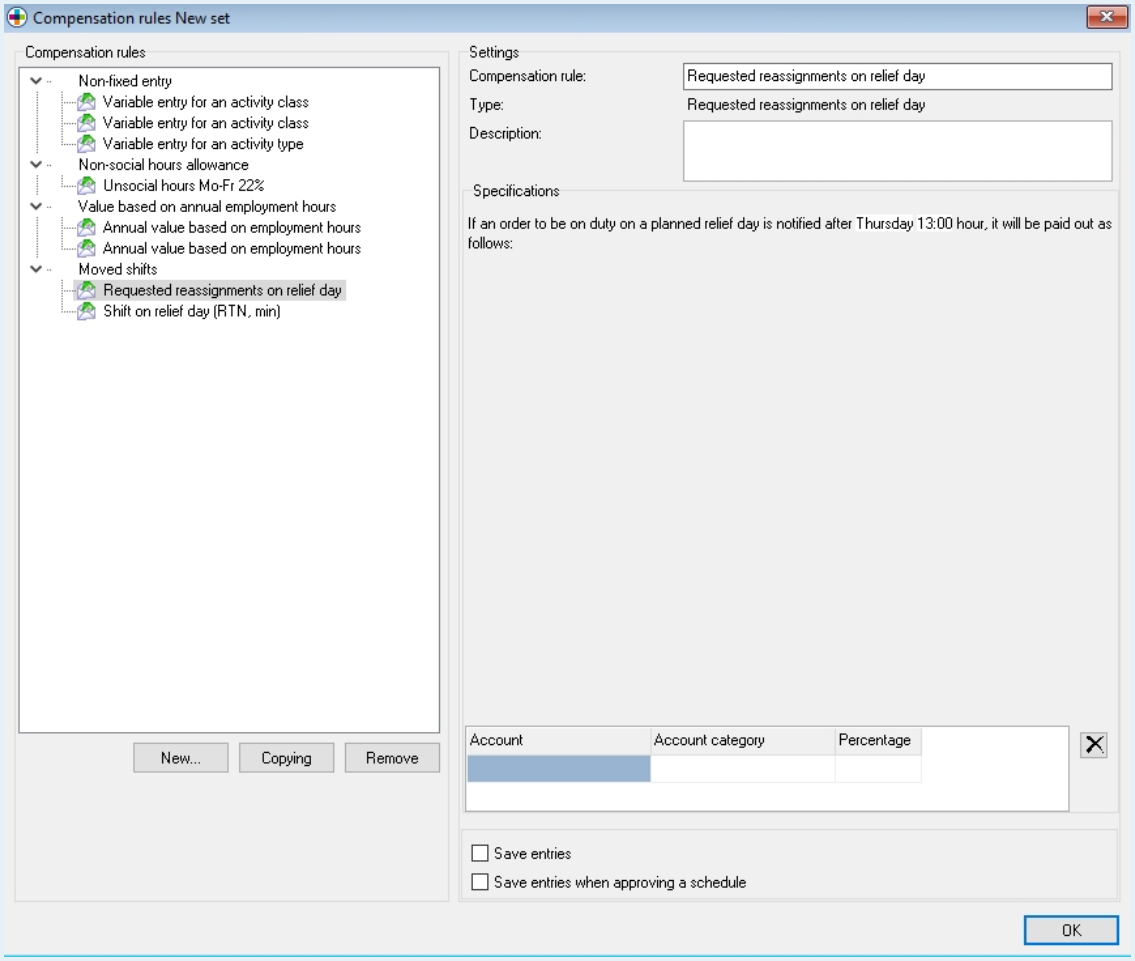
Depending on the configuration of your system, certain compensation rules may not be available. Please contact **ORTEC** if you would like to use compensation rules that aren't available in your system.

1.1 Compensation rule group

Compensation rules are organized in groups. Each group includes a number of related rules. For example, the rules in the group **Moved shifts** are generally used to gather information about changes in the schedule on a day the employee had no shift.

Example

An example of how compensation rules are grouped when defining them via **Employment conditions**.



Some rules depend on each other. For example, the rule **Threshold value for an entry per working day** has only affect on the rule **Entry for a working day**. Because of this relationship these two rules are put together in the group **Fixed entry for a day**.

A full list of the rule groups is presented below, indicating which compensation rules belong to each group.

Availability allowance

- Availability allowance

Fixed entries for public holidays

- Fixed entries for public holidays

Moved shifts

- Requested reassignments on relief day
- Shift on relief day (RTN)
- Shift on relief day (RTN, min)
- Shift on relief day (RTN, tft)

Non-social hours allowances

- Allowances
- Allowances per salary code
- Allowances per salary code for activity classes
- Allowances per time interval
- Non-social hours allowances in a period before or after a public holiday
- Non-social hours allowances in period on date
- Non-social hours allowances per month per day
- Public holidays
  - Public holiday definition
- Public holidays on days of week
- Public holidays on weekdays

Required reassignment allowance

- Required reassignment allowance

Shift types

- Instructor shift allowance

Sleeping hours

- Sleeping hours

In total there are seven groups that contains rules that are in general used to determine common allowances (apart from overtime): **Availability allowances**, **Fixed entries for public holidays**, **Moved shifts**, **Non-social hours allowances**, **Required reassignment allowance**, **Shift types** and **Sleeping hours**. This diversification is made, because the way in which the information of these different groups is ascertained and calculated differs and so also the practical use.

Carry over

- Account category to carry over (maximum)
- Account category to which to be carried over
- Carry over all account categories with expression
- Carry over per cost center
- Enter time for time
- Transfer balance per salary period
- Transfer maximum
- Transferbooking based upon an expression

Carry over balance

- Carry over balance (1)
- Carry over balance (2)

Clocking entry with function key

- Clocking entry with function key

Duty registration

- Duty registration

Fixed entry

- Fixed entries for activity class in a period
- Fixed entries for an activity class per salary code
- Fixed entries for an activity treatment on a working day
- Fixed entry for an activity sort
- Fixed entry for an activity type
- Time between shifts

Fixed entry for a day

- Entry for a working day
- Threshold value for an entry per working day

Fixed entry for leave request per calender day

- Fixed entry for leave request per calender day

Min/max employments

- Min/max: quarterly hours exceeding minimum

Non-fixed entry

- Clocking norm, min, max
- Difference between worked hours and employment-hours
- Entry based on employee property
- Kind of activity with respect to cyclical schedule
- Minimum availability
- Minimum availability per period
- Multiple activity kinds per period
- Spent time according to cyclical schedule
- Variable entries for an activity class (sal.)
- Variable entry for an activity class
- Variable entry for an activity type

Overtime

- Carry over kilometers
- Compensation
- Continuous overtime
- Continuous overtime (comp time)
- Continuous overtime (sal. code)
- Do not work as in cyclical schedule
- Enter travel time
- From another account
- Monthly transfer-booking
- Non-contiguous overtime
- Non-contiguous overtime (sal. group)
- Non-contiguous overtime (tft)
- Overtime
- Overtime (tft)
- Overtime adjusted with average shift percentage
- Overtime per salary code
- Overtime per salary code (tft)
- Overtime per time period
- Overtime per time period with threshold

- Overtime: based on account balance
- Overtime: calculated from cyclical schedule
- Overtime: compare employment hours
- Overtime: compare employment hours per x weeks
- Overtime: days according to cyclical schedule
- Overtime: fixed in regard to publication
- Overtime: hours above day-norm per month
- Overtime: hours above norm per week
- Overtime: number of days from published master schedule
- Overtime: threshold value

Overtime: rounded off

- Overtime: rounded off

Pass on call-out

- Pass on call-out

Round off the account balances

- Round off the account balances

Shifted time windows

- Shifted time windows

Skills

- Work with skill

Time window

- Time window hours

Total working hours

- Entry based on activity type or workstation

Travel distance

- Travel distance in shift
- Travel distance at the start of the shift
- Travel distance at the end of the shift

Travel expenses

- Travel expenses

Value based on annual employment hours

- Annual value based on age
- Annual value based on age and time in the company
- Annual value based on employment hours
- Correction in case of long term sick leave

Value based on annual hours of employment

- Value based on annual hours of employment

Variable entry for leave request per calender day

- Variable entry for leave request per calender day

Waiting day in case of sick-leave

- Waiting day in case of sick-leave

Working pattern

- Repetition

1.2 Expressions

Use various types of expressions within the following two compensation rules:

- Transferbooking based upon an expression
- Carry over all account categories with expression

The expressions that can be used in those compensation rules are explained in "Expressions" on page 136.

2 Availability allowance

The rule in the group **Availability allowance** can be used to calculate the amount of hours an employee has right on availability allowance. In this the rule make use of the time an employee is continuous available for work.

2.1 Availability allowance

Stipulates that a specified number of hours are to be credited if an employee has to be continuously available for a specified period.

How to use

Employees who need to be available for work for a period of [**Duration**] hours, will receive [**Hours**] hours of paid leave.

[**Target account** | **Target category**]

- **Duration** is the number of hours of continuous availability necessary to trigger a booking.

  An employee is treated as available if his/her activities are of the following kinds: Attendance, Consignation, Operational, On call, Standby or Work.

- **Hours** is the fixed number of hours to be credited to the target category.

Example

On Saturday 6 May 2017 and Sunday 7 May 2017, an employee is assigned to an availability shift B, with the following makeup:

Activity type	Start	End
Available	0:00	0:00

The activity kind of the activity type 'Available' is 'Consignation'.

Employees who need to be available for work for a period of [**40:00**] hours, will receive [**04:00**] hours of paid leave.

[**Hours** | **Availability**]

Hours	01-01-2017 to 01-01-2018
Availability	
> 06-05-2017	4:00
Total	4:00

3 Fixed entries for public holidays

The compensation rule in this group is in general used to determine corresponding allowances in case an employee works on a public holiday.

3.1 Fixed entries for public holidays

Stipulates that a value proportional to the employee's contractual hours is to be booked for each public holiday that falls on a weekday.

How to use

For public holidays on weekdays, [**Hours**] hours are added to or deducted from the following accounts. Under a full-time contract the employment hours per week are [**Hours per week**].

[**Target account** | **Target category**] | [**Percentage**]

- **Hours** is the number of hours added to or deducted from, in case a public holiday falls on a weekday.
- **Hours per week** is number of hours an employee with a full-time contract is required to work.

A weekday is a Monday, Tuesday, Wednesday, Thursday or Friday.

Example

Friday 5 May 2017 is a public holiday. The employee concerned has a full-time contract.

For public holidays on weekdays, [**24**] hours are added to or deducted from the following accounts. Under a full-time contract the employment hours per week are [**36**].

[**Public holidays** | **Hours**] | [**100**]

Public holiday	01-01-2017 to 01-01-2018
Hours	
> 05-05-2017	24:00
Total	24:00

4 Moved shifts

The compensation rules in this group **Moved shifts** can be used to identify the allowances in hours in case an employee is assigned to a shift on its relief day.

4.1 Requested reassignments on relief day

Stipulates that the number of hours that an employee works is to be booked, if he or she works on a day when he/she was not originally supposed to work, and if the change to his/her day off is made after a specified point in the week.

How to use

If an order to be on duty on a planned relief day is notified after [**Day of the week**] [**Time**] hour, it will be paid out as follows:

[**Target account** | **Target category** | **Percentage**]

- **Day of the week** is the day of the week [Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday] on which the cut-off time falls for making changes without a booking being necessary.
- **Time** is the cut-off time for making changes without a booking being necessary.

The timing of a change is determined from the Swap and Amendment History.

Example

In the cyclical schedule, a particular employee is assigned a 'Day off' shift on Friday 5 May 2017. After publication of the schedule, however, the employee is assigned shift A on that day. The change is made on Thursday 27 April, at 14:35 hours. Shift A is made up of the following activities:

Activity type	Start	End
Work	08:00	12:00
Break	12:00	12:30
Work	12:30	16:00

If an order to be on duty on a planned relief day is notified after [**Thursday**] [**12:00**] hour, it will be paid out as follows:

[**Hours** | **Extra hours** | **100**]

Hours	01-01-2017 to 01-01-2018
Extra hours	
> 05-05-2017 A	08:00
Total	08:00

4.2 Shift on relief day(RTN)

Stipulates that a fixed number of hours are to be booked if an employee works at a time when, according to the cyclical schedule, he or she should not be working.

How to use

Whenever an employee works on a day that is a relief day according to the cyclical schedule, he/she will be paid out for [**Hours**] hours in the categories indicated below.

[**Target account** | **Target category**]

- **Hours** is the fixed number of hours to be booked if the employee works.

When ascertaining whether a booking is necessary, ORTEC WS may check the rotation plan or the published schedule for shifts, depending on the system configuration settings.
The target account has to be one to which bookings of the 'Time' type can be made.

Example

In the cyclical schedule, a particular employee is assigned a 'Day off' shift on Friday 5 May 2017. After publication of the schedule, however, the employee is assigned shift A on that day. Shift A is made up of the following activities:

Activity type	Start	End
Work	08:00	12:00
Break	12:00	12:30
Work	12:30	16:00

Whenever an employee works on a day that is a relief day according to the cyclical schedule, he/she will be paid out for [**08:00**] hours in the categories indicated below.

[**Hours** | **Extra hours**]

Hours		01-01-2017 to 01-01-2018
Extra hours		
> 05-05-2017	A	08:00
Total		08:00

4.3 Shift on relief day (RTN, min)

Stipulates that the number of hours worked are to be booked (subject to a minimum) if an employee works at a time when, according to the cyclical schedule, he or she should not be working.

How to use

Whenever an employee works on a day that is a relief day according to the cyclical schedule, he/she will be paid out for at least [**Hours**] hours in the categories indicated below.

[**Target account** | **Target category** | **Percentage**]

- **Hours** is the minimum number of hours to be booked if the employee works when he or she should have had a day off.

When ascertaining whether a booking is necessary, ORTEC WS may check the rotation plan or the published schedule for shifts, depending on the system configuration settings.

Example

In the cyclical schedule, a particular employee is assigned a 'Day off' shift on Friday 5 May 2017. After publication of the schedule, however, the employee is assigned shift A on that day. Shift A is made up of the following activities:

Activity type	Start	End
Work	08:00	12:00
Break	12:00	12:30
Work	12:30	16:00

Whenever an employee works on a day that is a relief day according to the cyclical schedule, he/she will be paid out for at least [**12:00**] hours in the categories indicated below.

[**Hours** | **Extra hours** | **100**]

Hours	01-01-2017 to 01-01-2018
Extra hours	
> 05-05-2017 A	12:00
Total	12:00

Example

Whenever an employee works on a day that is a relief day according to the cyclical schedule, he/she will be paid out for at least [**04:30**] hours in the categories indicated below.

[**Hours** | **Extra hours** | **100**]

Hours	01-01-2017 to 01-01-2018
Extra hours	
> 05-05-2017 A	07:30
Total	07:30

4.4 Shift on relief day (RTN, tft)

Stipulates that the number of hours worked are to be booked (subject to a minimum) if an employee works at a time when, according to the cyclical schedule, he or she should not be working, and if the employee's chosen time-for-time option is as specified.

How to use

Whenever an employee works on a day that is off according to the cyclical schedule and if he/she
☐ has
☐ has not
chosen for a 'Time For Time' arrangement, the employee will be paid out for at least [**Hours**] hour(s) in the categories indicated below.

[**Target account** | **Target category** | **Percentage**]

- **Hours** is the minimum number of hours to be booked if the employee works when he or she should have had a day off.

When ascertaining whether a booking is necessary, ORTEC WS may check the rotation plan or the published schedule for shifts, depending on the system configuration settings.
An employee's time-for-time choice is shown on the 'Employment conditions' tab in the 'Employee management' window.

Example

In the cyclical schedule, a particular employee is assigned a 'Day off' shift on Friday 5 May 2017. After publication of the schedule, however, the employee is assigned shift A on that day. Shift A is made up of the following activities:

Activity type	Start	End
Work	08:00	12:00
Break	12:00	12:30
Work	12:30	16:00

The employee has chosen pay for time.

Whenever an employee works on a day that is off according to the cyclical schedule and if he/she
☐ has
☒ has not
chosen for a 'Time For Time' arrangement, the employee will be paid out for at least [**12**] hour(s) in the categories indicated below.

[**Hours** | **Extra hours** | **100**]

Hours	01-01-2017 to 01-01-2018
Extra hours	
> 05-05-2017 A	12:00
Total	12:00

5 Non-social hours allowance

The compensation rules in the group **Non-social hours allowance** can be used to calculate the amount of hours an employee has right on non-social hours allowances. In this the rules make use of the hours employees spent on work or certain activities for hours of the day that are less favorite. For example an employee gain non-social hours allowance for the number of hours he works between 00:00 and 06:00.

The compensation rules in this group offer different formulations to calculate the amount of hours an employee has right on non-social hours allowance. The different formulations of the non-social hours allowances are explained separately in the sections below.

5.1 Allowances

Stipulates that a value is to be booked, based on the number of hours that an employee is scheduled to work within a specified time interval on one or more days of the week.

How to use

Employees who work between [**Start time**] and [**End time**] on a [**Days of the week**] will be paid out on:
Calculations will be based upon [**Schedule**].

[**Target account** | **Target category** | **Percentage**]

- **Start time** is the start of the time interval within which work needs to be scheduled in order to trigger a booking.
- **End time** is the end of the time interval within which work needs to be scheduled in order to trigger a booking.
- **Days of the week** are the days of the week [Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday, public holiday] on which work needs to be scheduled in order to trigger a booking.
- **Schedule** is the schedule [Published master schedule, Realized master schedule, Cyclical schedule] on which the calculations are based upon.

Depending on the system configuration settings, the booking date used in the context of this rule is the date on which the shift starts or the date of the hours to which the booking relates.

Example

Shift A, as scheduled for Saturday 8 April 2017, is made up of the following activities:

Activity type	Start	End
Work	23:00	03:00
Break	03:00	03:30
Work	03:30	07:00

Employees who work between [**05:00**] and [**08:00**] on a [**Saturday, Sunday**] will be paid out on: Calculations will be based upon [**Realized master schedule**].

[**Hours** | **Worked hours** | **100**]

Hours	01-01-2017 to 01-01-2018
Worked hours	
> 09-04-2017 A	02:00
Total	02:00

5.2 Allowances per salary code

Stipulates that a value is to be booked, based on the number of hours that an employee is scheduled to work within a specified time interval on one or more specified days of the week, provided that the employee's salary code is as specified.

How to use

Employees whose salary classification is between [**Min salary code**] and [**Max salary code**] and who work hours between [**Start time**] and [**End time**] on a [**Days of the week**] will be paid out on: Calculations will be based upon [**Schedule**].

[**Target account** | **Target category** | **Percentage**]

- **Min salary code** is the lowest salary code that an employee may have for a booking to be triggered.
- **Max salary code** is the highest salary code that an employee may have for a booking to be triggered.
- **Start time** is the start of the time interval within which work needs to be scheduled in order to trigger a booking.
- **End time** is the end of the time interval within which work needs to be scheduled in order to trigger a booking.
- **Days of the week** are the days of the week [Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday, public holiday] on which work needs to be scheduled in order to trigger a booking.
- **Schedule** is the schedule [Published master schedule, Realized master schedule, Cyclical schedule] on which the calculations are based upon.

Depending on the system configuration settings, the booking date used in the context of this rule is the date on which the shift starts or the date of the hours to which the booking relates.

Example

Shift A is scheduled for Saturday 8 April 2017. The shift is to be worked by an employee on salary code C and is made up of the following activities:

Activity type	Start	End
Work	23:00	03:00
Break	03:00	03:30
Work	03:30	07:00

Employees whose salary classification is between [**A**] and [**E**] and who work hours between [**05:00**] and [**08:00**] on a [**Saturday, Sunday**] will be paid out on:
Calculations will be based upon [**Realized master schedule**].

[**Hours** | **Worked hours** | **100**]

Hours	01-01-2017 to 01-01-2018
Worked hours	
> 09-04-2017 A	02:00
Total	02:00

5.3 Allowances per salary code for activity classes

Stipulates that a value is to be booked, based on the number of hours that an employee is scheduled to spend on an activity of a specified kind within a specified time interval on a particular day of the month gain from a specified schedule, provided that the employee's salary code is as specified.

How to use

Book the [**Status**] time between [**Start time**] and [**End time**] hours on a [**Days of the week**] where the activities have [**Kinds**]
for employees in a salary code between [**Min salary code**] and [**Max salary code**] the following entry is made.

☐Sum entries for consecutive activities within the same shift when the following properties are equal: activity type, cost center, workstation, kind of request.

For this rule [**do/do not**] take the transition from daylight savings time to standard time into account.

☐This rule applies only to employees for whom the employee property [**Property**] ... [**does/does not**] have the value [**Value**].

Calculations will be based upon [**Schedule**].

[**Target account** | **Target category** | **Percentage**]

- **Status** is the status [planned, spent] of the hours to be booked. Planned time is calculated on basis of the shift as originally scheduled, spent time is based on the shift as realized in the schedule.

- **Start time** is the start of the time interval within which work needs to be scheduled in order to trigger a booking.
- **End time** is the end of the time interval within which work needs to be scheduled in order to trigger a booking.
- **Kinds** are the kinds [classes, activity types, main activity types] to which the shift's activity types need to belong in order to trigger a booking.
- **Days of the week** are the days of the week [Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday, public holiday] on which work needs to be scheduled in order to trigger a booking.
- **Min salary code** is the lowest salary code that an employee may have for a booking to be triggered.
- **Max salary code** is the highest salary code that an employee may have for a booking to be triggered.
- **Property** is the employee property that is checked in order to determine if an entry is made.
- **Value** is the value an employee should have for the specified employee property in order to make an entry.
- **Schedule** is the schedule [Published master schedule, Realized master schedule, Cyclical schedule] on which the calculations are based upon.

Depending on the system configuration settings, the booking date used in the context of this rule is the date on which the shift starts or the date of the hours to which the booking relates.

Example

Shift A is scheduled for Saturday 8 April 2017. The shift is to be worked by an employee on salary code C and is made up of the following activities:

Activity type	Start	End
Work	08:00	12:00
Break	12:00	12:30
Work	12:30	16:30
Contactable	16:30	08:00

The activity type 'Work' belongs to the activity class 'Work'. The activity type 'Break' belongs to the activity class 'Break'. The activity class 'Contactable' is of the class 'Availability (no labor)'.

In practice, the employee was called on to work during the shift from 01:30 to 02:30. In the realization phrase, the activity type 'Work' is therefore assigned to him for that period.

Book the [**planned**] time between [**00:00**] and [**00:00**] hours on a [**Saturday, Sunday**] where the activities have [**classes Work**]
for employees in a salary code between [**A**] and [**E**] the following entry is made.

☐Sum entries for consecutive activities within the same shift when the following properties are equal: activity type, cost center, workstation, kind of request.

For this rule [**do not**] take the transition from daylight savings time to standard time into account.

☐This rule applies only to employees for whom the employee property [**(None)**] ...

Calculations will be based upon [**Realized master schedule**].

[**Hours** | **Worked hours** | **100**]

Hours	01-01-2017 to 01-01-2018
Worked hours	
> 08-04-2017 A	08:00
Total	08:00

Example

Book the [**spent**] time between [**00:00**] and [**00:00**] hours on a [**Saturday, Sunday**] where the activities have [**classes Work**]
for employees in a salary code between [**A**] and [**E**] the following entry is made.

☐Sum entries for consecutive activities within the same shift when the following properties are equal: activity type, cost center, workstation, kind of request.

For this rule [**do not**] take the transition from daylight savings time to standard time into account.

☐This rule applies only to employees for whom the employee property [**(None)**] ...

Calculations will be based upon [**Realized master schedule**].

[**Hours** | **Worked hours** | **100**]

Hours	01-01-2017 to 01-01-2018
Worked hours	
> 08-04-2017 A	08:00
> 09-04-2017 A	01:00
Total	09:00

5.4 Allowances per time interval

Stipulates that a value is to be booked, based on the number of hours that an employee spends on an activity of a specified kind on one or more specified days of the week within a specified time interval, provided that the employee's salary code and the start and/or end time of the shift are as specified.

How to use

Book the [**Status**] time between [**Start time**] and [**End time**] hours on a [**Days of the week**] where the activities have [**Kinds**].
for employees in a salary code between [**Min salary code**] and [**Max salary code**] the following entry is made.

Exceptions:
☐ This rule is only valid if the worked shift started before [**Latest start time**] hours.
☐ This rule is only valid if the worked shift started after [**Earliest start time**] hours.
☐ This rule is only valid if the worked shift has ended after [**Latest end time**] hours.
☐ This rule is only valid if the worked shift has ended before [**Earliest end time**] hours.
☐ This rule is only valid if at least half of a shift is after [**Half-after time**] hour.
☐ This rule is only valid in the first [**Number**] calendar week(s) of a sick leave.
☐ This rule is not valid on a [**Day type**].
☐ Calculate the begin and end time of the shift according to the selected activity kinds.

This rule is valid [**Reassignment history**].

☐Sum entries for consecutive activities within the same shift when the following properties are equal: activity type, cost center, workstation, kind of request.

☐All entries for a shift will be entered on the begin date of the shift.

For this rule, [**do/do not**] take the transition from daylight savings time to standard time into account.

☐This rule applies only to employees for whom the employee property [**Property**] ... [**does/does not**] have the value [**Value**].

Calculations will be based upon [**Schedule**].

[**Target account** | **Target category** | **Percentage**]

- **Status** is the status [planned, spent] of the hours to be booked. Planned time is calculated on basis of the shift as originally scheduled, spent time is based on the shift as realized in the schedule.
- **Start time** is the start of the time interval within which work needs to be scheduled in order to trigger a booking.
- **End time** is the end of the time interval within which work needs to be scheduled in order to trigger a booking.
- **Days of the week** are the days of the week [Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday, public holiday] in respect of which hours are to be booked.
- **Kinds** are the kinds [classes, activity types, main activity types] to which the shift's activity types need to belong in order to trigger a booking.
- **Min salary code** is the lowest salary code that an employee may have for a booking to be triggered.
- **Max salary code** is the highest salary code that an employee may have for a booking to be triggered.
- **Latest start time** is the latest time that a shift can start for a booking to be triggered.
- **Earliest start time** is the earliest time that a shift can start for a booking to be triggered.
- **Latest end time** is the latest time that a shift can end for a booking to be triggered.
- **Earliest end time** is the earliest time that a shift can end for a booking to be triggered.
- **Half-after time** is the time after which at least half a shift must fall for a booking to be triggered.
- **Number** is the maximum number of weeks that an employee may be off sick for a booking to be made.
- **Day type** describes the type of day [a public holiday, day before a public holiday, day after a public holiday] that hours must be scheduled for in order to trigger a booking.
- **Reassignment history** is a phrase describing the applicability of the rule by reference to the shift reassignment history [always, only in case of an exchange, not in case of an exchange].
- **Property** is the employee property that is checked in order to determine if an entry is made.
- **Value** is the value an employee should have for the specified employee property in order to make an entry.
- **Schedule** is the schedule [Published master schedule, Realized master schedule, Cyclical schedule] on which the calculations are based upon.

Example

Shift A is scheduled for Saturday 8 April 2017. The shift is to be worked by an employee on salary code C and is made up of the following activities:

Activity type	Start	End
Work	23:00	03:00
Break	03:00	03:30
Work	03:30	07:00

The activity type 'Work' belongs to the activity class 'Work'. The activity type 'Break' belongs to the activity class 'Break'.

Book the [**planned**] time between [**05:00**] and [**08:00**] hours on a [**Saturday, Sunday**] where the activities have [**classes Work**].
for employees in a salary code between [**A**] and [**E**] the following entry is made.

Exceptions:
☒ This rule is only valid if the worked shift started before [**1:00**] hours.
☐ This rule is only valid if the worked shift started after [**0:00**] hours.
☐ This rule is only valid if the worked shift has ended after [**0:00**] hours.
☐ This rule is only valid if the worked shift has ended before [**0:00**] hours.
☐ This rule is only valid if at least half of a shift is after [**0:00**] hour.
☐ This rule is only valid in the first [**0**] calendar week(s) of a sick leave.
☐ This rule is not valid on a [**None**].
☐ Calculate the begin and end time of the shift according to the selected activity kinds.

This rule is valid [**always**].

☐Sum entries for consecutive activities within the same shift when the following properties are equal: activity type, cost center, workstation, kind of request.

☐All entries for a shift will be entered on the begin date of the shift.

For this rule, [**do not**] take the transition from daylight savings time to standard time into account.

☐This rule applies only to employees for whom the employee property [**(None)**] ...
Calculations will be based upon [**Realized master schedule**].

[**Hours** | **Worked hours** | **100**]

Hours	01-01-2017 to 01-01-2018
Worked hours	
> 09-04-2017 A	02:00
Total	02:00

5.5 Non-social hours allowances in a period before or after a public holiday

Stipulates that a value is to be booked, based on the number of hours that an employee is scheduled to spend on an activity of a specified kind within a specified time interval on a day that precedes or follows a public holiday, provided that the employee's salary code and the start and/or end time of the shift are as specified.

How to use

For [**Status**] time between [**Start**] and [**End**] hours on [**the day before a public holiday/the day after a public holiday**], if this day is on a [**Days of the week**], where the activities have [**Kinds**]

for employees in a salary code between [**Min salary code**] and [**Max salary code**] the following entry is made.

Exceptions:

☐This rule is only valid if the worked shift has started before [**Latest start time**] hours.
☐This rule is only valid if the worked shift has started after [**Earliest start time**] hours.
☐This rule is only valid if the worked shift has ended before [**Latest end time**] hours.
☐This rule is only valid if the worked shift has ended after [**Earliest end time**] hours.
☐This rule is only valid if at least half of a shift is after [**Half-after time**] hour.
☐This rule is only valid in the first [**Number**] calendar week(s) of a sick leave.
☐Calculate the begin and end time of the shift according to the selected activity kinds.

This rule is valid [**Reassignment history**].

☐Sum entries for consecutive activities within the same shift when the following properties are equal: activity type, cost center, workstation, kind of request.

☐All entries for a shift will be entered on the begin date of the shift.

For this rule, [**do/do not**] take the transition from daylight savings time to standard time into account.

☐This rule applies only to employees for whom the employee property [**Property**] ... [**does/does not**] have the value [**Value**].

Calculations will be based upon [**Schedule**]

[**Target account** | **Target category** | **Percentage**]

- **Status** is the status [spent, planned] of the hours to be booked. Spent time is calculated on the basis of the realized shift, planned time on the basis of the shift as originally scheduled.
- **Start** is the start of the time interval within which work needs to be scheduled in order to trigger a booking.
- **End** is the end of the time interval within which work needs to be scheduled in order to trigger a booking.
- **Days of the week** are the days [Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday] of the week.
- **Kinds** are the kinds [classes, activity types, main activity types] to which the shift's activity types need to belong in order to trigger a booking.
- **Min salary code** is the lowest salary code that an employee may have for a booking to be triggered.
- **Max salary code** is the highest salary code that an employee may have for a booking to be triggered.
- **Latest start time** is the latest time that a shift can start for a booking to be triggered.
- **Earliest start time** is the earliest time that a shift can start for a booking to be triggered.
- **Latest end time** is the latest time that a shift can end for a booking to be triggered.
- **Earliest end time** is the earliest time that a shift can end for a booking to be triggered.
- **Half-after time** is the time after which at least half a shift must fall for a booking to be triggered.

- **Number** is the maximum number of weeks that an employee may be off sick for a booking to be made.
- **Reassignment history** is a phrase describing the applicability of the rule by reference to the shift reassignment history [always, only in case of an exchange, not in case of an exchange].
- **Property** is the employee property that is checked in order to determine if an entry is made.
- **Value** is the value an employee should have for the specified employee property in order to make an entry.
- **Schedule** is the schedule [Published master schedule, Realized master schedule, Cyclical schedule] from which the information is obtained.

In case the rule **Public holiday definition** is available within ORTEC WS the compensation rules covering public holidays do only have affect on the public holiday chosen in the rule **Public holiday definition**.

Example

Shift A is scheduled for Tuesday 26 December 2017. The shift is to be worked by an employee on salary code C and is made up of the following activities:

Activity type	Start	End
Work	23:00	03:00
Break	03:00	03:30
Work	03:30	07:00

The activity type 'Work' belongs to the activity class 'Work'. The activity type 'Break' belongs to the activity class 'Break'. Tuesday 26 December 2017 is defined as public holiday.

For [**Planned**] time between [**5:00**] and [**8:00**] hours on [**the day after a public holiday**], if this day is on a [**Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Monday**], where the activities have [**classes Work**]

for employees in a salary code between [**A**] and [**F**] the following entry is made.

Exceptions:

☒ This rule is only valid if the worked shift has started before [**1:00**] hours.
☐This rule is only valid if the worked shift has started after [**0:00**] hours.
☐This rule is only valid if the worked shift has ended before [**0:00**] hours.
☐This rule is only valid if the worked shift has ended after [**0:00**] hours.
☐This rule is only valid if at least half of a shift is after [**0:00**] hour.
☐This rule is only valid in the first [**0**] calendar week(s) of a sick leave.
☐Calculate the begin and end time of the shift according to the selected activity kinds.

This rule is valid [**always**].
For this rule, [**do**] take the transition from daylight savings time to standard time into account.

☐This rule applies only to employees for whom the employee property [**(None)**] ...

Calculations will be based upon [**Realized master schedule**]

[**Hours** | **Hours worked** | **100**]

Hours	01-01-2017 to 01-01-2018
Worked hours	
27-12-2017 A	02:00
Total	02:00

5.6 Non-social hours allowances in period on date

Stipulates that a value is to be booked, based on the number of hours that an employee is scheduled to spend on an activity of a specified kind within a specified time interval on a particular day of the month, provided that the employee's salary code and the start and/or end time of the shift are as specified.

How to use

For [**Status**] time between [**Start**] and [**End**] hours on [**date**] where activities have [**Kinds**]

,for employees in a salary code between [**Min salary code**] and [**Max salary code**] the following entry is made.

Exceptions:
☐This rule is only valid if the worked shift has started before [**Latest start time**] hours.
☐This rule is only valid if the worked shift has started after [**Earliest start time**] hours.
☐This rule is only valid if the worked shift has ended before [**Latest end time**] hours.
☐This rule is only valid if the worked shift has ended after [**Earliest end time**] hours.
☐This rule is only valid if at least half of a shift is after [**Half-after time**] hour.
☐This rule is only valid in the first [**Number**] calendar week(s) of a sick leave.
☐This rule is not valid on a [**Day type**].
☐Calculate the begin and end time of the shift according to the selected activity kinds.

This rule is valid [**Reassignment history**].

☐Sum entries for consecutive activities within the same shift when the following properties are equal: activity type, cost center, workstation, kind of request.

☐All entries for a shift will be entered on the begin date of the shift.

For this rule, [**do/do not**] take the transition from daylight savings time to standard time into account.

☐This rule applies only to employees for whom the employee property [**Property**] ... [**does/does not**] have the value [**Value**].

Calculations will be based upon [**Schedule**]

[**Target account** | **Target category** | **Percentage**]

- **Status** is the status [spent, planned] of the hours to be booked. Spent time is calculated on the basis of the realized shift, planned time on the basis of the shift as originally scheduled.
- **Start** is the start of the time interval within which work needs to be scheduled in order to trigger a booking.
- **End** is the end of the time interval within which work needs to be scheduled in order to trigger a booking.
- **Date** is the date in respect of which hours are to be booked.
- **Kinds** are the kinds [classes, activity types, main activity types] to which the shift's activity types need to belong in order to trigger a booking.

- **Min salary code** is the lowest salary code that an employee may have for a booking to be triggered.
- **Max salary code** is the highest salary code that an employee may have for a booking to be triggered.
- **Latest start time** is the latest time that a shift can start for a booking to be triggered.
- **Earliest start time** is the earliest time that a shift can start for a booking to be triggered.
- **Latest end time** is the latest time that a shift can end for a booking to be triggered.
- **Earliest end time** is the earliest time that a shift can end for a booking to be triggered.
- **Half-after time** is the time after which at least half a shift must fall for a booking to be triggered.
- **Number** is the maximum number of weeks that an employee may be off sick for a booking to be made.
- **Day type** describes the type of day [a public holiday, the day before a public holiday, the day after a public holiday] that hours must be scheduled for in order to trigger a booking.
- **Reassignment history** is a phrase describing the applicability of the rule by reference to the shift reassignment history [always, only in case of an exchange, not in case of an exchange].
- **Property** is the employee property that is checked in order to determine if an entry is made.
- **Value** is the value an employee should have for the specified employee property in order to make an entry.
- **Schedule** is the schedule [Published master schedule, Realized master schedule, Cyclical schedule] from which the information is obtained.

In case the rule **Public holiday definition** is available within ORTEC WS the compensation rules covering public holidays do only have affect on the public holiday chosen in the rule **Public holiday definition**.

Date consists of a day number and a month number. Bookings are therefore made annually.

Example

Shift A is scheduled for Saturday 8 April 2017. The shift is to be worked by an employee on salary code C and is made up of the following activities:

Activity type	Start	End
Work	23:00	03:00
Break	03:00	03:30
Work	03:30	07:00

The activity type 'Work' belongs to the activity class 'Work'. The activity type 'Break' belongs to the activity class 'Break'.

For [**planned**] time between [**05:00**] and [**08:00**] hours on [**9 April**] where activities have [**classes Work**]

,for employees in a salary code between [**A**] and [**F**] the following entry is made.

Exceptions:
☒ This rule is only valid if the worked shift has started before [**01:00**] hours.
☐This rule is only valid if the worked shift has started after [**00:00**] hours.
☐This rule is only valid if the worked shift has ended before [**00:00**] hours.
☐This rule is only valid if the worked shift has ended after [**00:00**] hours.
☐This rule is only valid if at least half of a shift is after [**00:00**] hour.
☐This rule is only valid in the first [**0**] calendar week(s) of a sick leave.
☐This rule is not valid on a [**None**].
☐Calculate the begin and end time of the shift according to the selected activity kinds.

This rule is valid [**always**].

☐Sum entries for consecutive activities within the same shift when the following properties are equal: activity type, cost center, workstation, kind of request.

☐All entries for a shift will be entered on the begin date of the shift.

For this rule, [**do**] take the transition from daylight savings time to standard time into account.

☐This rule applies only to employees for whom the employee property [**(None)**] ...

Calculations will be based upon [**Realized master schedule**]

[**Hours** | **Worked hours** | **100**]

Hours	01-01-2017 to 01-01-2018
Worked hours	
> 09-04-2017 A	02:00
Total	02:00

5.7 Non-social hours allowances per month per day

Stipulates that a value is to be booked, based on the number of hours that an employee is scheduled to spend on an activity of a specified kind within a specified time interval on a particular day of the month, provided that the employee's salary code and the start and/or end time of the shift are as specified.

How to use

For [**Status**] time between [**Start time**] and [**End time**] hours at [**Ordinal**] [**Days of the week**] of [**Month**] where the activities have [**Kinds**]

for employees in a salary code between [**Min salary code**] and [**Max salary code**] the following entry is made.

Exceptions:
☐This rule is only valid if the worked shift starts before [**Latest start time**] hours.
☐This rule is only valid if the worked shift starts after [**Earliest start time**] hours.
☐This rule is only valid if the worked shift has ended before [**Latest end time**] hours.
☐This rule is only valid if the worked shift has ended before [**Earliest end time**].
☐This rule is only valid if at least half of a shift is after[**Half-after time**] hour.
☐This rule is only valid in the first [**Number**] calendar week(s) of a sick leave.
☐Calculate the begin and end time of the shift according to the selected activity kinds.

This rule is valid [**Reassignment history**].

☐Sum entries for consecutive activities within the same shift when the following properties are equal: activity type, cost center, workstation, kind of request.

☐All entries for a shift will be entered on the begin date of the shift.

For this rule [**do/do not**] take the transition from daylight savings to standard time into account.

☐This rule applies only to employees for whom the employee property [**Property**] ... [**does/does not**] have the value [**Value**].

Calculations will be based upon [**Schedule**].

[**Target account** | **Target category** | **Percentage**]

- **Status** is the status [spent, planned] of the hours to be booked. Spent time is calculated on the basis of the realized shift, planned time on the basis of the shift as originally scheduled.
- **Start time** is the start of the time interval within which work needs to be scheduled in order to trigger a booking.
- **End time** is the end of the time interval within which work needs to be scheduled in order to trigger a booking.
- **Ordinal** is a descriptor [first, second, third, fourth, last] indicating the occurrence of the specified day(s) of the week in respect of which hours are to be booked.
- **Days of the week** are the days of the week [Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday] in respect of which hours are to be booked.
- **Month** is the month [January, February,… ,December] in respect of which hours are to be booked.
- **Class** is the class [class, activity types, main activity type] from which you wish to select the kinds, to which the shift's activity types need to belong in order to trigger a booking, from.
- **Kinds** are the kinds [Amplitude, Availability (no labor),…, Work (stand by), WTR] to which the shift's activity types need to belong in order to trigger a booking.
- **Min salary code** is the lowest salary code that an employee may have for a booking to be triggered.
- **Max salary code** is the highest salary code that an employee may have for a booking to be triggered.
- **Latest start time** is the latest time that a shift can start for a booking to be triggered.
- **Earliest start time** is the earliest time that a shift can start for a booking to be triggered.

- **Latest end time** is the latest time that a shift can end for a booking to be triggered.
- **Earliest end time** is the earliest time that a shift can end for a booking to be triggered.
- **Half-after time** is the time after which at least half a shift must fall for a booking to be triggered.
- **Number** is the maximum number of weeks that an employee may be off sick for a booking to be made.
- **Day type** describes the type of day [public holiday, day before a public holiday, day after a public holiday] that hours must be scheduled for in order to trigger a booking.
- **Reassignment history** is a phrase describing the applicability of the rule by reference to the shift reassignment history [always, only in case of an exchange, not in case of an exchange].
- **Property** is the employee property that is checked in order to determine if an entry is made.
- **Value** is the value an employee should have for the specified employee property in order to make an entry.
- **Schedule** is the schedule [Published master schedule, Realized master schedule, Cyclical schedule] on which the bookings are based.

Bookings are made annually.

Example

Shift A is scheduled for Saturday 8 April 2017. The shift is to be worked by an employee on salary code C and is made up of the following activities:

Activity type	Start	End
Work	23:00	03:00
Break	03:00	03:30
Work	03:30	07:00

The activity type 'Work' belongs to the activity class 'Work'. The activity type 'Break' belongs to the activity class 'Break'.

For [**planned**] time between [**05:00**] and [**08:00**] hours at [**second**] [**Sunday**] of [**April**] where the activities have [**classes Work**]

for employees in a salary code between [**A**] and [**F**] the following entry is made.

Exceptions:
☒ This rule is only valid if the worked shift has started before [**01:00**] hours.
☐This rule is only valid if the worked shift has started after [**00:00**] hours.
☐This rule is only valid if the worked shift has ended before [**00:00**] hours.
☐This rule is only valid if the worked shift has ended after [**00:00**] hours.
☐This rule is only valid if at least half of a shift is after [**00:00**] hour.
☐This rule is only valid in the first [**0**] calendar week(s) of a sick leave.
☐Calculate the begin and end time of the shift according to the selected activity kinds.

This rule is valid [**always**].

☐Sum entries for consecutive activities within the same shift when the following properties are equal: activity type, cost center, workstation, kind of request.

☐All entries for a shift will be entered on the begin date of the shift.

For this rule, [**do**] take the transition from daylight savings time to standard time into account.

☐This rule applies only to employees for whom the employee property [**(None)**] ...

Calculations will be based upon [**Realized master schedule**]

[**Hours** | **Worked hours** | **100**]

Hours	01-01-2017 to 01-01-2018
Worked hours	
> 09-04-2017 A	02:00
Total	02:00

5.8 Public holidays

Stipulates that a value is to be booked, based on the number of hours that an employee is scheduled to spend on activities of certain specified kinds in the context of a shift worked on or adjacent to a public holiday, provided that the employee's salary code is as specified.

In case the rule **Public holiday definition** is available within ORTEC WS the compensation rules covering public holidays do only have affect on the public holiday chosen in the rule **Public holiday definition**.

How to use

Whenever an employee has a shift between [**Time before**] hours before a public holiday and [**Time after**] hours after a public holiday and contains [**Kinds**]

and the employee is classified in salary code [**Min salary code**] and [**Max salary code**], credits will be paid out as follows:
This rule sums entries for activities within the same shift when the following properties are equal: activity type, cost center, workstation.
For this rule [**do/do not**] take the transition from daylight savings to standard time into account.

[**Target account** | **Target category** | **Percentage**]

- **Time before** and **time after** are the starting and end times of the interval around the public holiday, within which the shift needs to be scheduled in order to trigger a booking.
- **Kinds** are the classes [Amplitude, Availability (no labor), ..., Work (stand by), WTR] to which the shift's activity types need to belong in order to trigger a booking.
- **Min salary code** is the lowest salary code that an employee may have for a booking to be triggered.
- **Max salary code** is the highest salary code that an employee may have for a booking to be triggered.

Depending on the system configuration settings, the booking date used in the context of this rule is the date on which the shift starts or the date of the hours to which the booking relates.

Activation of this rule results in booking of the total duration of activities of the specified kinds that take place within the time interval, even if part of the shift is outside the interval.

Example

Shift A is scheduled for 27 December 2017 (the day after Boxing Day). The shift is to be worked by an employee on salary code C and is made up of the following activities:

Activity type	Start	End
Work	08:00	12:00
Break	12:00	12:30
Work	12:30	16:00

The activity type 'Work' belongs to the activity class 'Work'. The activity type 'Break' belongs to the activity class 'Break'.

Whenever an employee has a shift between [**00:00**] hours before a public holiday and [**10:00**] hours after a public holiday and contains [**Work**]

and the employee is classified in salary code [**A**] and [**F**], credits will be paid out as follows:
This rule sums entries for activities within the same shift when the following properties are equal: activity type, cost center, workstation.
For this rule [**do**] take the transition from daylight savings to standard time into account.

[**Hours** | **Public holiday hours** | **100**]

Hours		01-01-2017 to 01-01-2018
Public holiday hours		
> 27-12-2017	A	02:00
Total		02:00

5.9 Public holidays on days of week

Stipulates that a value is to be booked, based on the number of hours that an employee is scheduled to spend on activities of certain specified kinds in the context of a shift worked on or adjacent to a public holiday that falls on a particular day or days of the week, provided that the employee's salary code is as specified.

How to use

Whenever an employee has a shift between [**Time before**] hour before and [**Time after**] after a public holiday, that public holiday is a [**Days of the week**] and the shift contains [**Kinds**].

, and the employee is classified in salary code [**Min salary code**] to [**Max salary code**], credits will be made as follows:
For this rule [**do not/do**] take the transition from daylight savings time to standard time into account.

[**Target account** | **Target category** | **Percentage**]

- **Time before** and **Time after** are the starting and end times of the interval around the public holiday, within which the shift needs to be scheduled in order to trigger a booking.
- **Days of the week** are the days of the week [Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday] on which the public holiday needs to fall in order to trigger a booking.
- **Kinds** are the kinds [Amplitude, Availability (no labor), Break, ..., Overtime, Overtime (for free time)] to which the shift's activity types need to belong in order to trigger a booking.
- **Min salary code** is the lowest salary code that an employee may have for a booking to be triggered.
- **Max salary code** is the highest salary code that an employee may have for a booking to be triggered.

Depending on the system configuration settings, the booking date used in the context of this rule is the date on which the shift starts or the date of the hours to which the booking relates.
Activation of this rule results in booking of the total duration of activities of the specified kinds that take place within the time interval, even if part of the shift is outside the interval.

In case the rule **Public holiday definition** is available within ORTEC WS the compensation rules covering public holidays do only have affect on the public holiday chosen in the rule **Public holiday definition**.

Example

Shift A is scheduled for Tuesday 27 December 2017 (the day after Boxing Day). The shift is to be worked by an employee on salary code C and is made up of the following activities:

Activity type	Start	End
Work	08:00	12:00
Break	12:00	12:30
Work	12:30	16:00

The activity type 'Work' belongs to the activity class 'Work'. The activity type 'Break' belongs to the activity class 'Break'.

Whenever an employee has a shift between [**0:00**] hour before and [**10:00**] after a public holiday, that public holiday is a [**Tuesday, Wednesday**] and the shift contains [**Work**].

, and the employee is classified in salary code [**A**] to [**F**], credits will be made as follows:
For this rule [**do not**] take the transition from daylight savings time to standard time into account.

[**Hours** | **Public holiday hours** | **100**]

Hours	01-01-2017 to 01-01-2018
Public holiday hours	
> 27-12-2017 A	02:00
Total	02:00

5.10 Public holidays on weekdays

Stipulates that a value is to be booked, based on the number of hours that an employee is scheduled to spend on activities of certain specified kinds in the context of a shift worked on or adjacent to a public holiday that falls on a weekday, provided that the employee's salary code is as specified.

How to use

Whenever an employee has a shift between [**Time before**] hour(s) before a public holiday on a weekday and [**Time after**] after a public holiday on a weekday and contains [**Kinds**].
and the employee is classified in salary code [**Min salary code**] to [**Max salary code**], credits will be paid out as follows:

[**Target account** | **Target category** | **Percentage**]

- **Time before** and **Time after** are the starting and end times of the interval around the public holiday, within which the shift needs to be scheduled in order to trigger a booking.
- **Kinds** are the kinds [Amplitude, Availability (no labor), Break, ..., Overtime, Overtime (for free time)] to which the shift's activity types need to belong in order to trigger a booking.
- **Min salary code** is the lowest salary code that an employee may have for a booking to be triggered.
- **Max salary code** is the highest salary code that an employee may have for a booking to be triggered.

Activation of this rule results in booking of the total duration of activities of the specified kinds that take place within the time interval, even if part of the shift is outside the interval.

In case the rule **Public holiday definition** is available within ORTEC WS the compensation rules covering public holidays do only have affect on the public holiday chosen in the rule **Public holiday definition**.

Example

Shift A is scheduled for Tuesday 27 December 2017 (the day after Boxing Day). The shift is to be worked by an employee on salary code C and is made up of the following activities:

Activity type	Start	End
Work	08:00	12:00
Break	12:00	12:30
Work	12:30	16:00

The activity type 'Work' belongs to the activity class 'Work'. The activity type 'Break' belongs to the activity kind 'Break'.

Whenever an employee has a shift between [**0:00**] hour(s) before a public holiday on a weekday and [**10:00**] after a public holiday on a weekday and contains [**Work**].
and the employee is classified in salary code [**A**] to [**F**], credits will be paid out as follows:

[**Hours** | **Public holiday hours** | **100**]

Hours	01-01-2017 to 01-01-2018
Public holiday hours	
> 27-12-2017 A	02:00
Total	02:00

6 Required reassignment allowance

The compensation rule in this group can be used to determine corresponding allowances in case a planner give compensation to employees when the working hours of the cyclical schedule do not match the working hours of the realized master schedule.

6.1 Required reassignment allowance

Stipulates that a value is to be booked, based on the number of hours that an employee works within a specified time interval on one or more weekdays as a result of a schedule revision.

How to use

Whenever a shift is different from the shift according to the cyclical schedule, the employee will be credited for the **Time** of the shift actually worked that does not overlap the original shift in the cyclical schedule between [**Start time**] and [**End time**] on a [**Days of the week**].
This rule applies: [**Reassignment history**]

☐ This rule does [**only/not**] apply to the first [**Hours limit**] hours.
☐ This rule does [**only/not**] apply if the shift started before [**Start time limit**].

[**Target account** | **Target category** | **Percentage**]

- **Time** is a descriptor [working time, duration] specifying the time to be booked.
- **Start time** is the start of the qualifying time interval.
- **End time** is the end of the qualifying time interval.
- **Days of the week** are the days of the week [Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday, public holiday] in respect of which hours are to be booked.
- **Reassignment history** is a phrase describing the applicability of the rule by reference to the shift reassignment history [always, only in case of an exchange, not in case of an exchange].
- **Hours limit** is the number of hours a shift is different from the shift according to the cyclical schedule; depending on the rule its setting the hours triggered are the hours before or after this limit is passed.
- **Start time limit** depending on the rule its settings the start time limit is the earliest or latest time that a shift can start for a booking to be triggered.

Depending on the system configuration settings, the booking date used in the context of this rule is the date on which the shift starts or the date of the hours to which the booking relates.

Example

The cyclical schedule provides for an employee to work shift A on Wednesday 8 February 2017. Shift A is made up of the following activities:

Activity type	Start	End
Work	09:00	13:00
Break	13:00	13:30
Work	13:30	17:00

However, the schedule provides for the employee to work shift B on the same day. Shift B is made up of the following activities:

Activity type	Start	End
Work	13:00	17:00
Break	17:00	17:30
Work	17:30	21:00

Whenever a shift is different from the shift according to the cyclical schedule, the employee will be credited for the **working time** of the shift actually worked that does not overlap the original shift in the cyclical schedule between [**16:00**] and [**18:00**] on a [**Monday, Tuesday, Wednesday, Thursday, Friday**]. This rule applies: [**always**]

☐ This rule does[**only**] apply to the first [**00:00**] hours.
☐ This rule does [**only**] apply if the shift started before [**00:00**].

[**Hours** | **Worked hours** | **100**]

Hours		01-01-2017 to 01-01-2018
Worked hours		
> 08-02-2017	B	00:30
Total		00:30

7 Shift types

The compensation rule in this group can be used to filter on entries made for instructor shifts.

7.1 Instructor shift allowance

Stipulates that a value is to be booked when an employee performs an instructor shift. Allows for rounding off.

How to use

The following entry is made for an instructor shift.
If the entry is a percentage the entry is rounded to the [**Relationship**] [**Multiple**] minutes.

☐ The entry is a fixed value

[**Target account** | **Target category** | **Percentage**]

- **Relationship** is an expression [previous, nearest, next] indicating the direction in which rounding off is performed.
- **Multiple** is the minute-multiple to which the booking must be rounded off.

If the checkmark **The entry is a fixed value** is selected a fixed value is booked for each instructor shift. In this case the variable **%** below is replaced by the variable **Value**, here it is possible to define the value that should be made for each entry.

Example

Shift A, as scheduled for 25 February 2017, is made up of the following activities:

Activity type	Start	End
Work	08:00	15:30

The shift is designated instructor shift.

The following entry is made for an instructor shift.

If the entry is a percentage the entry is rounded to the [**next**] [**60**] minutes.

☐ The entry is a fixed value

[**Hours** | **Instruction** | **100**]

Hours	01-01-2017 to 01-01-2018
Instruction	
> 25-02-2017 A	08:00
Total	08:00

8 Sleeping hours

The compensation rule in the group **Sleeping hours** can be used to determine the allowances in case the rest period between shifts is too short.

8.1 Sleeping hours

Stipulates that, if a rest is followed by a rest period of insufficient length, the shortfall is to be credited to the employee.

How to use

Whenever the rest period following a shift is less than [**Duration**] hours an employee will be compensated with time off.
[**Reassignment history**]

[**Target account** | **Target category** | **Percentage**]

- **Duration** is the minimum rest period length.
- **Reassignment history** is a phrase describing the applicability of the rule by reference to the shift reassignment history [always, only in case of an exchange, not in case of an exchange].

The booking date is the date on which the inadequate rest period starts.

Example

On 9 and 10 February 2017, an employee is scheduled to work shift A and shift B in succession. Shift A is made up of the following activities:

Activity type	Start	End
Work	15:00	23:00

Shift B is made up of the following activities:

Activity type	Start	End
Work	07:00	15:00

Whenever the rest period following a shift is less than [**11:00**] hours an employee will be compensated with time off.
[**always**]

[**Hours** | **Worked hours** | **100**]

Hours	01-01-2017 to 01-01-2018
Worked hours	
> 09-02-2017 A	03:00
Total	03:00

9 Carry over

The compensation rules in this group can be used to carry over bookings from one category to another. The way in which you wish to carry over a booking can differ per occasion. The different compensation rules within this group offer different ways to carry over bookings from one category to another. How these compensation rules carry over bookings is explained separately in the sections below.

> It is possible to carry over bookings from one category to another, for categories that are in the same account and categories that are in different accounts.

9.1 Account category to carry over (maximum)

Stipulates that – subject to a monthly maximum – the sum of the bookings under a particular category in a particular account are to be carried over to a category in another account, after being divided by a fixed value and rounded up to a whole number.

How to use

The sum of the entries on account [**Account**], category [**Category**] will be entered on the categories below. The value that will be entered is the mentioned sum / [**Denominator**] (from [**Significance**] rounded to the next highest unit) with a maximum of [**Maximum**] each month.

[**Target account** | **Target category**]

- **Account** is the account from which the balance is to be carried over to the target category.
- **Category** is the category in the source account from which the balance is to be carried over to the target category.
- **Denominator** is the number by which the sum of the bookings is to be divided.
- **Significance** is the fraction above which a value is to be rounded up.
- **Maximum** is the maximum value that may be booked per month.

Example

The ‘Hours’ account includes the following bookings:

Hours	01-03-20175 to 01-04-2018
Hours worked	
> 14-03-2017	08:00
> 26-03-2017	08:00
Total	16:00

The sum of the entries on account [**Hours**], category [**Hours worked**] will be entered on the categories below. The value that will be entered is the mentioned sum / [**7.2**] (from [**0.1**] rounded to the next highest unit) with a maximum of [**22**] each month.

[**Compensation** | **Hours**]

Compensation	01-03-2017 to 01-04-2018
Hours	
> 31-03-2017	3
Total	3

In this example, the amount booked will be 16.0/7.2 = 2.22, which will be rounded up to 3.

9.2 Account category to which to be carried over

Stipulates that bookings under a particular category in one account are to be carried over to a particular category in another account, if the value of an employee attribute is as specified.

How to use

Entries on account [**Account**], category [**Category**] will be entered as follows.

☐ Enter Hours

☐ Enter difference total - hours

☐ This rule is only valid if the employee property [**Property**] [**does/does not**] have value [**Value**].

☐ Transfer bookings that have not been approved.

[**Target account** | **Target category** | **%**]

- **Account** is the account from which the bookings are to be carried over.
- **Category** is the category in the source account from which the bookings are to be carried over.
- **Property** is the desirable property to make distinction with.
- **Value** is an value related to the employee property; bookings relating to an employee are carried over only if the value of the employee’s property is or is not as specified.
- If the checkmark **Book hours** is selected, the actual number of hours booked under the relevant category will be carried over.
- If **Book difference total-hours** is selected, the specified percentage of the difference between the value and the number of hours booked will be carried over.

If both options are activated, the specified percentage of the value will be carried over.

- The checkmark **This rule is only valid if the employee property...** is selected if there is made a distinction in the employee properties.
  - **Property** is the desirable property to make distinction with.
  - **Value** is an value related to the employee property; bookings relating to an employee are carried over only if the value of the employee's property is or is not as specified.
- The checkmark **Transfer bookings that have not been approved** is selected if the bookings are made before the entries on the source account are approved.

Example

An employee has the following attribute:

Monetary overtime hours compensation	no	01-01-2017

The account 'Overtime hours' includes the following booking:

Overtime hours	01-01-2017 to 01-01-2018
Extra hours	
> 01-01-2017	10:00 / 140% / 14:00
Total	14:00

Entries on account [**Overtime hours**], category [**Extra hours**] will be entered as follows.

☒ Enter Hours

☐ Enter difference total - hours

☒ This rule is only valid if the employee property [**Monetary overtime hours compensation**] [**does not**] have value [**yes**].

☐ Transfer bookings that have not been approved.

[**Hours** | **Compensation hours** | **100**]

Hours	01-01-2017 to 01-01-2018
compensation hours	
> 01-01-2017	10:00 / 100% / 10:00
Total	10:00

Example

Entries on account [**Overtime hours**], category [**Extra hours**] will be entered as follows.

☐ Enter Hours

☒ Enter difference total - hours

☒ This rule is only valid if the employee property [**Monetary overtime hours compensation**] [**does not**] have value [**yes**].

☐ Transfer bookings that have not been approved.

[**Hours** | **Compensation hours** | **100**]

Hours	01-01-2017 to 01-01-2018
compensation hours	
> 01-01-2017 A	10:00 / 40% / 04:00
Total	04:00

Example

Entries on account [**Overtime hours**], category [**Extra hours**] will be entered as follows.

☒ Enter Hours

☒ Enter difference total - hours

☒ This rule is only valid if the employee property [**Monetary overtime hours compensation**] [**does not**] have value [**yes**].

☐ Transfer bookings that have not been approved.

[**Hours** | **Compensation hours** | **100**]

Hours	01-01-2017 to 01-01-2018
compensation hours	
> 01-01-2017 A	10:00 / 140% / 14:00
Total	14:00

9.3 Carry over all account categories with expression

Stipulates that all bookings under all categories in one account are to be carried over to the categories of the same names in another account, and that an expression is to be applied to the bookings made in the target account.

How to use

For entries on all categories of account [**Account**] a new entry will be created per [**Period**] according to the following expression on the accounts below. For each category, an entry will be made on the category with the same name on the accounts below. If such a category does not exist, it will be made.

☐ Enter on the last day of the period

Expression

[**Target account** | **Percentage**]

- **Account** is the account from which entries are to be carried over.
- **Period** is the period [day, week, month, quarter, year, entry, shift, salary period, equal properties, entry day, period type] for which entries are to be carried over.

  The period period type can be customized by the use itself in the window Maintenance, Period types.

- If the checkmark **Enter the amount on the last day of the period** is selected the entry date is the last day of the selected period. The default booking date is the first day of the specified period.
- **Expression** is a formula to be applied to entries made in the target account.

For an explanation of expressions, see Expressions.

Example

Account Hours includes the following entries:

Hours	30-01-2017 to 06-02-2017
Category A	
> 31-01-2017	10:00
> 01-02-2017	14:00
Category B	
> 31-01-2017	10:00
> 01-02-2017	10:00
Total	44:00

For entries on all categories of account [**Hours**] a new entry will be created per [**week**] according to the following expression on the accounts below. For each category, an entry will be made on the category with the same name on the accounts below. If such a category does not exist, it will be made.

☐ Enter on the last day of the period

VALUE(ALL_CATEGORIES)

[**Hours 2** | **100**]

Hours 2	30-01-2017 to 06-02-2017
Category A	
> 30-01-2017	24:00
Category B	
> 30-01-2017	20:00
Total	44:00

9.4 Carry over per cost center

Stipulates that bookings under a particular category in one account are to be carried over to a particular category in another account, and that the bookings are grouped by cost center.

How to use

The sum of the entries on account [**Account**], category [**Category**] will be entered on the categories below.
The entries will be grouped by cost center.

☐ The entries will also be grouped by department.

The entries will be made on the last day of every **month/salary period**.

☐ Use fast calculation. Turn off this option when the selected categories above are dependent on one of the categories below.

[**Target account**| [**Target category**] | **Percentage**]

- **Account** is the account from which entries are to be carried over.
- **Category** is the category in the source account from which the bookings are to be carried over.

Example

Account Hours includes the following entries:

Hours	01-01-2017 to 01-02-2017
Hours worked	
> 03-01-2017	8:00 / Cost center 1
> 04-01-2017	6:00 / Cost center 2
> 06-01-2017	4:00 / Cost center 1
> 06-01-2017	4:00 / Cost center 2
Total	22:00

The sum of the entries on account [**Hours**], category [**Hours worked**] will be entered on the categories below.
The entries will be grouped by cost center.

☐ The entries will also be grouped by department.

The entries will be made on the last day of every **month**.

☒ Use fast calculation. Turn off this option when the selected categories above are dependent on one of the categories below.

[**Costs**| [**Working hours**] | **100**]

Costs	01-01-2017 to 01-02-2017
Working hours	
> 31-01-2017	12:00 / Cost center 1
> 31-01-2017	10:00 / Cost center 2
Total	22:00

9.5 Enter time for time

Stipulates that bookings under a particular category in one account are to be carried over to a particular category in another account, if the value of an employee attribute and the chosen time-for-time option are as specified.

How to use

Entries on account [**Account**], category [**Category**] will be entered as follows, when the chosen method of payment for time for time is [**Option**].

☐ Enter hours

☐ Enter difference total - hours

☐ This rule is only valid if the employee property [**Property**]...

☐ Transfer bookings that have not been approved

[**Target account** | **Target category** | **Percentage**]

- **Account** is the account from which the bookings are to be carried over.
- **Category** is the category in the source account from which the bookings are to be carried over.

- **Option** is the chosen time-for-time option [no time for time, time for time] on which a booking is based.
- If the checkmark **Enter hours** is selected, the actual number of hours booked under the relevant category will be carried over.
- If **Enter difference total-hours** is selected, the specified percentage of the difference between the value and the number of hours booked will be carried over.

> If both options are activated, the specified percentage of the value will be carried over.

- The checkmark **This rule is only valid if the employee property [Property] [does/does not] have value [Value]** is selected if there is made a distinction in the employee properties.
  - **Property** is the desirable property to make distinction with.
  - **Value** is an value related to the employee property; bookings relating to an employee are carried over only if the value of the employee's property is or is not as specified.
- The checkmark **Transfer bookings that have not been approved** is selected if the bookings are made before the entries on the source account are approved.

Example

An employee has the following attribute:

Lease car	yes	01-01-2017

The employee has opted **time for time**.

The account Overtime hours includes the following booking:

Overtime hours	01-01-2017 to 01-01-2018
Extra hours	
> 01-01-2017 A	10:00 / 140% / 14:00
Total	14:00

Entries on account [**Overtime hours**], category [**Extra hours**] will be entered as follows, when the chosen method of payment for time for time is [**time for time**].

☒ Enter hours

☒ Enter difference total - hours

☒ This rule is only valid if the employee property [**lease car**] [**does**] have value [**yes**]

☐ Transfer bookings that have not been approved

[**Hours** | **Compensation hours** | **100**]

Hours	01-01-2017 to 01-01-2018
Compensation hours	
> 01-01-2017 A	10:00 / 140% / 14:00
Total	14:00

9.6 Transfer balance per salary period

Stipulates that the positive or negative balance accumulated under a particular category in a particular account in the course of a salary period is to be carried over to the first day of the following salary period.

How to use

The [**positive/negative**] balance of account [**Account**], category [**Category**] over the salary period will be entered on the first day after the salary period accordingly.

☐ Transfer bookings that have not been approved

[**Target account** | **Target category** | **Percentage**]

- **Positive/negative** is a descriptor determining whether a positive or negative balance or both balances should be carried over.
- **Account** is the account from which the balance is to be carried over to the target category.
- **Category** is the category in the source account from which the balance is to be carried over to the target category.

If the checkmark **Transfer bookings that have not been approved** is selected also the entries that are based upon shifts of schedules that have a status lower than approved are transferred.

Example

The salary period 03 2017 runs from 28 February 2017 to 3 April 2017.

Hours	01-03-2017 to 04-04-2017
Hours worked	
> 14-03-2017	08:00
> 26-03-2017	07:30
Total	15:30

The [**positive**] balance of account [**Hours worked**], category [**Hours**] over the salary period will be entered on the first day after the salary period accordingly.

☐ Transfer bookings that have not been approved

[**Compensation** | **Hours** | **100**]

Compensation	01-03-2017 to 04-04-2017
Hours	
> 03-04-2017	15:30
Total	15:30

9.7 Transfer maximum

Stipulates that the highest or lowest of the values booked under two categories in a specified account is to be carried over to a specified category in another account.

How to use

The [**minimum/maximum**] of entries on account [**Account**], category [**1st category**] and category [**2nd category**] will be entered as described below.

Compare entries per [**shift/day**].

☐ Enter hours

☐ Enter difference - total hours

☐ Transfer bookings that have not been approved

[**Target account** | **Target category** | **Percentage**]

- **Min/max** indicates whether the highest or lowest [maximum, minimum] of the two bookings is to be carried over.
- **Account** is the account of the categories from which the bookings are to be carried over.
- **1st category** and **2nd category** are the categories of the bookings that are to be compared.
- **Shift/day** is a statement of the comparison interval.
- If the checkmark **Book hours** is selected, the actual number of hours booked under the relevant category will be carried over.
- If **Book difference total-hours** is selected, the specified percentage of the difference between the value and the number of hours booked will be carried over.

> If both checkmarks are selected, the specified percentage of the value will be carried over.

If the checkmark **Transfer bookings that have not been approved** is selected also the entries that are based upon shifts of schedules that have a status lower than approved are transferred.

Example

The account Hours includes the following booking:

Hours	01-01-2017 to 01-01-2018
Hours A	
> 01-01-2017	10:00
Hours B	
> 01-01-2017	14:00
Total	24:00

The [**maximum**] of entries on account [**Hours**], category [**Hours A**] and category [**Hours B**] will be entered as described below.

Compare entries per [**day**].

☒Enter hours

☒ Enter difference - total hours

☐ Transfer bookings that have not been approved

[**Overtime hours** | **Compensation hours** | **100**]

Overtime hours	01-01-2017 to 01-01-2018
Compensation hours	
> 01-01-2017	14:00
Total	14:00

9.8 Transferbooking based upon an expression

Stipulates that bookings under a specified category are to be carried over to one or more other categories, and that an expression is to be applied to the bookings made in the target account.

How to use

For entries on the [**Account**], [**Categories**] the value following from the following expression will be carried over to the following accounts per [**Period**].

☐ Enter on the last day of the period
☐ Enter per department

Expression

[**Target account** | **Target category** | **Percentage**]

- **Account** is the account from which all bookings are carried over.
- **Categories** is the category or are the categories from which all bookings are carried over.
- **Period** is the period [day, week, month, quarter, year, entry, shift, salary period, equal properties, entry day, period type] for which bookings are to be carried over.

  The period type can be customized by the use itself in the window Maintenance, Period types.

- If the checkmark Enter per department is selected the booking is carried over the balances per department to another account or category
- **Expression** is a formula to be applied to bookings made in the target account.

Expressions may be entered on screen by completing the [expression] field. On-screen expression entry is supported by a menu that pops up when the user right-clicks on the [expression] field. From this menu, the user may select operators, functions, expressions and categories. For more information about the expressions see "Expressions" on page 1.

The default booking date is the first day of the specified period. If the checkmark **Enter on the last day of the period** is selected the booking date is the last day of the selected period.

Example

Account 1, within which categories 1A and 1B are defined, includes the following bookings:

Account 1	01-05-2017 to 08-05-2017
1A	
> 02-05-2017 A	10:30 / 100% / 10:30
> 05-05-2017 A	-5:00 / -50% / 02:30
1B	
> 03-05-2017 B	10:00 / 100% / 14:00
Total	27:00

For entries on the [**Account 1**], [**All**] the value following from the following expression will be carried over to the following accounts per [**week**].

☐ Enter on the last day of the period
☐ Enter per department

MAX(VALUE('1A'),VALUE('2A'))

[**Account 2** | **2A** | **100**]

Account 2	01-05-2017 to 08-05-2017
2A	
> 02-05-2017	14:00 / 100% / 14:00
Total	14:00

10 Carry over balance

The compensation rule in this group can be used to carry over the balance of an account category. The difference of the compensation rules within this group with the compensation rules in the group **Carry over** is that the entries of the group **Carry over balance**are made on the start times of the balance periods and not on the dates the entries are made.

10.1 Carry over balance (1)

Stipulates that the balance under a particular category or categories is/are to be carried over to the following year in response to a manual carryover command.

How to use

At the end of a year balances on these account categories will be transferred to the next year as follows:

[**Target account** | **Target category** | **Percentage**]

This rule is activated manually by selecting the option **Carry over** from the menu **Accounts**. If further adjustments to the source category balance are made after the carryover has been performed, it should be repeated.
Only one instance of this rule may be included in a given rule set.
Where possible, the rule 'Carry balance (2)' should be used in preference to this rule.

Example

Leave	01-01-2016 to 01-01-2017
Current year's entitlement	
01-01-2016 Correction	180:00
Total	180:00

At the end of a year balances on these account categories will be transferred to the next year as follows:

[**Leave** | **Current year's entitlement** | **100**]

If from the menu **Accounts**, the option **Carry over** and the relevant department is selected, the account Leave and year 2016 are selected, the result is as follows:

Leave	01-01-2016 to 01-01-2018
Current year's entitlement	
01-01-2016 Correction	180:00
31-12-2016 Carried over to next year	-180:00
01-01-2017 Carried over from previous year	180:00
Total	180:00

10.2 Carry over balance (2)

Stipulates that, at the end of a period, the balance on a particular account is to be carried forward to the following period.

How to use

The balance on account [**Account**], categories [**Categories**] will be transferred as follows: [**Period**]

☐ All entries are sorted and transferred per department
from [**Start date**] to [**End date**] on [**Booking date**] to:

[**Target account** | **Target category** | **Percentage**]

- **Account** is the account from which the balance is to be carried over to the target category.
- **Categories** is the category (or categories) of the selected account for which the balance should be carried over.

> It is possible to transfer the balance of one, multiple or all account categories of the selected account. This is possible by selecting the checkmark of multiple categories or by selecting the checkmark All.

- **Period** is the period [period and date as below, per year, per quarter, per month, per week, per salary period] for which bookings are to be carried forward.
- If the checkmark **All entries are sorted and transferred per department** is selected the entries are made for the departments separately.
- **Start date** is the start date of the period for which bookings are to be carried forward.
- **End date** is the end date of the period for which bookings are to be carried forward.
- **Booking date** is the date on which the result is to be booked under the target category.

> The string "from [**Start date**] to [**End date**] on [**Booking date**] to:" is only available if for **Period** the option **Period and date as below** is selected.
> If the Period is the [per year, per quarter, per month, per week, per salary period], then the booking date is the first day after the period for which bookings are to be carried forward.
> The target category may be a category in the source account.

Example

Leave	01-01-2016 to 01-01-2017
Current year's entitlement	180:00
Carried forward from previous year	0:00
Taken in current year	-120:00
Total	60:00

The balance on account [**Leave**] will be transferred as follows: [**per year**]

☐ All entries are sorted and transferred per department

[**Leave** | **Carried forward from previous year** | **100**]

Leave	01-01-2017 to 01-01-2018
Current year's entitlement	180:00
Carried forward from previous year	
> 01-01-2017	60:00
Taken in current year	0:00
Total	240:00

Example

The balance on account [**Leave**] will be transferred as follows: [**period and date as below**]

☐ All entries are sorted and transferred per department
from [**01-01-2016**] to [**01-01-2017**] on [**01-01-2017**] to:

[**Unused leave** | **Unused in 2016** | **100**]

Unused leave	01-01-2017 to 01-01-2018
Unused in 2016	
> 01-01-2017	60:00
Total	60:00

11 Clocking entry with function key

The compensation rule in this group can be used to create an entry based on a clock-in or clock-out booking of a specific function key.

11.1 Clocking entry with function key

Stipulates that a fixed value is to be booked if a function key is used when making an incoming or outgoing clocking booking.

This rule is available only in the Time Registration Module.

How to use

This rule applies for [**Direction**] in case the clocking is for one of the following basic activity types: [**Activity types**]
If the function key [**Key**] is selected, a value of [**Value**] is added.

[**Target account** | **Target category** | **Percentage**]

- **Direction** is a descriptor [Clock in, booking out] indicating the direction of the booking.
- **Activity types** are the (primary) activity types that the clocked time must include for a booking to be made.
- **Key** is the function key to be used.
- **Value** is the fixed value to be booked if the specified function key is used.

Example

Shift A, as scheduled for Saturday 6 May 2017, is made up of the following activities:

Activity type	Start	End
Work	08:00	12:00
Break	12:00	12:30
Work	12:30	16:00

This rule applies for [**Clock in**] in case the clocking is for one of the following basic activity types: [**work**]
If the function key [**F1**] is selected, a value of [**7**] is added.

[**Claims** | **Cycle allowance** | **100**]

Claims	01-01-2017 to 01-01-2018
Cycle allowance	
> 06-05-2017	7
Total	7

12 Duty registration

The rule in the group **Duty registration** can be used to book the accounting information from shifts to an account.

12.1 Duty registration

Stipulates the value of the duty registration is to be credited.

This rule is only available in the Accounting Module.

How to use

Enter the [**times/amount**] from the registration of a shift.

[**Target account** | **Target category**| **Percentage**]

Example

On Saturday 6 May 2017 an employee is assigned to the shift A, with the following makeup:

Activity type	Start	End
Work	8:00	12:00
Break	12:00	12:30
Work	12:30	16:30

Registration tab of the shift:

Cost center	Amounts and hours
Cost center 1	6:00
Cost center 2	1:00
Cost center 3	1:00

Enter the [**times**] from the registration of a shift.

[**Costs** | **Registration**| **100**]

Costs	01-01-2017 to 01-01-2018
Registration	
> 06-05-2017	6:00 / Cost center 1
> 06-05-2017	1:00 / Cost center 2
> 06-05-2017	1:00 / Cost center 3
Total	8:00

13 Fixed entry

All the various types of compensation rules in the group **Fixed entry** will always make a fixed entries. For example if a shift contains the activity type **Work** a fixed entry of 8 will be made.

> For the compensation rules, outside the group **Fixed entry**, the entry made is generally the number of hours for which the compensation rule returns a violation.

13.1 Fixed entries for activity class in a period

Stipulates that a fixed value is to be booked for each activity of a specified type or for each shift involving an activity of a specified type that an employee undertakes, if the timing of the activity and the employee's salary code are as specified.

How to use

Applies to [**Status**] time with [**Type**] [**Attribute**]

Enter [**Value**] hours per [**Basis**] of which the [**Timing**] for the above mentioned activities is between [**Start time**] and [**End time**] hours on one of the following days: [**Days of the week**].

This rule applies to all employees with salary codes between [**Min salary code**] and [**Max salary code**].

☐This rule applies only to employees for whom the employee property [**Property**] ... [**does/does not**] have the value [**Value**].

Calculations will be base upon [**Schedule**]

[**Target account** | **Target category** | **Percentage**]

- **Status** is the status [spent, planned] of the hours to be booked. Spent time is calculated on the basis of the realized shift, planned time on the basis of the shift as originally scheduled.
- **Type** is the type [class, activity types, main activity type] to which the attributes, for which the bookings are to be made, belongs to.
- **Attribute** is the attribute for which the bookings are to be made.
- **Value** is the fixed value of the booking.
- **Basis** is the basis [activity, shift] for the fixed booking.
- **Timing** specifies the reference point of the activity's timing [time in between, begin time, end time, begin time and end time, begin time or end time], which needs to fall between start and end for a booking to be triggered.
- **Start time** is the start of the time interval within which work needs to be scheduled in order to trigger a booking.
- **End time** is the end of the time interval within which work needs to be scheduled in order to trigger a booking.
- **Days of the week** are the days of the week [Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday, public holiday] in respect of which hours are to be booked.

- **Min salary code** is the lowest salary code that an employee may have for a booking to be triggered.
- **Max salary code** is the highest salary code that an employee may have for a booking to be triggered.
- **Property** is the employee property that is checked in order to determine if an entry is made.
- **Value** is the value an employee should have for the specified employee property in order to make an entry.
- **Schedule** is the schedule [Published master schedule, Realized master schedule, Cyclical schedule] on which the calculations are based upon.

Example

Shift A, as scheduled for 9 April 2017, is made up of the following activities:

Activity type	Start	End
Work	08:00	12:00
Break	12:00	12:30
Work	12:30	16:00

The activity type 'Work' is associated with the highest level of the organizational hierarchy ('Organization). The employee's salary code is C.

Applies to [**planned**] time with [**Activity types**] [**Work**]

Enter [**2**] hours per [**activity**] of which the [**start time**] for the above mentioned activities is between [**07:00**] and [**19:00**] hours on one of the following days: [**Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday**].

This rule applies to all employees with salary codes between [**A**] and [**F**].

☐This rule applies only to employees for whom the employee property [**(None)**] ...

Calculations will be base upon [**Realized master schedule**]

[**Hours** | **Allowance hours** | **100**]

Hours	01-01-2017 to 01-01-2018
Worked hours	
> 09-04-2017 A	04:00
Total	04:00

13.2 Fixed entries for an activity class per salary code

Stipulates that a fixed value is to be booked for each occasion that an employee is scheduled to undertake an activity of a specified kind, if the employee's salary code is as specified.

How to use

Whenever a shift contains elements like [**Kinds**]

and the employees salary classification is between [**Min salary code**] and [**Max salary code**], a [**Value**] entry will be made on.

☐This rule applies only to employees for whom the employee property [**Property**] ... [**does/does not**] have the value [**Property value**].

[**Target account** | **Target category**]

- **Kinds** are the classes [Attendance, Occasional Leave, Special leave, ..., Sick leave, Sick leave (no work)] to which the shift's activity types need to belong in order to trigger a booking.
- **Min salary code** is the lowest salary code that an employee may have for a booking to be triggered.
- **Max salary code** is the highest salary code that an employee may have for a booking to be triggered.
- **Value** is the fixed value of the booking.
- **Property** is the employee property that is checked in order to determine if an entry is made.
- **Property value** is the value an employee should have for the specified employee property in order to make an entry.

Example

Shift A is scheduled for Saturday 8 April 2017. The shift is to be worked by an employee on salary code C and is made up of the following activities:

Activity type	Start	End
Work	8:00	12:00
Break	12:00	12:30
Work	12:30	16:00

The activity type 'Work' belongs to the activity class 'Work'. The activity type 'Break' belongs to the activity class 'Break'.

Whenever a shift contains elements like [**Work**]

and the employees salary classification is between [**A**] and [**F**], a [**8**] entry will be made on.

☐This rule applies only to employees for whom the employee property [**(None)**] ...

[**Hours** | **Worked hours**]

Hours	01-01-2017 to 01-01-2018
Worked hours	
> 08-04-2017 A	08:00
Total	08:00

13.3 Fixed entries for an activity treatment on a working day

Stipulates that a fixed value is to be booked for each occasion that an employee is scheduled to undertake an activity of a specified kind, with a maximum of the value booked in a day and/or a week.

How to use

If a shift contains an activity type of class: [**Kinds**]

A fixed value of [**Value**] will be entered:

The maximum value per week is: [**Max per week**].
The maximum value per day is: [**Max per day**].

☐This rule applies only to employees for whom the employee property [**Property**] ... [**does/does not**] have the value [**Property value**].

[**Target account** | **Target category**]

- **Kinds** are the classes [Amplitude, Availability (no labor),..., Work (stand by), WTR] to which the shift's activity types need to belong in order to trigger a booking.
- **Value** is the fixed value of the booking.
- **Max per week** is the maximum value to be booked in one week.
- **Max per day** is the maximum value to be booked in one day.
- **Property** is the employee property that is checked in order to determine if an entry is made.
- **Property value** is the value an employee should have for the specified employee property in order to make an entry.

Example

Shift A is scheduled for Saturday 8 April 2017 and made up of the following activities:

Activity type	Start	End
Work	8:00	11:00
Break	11:30	12:00
Work	12:00	14:00

Shift B is also scheduled for Saturday 8 April 2017 and made up of the following activity:

Activity type	Start	End
Work	15:00	18:00

The activity type 'Work' belongs to the activity class 'Work'. The activity type 'Break' belongs to the activity class 'Break'.

If a shift contains an activity type of class: [**Work**]

A fixed value of [**8**] will be entered:

The maximum value per week is: [**50**].
The maximum value per day is: [**10**].

☐This rule applies only to employees for whom the employee property [**(None)**] ...

[**Hours** | **Worked hours**]

Hours	01-01-2017 to 01-01-2018
Worked hours	
> 08-04-2017 A	08:00
> 08-04-2017 B	02:00
Total	10:00

13.4 Fixed entry for an activity sort

Stipulates that a fixed value is to be booked for each occasion that an employee is scheduled to undertake an activity of a specified class.

How to use

Whenever a shift contains the following classes [**Kinds**]

a [**Value**] entry will be made on:

☐This rule applies only to employees for whom the employee property [**Property**] ... [**does/does not**] have the value [**Property value**].

[**Target account** | **Target category**]

- **Kinds** are the classes [Attendance, Occasional Leave, Special leave, ..., Work (stand by), WTR] to which the shift's activity types need to belong in order to trigger a booking.
- **Value** is the fixed value of the booking.
- **Property** is the employee property that is checked in order to determine if an entry is made.
- **Property value** is the value an employee should have for the specified employee property in order to make an entry.

Example

Shift A, as scheduled for Saturday 8 April 2017, is made up of the following activities:

Activity type	Start	End
Work	8:00	12:00
Break	12:00	12:30
Work	12:30	16:00

The activity type 'Work' belongs to the activity class 'Work'. The activity type 'Break' belongs to the activity class 'Break'.

Whenever a shift contains the following classes [**Work**]

a [**8**] entry will be made on:

☐This rule applies only to employees for whom the employee property [**(None)**] ...

[**Hours** | **Worked hours**]

Hours	01-01-2017 to 01-01-2018
Worked hours	
> 08-04-2017 A	08:00
Total	08:00

13.5 Fixed entry for an activity type

Stipulates that a fixed value is to be booked for each occasion that an employee is scheduled to undertake an activity of a specified type.

How to use

For shifts with activities belonging to department [**Department**]:

, with type [**Activity type**], or secondary activities belonging to this type, a fixed amount of [**Value**] per [**Basis**] will be entered.

☐This rule applies only to employees for whom the employee property [**Property**] ... [**does/does not**] have the value [**Property value**].

[**Target account** | **Target category**]

- **Department** is the department associated with the activity type.
- **Activity type** is the type of the activities for which bookings are to be made.
- **Value** is the fixed value of the booking per shift or activity.
- **Basis** is the basis [activity, shift] for the fixed booking.
- **Property** is the employee property that is checked in order to determine if an entry is made.
- **Property value** is the value an employee should have for the specified employee property in order to make an entry.

Example

Shift A, as scheduled for 8 April 2017, is made up of the following activities:

Activity type	Start	End
Work	23:00	03:00
Break	03:00	03:30
Work	03:30	07:00

The activity type 'Work' is associated with the highest level of the Department ('Organization').

For shifts with activities belonging to department [**Organization**]:

, with type [**Work**], or secondary activities belonging to this type, a fixed amount of [**1**] per [**Activity**] will be entered

☐This rule applies only to employees for whom the employee property [**(None)**] ...

[**Days** | **Half days**]

Days	01-01-2017 to 01-01-2018
Half days	
> 08-04-2017 A	1
> 09-04-2017 A	1
Total	2

13.6 Time between shifts

Stipulates that a fixed value is to be booked if the time between a shift and the previous or next shift is within specific time.

How to use

If the time between a shift and the [**Direction**] shift is greater than [**Minimum time**] and less than [**Maximum time**], a fixed amount of [**Value**] will be entered.
Activities with the following activity types are not considered.
[**Activity types**]
In the event a shift consists of only the activity types mentioned above, the previous or next shifts are used.

[**Target account** | **Target category** | **Percentage**]

- **Direction** is the direction [next, previous, previous or next] for which the time between shifts is checked.

- **Minimum time** is the minimum time that should be within two subsequent shifts in order to trigger a booking.
- **Maximum time** is the maximum time that is allowed to be within two subsequent shifts in order to trigger a booking.
- **Activity types** are the activity types you do wish not to include within this rule.

If the direction **previous or next** is chosen, each time shifts follow each other within the specific time intervals two bookings are made; one booking for each of the shifts.

Example

On 9 and 10 February 2017, an employee is scheduled to work shift A and shift B in succession. Shift A is made up of the following activities:

Activity type	Start	End
Work	15:00	23:00

Shift B is made up of the following activities:

Activity type	Start	End
Work	07:00	15:00

If the time between a shift and the [**next**] shift is greater than [**00:00**] and less than [**10:00**], a fixed amount of [**1**] will be entered.
Activities with the following activity types are not considered.
[**(None)**]
In the event a shift consists of only the activity types mentioned above, the previous or next shifts are used.

[**Allowances** | **Rest** | **100**]

Allowances		01-01-2017 to 01-01-2018
Rest		
> 09-02-2017	A	01:00
Total		01:00

Example

Now the direction **previous or next** is chosen.

If the time between a shift and the [**previous or next**] shift is greater than [**00:00**] and less than [**10:00**], a fixed amount of [**1**] will be entered.
Activities with the following activity types are not considered.
[**(None)**]
In the event a shift consists of only the activity types mentioned above, the previous or next shifts are used.

[**Allowances** | **Rest** | **100**]

Allowances		01-01-2017 to 01-01-2018
Rest		
> 09-02-2017	A	01:00
> 10-02-2017	B	01:00
Total		02:00

14 Fixed entry for a day

The compensation rules in this group can be used to enter a fixed value per day with a shift.

The compensation rule "Threshold value for an entry per working day" is only useful in case in the compensation rule "Entry for a working day" the checkmark "Boundary values apply to this rule" is selected within the same set of rules.

14.1 Entry for a working day

Stipulates that a fixed value is to be booked for each day on which at least one shift that involves working time is scheduled.

How to use

Enter a fixed value of [**Value**] on a day in case at least 1 shift starts on that day that counts as work.

☐ Boundary values apply to this rule.
☐ Include shifts for the same employee in different employee conditions groups.

Calculations will be based upon [**Schedule**]

[**Target account** | **Target category**]

- **Value** is the value to be booked if a day includes working time.
- **Schedule** is the schedule [Published master schedule, Realized master schedule, Cyclical schedule] on which the calculations are based upon.

If the checkmark **Boundary values apply to this rule** is selected the threshold values indicated in the rule "Threshold value for an entry per working day" will be used to make a correction if necessary.

If the checkmark **Include shifts for the same employee in different employee conditions groups** is selected, the time an employee is assigned to a work shift is a summation of the work time of different employee conditions groups. This can be helpful in case the checkmark **Boundary values apply to this rule** is selected.

Example

Shift A, as scheduled for Monday 8 May 2017 and Wednesday 10 May 2017, is made up of the following activities:

Activity type	Start	End
Work	08:00	12:00
Break	12:00	12:30
Work	12:30	16:00

The activity type 'Work' belongs to the activity class 'Work'. The activity type 'Break' belongs to the activity class 'Break'.

Enter a fixed value of [**2**] on a day in case at least 1 shift starts on that day that counts as work.

☐ Boundary values apply to this rule.
☐ Include shifts for the same employee in different employee conditions groups.

Calculations will be based upon [**Realized master schedule**]

[**Days** | **Working days**]

Hours	01-01-2017 to 01-01-2018
Working days	
> 08-05-2017	2
> 10-05-2017	2
Total	4

14.2 Threshold value for an entry per working day

Defines the threshold value that applies in the context of bookings per worked day.

How to use

The total per week of entries on an account created by "Entry for a working day" type for which boundary values apply must lie between [**Min value**] and [**Max value**]. If this is not the case, a correction follows by means of contra entries.

- **Min value** is the minimum weekly value of the sum; if the sum is below this value, a counter-booking will be made.
- **Max value** is the maximum weekly value of the sum; if the sum is above this value, a counter-booking will be made.

No more than one threshold value for booking per worked day may be defined per rule set.
The rule will trigger a counter-booking only if the whole of the week in question is within the period to which the account relates.

Example

Shift A, as scheduled for Monday 8 May 2017 and Wednesday 10 May 2017, is made up of the following activities:

Activity type	Start	End
Work	08:00	12:00
Break	12:00	12:30
Work	12:30	16:00

The activity type ‘Work’ belongs to the activity class ‘Work’. The activity type ‘Break’ belongs to the activity class ‘Break’.

Enter a fixed value of [**2**] on a day in case at least 1 shift starts on that day that counts as work.

☒ Boundary values apply to this rule.
☐ Include shifts for the same employee in different employee conditions groups.

Calculations will be based upon [**Realized master schedule**]

[**Days** | **Working days**]

The checkmark **Boundary values apply to this rule** is selected.

The total per week of entries on an account created by "Entry for a working day" type for which boundary values apply must lie between [**-3**] and [**3**]. If this is not the case, a correction follows by means of contra entries.

Hours	08-05-2017 to 15-05-2017
Availability	
> 08-05-2017	2
> 10-05-2017	2
> 14-05-2017	-1
Total	3

15 Fixed entry for leave request per calender day

This rule creates an account entry with a fixed value of 1 for each day of a full day, non-partial leave request, begin- and end time of 00:00 hours.

Additional expression rules are required to convert the fixed value of 1 to the actual number of hours the leave day is worth. The end value booked to the leave balance check account should be equal to the setup of the replace leave shift functionality (replace existing shifts, cyclical schedule, contract hours, work pattern).

16 Minimum availability

The compensation rules in this group can be used to make entries related to the employee its 'min/max' contract.

16.1 Min/max: quarterly hours exceeding minimum

Stipulates that, if an employee is on a min/max contract, the hours that he or she works in excess of the minimum during a quarter are to be booked.

How to use

Employees on a 'min/max' contract are paid out quarterly for all hours that were realized above the employee's contractual minimum, as indicated below:

[**Target account** | **Target category** | **%**]

In the context of this rule, the entry date is the first day of the quarter following that to which the entry relates.

HARMONY calculates the number of hours to be worked in a quarter by dividing the number of days in the quarter by seven and multiplying by the minimum number of hours to be worked per week.

Example

An employee has a min/max contract (see 'Personnel management', 'Employment rules' tab) with the following parameters:

- Minimum: 8 hours/week
- Maximum: 24 hours/week

In the first quarter of 2005, the employee works 150 hours.

Employees on a 'min/max' contract are paid out quarterly for all hours that were realized above the employee's contractual minimum, as indicated below:

[**Hours** | **Above minimum** | **100**]

Hours	01-01-2005 to 01-01-2006
Above minimum	
> 01-04-2005	47:25
Total	47:25

17 Non-fixed entry

The compensation rules within this group can be used to determine all kind of non-fixed entries. These compensation rules make use of the different types of activities that are available within ORTEC WS.

In this way it is possible to create different variables of your own choice. For example if you want to determine the variable **Working hours** you can create the category **Working hours** and book the cumulative value of the entries made for the different activities on this category. In this way you can choose to include only the activity **Work**, but also include the variables **Work(stand by)**, **Education**, **Break** etc.

17.1 Clocking: norm, min, max.

Stipulates that the standard, daily minimum or daily maximum time of a shift is to be booked.

This rule is available only in the Time and Attendance Module.

How to use

Take the [**Duration**] of a shift into account. Use the actual labor-time instead, in case

☐ A clocking is processed for this shift.
☐ The norm time equals 0:00 hours.
☐ Count 0:00 hours in case there is no processed clocking and the shift ends in the past.

[**Target account** | **Target category** | **Percentage**]

- **Duration** is a specification of the time [Norm time, Daily minimum, Daily maximum] to be booked.

It is possible to use the actual labor-time as **Duration** in the following cases: 1) A clocking time is processed for this shift; 2) The normtime equals 0:00 hours; 3) Count 0:00 hours in case there is no processed clocking and the shift ends in the past, by selecting the corresponding checkmarks.

Example

Shift A, as scheduled for Friday 5 May 2017, is made up of the following activities:

Activity type	Start	End
Work	08:00	12:00
Break	12:00	12:30
Work	12:30	16:00

The norm time of the shift is 07:45 hours.

Take the [**Norm time**] of a shift into account. Use the actual labor-time instead of the norm time, in case

☐ A clocking time is processed for this shift.
☐ The normtime equals 0:00 hours.
☐ Count 0:00 hours in case there is no processed clocking and the shift ends in the past.

[**Hours** | **Hours to be worked** | **100**]

Hours	01-01-2017 to 01-01-2018
Hours to be worked	
> 05-05-2017 A	07:45
Total	07:45

17.2 Difference between worked hours and employment-hours

Stipulates that the difference between an employee's worked and contractual hours is to be booked on a weekly basis.

How to use

The difference between an employees working hours and contract hours (employment conditions) is entered weekly as follows:
Calculations will be based upon [**Schedule**]

[**Target account** | **Target category** | **Percentage**]

- **Schedule** is the schedule [Published master schedule, Realized master schedule, Cyclical schedule] on which the calculations are based upon.

The booking date is the last day of the week to which the booking relates.

Example

An employee is contracted to work thirty-six hours a week. In the week 6 to 13 February 2017, the employee works five shifts of 7.5 hours.

The difference between an employees working hours and contract hours (employment conditions) is entered weekly as follows:
Calculations will be based upon [**Realized master schedule**]

[**Hours** | **Overtime hours**| **100**]

Hours	01-02-2017 to 01-03-2017
Hours to be worked	
> 12-02-2017	01:30
Total	01:30

17.3 Entry based on employee property

Stipulates that a value is to be booked, based on a specified employee attribute, if an employee works a shift that involves an activity of a specified kind.

How to use

If an employee carries out a shift with at least one activity of kind [**Kinds**]

The entry below is made based in employee property [**Property**]:

[**Target account** | **Target category** | **Percentage**]

- **Kinds** are the kinds [Amplitude, Availability (no labor), Break, ..., Work (stand by), WTR] to which the shift's activity types need to belong in order to trigger a booking.
- **Property** is the name of the employee property upon which the booking will be based.

The specified employee property must be of the 'Integer' type or the 'Decimal number' type.

Example

Shift A, as scheduled for Saturday 8 April 2017, is made up of the following activities:

Activity type	Start	End
Work	8:00	12:00
Break	12:00	12:30
Work	12:30	16:00

The activity type 'Work' belongs to the activity kind 'Work'. The activity type 'Break' belongs to the activity kind 'Break'. The employee has been assigned the attribute 'Distance to work' with the value 23.

If an employee carries out a shift with at least one activity of kind [**Attendance, Operational, Work**]

The entry below is made based in employee property [**Distance to work**]:

[**Travel expenses** | **Distance to work** | **100**]

Travel expenses	01-01-2017 to 01-01-2018
Distance to work	
> 08-04-2017 A	23:00
Total	23:00

17.4 Kind of activity with respect to cyclical schedule

Stipulates that a value is to be booked, based on the number of hours that an employee is scheduled to spend on activities of a specified kind when the cyclical schedule indicates that the employee should undertake an activity of a different kind.

How to use

Whenever an employee has activities of kind [**Kinds**]
, while according to the cyclical schedule this employee should have activities of kind [**Cyclical kinds**]
, this will be paid out on:

[**Target account** | **Target category** | **Percentage**]

- Kinds are the kinds [Amplitude, Availability (no labor),..., Work (stand by), WTR] of activities that are to be booked.
- Cyclical kinds are the kinds [[Amplitude, Availability (no labor),..., Work (stand by), WTR]] to which the activity types of the shifts in the cyclical schedule need to belong in order to trigger a booking.

This rule will generate a booking only if the times of the shift included in the schedule match those of the shift in the cyclical schedule.

The rule generates a booking for each activity.

Example

Shift A, as scheduled for Saturday 8 April 2017, is made up of the following activities:

Activity type	Start	End
Work	08:00	12:00
Break	12:00	12:30
Work	12:30	16:00

The activity type 'Work' belongs to the activity kind 'Work'. The activity type 'Break' belongs to the activity kind 'Break'. However, the cyclical schedule indicates that the employee should have worked the following shift:

Activity type	Start	End
Free time	08:00	12:00
Break	12:00	12:30
Free time	12:30	16:00

The activity 'Free time' belongs to the activity kind 'Free time'.

Whenever an employee has activities of kind [**Work**]
, while according to the cyclical schedule this employee should have activities of kind [**Free time**]
, this will be paid out on:

[**Hours** | **Extra hours** | **100**]

Hours	01-01-2017 to 01-01-2018
Extra hours	
> 08-04-2017	04:00
> 08-04-2017	03:30
Total	07:30

17.5 Minimum availability

Stipulates that the number of hours that an employee is contractually required to work is to be booked each month.

How to use

On the first day of the month the minimum availability of an employee in a working week containing [**Days of the week**] is calculated as follows:

[**Target account** | **Target category** | **Percentage**]

- **Days of the week** are the days of the week [Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday] that count as working days.

This rule can be used to ascertain the hours to be worked in a given period.

Example

An employee is contracted to work thirty-six hours a week. In March 2017, there are twenty-three non-weekend days.

On the first day of the month the minimum availability of an employee in a working week containing [**Monday, Tuesday, Wednesday, Thursday, Friday**] is calculated as follows:

[**Hours** | **Hours to be worked** | **100**]

Hours	01-03-2017 to 01-04-2017
Hours to be worked	
01-03-2017	165:36
Total	165:36

17.6 Minimum availability per period

Stipulates that, the number of hours that an employee is contractually expected to work is booked, for each day that counts as a working day for the employee in question.

How to use

On the first day of a [**Booking period**], the [**Contract duration**] of an employee in a working week containing [**Days of the week**] is calculated as follows.

☐ Take public holidays into account when calculating availability.

[**Target account** | **Target category** | **Percentage**]

- **Booking period** is the period [month, week, day] whose calculated hours are to be aggregated.
- **Contract duration** is a specification of the contractual hours figure [contract duration, min_ contract duration, max_contract duration] that is to serve as the basis for the booking.
- **Days of the week** are the days of the week [Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday] that count as working days.

This rule can be used to ascertain the hours to be worked in a given period.

Example

An employee is contracted to work thirty-six hours a week. Friday 5 May 2017 is a public holiday.

On the first day of [**day**], the [**contract duration**] of an employee in a working week containing [**Monday, Tuesday, Wednesday, Thursday, Friday**] is calculated as follows.

☐ Take public holidays into account when calculating availability.

[**Hours** | **Hours to be worked** | **100**]

Hours	01-05-2017 to 08-05-2017
Hours to be worked	
> 01-05-2017	07:12
> 02-05-2017	07:12
> 03-05-2017	07:12
> 04-05-2017	07:12
Total	28:48

17.7 Multiple activity kinds per period

Stipulates that a value is to be booked, based on the number of hours that an employee is scheduled to spend on activities of certain specified kinds, if the activities in question form part of an uninterrupted period made up of activities of certain specified kinds.

How to use

For each uninterrupted period of the following activity kinds: [**Uninterrupted kinds**]

an entry will be made for the duration of the activity kinds: [**Kinds**]

for the following categories:

[**Target account** | **Target category** | **Percentage**]

- **Uninterrupted kinds** are the kinds [Attendance, Occasional Leave, Special leave, ..., Sick leave, Sick leave (no work)] to which activity types in the uninterrupted period need to belong in order to trigger a booking.
- **Kinds** are the kinds [Attendance, Occasional Leave, Special leave, ..., Sick leave, Sick leave (no work)] of activities that are to be booked.

Example

Shift A, as scheduled for Saturday 8 April 2017, is made up of the following activities:

Activity type	Start	End
Work	08:00	12:00
Break	12:00	12:30
Work	12:30	16:00

The activity type 'Work' belongs to the activity kind 'Work'. The activity type 'Break' belongs to the activity kind 'Break'.

For each uninterrupted period of the following activity kinds: [**Break, Work**]

an entry will be made for the duration of the activity kinds: [**Work**]

for the following categories:

[**Hours** | **Worked hours** | **100**]

Hours	01-01-2017 to 01-01-2018
Worked hours	
> 08-04-2017	07:30
Total	07:30

17.8 Spent time according to cyclical schedule

Stipulates that the average number of hours an employee spent on an activity per week according to his cyclical schedule are booked per day.

How to use

Enter per day the average number of hours an employee spends weekly on the activity types: [**Activity types**] according to the cyclical schedule.

[**Target account** | **Target category** | **Percentage**]

Activity types are the activity types within the cyclical schedule for which an booking is triggered.

The activity types able to select for the variable **Activity types** can be configured in the menu **Organization**.

Example

The rotation plan for the period 01-05-2017 to 29-05-2017 (four weeks) provides for a particular employee to work shift A every weekday (Monday to Friday). Shift A is made up of the following activities:

Activity type	Start	End
Work	09:00	12:30
Break	12:30	13:00
Work	13:00	17:00

Enter per day the average number of hours an employee spends weekly on the activity types: [**Work**] according to the cyclical schedule.

[**Hours** | **Weekly average** | **100**]

The account Hours include the following bookings for the week 01-05-2017 to 08-05-2017.

Hours	01-05-2017 to 08-05-2017
Weekly average	
> 01-05-2017	37.50
> 02-05-2017	37.50
> 03-05-2017	37.50
> 04-05-2017	37.50
> 05-05-2017	37.50
> 06-05-2017	37.50
> 07-05-2017	37.50
Total	262.50

The average number of hours the employee spends weekly on the activity type work according to the cyclical schedule is 5x7.50=37.50.

17.9 Variable entries for an activity class (sal.)

Stipulates that a value is to be booked, based on the number of hours that an employee is scheduled to spend on an activity of a specified kind, if the employee's salary code is as specified.

How to use

Whenever a shift contains elements like [**Kinds**]

, and the employees salary classification is between [**Min salary code**] and [**Max salary code**], entries will be made as indicated below:

[**Target account** | **Target category** | **Percentage**]

Depending on the system configuration settings, the booking date used in the context of this rule is the date on which the shift starts or the date of the hours to which the booking relates.

Example

Shift A is scheduled for Saturday 9 April 2005. The shift is to be worked by an employee on salary code C and is made up of the following activities:

Activity type	Start	End
Work	08:00	12:00
Break	12:00	12:30
Work	12:30	16:00

The activity type 'Work' belongs to the activity class 'Work'. The activity type 'Break' belongs to the activity class 'Break'.

Whenever a shift contains elements like [**Work**]

, and the employees salary classification is between [**A**] and [**F**], entries will be made as indicated below:

[**Hours** | **Worked hours** | **100**]

Hours		01-01-2017 to 01-01-2018
Worked hours		
> 08-04-2017	A	07:30
Total		07:30

17.10 Variable entry for an activity class

Stipulates that a value is to be booked, based on the number of hours that an employee is scheduled to spend on an activity of a specified kind.

How to use

Whenever a shift contains elements like [**Kinds**]

, entries will be made as indicated below:
This rule sums entries for activities within the same shift when the following properties are equal: activity type, cost center, workstation.
For this rule, [**do/do not**] take the transition from daylight savings time into account.

[**Target account** | **Target category** | **Percentage**]

- **Kinds** are the kinds [Amplitude, Availability (no labor), ..., Work (stand by), WTR] to which the shift's activity types need to belong in order to trigger a booking.

Depending on the system configuration settings, the booking date used in the context of this rule is the date on which the shift starts or the date of the hours to which the booking relates.

Example

Shift A, as scheduled for Saturday 8 April 2017, is made up of the following activities:

Activity type	Start	End
Work	08:00	12:00
Break	12:00	12:30
Work	12:30	16:00

The activity type 'Work' belongs to the activity kind 'Work'. The activity type 'Break' belongs to the activity kind 'Break'.

Whenever a shift contains elements like [**Work**]

, entries will be made as indicated below:
This rule sums entries for activities within the same shift when the following properties are equal: activity type, cost center, workstation.
For this rule, [**do not**] take the transition from daylight savings time into account.

[**Hours** | **Worked hours** | **100**]

Hours		01-01-2017 to 01-01-2018
Worked hours		
> 08-04-2017	A	07:30
Total		07:30

17.11 Variable entry for an activity type

Stipulates that a value is to be booked, based on the number of hours that an employee is scheduled to spend on an activity of a specified type.

How to use

If a shift contains an activity identical to the [**Activity type**] for organization unit [**Organization unit**] , or has an activity type that is derived from this, the following calculation is performed:

☐ Add activities with the dame activity type and workplace together.
☐ The entries are rounded to the [**Direction**] [**Significant**] minutes.
☐ In case of illness shifts, look at the shifts before sick leave.
☐ In case of leave-of-absence shifts, look at the shifts before leave-of-absence.

[**Target account** | **Target category** | **Percentage**]

- **Activity type** is the type of the activities for which bookings are to be made.
- **Organization unit** is the organization unit (or department) associated with the activity type.
- **Direction** is the direction [next, previous, nearest] to which the entries should rounded.
- **Significant** is the significant in minutes at which entries are rounded off.

Depending on the system configuration settings, the booking date used in the context of this rule is the date on which the shift starts or the date of the hours to which the booking relates.

Example

Shift A, as scheduled for 8 April 2017, is made up of the following activities:

Activity type	Start	End
Work	23:00	03:00
Break	03:00	03:30
Work	03:30	07:00

The activity type 'Work' is associated with the highest level of the organizational hierarchy ('Organization').

If a shift contains an activity identical to the [**Work**] for organization unit [**Organization**] , or has an activity type that is derived from this, the following calculation is performed:

☒ Add activities with the same activity type and workplace together.
☐ The entries are rounded to the [**next**] [**30**] minutes.
☐ In case of illness shifts, look at the shifts before sick leave.
☐ In case of leave-of-absence shifts, look at the shifts before leave-of-absence.

[**Hours** | **Worked hours** | **100**]

Hours		01-01-2017 to 01-01-2018
Worked hours		
> 08-04-2017	A	01:00
> 09-04-2017	A	06:30
Total		07:30

18 Overtime

The compensation rules in this group can be used to determine the amount of overtime employees made by use of different overtime definitions and to determine the entries that should be made in order to compensate the overtime made by employees.

Important is that within one set of compensation rules it is in ORTEC WS possible to select just one rule to define the variable overtime.

> The compensation rules that can be used to determine an overtime definition all starts with the syntax "Overtime: ...".

18.1 Carry over kilometers

Stipulates that the number of kilometers booked under a specified category of one account is to be carried over to a specified category in another account.

How to use

Entries in kilometers from account [**Account**] in category [**Category**] are to be made as follows:

☐ Transfer bookings that have not been approved.

[**Target account** | **Target category** | **Percentage**]

> This rule makes use of the property 'DISTANCE_TO_WORK', as defined for the relevant employee. Also it is possible to use the 'Pre-shift kilometers' rule or the 'Post-shift kilometers' rule, or to use the 'Carry over and apply an expression' rule incorporating an employee attribute relating to distance to work.

- **Account** is the account to which the desirable category belongs.
- **Category** is the category from which the number of kilometers is to be carried over.
- If **Transfer bookings that have not been approved** is selected, the number of kilometers are carried over to another account before these kilometers have been approved.

  > In default the number of kilometers are carried over to another account after they are approved.

Example

The distance to work (DISTANCE_TO_WORK) defined for an employee is 16.

The account Days includes the following booking:

Days	01-01-2017 to 01-01-2018
Extra days	
> 09-04-2017	1:00
Total	1:00

For this booking, the value of **Kilometer allowance** is 2.

Entries in kilometers from account [**Days**] in category [**Extra days**] are to be made as follows:

☐ Transfer bookings that have not been approved.

[**Kilometers** | **Distance from work** | **100**]

Kilometers	01-01-2017 to 01-01-2017
Distance from work	
> 09-04-2017	32:00
Total	32:00

The value booked to the target account equals DISTANCE_TO_WORK multiplied by **Kilometer allowance**.

18.2 Compensation

Provides for compensation in the form of a correction based on a value recorded under another category.

How to use

Within a period of [**Number**] weeks a positive balance on account [**Account**], category [**Category**] will be used as compensation for category [**Target category**].

- **Number** is the number of weeks within which compensation may take place.
- **Account** is the account to which source category belongs.
- **Category** is the category whose calculated balance is available for compensation.
- **Target category** is the category to be corrected.

Compensation is booked only if the value to be compensated has been generated by an overtime rule.
It is not necessary to specify a target account and target category in the context of this rule.
This rule automatically generates a correction, which is recorded in the database.

Example

The following overtime definition is included in the rule set:

Overtime is to be calculated on a daily basis by determining the difference between the hours worked according to the master schedule and the hours according to the published schedule.

☐ No non-social hours allowance is granted to overtime hours.

☐ In case of an on call week - a week with at least 5 on call shifts - the difference in hours worked over the whole week is considered. A non-social hour allowance is granted to all hours worked.

The following rule is defined for booking overtime:

Overtime hours will be paid out as indicated below.

☐ A threshold value applies in the context of this rule.
☒ Only positive differences (worked more) are included.
☐ Only negative differences (worked less) are included.
☒ Include overtime for outlet.
☒ Include overtime for swapped shifts.
☒ Include overtime for extra shifts
☐ Overtime reward is a fixed value.

[**Overtime** | **Overtime hours** | **100**]

The published schedule for the period 30-01-2017 to 27-02-2017 (four weeks) provides for a particular employee to work shift A every weekday (Monday to Friday). Shift A is made up of the following activities:

Activity type	Start	End
Work	08:00	16:00

In practice, however, the employee works until 17:00 hours on 25 February.

The 'Compensation hours' account includes the following booking:

Compensation hours	01-01-2017 to 01-01-2018
To compensate	
25-02-2017 A	04:00
Total	04:00

Within a period of [**1**] weeks a positive balance on account [**Compensation hours**], category [**To compensate**] will be used as compensation for category [**Overtime hours**].

Overtime	01-01-2017 to 01-01-2018
Overtime hours	
> 25-02-2017 A	01:00
> 25-02-2017 Correction	-01:00
Total	00:00

18.3 Continuous overtime

Stipulates that the number of hours of continuous overtime worked (within a specified range) is to be booked.

How to use

An employee who works between [**Minimum duration**] and [**Maximum duration**] hours, immediately following regular work, will be paid out as follows:

[**Target account** | **Target category** | **Percentage**]

- **Minimum duration** is the minimum number of hours' continuous overtime that triggers a booking.
- **Maximum duration** is the maximum number of hours' continuous overtime that triggers a booking.

Continuous overtime is increase in working time brought about by extension of a shift, i.e. the difference between the working time in the realized shift and the working time in the shift as originally scheduled.

Example

The published schedule for the period 30-01-2017 to 27-02-2017 (four weeks) provides for a particular employee to work shift A every weekday (Monday to Friday). Shift A is made up of the following activities:

Activity type	Start	End
Work	08:00	16:00

In practice, however, the employee works until 17:00 hours on 25 February.

An employee who works between [**0:00**] and [**2:00**] hours, immediately following regular work, will be paid out as follows:

[**Overtime** | **Overtime hours** | **100**]

Overtime	01-01-2017 to 01-01-2018
Overtime hours	
25-02-2017 A	01:00
Total	01:00

18.4 Continuous overtime (comp time)

Stipulates that the number of hours of continuous overtime worked (within a specified range) is to be booked, if the employee's salary code and time-for-time choice are as specified.

How to use

An employee who is classified in salary code [**Min salary code**] to [**Max salary code**] and who works overtime for [**Minimum duration**] and [**Maximum duration**] hours immediately following regular work and who has

☐ has
☐ has not chosen for a Time For Time arrangement, will be paid out as follows:

[**Target account** | **Target category** | **Percentage**]

- **Min salary code** is the lowest salary code that an employee may have for a booking to be triggered.

- **Max salary code** is the highest salary code that an employee may have for a booking to be triggered.
- **Minimum duration** is the minimum number of hours' contiguous overtime that triggers a booking.
- **Maximum duration** is the maximum number of hours' contiguous overtime that triggers a booking.

If the checkmark **has** is selected the booking is made when the employee has chosen a Time for Time arrangement.
Continuous overtime is increase in working time brought about by extension of a shift, i.e. the difference between the working time in the realized shift and the working time in the shift as originally scheduled.

Example

The published schedule for the period 30-01-2017 to 27-02-2017 (four weeks) provides for a particular employee on salary code C who has decided against the time-for-time option to work shift A every weekday (Monday to Friday). Shift A is made up of the following activities:

Activity type	Start	End
Work	08:00	16:00

In practice, however, the employee works until 17:00 hours on 25 February.

An employee who is classified in salary code [**A**] to [**F**] and who works overtime for [**0:00**] and [**2:00**] hours immediately following regular work and who has

☐ indeed
☒ not chosen for a Time For Time arrangement, will be paid out as follows:

[**Overtime** | **Overtime payout** | **100**]

Overtime	01-01-2017 to 01-01-2018
Overtime payout	
25-02-2017 A	01:00
Total	01:00

18.5 Continuous overtime (sal. code)

Stipulates that the number of hours of contiguous overtime worked (within a specified range) is to be booked, if the employee's salary code is as specified.

How to use

An employee, whose salary code is between [**Min salary code**] and [**Max salary code**] and who works overtime between [**Minimum duration**] and [**Maximum duration**] hours, immediately following regular work, will be paid out as follows:

[**Target account** | **Target category** | **Percentage**]

- **Min salary code** is the lowest salary code that an employee may have for a booking to be triggered.
- **Max salary code** is the highest salary code that an employee may have for a booking to be triggered.

- **Minimum duration** is the minimum number of hours' contiguous overtime that triggers a booking.
- **Maximum duration** is the maximum number of hours' contiguous overtime that triggers a booking.

Continuous overtime is increase in working time brought about by extension of a shift, i.e. the difference between the working time in the realized shift and the working time in the shift as originally scheduled.

Example

The published schedule for the period 30-01-2017 to 27-02-2017 (four weeks) provides for a particular employee on salary code C to work shift A every weekday (Monday to Friday). Shift A is made up of the following activities:

Activity type	Start	End
Work	08:00	16:00

In practice, however, the employee works until 17:00 hours on 25 February.

An employee, whose salary code is between [**A**] and [**F**] and who works overtime between [**0:00**] and [**2:00**] hours, immediately following regular work, will be paid out as follows:

[**Overtime** | **Overtime hours** | **100**]

Overtime	01-01-2017 to 01-01-2018
Overtime hours	
> 25-02-2017 A	01:00
Total	01:00

18.6 Do not work as in cyclical schedule

Stipulates that the number of working hours are to be booked if an employee works at a time when, according to the cyclical schedule, he/she should not be working.

How to use

Whenever an employee has to work on a day that would be his/her day-off according to the cyclical team schedule, he or she will be credited as indicated below:

[**Target account** | **Target category** | **%**]

Example

In the cyclical schedule, a particular employee is assigned a 'Day off' shift on Saturday 6 May 2005. After publication of the schedule, however, the employee is assigned shift A on that day. Shift A is made up of the following activities:

Activity type	Start	End
Work	08:00	12:00
Break	12:00	12:30
Work	12:30	16:00

Whenever an employee has to work on a day that would be his/her day-off according to the cyclical team schedule, he or she will be credited as indicated below:

[**Hours** | **Extra hours** | **100**]

Hours	01-01-2017 to 01-01-2018
Extra hours	
> 06-05-2017 A	07:30
Total	07:30

18.7 Enter travel time

Stipulates that travel expense bookings are to be carried over to another category.

How to use

For entries on account [**Account**], category [**Category**] any travel expenses can be credited on:

☐ Transfer bookings that have not been approved.

[**Target account** | **Target category** | **Percentage**]

- **Account** is the account to which the category you wish to carry over belongs.
- **Category** is the category from which the kilometers are to be carried over.

This rule results in a booking being carried over from the source category if it has the property 'travel expenses'.
Where possible, it is preferable to use the 'Pre-shift kilometers' rule or the 'Post-shift kilometers' rule, or to use the 'Carry over and apply an expression' rule incorporating an employee attribute relating to distance to work.

If the checkmark **Transfer bookings that have not been approved** is selected, bookings are also transferred for schedules that have a status lower than approved.

Example

The ‘Days’ account includes the following booking:

Days	01-01-2017 to 01-01-2018
Extra days	
> 09-04-2017	1:00
Total	1:00

The property ‘travel expenses’ has been assigned to this booking.

For entries on account [**Days**], category [**Extra days**] any travel expenses can be credited on:

☐ Transfer bookings that have not been approved.

[**Kilometers** | **Days allowance** | **100**]

Kilometers	01-01-2017 to 01-01-2018
Days allowance	
> 09-04-2017	1:00
Total	1:00

18.8 From another account

Stipulates that the balance under a particular category is to be carried over for each period of a specified number of weeks.

How to use

The balance of account [**Account**], category [**Category**] will entered after every [**Number**] weeks to the accounts below:

[**Target account** | **Target category** | **Percentage**]

- **Account** is the account from which the balance is to be carried over to the target category.
- **Category** is the category of the account from which the balance is to be carried over to the target category.
- **Number** is the length (in weeks) of the period for which the balance is to be carried over.

The booking date is the last day of the period for which the balance is to be carried over.

Example

The 'Hours' account includes the following bookings:

Hours	01-01-2017 to 01-02-2017
Hours worked	
> 16-01-2017	08:00
> 26-01-2017	07:30
Total	15:30

26-03-2017 is the thirteenth Sunday of 2017.

The balance of account [**Hours**], category [**Hours worked**] will entered after every [**13**] weeks to the accounts below:

[**Compensation** | **Hours** | **100**]

Compensation	01-01-2017 to 01-01-2018
Hours	
> 26-03-2017	15:30
Total	15:30

18.9 Monthly transfer-booking

Stipulates that the balance under a particular category is to be carried over to a category in another account once a month, given the maximum value of an entry.

How to use

The balance of account [**Account**] with category [**Category**] will be entered monthly on the accounts below. An entry has a maximum of [**Value**] hour(s).

[**Target account** | **Target category** | **Percentage**]

- **Account** is the account from which the balance is to be carried over to the target category.
- **Category** is the category in the source account from which the balance is to be carried over to the target category.
- **Value** is the maximum value of the entry to be booked over.

Example

Hours	01-01-2017 to 01-02-2017
Hours worked	
> 16-01-2017	08:00
> 26-01-2017	07:30
Total	15:30

The balance of account [**Hours**] with category [**Hours worked**] will be entered monthly on the accounts below. An entry has a maximum of [**100**] hour(s).

[**Compensation** | **Hours** | **100**]

Compensation	01-02-2017 to 01-03-2018
Hours	
> 01-02-2017	15:30
Total	15:30

18.10 Non-contiguous overtime

Stipulates that the number of hours of non-contiguous overtime worked is to be booked, subject to a certain minimum.

How to use

Hours worked in overtime that do not immediately follow regular hours are credited using a minimum value of [**Minimum booking**] hours the overtime will be paid out as indicated below:

[**Target account** | **Target category** | **Percentage**]

- **Minimum booking** is the minimum number of hours of non-contiguous overtime to be booked.

Non-contiguous overtime is working time which is additional to that included in the shift as originally scheduled, but which does not overlap with the original shift and is immediately preceded or followed by an activity that does not consist of working time.

Example

Shift A, as scheduled for 25 February 2017, is made up of the following activities:

Activity type	Start	End
Work	08:00	16:00

In practice, shift A on 25 February is made up of the following activities:

Activity type	Start	End
Work	08:00	16:00
Free time	16:00	17:00
Work	17:00	17:45

Hours worked in overtime that do not immediately follow regular hours are credited using a minimum value of [**01:00**] hours the overtime will be paid out as indicated below:

[**Overtime** | **Overtime hours** | **100**]

Overtime	01-01-2017 to 01-01-2018
Overtime hours	
> 25-02-2017 A	01:00
Total	01:00

18.11 Non-contiguous overtime (sal. group)

Stipulates that the number of hours of non-contiguous overtime worked is to be booked, subject to a certain minimum and if the employee's salary code is as specified.

How to use

Hours worked over in overtime that do not immediately follow regular hours are credited using a minimum value of [**Minimum booking**] hours. This rule will be applied to employees with salary code between [**Min salary code**] and [**Max salary code**]. The overtime will be paid out as indicated below:

[**Target account** | **Target category** | **Percentage**]

- **Minimum booking** is the minimum number of hours of non-contiguous overtime to be booked.
- **Min salary code** is the lowest salary code that an employee may have for a booking to be triggered.
- **Max salary code** is the highest salary code that an employee may have for a booking to be triggered.

Non-contiguous overtime is working time which is additional to that included in the shift as originally scheduled, but which does not overlap with the original shift and is immediately preceded or followed by an activity that does not consist of working time.

Example

The published schedule for 25 February 2017 provides for a particular employee on salary code C to work shift A, which is made up of the following activities:

Activity type	Start	End
Work	08:00	16:00

In practice, shift A on 25 February is made up of the following activities:

Activity type	Start	End
Work	08:00	16:00
Free time	16:00	17:00
Work	17:00	17:45

Hours worked over in overtime that do not immediately follow regular hours are credited using a minimum value of [**01:00**] hours this rule will be applied to employees with salary code between [**A**] and [**F**] the overtime will be paid out as indicated below:

[**Overtime** | **Overtime hours** | **100**]

Overtime	01-01-2017 to 01-01-2018
Overtime hours	
> 25-02-2017 A	01:00
Total	01:00

18.12 Non-contiguous overtime (tft)

Stipulates that the number of hours of non-contiguous overtime worked is to be booked, subject to a certain minimum and if the employee's salary code and time-for-time choice are as specified.

How to use

Hours worked in overtime that do not immediately follow regular hours are credited using a minimum value of [**Minimum booking**] hours. This rule will be applied to employees with salary code between [**Min salary code**] and [**Max salary code**] who

☐ have

☐ have not

chosen for a Time For Time arrangement. The overtime will be paid out as indicated below:

[**Target account** | **Target category** | **Percentage**]

- **Minimum booking** is the minimum number of hours of non-contiguous overtime to be booked.
- **Min salary code** is the lowest salary code that an employee may have for a booking to be triggered.
- **Max salary code** is the highest salary code that an employee may have for a booking to be triggered.

Non-contiguous overtime is working time which is additional to that included in the shift as originally scheduled, but which does not overlap with the original shift and is immediately preceded or followed by an activity that does not consist of working time.

With Time For Time arrangement it is mean that overtime is swapped with the availability of extra free time. Another option than Time For Time arrangement could be to pay out the overtime.

Example

The published schedule for 25 February 2017 provides for a particular employee on salary code C who has decided against the time-for-time option to work shift A, which is made up of the following activities:

Activity type	Start	End
Work	08:00	16:00

In practice, shift A on 25 February is made up of the following activities:

Activity type	Start	End
Work	08:00	16:00
Free time	16:00	17:00
Work	17:00	17:45

Hours worked in overtime that do not immediately follow regular hours are credited using a minimum value of [**01:00**] hours. This rule will be applied to employees with salary code between [**A**] and [**F**] who

☐ have

☒ have not

chosen for a Time For Time arrangement the overtime will be paid out as indicated below:

[**Overtime** | **Overtime payout** | **100**]

Overtime	01-01-2017 to 01-01-2018
Overtime payout	
> 25-02-2017 A	01:00
Total	01:00

18.13 Overtime

Stipulates that overtime (as defined in the rule set) is to be booked.

How to use

Overtime hours will be paid out as indicated below.

☐A threshold value is applied for this rule.
☐Only positive differences (worked more) are included.
☐Only negative differences (worked less) are included.
☐Include overtime for outlet.
☐Include overtime for swapped shifts.
☐Include overtime for extra shifts.
☐Overtime reward is a fixed value.

[**Target account** | **Target category** | **Percentage**]

This rule makes use of an overtime definition. The rule results in the booking of all hours that count as working time and meet the applicable overtime definition.

In case the checkmark **Overtime reward is a fixed value** is selected the variable **Percentage**, next to the variable **Target category**, is replaced by [**Value**].

Example

The following overtime definition is included in the rule set (Overtime - hours above norm per week):

Overtime is defined as the total performed working hours in a roster minus the average number of hours of [**40:00**] hours per week.

☐ No non-social hour bonus is granted to overtime hours.

In the scheduling period 30-01-2017 to 27-02-2017 (four weeks), a particular employee works shift A every weekday (Monday to Friday). Shift A is made up of the following activities:

Activity type	Start	End
Work	08:00	16:30

Overtime hours will be paid out as indicated below.

☐A threshold value is applied for this rule.
☒ Only positive differences (worked more) are included.
☐Only negative differences (worked less) are included.
☒ Include overtime for outlet.
☒ Include overtime for swapped shifts.
☒ Include overtime for extra shifts.
☐Overtime reward is a fixed value.

[**Overtime** | **Overtime hours** | **100**]

Overtime	01-01-2017 to 01-01-2018
Overtime hours	
> 23-02-2017 A	01:30
> 24-02-2017 A	08:30
Total	10:00

Example

Now the checkmark **Overtime reward is a fixed value** is selected.

Overtime hours will be paid out as indicated below.

☐A threshold value is applied for this rule.
☒ Only positive differences (worked more) are included.
☐Only negative differences (worked less) are included.
☒ Include overtime for outlet.
☒ Include overtime for swapped shifts.
☒ Include overtime for extra shifts.
☒ Overtime reward is a fixed value.

[**Overtime** | **Overtime hours** | **12**]

Overtime	01-01-2017 to 01-01-2018
Overtime hours	
> 23-02-2017 A	12:00
> 24-02-2017 A	12:00
Total	24:00

18.14 Overtime (tft)

Stipulates that overtime (as defined in the rule set) is to be booked, if the employee's salary code and time-for-time choice are as specified and if a reassignment history criterion is met.

How to use

Employees who work overtime and who have a salary group which lies between [**Min salary code**] and [**Max salary code**] and who
☐have
☐not have chosen for a 'Time For Time' [**Reassignment history**] arrangement will be paid out as indicated below.

☐A threshold value is applied for this rule.
☐Only positive differences (worked more) are included.
☐Only negative differences (worked less) are included.
☐Include overtime for outlet.
☐Include overtime for swapped shifts.
☐Include overtime for extra shifts.
☐Overtime reward is a fixed value.

[**Target account** | **Target category** | **Percentage**]

- **Min salary code** is the lowest salary code that an employee may have for a booking to be triggered.
- **Max salary code** is the highest salary code that an employee may have for a booking to be triggered.
- **Reassignment history** is a phrase describing the applicability of the rule by reference to the shift reassignment history [always, only in case of an exchange, not in case of an exchange].

This rule makes use of an overtime definition. The rule results in the booking of all hours that count as working time and meet the applicable overtime definition.

In case the checkmark **Overtime reward is a fixed value** is selected the variable **Percentage**, next to the variable **Target category**, is replaced by [**Value**].

Example

The following overtime definition is included in the rule set (Overtime - hours above norm per week):

Overtime is defined as the total performed working hours in a roster minus the average number of hours of [**40:00**] hours per week.

☐ No non-social hour bonus is granted to overtime hours.

In the scheduling period 30-01-2017 to 27-02-2017 (four weeks), a particular employee works shift A every weekday (Monday to Friday). Shift A is made up of the following activities:

Activity type	Start	End
Work	08:00	16:30

The employee is on salary code C and has not chosen the time-for-time option.

Employees who work overtime and who have a salary group which lies between [**A**] and [**F**] and who
☐Indeed
☒ not have chosen for a 'Time For Time' [**always**] arrangement will be paid out as indicated below.

☐A threshold value is applied for this rule.
☒ Only positive differences (worked more) are included.
☐Only negative differences (worked less) are included.
☒ Include overtime for outlet.
☒ Include overtime for swapped shifts.
☒ Include overtime for extra shifts.
☐Overtime reward is a fixed value.

[**Overtime** | **Overtime hours** | **100**]

Overtime	01-01-2017 to 01-01-2018
Overtime hours	
> 23-02-2017 A	01:30
> 24-02-2017 A	08:30
Total	10:00

18.15 Overtime adjusted with average shift percentage

Stipulates that overtime (as defined in the rule set) is to be booked, if the employee's salary code and time-for-time choice are as specified, if a reassignment history criterion is met and if the timing of the overtime (both within the week and relative to the original shift) is as specified.

How to use

This rule is for overtime between [**Start time**] and [**End time**] hours on a [**Days of the week**] Employees will be paid out over a period from [**Duration**] from the start of a shift or the same

duration after the end of a shift. The employees salary group lies between [**Min salary code**] and [**Max salary code**]. The employee has
☐indeed
☐not
chosen for a 'Time For Time' arrangement.
The overtime hours will be paid out, according to: [**Reassignment history**]

☐A threshold value is applied for this rule.
☐Only positive differences (worked more) are included.
☐Only negative differences (worked less) are included.
☐Include overtime for outlet.
☐Include overtime for swapped shifts.
☐Include overtime for extra shifts.
☐Overtime reward is a fixed value.

[**Target account** | **Target category** | **%**]

- **Min salary code** is the lowest salary code that an employee may have for a booking to be triggered.
- **Max salary code** is the highest salary code that an employee may have for a booking to be triggered.
- **Days of the week** are the days of the week [Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday, public holiday] in respect of which hours are to be booked.
- **Duration** is the length of the period of time before and after a shift, during which any overtime worked is not to be booked.
- **Min salary code** is the lowest salary code that an employee may have for a booking to be triggered.
- **Max salary code** is the highest salary code that an employee may have for a booking to be triggered.
- **Reassignment history** is a phrase describing the applicability of the rule by reference to the shift reassignment history [always, only in case of an exchange, not in case of an exchange].

This rule makes use of an overtime definition.
The rule results in the booking of all hours that count as working time and meet the applicable overtime definition.
Where possible, it is preferable to use the ‘Overtime per time interval with threshold’ rule.

In case the checkmark **Overtime reward is a fixed value** is selected the variable **%**, next to the variable **Target category**, is replaced by [**Value**].

Example

The following overtime definition is included in the rule set (Overtime: fixed in regard to publication).

Overtime is calculated on a daily basis by determining the difference between the hours worked according to the master schedule and the hours according to the published schedule.
☐ No non-social hour allowance is granted to overtime hours

On call weeks:
☐ In case of an on call week - a week with at least [**5**] on call shifts - the difference in hours worked over the whole week is considered. A non-social hour allowance is granted to all hours worked.

The published schedule for the period 31-01-2005 to 28-02-2005 (four weeks) provides for a particular employee to work shift A every weekday (Monday to Friday). Shift A is made up of the following activities:

Activity type	Start	End
Work	08:00	16:00

In practice, however, the employee works until 17:00 hours on 25 February. The employee is on salary code C and has not chosen the time-for-time option.

This rule is for overtime between [**00:00**] and [**00:00**] hours on a [**Monday, Tuesday, Wednesday, Thursday, Friday**]
Employees will be paid out over a period from [**00:30**] from the start of a shift or the same duration after the end of a shift. The employees salary group lies between [**A**] and [**F**]. The employee has
☐ indeed
☒ not
chosen for a 'Time For Time' arrangement.
The overtime hours will be paid out, according to: [**always**]

☐ A threshold value is applied for this rule.
☒ Only positive differences (worked more) are included.
☐ Only negative differences (worked less) are included.
☒ Include overtime for outlet.
☒ Include overtime for swapped shifts.
☒ Include overtime for extra shifts.
☐ Overtime reward is a fixed value.

[**Overtime** | **Overtime hours** | **100**]

Overtime	01-01-2017 to 01-01-2018
Overtime hours	
> 25-02-2017 A	00:30
Total	00:30

18.16 Overtime per salary code

Stipulates that overtime (as defined in the rule set) is to be booked, if the employee's salary code is as specified.

How to use

For employees in a salary group between [**Min salary code**] and [**Max salary code**], overtime hours will be paid as indicated below.

☐A threshold value is applied for this rule.
☐Only positive differences (worked more) are included.
☐Only negative differences (worked less) are included.
☐Include overtime for outlet.
☐Include overtime for swapped shifts.
☐Include overtime for extra shifts.
☐Overtime reward is a fixed value.

[**Target account** | **Target category** | **Percentage**]

- **Min salary code** is the lowest salary code that an employee may have for a booking to be triggered.
- **Max salary code** is the highest salary code that an employee may have for a booking to be triggered.

This rule makes use of an overtime definition.
The rule results in the booking of all hours that count as working time and meet the applicable overtime definition.

Example

The following overtime definition is included in the rule set (Overtime - hours above norm per week):

Overtime is defined as the total performed working hours in a roster minus the average number of hours of [**40:00**] hours per week.

☐ No non-social hour bonus is granted to overtime hours.

In the scheduling period 30-01-2017 to 27-02-2017 (four weeks), a particular employee works shift A every weekday (Monday to Friday). Shift A is made up of the following activities:

Activity type	Start	End
Work	08:00	16:30

The employee is on salary code C.

For employees in a salary group between [**A**] and [**F**], overtime hours will be paid as indicated below.

☐A threshold value is applied for this rule.
☒ Only positive differences (worked more) are included.
☐Only negative differences (worked less) are included.
☒ Include overtime for outlet.
☒ Include overtime for swapped shifts.
☒ Include overtime for extra shifts.
☐Overtime reward is a fixed value.

[**Overtime** | **Overtime hours** | **100**]

Overtime		01-01-2017 to 01-01-2018
Overtime hours		
> 23-02-2017	A	01:30
> 24-02-2017	A	08:30
Total		10:00

18.17 Overtime per salary code (tft)

Stipulates that overtime (as defined in the rule set) is to be booked, if the employee's salary code and time-for-time choice are as specified.

How to use

For employees, with salary group between [**Min salary code**] and [**Max salary code**], who have
☐ have
☐ have not
chosen for a 'Time For Time' arrangement, overtime hours will be paid as indicated below.

☐A threshold value is applied for this rule.
☐Only positive differences (worked more) are included.
☐Only negative differences (worked less) are included.
☐Include overtime for outlet.
☐Include overtime for swapped shifts.
☐Include overtime for extra shifts.
☐Overtime reward is a fixed value.

[**Target account** | **Target category** | **Percentage**]

- **Min salary code** is the lowest salary code that an employee may have for a booking to be triggered.
- **Max salary code** is the highest salary code that an employee may have for a booking to be triggered.

This rule makes use of an overtime definition.
The rule results in the booking of all hours that count as working time and meet the applicable overtime definition.

Example

The following overtime definition is included in the rule set (Overtime - hours above norm per week):

Overtime is defined as the total performed working hours in a roster minus the average number of hours of [**40:00**] hours per week.

☐ No non-social hour bonus is granted to overtime hours.

In the scheduling period 30-01-2017 to 27-02-2017 (four weeks), a particular employee works shift A every weekday (Monday to Friday). Shift A is made up of the following activities:

Activity type	Start	End
Work	08:00	16:30

The employee is on salary code C and has not chosen the time-for-time option.

For employees, with salary group between [**A**] and [**F**], who have
☐ have
☒ have not
chosen for a 'Time For Time' arrangement, overtime hours will be paid as indicated below.

☐A threshold value is applied for this rule.
☒ Only positive differences (worked more) are included.
☐Only negative differences (worked less) are included.
☒ Include overtime for outlet.
☒ Include overtime for swapped shifts.
☒ Include overtime for extra shifts.
☐Overtime reward is a fixed value.

[**Overtime** | **Overtime hours** | **100**]

Overtime	01-01-2017 to 01-01-2018
Overtime hours	
> 23-02-2017 A	01:30
> 24-02-2017 A	08:30
Total	10:00

18.18 Overtime per time period

Stipulates that overtime (as defined in the rule set) is to be booked, if the employee's salary code and time-for-time choice are as specified, if a reassignment history criterion is met and if the timing of the overtime within the week is as specified.

How to use

Employees who work overtime between [**Start time**] and [**End time**] hours on a [**Days of the week**] will [**Reassignment history**] be paid as described below. This rule applies to employees whose salary group lies between [**Min salary code**] and [**Max salary code**] and who
☐ have
☐ have not
chosen for the time for time arrangement.
This rule only applies to activities with [**Type**] [**Kinds**]

☐This rule applies [**only/does not**] to the first [**Duration**] hours of overtime.
☐This rule applies [**only/does not**] if the overtime starts before [**Time**].

☐A threshold value is applied for this rule.
☐Only positive differences (worked more) are included.
☐Only negative differences (worked less) are included.
☐Include overtime for outlet.
☐Include overtime for swapped shifts.
☐Include overtime for extra shifts.
☐Overtime reward is a fixed value.
This rule sums entries for consecutive activities within the same shift when the following properties are equal: activity type, cost center, workstation.

[**Target account** | **Target category** | **Percentage**]

- **Start time** is the start of the time interval within which work needs to be scheduled in order to trigger a booking.
- **End time** is the end of the time interval within which work needs to be scheduled in order to trigger a booking.
- **Days of the week** are the days of the week [Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday, public holiday] in respect of which hours are to be booked.
- **Reassignment history** is a phrase describing the applicability of the rule by reference to the shift reassignment history [always, only in case of an exchange, not in case of an exchange].
- **Min salary code** is the lowest salary code that an employee may have for a booking to be triggered.
- **Max salary code** is the highest salary code that an employee may have for a booking to be triggered.
- **Type** is the type [class, activity types, main activity type] to which the kinds you wish to choose from belongs.
- **Kinds** are the kinds [Amplitude, Availability (no labor), Break, …, Overtime, Overtime (for free time)] to which the shift's activity types need to belong in order to trigger a booking.
- **Duration** depending on the configuration duration is the minimum time an employee work need to work over before an booking is triggered or the maximum time that is booked.
- **Time** is a clocking time between 00:00 and 23:59; depending on the configuration only the overtime that is made before (or after) Time is booked on the target account.

This rule makes use of an overtime definition. The rule results in the booking of all hours that count as working time and meet the applicable overtime definition.
The part **This rule only applies to activities with [Type] [Kinds]**is only visible when the overtime definition is **Overtime - based on account balance**.

Example

The following overtime definition is included in the rule set (Overtime - fixed in regard to publication):

Overtime is calculated on a daily basis by determining the difference between the hours worked according to the master schedule and the hours according to the published schedule.
☐ No non-social hour allowance is granted to overtime hours

On call weeks:
☐ In case of an on call week - a week with at least [**5**] on call shifts - the difference in hours worked over the whole week is considered. A non-social hour allowance is granted to all hours worked.

The published schedule for the period 30-01-2017 to 27-02-2017 (four weeks) provides for a particular employee to work shift A every weekday (Monday to Friday). Shift A is made up of the following activities:

Activity type	Start	End
Work	08:00	16:00

In practice, however, the employee works until 17:00 hours on 25 February.

The employee is on salary code C and has not chosen the time-for-time option.

Employees who work overtime between [**00:00**] and [**00:00**] hours on a [**Monday, Tuesday, Wednesday, Thursday, Friday**] will [**always**] be paid as described below. This rule applies to employees whose salary group lies between [**A**] and [**F**] and who
☐ have
☒ have not
chosen for the time for time arrangement.

☒This rule applies [**does not**] to the first [**00:45**] hours of overtime.
☐This rule applies [**only**] if the overtime starts before [**00:00**].

☐A threshold value is applied for this rule.
☐Only positive differences (worked more) are included.
☐Only negative differences (worked less) are included.
☒Include overtime for outlet.
☒Include overtime for swapped shifts.
☒Include overtime for extra shifts.
☐Overtime reward is a fixed value.
This rule sums entries for consecutive activities within the same shift when the following properties are equal: activity type, cost center, workstation.

[**Overtime** | **Overtime hours** | **100**]

Overtime	01-01-2017 to 01-01-2018
Overtime hours	
> 25-02-2017 A	00:15
Total	00:15

18.19 Overtime per time period with threshold

Stipulates that overtime (as defined in the rule set) is to be booked, if the employee's salary code and time-for-time choice are as specified, if a reassignment history criterion is met and if the timing of the overtime (both within the week and relative to the original shift) is as specified.

How to use

In case of overtime, between [**Start time**] and [**End time**] on a [**Days of the week**] for hours beyond a period between [**Duration**] hours before a shift starts and the same period after a shift ends will [**Reassignment history**] be paid out as described below. This rule applies to employees whose salary group lies between [**Min salary code**] and [**Max salary code**] and who
☐ have
☐ have not chosen for the time for time arrangement.
This rule only applies to activities with [**Type**] [**Kinds**]

☐A threshold value is applied for this rule.
☐Only positive differences (worked more) are included.
☐Only negative differences (worked less) are included.
☐Include overtime for outlet.
☐Include overtime for swapped shifts.
☐Include overtime for extra shifts.
☐Overtime reward is a fixed value.
This rule sums entries for consecutive activities within the same shift when the following properties are equal: activity type, cost center, workstation.

[**Target account** | **Target category** | **Percentage**]

- **Start time** is the start of the time interval within which work needs to be scheduled in order to trigger a booking.
- **End time** is the end of the time interval within which work needs to be scheduled in order to trigger a booking.
- **Duration** is the threshold value that is set on the period of overtime. The duration time is not booked as overtime, overtime more than the threshold is booked.
- **Days of the week** are the days of the week [Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday, public holiday] in respect of which hours are to be booked.
- **Reassignment history** is a phrase describing the applicability of the rule by reference to the shift reassignment history [always, only in case of an exchange, not in case of an exchange].
- **Min salary code** is the lowest salary code that an employee may have for a booking to be triggered.
- **Max salary code** is the highest salary code that an employee may have for a booking to be triggered.
- **Type** is the type [class, activity types, main activity type] to which the kinds you wish to choose from belongs.
- **Kinds** are the kinds [Amplitude, Availability (no labor), Break, …, Overtime, Overtime (for free time)] to which the shift's activity types need to belong in order to trigger a booking.

This rule makes use of an overtime definition.
The rule results in the booking of all hours that count as working time and meet the applicable overtime definition.
The part **This rule only applies to activities with [Type] [Kinds]**is only visible when the overtime definition is **Overtime - based on account balance**.

Example

The following overtime definition is included in the rule set (Overtime - fixed in regard to publication):

Overtime is calculated on a daily basis by determining the difference between the hours worked according to the master schedule and the hours according to the published schedule.
☐ No non-social hour allowance is granted to overtime hours

On call weeks:
☐ In case of an on call week - a week with at least [**5**] on call shifts - the difference in hours worked over the whole week is considered. A non-social hour allowance is granted to all hours worked.

The published schedule for the period 30-01-2017 to 27-02-2017 (four weeks) provides for a particular employee to work shift A every weekday (Monday to Friday). Shift A is made up of the following activities:

Activity type	Start	End
Work	08:00	16:00

In practice, however, the employee works until 17:00 hours on 25 February.

The employee is on salary code C and has not chosen the time-for-time option.

Employees who work overtime between [**00:00**] and [**00:00**] hours on a [**Monday, Tuesday, Wednesday, Thursday, Friday**] for hours beyond a period between [**0:15**] hours before a shift starts and the same period after a shift ends will [**always**] be paid as described below. This rule applies to employees whose salary group lies between [**A**] and [**F**] and who
☐ have
☒ have not chosen for the time for time arrangement.

☐This rule applies [**only**] to the first [**00:00**] hours of overtime.
☐This rule applies [**only**] if the overtime starts before [**00:00**].

☐A threshold value is applied for this rule.
☐Only positive differences (worked more) are included.
☐Only negative differences (worked less) are included.
☒Include overtime for outlet.
☒Include overtime for swapped shifts.
☒Include overtime for extra shifts.
☐Overtime reward is a fixed value.
This rule sums entries for consecutive activities within the same shift when the following properties are equal: activity type, cost center, workstation.

[**Overtime** | **Overtime hours** | **100**]

Overtime	01-01-2017 to 01-01-2018
Overtime hours	
> 25-02-2017 A	00:45
Total	00:45

18.20 Overtime: based on account balance

Defines the definition of overtime making use of the account balance per day and week for a pre-selected account and category.

How to use

Overtime is calculated based on the balance on [**Account**], category [**Category**]. To determine the exact times of overtime the realized schedule will be used.

Step 1: If the account balance per day is more than [**Day maximum**] hours, any additional hours are overtime.
Determining the exact times of overtime:
- The hours at the end of the realized shifts.

Step 2: If the account balance (minus the overtime resulting from step 1) per week (starting on [**Start day**]) is more than [**Week maximum**] hours are overtime.
Determining the exact time of overtime:
- First the hours on Sunday, next on Saturday, next on Friday, Thursday, Wednesday, Tuesday, Monday.
- The hours at the end of the realized shifts.

- Shifts are recorded completely on the day they start.
- Hours can only be marked as overtime once.
- Hours in between the begin- and end times of work shifts according to the cyclical schedule cannot be marked as overtime.
- Only activities with the following activity kind(s) can be marked as overtime: [**Kinds**]

☐No unsocial hours allowance is calculated for overtime.

Remark: it is not possible to determine the reason for overtime (reassignment, extra shift, outlet) when using this overtime definition.

- **Account** is the account to which source category belongs.
- **Category** is the category whose calculated balance is used to define the overtime definition.
- **Day maximum** is the maximum number of hours of the account its balance in one day, any additional hours more than the day maximum are seen as overtime.
- **Start day** is the first day of the week for which the week balance is calculated [Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday].
- **Week maximum** is the maximum number of hours of the account its balance in one week, any additional hours more than the week maximum are seen as overtime.
- **Kinds** are the kinds [Amplitude, Availability (no labor),..., Work (stand by), WTR] to which the shift's activity types need to belong in order to be marked for overtime.

This rule is a definition. No more than one definition of overtime may be included in a rule set.
A definition does not need to specify a target account or target category.

Example

The following rule is defined for booking overtime:

Overtime hours will be paid out as indicated below.

☐A threshold value is applied for this rule.
☒ Only positive differences (worked more) are included.
☐Only negative differences (worked less) are included.
☒ Include overtime for outlet.
☒ Include overtime for swapped shifts.
☒ Include overtime for extra shifts.
☐Overtime reward is a fixed value.

[**Overtime** | **Overtime hours** | **100**]

In the scheduling period 02-01-2017 to 08-01-2017 (one week), a particular employee works shift A on the weekdays Monday, Wednesday, Thursday and Friday. Shift A is made up of the following activities:

Activity type	Start	End
Work	09:00	17:00

For the same scheduling period 02-01-2017 to 08-01-2017 (one week), a particular employee works shift B on the weekday Tuesday. Shift B is made up of the following activities:

Activity type	Start	End
Work	09:00	19:00

The account Actual hours includes the following booking:

Actual hours		02-01-2017 to 08-01-2017
Worked hours		
> 02-01-2017	A	08:00
> 03-01-2017	B	10:00
> 04-01-2017	A	08:00
> 05-01-2017	A	08:00
> 06-01-2017	A	08:00
Total		42:00

Overtime is calculated based on the balance on [**Actual hours**], category [**Worked hours**]. To determine the exact times of overtime the realized schedule will be used.

Step 1: If the account balance per day is more than [**08:00**] hours, any additional hours are overtime.
Determining the exact times of overtime:
- The hours at the end of the realized shifts.

Step 2: If the account balance (minus the overtime resulting from step 1) per week (starting on [**Monday**]) is more than [**36:00**] hours are overtime.
Determining the exact time of overtime:
- First the hours on Sunday, next on Saturday, next on Friday, Thursday, Wednesday, Tuesday, Monday.
- The hours at the end of the realized shifts.

- Shifts are recorded completely on the day they start.
- Hours can only be marked as overtime once.
- Hours in between the begin- and end times of work shifts according to the cyclical schedule cannot be marked as overtime.
- Only activities with the following activity kind(s) can be marked as overtime: [**Work**]

☐No unsocial hours allowance is calculated for overtime.

Remark: it is not possible to determine the reason for overtime (reassignment, extra shift, outlet) when using this overtime definition.

Overtime	02-01-2017 to 08-01-2017
Overtime hours	
> 03-01-2017 B	02:00
> 06-01-2017 A	04:00
Total	06:00

18.21 Overtime: calculated from cyclical schedule

Defines overtime as the difference between working time in the realized schedule and the cyclical schedule.

How to use

Overtime is determined per day and is based on the difference between the hours worked according to the realized schedule and the hours work according to the cyclical schedule.
☐ Shift differential does not apply for overtime.

☐ When working as a spare (a week with at least [**Number**] spare shifts) overtime is determined per week and is based on the difference in working hours over the whole week. Shift differential applies to both regular and overtime.

- **Number** is the minimum number of spare shifts that a week must include to be considered as a spare week.

This rule is a definition. No more than one definition of overtime may be included in a rule set.
A definition does not need to specify a target account or target category.

Example

The following rule is defined for booking overtime (Overtime):

Overtime hours will be paid out as indicated below.

☐A threshold value is applied for this rule.
☒ Only positive differences (worked more) are included.
☐Only negative differences (worked less) are included.
☒ Include overtime for outlet.
☒ Include overtime for swapped shifts.
☒ Include overtime for extra shifts.
☐Overtime reward is a fixed value.

[**Overtime** | **Overtime hours** | **100**]

The rotation plan for the period 30-01-2017 to 27-02-2017 (four weeks) provides for a particular employee to work shift A every weekday (Monday to Friday). Shift A is made up of the following activities:

Activity type	Start	End
Work	08:00	16:00

In practice, however, the employee works until 17:00 hours on 24 February.

Overtime is determined per day and is based on the difference between the hours worked according to the realized schedule and the hours work according to the cyclical schedule.
☐ Shift differential does not apply for overtime.

Spare:
☐ When working as a spare (a week with at least [**5**] spare shifts) overtime is determined per week and is based on the difference in working hours over the whole week. Shift differential applies to both regular and overtime.

Overtime	01-01-2017 to 01-01-2018
Overtime hours	
> 24-02-2017 A	01:00
Total	01:00

18.22 Overtime: compare employment hours

Defines overtime as working time in excess of employment hours.

How to use

For calculating overtime hours, the hours, according to the realized shift schedule in this period and the total number of hours that should be worked in this roster according to the employees employment, need to be compared.

☐No non-social hour bonus is granted to overtime hours.

This rule is a definition. No more than one definition of overtime may be included in a rule set.
A definition does not need to specify a target account or target category.

Example

The following rule is defined for booking overtime (Overtime):

Overtime hours will be paid out as indicated below.

☐A threshold value is applied for this rule.
☒ Only positive differences (worked more) are included.
☐Only negative differences (worked less) are included.
☒ Include overtime for outlet.
☒ Include overtime for swapped shifts.
☒ Include overtime for extra shifts.
☐Overtime reward is a fixed value.

[**Overtime** | **Overtime hours** | **100**]

An employee is contracted to work forty hours per week (see 'Employee management', 'Employment rules' tab).

In the period 27-02-2017 to 27-03-2017 (four weeks), the employee works shift A every weekday (Monday to Friday). Shift A is made up of the following activities:

Activity type	Start	End
Work	08:00	16:30

For calculating overtime hours, the hours, according to the realized shift schedule in this period and the total number of hours that should be worked in this roster according to the employees employment, need to be compared.

☐No non-social hour bonus is granted to overtime hours.

Overtime		27-02-2017 to 27-03-2017
Overtime hours		
> 23-03-2017	A	01:30
> 24-03-2017	A	08:30
Total		10:00

The employee was contracted to work 160:00 hours in the scheduling period, but actually worked 170:00 hours.

18.23 Overtime: compare employment hours per x weeks

Defines overtime as working time in excess of contractual hours.

How to use

To calculate overtime hours a period of [**Number**] weeks has to be applied. The hours according to the realized shift schedule in this period and the total number of hours that should be worked in this period according to the employee's contract need to be compared.

- **Number** is the number of weeks comprising the period for which overtime is to be calculated.

This rule is a definition. No more than one definition of overtime may be included in a rule set.
A definition does not need to specify a target account or target category.
Calculation periods are defined by counting the relevant number of weeks from week 1 of the year to which the account relates.

Example

The following rule is defined for booking overtime (Overtime):

Overtime hours will be paid out as indicated below.

☐A threshold value is applied for this rule.
☒ Only positive differences (worked more) are included.
☐Only negative differences (worked less) are included.
☒ Include overtime for outlet.
☒ Include overtime for swapped shifts.
☒ Include overtime for extra shifts.
☐Overtime reward is a fixed value.

[**Overtime** | **Overtime hours** | **100**]

An employee is contracted to work thirty-six hours per week (see 'Employee management', 'Employment rules' tab).

In the period 27-02-2017 to 27-03-2017 (four weeks), the employee works shift A every weekday (Monday to Friday). Shift A is made up of the following activities:

Activity type	Start	End
Work	09:00	16:30

To calculate overtime hours a period of [**2**] weeks has to be applied. The hours according to the realized shift schedule in this period and the total number of hours that should be worked in this period according to the employee's contract need to be compared.

Overtime	27-02-2017 to 27-03-2017
Overtime hours	
> 10-03-2017 A	03:00
> 24-03-2017 A	03:00
Total	06:00

The employee was contracted to work 72:00 hours every two weeks, but actually worked 75:00 hours in the relevant period.

18.24 Overtime: days according to cyclical schedule

Defines the days to be taken into account when calculating overtime included in the published schedule.

How to use

Which days count as Saturdays, Sundays and Mondays is determined by the cyclical schedule. This is needed for calculating overtime hours.
☐Actual calendar Saturdays, Sundays and Mondays always count, regardless whether there should

have been worked on this day according to the cyclical schedule.

This rule is a definition. No more than one definition of overtime may be included in a rule set.
A definition does not need to specify a target account or target category.

Example

The following rule is defined for booking overtime (Overtime per time period):

Employees who work overtime between [**00:00**] and [**00:00**] hours on a [**Saturday**] will [**always**] be paid as described below. This rule applies to employees whose salary group lies between [**A**] and [**F**] and who
☐ have
☒ have not
chosen for the time for time arrangement.

☐This rule applies [**only**] to the first [**00:00**] hours of overtime.
☐This rule applies [**only**] if the overtime starts before [**00:00**].

☐A threshold value is applied for this rule [**only**].
☒Only positive differences (worked more) are included.
☐Only negative differences (worked less) are included.
☒Include overtime for outlet.
☒Include overtime for swapped shifts.
☒Include overtime for extra shifts.
☐Overtime reward is a fixed value.

[**Overtime** | **Overtime hours** | **100**]

According to the published schedule, an employee has a free day on Friday 24 February, when the activity kind assigned to him/her is 'Free Saturday'. In practice, however, the employee works the following shift on the day in question:

Activity type	Start	End
Work	08:00	16:00

Which days count as Saturdays, Sundays and Mondays is determined by the cyclical schedule. This is needed for calculating overtime hours.
☐Actual calendar Saturdays, Sundays and Mondays always count, regardless whether there should have been worked on this day according to the cyclical schedule.

Overtime	01-01-2017 to 01-01-2018
Overtime hours	
> 24-02-2017 A	08:00
Total	08:00

18.25 Overtime: fixed in regard to publication

Defines overtime as the difference between working time in the realized schedule and the published schedule.

How to use

Overtime is calculated on a daily basis by determining the difference between the hours worked according to the master schedule and the hours according to the published schedule.
☐ No non-social hour allowance is granted to overtime hours

☐ In case of an on call week - a week with at least [**Number**] on call shifts - the difference in hours worked over the whole week is considered. A non-social hour allowance is granted to all hours worked.

- **Number** is the minimum number of standby shifts that a week must include to be considered a standby week.

This rule is a definition. No more than one definition of overtime may be included in a rule set.
A definition does not need to specify a target account or target category.

Example

The following rule is defined for booking overtime:

Overtime hours will be paid out as indicated below.

☐A threshold value is applied for this rule.
☒ Only positive differences (worked more) are included.
☐Only negative differences (worked less) are included.
☒ Include overtime for outlet.
☒ Include overtime for swapped shifts.
☒ Include overtime for extra shifts.
☐Overtime reward is a fixed value.

[**Overtime** | **Overtime hours** | **100**]

The published schedule for the period 30-01-2017 to 27-02-2017 (four weeks) provides for a particular employee to work shift A every weekday (Monday to Friday). Shift A is made up of the following activities:

Activity type	Start	End
Work	08:00	16:00

In practice, however, the employee works until 17:00 hours on 24 February.

Overtime is calculated on a daily basis by determining the difference between the hours worked according to the master schedule and the hours according to the published schedule.
☐ No non-social hour allowance is granted to overtime hours

☐ In case of an on call week - a week with at least [**5**] on call shifts - the difference in hours worked over the whole week is considered. A non-social hour allowance is granted to all hours worked.

Overtime	01-01-2017 to 01-01-2018
Overtime hours	
> 24-02-2017 A	01:00
Total	01:00

18.26 Overtime: hours above day-norm per month

Defines overtime as the number of hours worked minus the number of working days times the daily standard.

How to use

Overtime is defined as the total performed working hours in a month minus the number of working days in that month times [**Daily standard**] hours.

- **Daily standard** is the number of hours in a standard working day.

This rule is a definition. No more than one definition of overtime may be included in a rule set.
A definition does not need to specify a target account or target category.

Example

The following rule is defined for booking overtime:

Overtime hours will be paid out as indicated below.

☐A threshold value is applied for this rule.
☒ Only positive differences (worked more) are included.
☐Only negative differences (worked less) are included.
☒ Include overtime for outlet.
☒ Include overtime for swapped shifts.
☒ Include overtime for extra shifts.
☐Overtime reward is a fixed value.

[**Overtime** | **Overtime hours** | **100**]

The published schedule for March 2017 provides for a certain employee to work shift A every weekday (Monday to Friday). Shift A is made up of the following activities:

Activity type	Start	End
Work	08:00	16:00

Overtime is defined as the total performed working hours in a month minus the number of working days in that month times [**07:30**] hours.

Overtime	01-01-2017 to 01-01-2018
Overtime hours	
> 30-03-2017 A	03:30
> 31-03-2017 A	08:00
Total	11:30

The employee should have worked twenty-three standard days, making 172:30 hours, but actually worked 184:00 hours.

18.27 Overtime: hours above norm per week

Defines overtime as the number of hours worked in excess of a defined standard number.

How to use

Overtime is defined as the total performed working hours in a roster minus the average number of hours of [**Standard hours**] hours per week.

☐ No non-social hour bonus is granted to overtime hours.

- **Standard hours** is the number of hours in excess of which working time counts as overtime.

This rule is a definition. No more than one definition of overtime may be included in a rule set.
A definition does not need to specify a target account or target category.

Example

The following rule is defined for booking overtime (Overtime):

Overtime hours will be paid out as indicated below.

☐ A threshold value is applied for this rule.
☒ Only positive differences (worked more) are included.
☐ Only negative differences (worked less) are included.
☒ Include overtime for outlet.
☒ Include overtime for swapped shifts.
☒ Include overtime for extra shifts.
☐ Overtime reward is a fixed value.

[**Overtime** | **Overtime hours** | **100**]

In the scheduling period 30-01-2017 to 27-02-2017 (four weeks), a particular employee works shift A every weekday (Monday to Friday). Shift A is made up of the following activities:

Activity type	Start	End
Work	08:00	16:30

Overtime is defined as the total performed working hours in a roster minus the average number of hours of [**40:00**] hours per week.

☐ No non-social hour bonus is granted to overtime hours.

Overtime	01-01-2017 to 01-01-2018
Overtime hours	
> 23-02-2017 A	01:30
> 24-02-2017 A	08:30
Total	10:00

18.28 Overtime: number of days from published master schedule

Defines the days to be taken into account when calculating overtime included in the published schedule.

How to use

For calculating overtime the published schedule is used to determine whether a day counts as a Saturday, Sunday or Monday.

☐ Regular Saturdays, Sundays and Mondays always count, whether one works on this day according to the published master schedule or not.

This rule is a definition. No more than one definition of overtime may be included in a rule set.
A definition does not need to specify a target account or target category.

Example

The following rule is defined for booking overtime (Overtime per time period):

Employees who work overtime between [**00:00**] and [**00:00**] hours on a [**Saturday**] will [**always**] be paid as described below. This rule applies to employees whose salary group lies between [**A**] and [**F**] and who
☐ have
☒ have not
chosen for the time for time arrangement.

☐This rule applies [**only**] to the first [**00:00**] hours of overtime.
☐This rule applies [**only**] if the overtime starts before [**00:00**].

☐A threshold value is applied for this rule [**only**].
☒Only positive differences (worked more) are included.
☐Only negative differences (worked less) are included.
☒Include overtime for outlet.
☒Include overtime for swapped shifts.
☒Include overtime for extra shifts.
☐Overtime reward is a fixed value.

[**Overtime** | **Overtime hours** | **100**]

According to the published schedule, an employee has a free day on Friday 24 February, when the activity kind assigned to him/her is 'Free Saturday'. In practice, however, the employee works the following shift on the day in question:

Activity type	Start	End
Work	08:00	16:00

For calculating overtime the published schedule is used to determine whether a day counts as a Saturday, Sunday or Monday.

☐ Regular Saturdays, Sundays and Mondays always count, whether one works on this day according to the published master schedule or not.

Overtime	01-01-2017 to 01-01-2018
Overtime hours	
> 24-02-2017 A	08:00
Total	08:00

18.29 Overtime: threshold value

Stipulates that, on a weekly basis, the total overtime made by an employee is corrected by a threshold value.

How to use

[**Threshold value**] hours overtime in a week will not be paid out if these hours should have been credited for the lowest allowance percentage according to "Overtime" type of rules and if a threshold value is applied. The week starts on Monday at [**Start time**].

- **Threshold value** is the threshold value for the total overtime an employee made in a week; if the total overtime is lower than the threshold value a counter-booking is made.

- **Start time** is the time on Monday that a new week starts; this is useful to calculate the total overtime made in a week.

The rule **Overtime: threshold value** will only be used if you have selected the checkmark **A threshold value is applied for this rule** in the rule with type **Overtime**.
It is not necessary to specify a target account and target category in the context of this rule.

Example

The following overtime definition is included in the rule set (Overtime: fixed in regard to publication):

Overtime is calculated on a daily basis by determining the difference between the hours worked according to the master schedule and the hours according to the published schedule.
☐ No non-social hour allowance is granted to overtime hours

☐ In case of an on call week - a week with at least [**5**] on call shifts - the difference in hours worked over the whole week is considered. A non-social hour allowance is granted to all hours worked.

In the scheduling period 30-01-2017 to 13-02-2017 (two weeks), where 30-01-2017 and 06-02-2017 are the Mondays, a particular employee is planned to work shift A every weekday (Monday to Friday). Shift A is made up of the following activities:

Activity type	Start	End
Work	08:00	16:30

In practice, however, the employee works every Monday and Tuesday until 17:30.

Overtime hours will be paid out as indicated below.

☒ A threshold value is applied for this rule.
☒ Only positive differences (worked more) are included.
☐ Only negative differences (worked less) are included.
☒ Include overtime for outlet.
☒ Include overtime for swapped shifts.
☒ Include overtime for extra shifts.
☐ Overtime reward is a fixed value.

[**Overtime** | **Overtime hours** | **100**]

the checkmark **A threshold value is applied for this rule** is selected.

[**01:30**] hours overtime in a week will not be paid out if these hours should have been credited for the lowest allowance percentage according to "Overtime" type of rules and if a threshold value is applied. The week starts on Monday at [**00:00**].

Overtime		01-01-2017 to 01-01-2018
Overtime hours		
> 30-01-2017		-01:00
> 30-01-2017	A	01:00
> 31-01-2017		-00:30
> 31-01-2017	A	01:00
> 06-02-2017		-01:00
> 06-02-2017	A	01:00
> 07-02-2017		-00:30
> 07-02-2017	A	01:00
Total		01:00

19 Overtime: Round off

The compensation rule in this group can be used to round off the entries of balance of an account category.

The compensation rule "Overtime: rounded off" can be used for all kinds of entries (in contrary to what the name implies).

19.1 Overtime: rounded off

Stipulates that, on a monthly basis, the balance under a particular category is to be rounded off and the difference between the rounded-off balance and the unadjusted balance is to be booked to another category.

How to use

Entries on section [**Category**] will be rounded off monthly to the nearest [**Significance**] minutes in favor of category [**Target category**].

- **Category** is the category whose balance is to be rounded off.
- **Significance** is the hour-multiple to which the booking must be rounded off.
- **Target category** is the category to which the difference between the rounded-off balance and the unadjusted balance is to be booked.

It is not necessary to specify a target account and target category in the context of this rule.
The balance is rounded off to the nearest multiple of the significance.
The specified category may be in the same account as the source category.

Example

The account Hours includes the following booking for February 2017:

Hours	01-02-2017 to 01-03-2017
Extra hours	
> 01-02-2017	63:06
Total	63:06

Entries on section [**Extra hours**] will be rounded off monthly to the nearest [**02:00**] minutes in favor of category [**Rounded off**].

Hours	01-02-2017 to 01-03-2017
Rounded off	
> 28-02-2017	01:06
Extra hours	
> 01-02-2017	63:06
> 28-02-2017	-01:06
Total	63:06

20 Pass on call-out

The compensation rule in this group can be used to carry bookings from one category to another in case of a call-out. This compensation rules also carry over bookings just as the rules in the group **Carry over** only this compensation rule is more specific in the case that the carrying of bookings only includes call-outs. This is why this compensation rule is set in an apart group.

It is possible to carry over bookings from one category to another for categories within the same account and categories of different accounts.
Call-out is a booking property. Provided that the system configuration settings allow, this property can be viewed in the booking details displayed in the 'Accounts' window.

20.1 Pass on call-out

Stipulates that any call-out bookings under a particular category in one account are to be carried over to a particular category in another account.

How to use

Entries on account [**Account**], category [**Category**] of at least [**Minimum entry**] will, in case of 'Call-out', be credited to the following categories.

☐ Credit hours
☐ Credit difference total - hours

☐ Transfer bookings that have not been approved.

[**Target account** | **Target category** | **Percentage**]

- **Account** is the account from which the bookings are to be carried over.
- **Category** is the category in the source account from which the bookings are to be carried over.
- **Minimum entry** is the minimum value that a booking must have in order to be carried over in accordance with the rule.

Call-out is a booking property. Provided that the system configuration settings allow, this property can be viewed in the booking details displayed in the 'Accounts' window.

The amount booked will be the difference between the minimum booking and the original booking.

If the checkmark **Credit hours** is selected, the actual number of hours booked under the relevant category will be carried over.
If **Credit difference total-hours** is selected, the specified percentage of the difference between the value and the number of hours booked will be carried over.
If both options are activated, the specified percentage of the value will be carried over.

If the checkmark **Transfer bookings that have not been approved** is selected the bookings are made before the entries of the source account are approved.

Example

The account Overtime hours includes the following booking:

Overtime hours	01-01-2017 to 01-01-2018
Extra hours	
> 01-01-2017 A	10:00 / 140% / 14:00
Total	14:00

The booking has been assigned the property ‘callout’.

Entries on account [**Overtime hours**], category [**Extra hours**] of at least [**16:00**] will, in case of 'Call-out', be credited to the following categories.

☒ Credit hours
☒ Credit difference total - hours

☐ Transfer bookings that have not been approved

[**Hours** | **Callout** | **100**]

Hours	01-01-2017 to 01-01-2018
Callout	
> 01-01-2017 A	10:00 / 100% / 02:00
Total	02:00

21 Round off the account balances

The compensation rule in the group **Round off the account balances** can be used to round off account balances per specified period.

21.1 Round off the account balances

Stipulates that a category balance is to be rounded off for each period of a specified number of weeks or months.

How to use

Entries on the account mentioned below have to be rounded per [**Period**].
The entries will be rounded off to the nearest [**Significance**] hours.

[**Target account** | **Target category**]

- **Period** is the number of weeks or months over which the bookings are to be rounded up.If the specified period is a week, the booking will be rounded off on the last day of the week; if the specified period is a month, the booking will be rounded off on the last day of the month.
- **Significance** is the hour-multiple to which the booking must be rounded off.

> Rounding off is always to the next positive multiple of the significance figure. A negative balance will therefore always be rounded off to 0:00 hours.

Example

The account Hours includes the following booking for February 2017:

Hours	01-02-2017 to 01-03-2017
Extra hours	
> 01-02-2017	61:06
Total	61:06

Entries on the account mentioned below have to be rounded per [**1 Month**]. The entries will be rounded off to the nearest [**2:00**] hours:
[**Hours** | **Extra hours**]

Hours	01-02-2017 to 01-03-2017
Extra hours	
> 01-02-2017	61:06
> 28-02-2017	00:54
Total	62:00

22 Shifted time windows

The compensation rule in the group **Shifted time windows** can be used to generate compensations when hours in a previous published time window is changed.

22.1 Shifted time windows

Stipulates that the time that the time window is shifted is to be booked.

How to use

Hours in a time window outside a previously communicated time window will be paid out as described below.

This rule is only valid for hours that are communicated between [**Minimal**] and [**Maximum**] day(s) in advance.
This rule is not valid for time window changes of type 'requested' and 'mandatory'.

[**Target account** | **Target category** | **Percentage**]

- **Minimum** is the lower bound of the range in which mutations in the time window is selected.
- **Maximum** is the upper bound of the range in which mutations in the time window is selected.

The published date of a department's time window can be set in 'Schedule management' under the tab 'Time windows'.

Example

The time window published until date is 01-02-2017. In the schedule of February 2017, in status planned, Shift A is planned on each work day:

Activity type	Start	End
Work	08:00	12:00
Break	12:00	12:30
Work	12:30	16:00

A time window is generated for the whole month using the option 'Use the relevant begin- and end time of the shift'. This results in a time window on each working day from 08:00 - 16:00.
The time window published until date is set to 01-03-2017 for this department on 02-01-2017.

On 30-01-2017, the time window for this employee on 03-02-2017 is changed to 08:00 - 20:00.

Hours in a time window outside a previously communicated time window will be paid out as described below.

This rule is only valid for hours that are communicated between [**0**] and [**7**] day(s) in advance.
This rule is not valid for time window changes of type 'requested' and 'mandatory'.

[**Time windows** | **Shifted 0-7 days** | **100**]

Time windows	01-01-2017 to 01-01-2018
Shifted 0-7 days	
> 03-02-2017	04:00
Total	04:00

23 Skills

The compensation rule in the group **Skill** can be used to filter on the entries that required a certain skill.

23.1 Work with skill

Stipulates that the time that an employee is scheduled to spend working at a workstation to which a user-defined qualification applies is to be booked.

How to use

Consider department [**Department**] and skill [**Skill**].
Whenever an employee works at a workstation which requires this skill, the following entry is created:
For this rule, [**do/do not**] take the transition from daylights savings time to standard time into account.

[**Target account** | **Target category** | **Percentage**]

- **Department** is the organizational unit or department in respect of which the qualification is defined.
- **Skill** is the qualification that, if required for the workstation or activity type in question, triggers an entry.

Example

Shift A, as scheduled for Saturday 8 April 2017, is made up of the following activities:

Activity type	Workstation	Start	End
Work	Room 1	08:00	12:00
Break		12:00	12:30
Work	Room 2	12:30	16:00

To work in room 2, the qualification ABC is required.

Consider department [**Organization**] and skill [**ABC**].
Whenever an employee works at a workstation which requires this skill, the following entry is created:
For this rule, [**do**] take the transition from daylights savings time to standard time into account.

[**Allowances** | **Qualifications** | **100**]

Allowances	01-01-2017 to 01-01-2018
Qualifications	
> 08-04-2017 A	03:30
Total	03:30

24 Time window

The compensation rule in the group **Time window** can be used to give a good overview of the time and dates employees are assigned to time windows for a given period.

Time windows can be generated by use of the right mouse menu within the planboard.

24.1 Time window hours

Stipulates that the duration an employee is assigned to a time window is to be entered on the indicated account for a given period.

The compensation rule 'Time window hours' is in general used to give the planner an overview of the day and times employees are assigned to time windows.

How to use

Enter the hours of active time windows

[**Target account** | **Target category** | **Percentage**]

Example

Assume that for an employee time windows are generated between 09:00 to 17:00 for weekdays for the period 03-04-2017 to 10-04-2017 (Monday to Monday).

Enter the hours of active time windows

[**Hours** | **Time window** | **100**]

Hours	03-04-2017 to 10-04-2017
Time window	
> 03-04-2017	08:00
> 04-04-2017	08:00
> 05-04-2017	08:00
> 06-04-2017	08:00
> 07-04-2017	08:00
Total	40:00

25 Total working hours

The compensation rule in the group **Total working hours** can be used to book activities with specific activity property values.

25.1 Entry based on activity type/workstation

Stipulates that the number of hours spent or the number of times spent on an activity type and/or at a workstation is to be booked.

How to use

When the total of worked hours on a day for activities lies between [**Minimum duration**] and [**Maximum duration**], the calculation below is made.

☐ Only take into account activities of which the activity type has the property [**Property**].
☐ Only takes into account activities on workstations where the bicycle property has the value [**Yes/No**].
☐ This rule is only valid for continuous hours.
☐ The entry made is a fixed value.

[**Target account** | **Target category** | **Percentage**]

- **Property** defines the property that should be activated for an activity. The property activated for an activity, determines whether a booking is made.
- **Yes/No** indicates if the employee travels by bicycle. The option activated for an employee determines whether a booking is made.
- **Minimum duration** is the minimum number of hours that must be spent on the relevant activities for an entry to be made.
- **Maximum duration** is the maximum number of hours that must be spent on the relevant activities for an entry to be made.

If the checkmark **This rule is only valid for continuous hours** is selected, a booking is only made for the continuous sequence of hours for which the minimum and maximum durations are valid.
The checkmark **The entry made is a fixed value** is useful if you wish to count the number of times employees has worked on an activity with a certain property (given the minimum and maximum duration are valid).
If the checkmark is selected the option [**Percentage**] below will changes to [**Value**].

Example

Shift A, as scheduled for 8 April 2017, is made up of the following activities:

Activity type	Start	End
Indoor work	08:00	12:00
Break	12:00	12:30
Outdoor work	12:30	16:00

The activity type 'Outdoor work' has the property 'Outside'.

When the total of worked hours on a day for activities lies between [**0:00**] and [**0:00**], the calculation below is made.

☒ Only take into account activities of which the activity type has the property [**Outside**].
☒ Only takes into account activities on workstations where the bicycle property has the value [**No**].
☐ This rule is only valid for continuous hours.
☐ The entry made is a fixed value.

[**Hours** | **Outdoors** | **100**]

Hours	01-01-2017 to 01-01-2018
Outside	
08-04-2005 A	03:30
Total	03:30

26 Travel distance

The compensation rules in the group **Travel distance** can be used to determine the travel allowances of employees. In this the travel distance are based on the distances an employee has traveled. This could be the distances between the employee's home to work and back, but also the distances between the different workstations within a shifts.

In order to determine the travel distances the compensation rules make use of the postal codes of the employee's home address and the places to work.

26.1 Travel distance in shift

Stipulates that the number of kilometers that an employee travels in the course of a shift is to be booked.

How to use

The calculation of the distance between two consecutive addresses within a shift or between two shifts which an employee performs is as follows:
This rule does not apply if there is an interruption between the activities or shifts of over [**Duration**] hour consisting of activity kind(s): [**Kinds**]

☐ This rule applies only when the start address belongs to an activity with [**Type**][**Attribute**].
☐ This rule applies only when the end address belongs to an activity with [**Type**][**Attribute**].

☐ This rule [**applies/does not apply**] in case the property [**Property**] of the [**Workstation**] locations is set to value [**Value**]. When an activity has no workstation, the default value of the property is used.

[**Target account** | **Target category** | **Percentage**]

- **Duration** is the maximum length of interruption that will trigger a booking.
- **Kinds** are activity kinds [Amplitude, Availability (no labor), …, Work (stand by), WTR]; if an interruption consists of a specified activity kind, no booking will be made.
- **Type** is the type [class, activity types, main activity type] to which the attributes, for which the bookings are to be made, belongs to.
- **Attribute** is the attribute for which the bookings are to be made.
- **Property** is the property associated with an workstation; the property can be used to indicate if this workstation should be used for the rule "Travel distance at the end of the shift".
- **Workstation** is the workstation [begin workstation, end workstation, begin and end workstation] for which the locations properties are checked.
- **Value** is the value the property of the location should have in order to trigger a booking.

This rule can be used to generate entries only in an account containing bookings of the type ‘Number’.

Example

Shift A, as scheduled for Saturday 8 April 2017, is made up of the following activities:

Activity type	Address	Start	End
Work	1234 AA	08:00	12:00
Break		12:00	12:30
Work	5678 BB	12:30	16:00

The distance from 1234 AA to 5678 BB is ten kilometers.

The calculation of the distance between two consecutive addresses within a shift or between two shifts which an employee performs is as follows:
This rule does not apply if there is an interruption between the activities or shifts of over [**01:00**] hour consisting of activity kind(s): [**Break**]

☐ This rule applies only when the start address belongs to an activity with [**activity types**[**(None)**]].
☐ This rule applies only when the end address belongs to an activity with [**activity types**][**(None)**].

☐ This rule [**applies**] in case the property [**(None)**] of the [**begin workstation**] locations is set to value [**(None)**]. When an activity has no workstation, the default value of the property is used.

[**Kilometers** | **During shift** | **100**]

Kilometers	01-01-2017 to 01-01-2018
During shift	
> 08-04-2018 A	10:00
Total	10:00

26.2 Travel distance at the start of the shift

Stipulates that the number of kilometers is to be booked, which the employee has to travel from his/her home to the workplace associated with the first activity in the shift.

How to use

The calculation of the distance between address of an employee and the first address of a shift which an employee performs is as follows.
This rule also applies to the distance home-work after an interruption of the shift of over [**Duration**] hour consisting of activity kind(s): [**Kinds**]

☐ This rule applies only to activities with [**Type**][**Attribute**].

☐ This rule [**applies/does not apply**] in case the property [**Property**] of the location of the workstation is set to value [**Value**]. When an activity had no workstation, the default value of the property is used.

[**Target account** | **Target category** | **Percentage**]

- **Duration** is the maximum length of interruption that will trigger a booking.
- **Kinds** are activity kinds [Amplitude, Availability (no labor), ..., Work (stand by), WTR]; if an interruption consists of a specified activity kind, no booking will be made.
- **Type** is the type [class, activity types, main activity type] to which the attributes, for which the bookings are to be made, belongs to.
- **Attribute** is the attribute for which the bookings are to be made.

- **Property** is the property associated with an workstation; the property can be used to indicate if this workstation should be used for the rule "Travel distance at the end of the shift".
- **Value** is the value the property of an location should have in order to trigger a booking.

This rule can be used to generate entries only in an account containing bookings of the type 'Number'.

Example

An employee's home address is on postcode 9999 CC.

Shift A, which the employee is scheduled to work on Saturday 8 April 2017, is made up of the following activities:

Activity type	Address	Start	End
Work	1234 AA	08:00	12:00
Break		12:00	12:30
Work	5678 BB	12:30	16:00

The distance from 9999 CC to 1234 AA is ten kilometers.

The calculation of the distance between address of an employee and the first address of a shift which an employee performs is as follows.
This rule also applies to the distance home-work after an interruption of the shift of over [**01:00**] hour consisting of activity kind(s): [**Break**]

☒ This rule applies only to activities with [**activity types**][**Work**].

☐ This rule **applies** in case the property [**(None)**] of the location of the workstation is set to value [**(None)**]. When an activity had no workstation, the default value of the property is used.

[**Kilometers** | **Home-Work** | **100**]

Kilometers	01-01-2017 to 01-01-2018
Home-Work	
> 08-04-2017 A	10:00
Total	10:00

26.3 Travel distance at the end of the shift

Stipulates that the number of kilometers is to be booked, which the employee has to travel from the workplace associated with the last activity in the shift to his/her home.

How to use

The calculation of the distance between the last address of a shift which an employee performs an the address of an employee is as follows.
This rule also applies to the distance work-home after an interruption of the shift of over [**Duration**] hour consisting of activity kind(s): [**Kinds**]

☐ This rule applies only to activities with [**Type**][**Attribute**].

☐ This rule [**applies/does not apply**] in case the property [**Property**] of the location of the workstation is set to value [**Value**]. When an activity had no workstation, the default value of the property is used.

[**Target account** | **Target category** | **Percentage**]

- **Duration** is the maximum length of interruption that will trigger a booking.
- **Kinds** are activity kinds [Amplitude, Availability (no labor), …, Work (stand by), WTR]; if an interruption consists of a specified activity kind, no booking will be made.
- **Type** is the type [class, activity types, main activity type] to which the attributes, for which the bookings are to be made, belongs to.
- **Attribute** is the attribute for which the bookings are to be made.
- **Property** is the property associated with an workstation; the property can be used to indicate if this workstation should be used for the rule "Travel distance at the end of the shift".
- **Value** is the value the property of an location should have in order to trigger a booking.

This rule can be used to generate entries only in an account containing bookings of the type 'Number'.

Example

An employee's home address is on postcode 9999 CC.

Shift A, which the employee is scheduled to work on Saturday 8 April 2017, is made up of the following activities:

Activity type	Address	Start	End
Work	1234 AA	08:00	12:00
Break		12:00	12:30
Work	5678 BB	12:30	16:00

The distance from 5678 BB to 9999 ZZ is twenty kilometers.

The calculation of the distance between the last address of a shift which an employee performs an the address of an employee is as follows.
This rule also applies to the distance work-home after an interruption of the shift of over [**01:00**] hour consisting of activity kind(s): [**Break**]

☒ This rule applies only to activities with [**activity types**][**Work**].

☐ This rule **applies** in case the property [**(None)**] of the location of the workstation is set to value [**(None)**]. When an activity had no workstation, the default value of the property is used.

[**Kilometers** | **Work-home** | **100**]

Kilometers	01-01-2017 to 01-01-2018
Work-home	
> 08-04-2017 A	20:00
Total	20:00

27 Travel expenses

The compensation rule in the group **Travel expenses** can be used to determine the travel allowances of employees, in case no public transport is available.

> In order to determine the travel distances the compensation rules make use of the postal codes of the employee's home address and the places to work.

27.1 Travel expenses

Stipulates that the number of kilometers that an employee travels from home to work or from work to home at the start or end of shift are to be booked if the shift start time or end time is outside the applicable public transport running times.

How to use

Employee will receive traveling allowances for shifts at hours where no public transportation is available. The allowance is based upon the distance between the employee's home address and the place to work and back home again, using postal codes as reference. The employee will be paid out only if this return distance is [**Sign**] than [**Number**] kilometers.

[**Target account** | **Target category** | **Percentage**]

- **Sign** [<,<=,>,>=] is a mathematical sign used to specify the travel distance for which expenses are payable.
- **Number** is a numeric value which, in conjunction with the sign, specifies the travel distance for which expenses are payable.

> Public transport running times can be defined for individual employees, provided that the system configuration settings allow.

Example

An employee's home postcode is 9999 ZZ. The public transport running times defined for the employee indicate that he/she cannot get to work by public transport before 07:00 on a Saturday.

Shift A, which the employee is scheduled to work on Saturday 8 April 2017, is made up of the following activities:

Activity type	Address	Start	End
Work	1234 AA	04:00	12:00

The one-way distance between 9999 ZZ to 1234 AA is 12 kilometers.

Employee will receive traveling allowances for shifts at hours where no public transportation is available. The allowance is based upon the distance between the employee's home address and the place to work and back home again, using postal codes as reference. The employee will be paid out only if this return distance is [**>**] than [**10**] kilometers.

[**Kilometers** | **Home-work** | **100**]

Kilometers	01-01-2017 to 01-01-2018
Home-work	
08-04-2017 A	24:00
Total	24:00

28 Value based on annual employment hours

The compensation rules in the group **Value based on annual employment hours** can all be used to calculate the annual value of an employee. The different rules in this group offers different formulations to calculate the annual value.

> In ORTEC WS the annual value is the number of hours an employee should work during the rest of the year according to its contract. For example if an employee is contracted to work 1200 hours in a year and the schedule of exactly half the year is published already, the concerning employee has a annual value of 600 hours.

28.1 Annual value based on age

Stipulates that a value is to be booked, based on an employee's working hours (working week), provided that the employee's age is as specified.

How to use

The labor-time per week equals [**Hours per week**] hours for a full-time contract. The labor-time per year equals [**Hours per year**] hours for a full-time contract. For employees between the ages of [**Min age**] and [**Max age**] the accounts below are based upon the annual labor-time. Part-time employees are credited proportionally.
The age that is reached [**Moment**] is considered here.
This rule applies for employees with a salarygroup between [**Min salary code**] and [**Max salary code**].

[**Target account** | **Target category** | **Percentage**]

- **Hours per week** is the number of hours in a full-time employee's contractual working week.
- **Hours per year** is the number of hours in a full-time employee's contractual working year.
- **Min age** is the minimum age (in years) that an employee needs to have reached for a booking to be made.
- **Max age** is the maximum age (in years) that an employee may have reached for a booking to be made.
- **Moment** is the reference date [at the end of the year, at the end of the month] for determining the employee's age.
- **Min salary code** is the lowest salary code that an employee may have for a booking to be triggered.
- **Max salary code** is the highest salary code that an employee may have for a booking to be triggered.

> In the context of this rule, the booking date is the date on which the employee's employment conditions took effect, provided that this date is after 1 January of the year in question. Otherwise, the booking date is 1 January of the year in question.

Example

Employee's salary code:	C
Hours per week:	24
Employee's date of birth:	12-03-1965

The labor-time per week equals [**36**] hours for a full-time contract. The labor-time per year equals [**1872**] hours for a full-time contract. For employees between the ages of [**18**] and [**65**] the accounts below are based upon the annual labor-time. Part-time employees are credited proportionally.
The age that is reached [**at the end of the year**] is considered here.
This rule applies for employees with a salarygroup between [**A**] and [**E**].

[**Annual hours** | **Hours to be credited** | **100**]

Annual hours	01-01-2017 to 01-01-2018
Hours to be credited	
> 01-01-2017	1248:00
Total	1248:00

28.2 Annual value based on age and time in the company

Stipulates that a value is to be booked, based on an employee's working hours (working week), provided that the employee's age and joining date are as specified.

How to use

The labor-time per week equals [**Hours per week**] hours for a full-time contract. The labor-time per year equals [**Hours per year**] for a full-time contract. For employees aged [**Min age**] to [**Max age**] and who took up their duties between [**Min date**] and [**Max date**] the credits below are based upon the annual labor-time. Part-time employees are credited proportionally.
The age that is reached is considered here: [**Moment**]

[**Target account** | **Target category** | **Percentage**]

- **Hours per week** is the number of hours in a full-time employee's contractual working week.
- **Hours per year** is the number of hours in a full-time employee's contractual working year.
- **Min age** is the minimum age (in years) that an employee needs to have reached for a booking to be made.
- **Max age** is the maximum age (in years) that an employee may have reached for a booking to be made.
- **Min date** is the earliest joining date that will trigger a booking.
- **Max date** is the latest joining date that will trigger a booking.
- **Moment** is the reference date [at the end of the year, at the end of the month] for determining the employee's age.

In the context of this rule, the booking date is the date on which the employee's employment conditions took effect, provided that this date is after 1 January of the year in question. Otherwise, the booking date is 1 January of the year in question.

Example

Employee's salary code:	C
Hours per week:	24
Employee's date of birth:	12-03-1965

The employee entered service on 1 May 2003.

The labor-time per week equals [**36**] hours for a full-time contract. The labor-time per year equals [**1872**] for a full-time contract. For employees aged [**18**] to [**65**] and who took up their duties between [**01-01-2000**] and [**01-01-2004**] the accounts below are based upon the annual labor-time. Part-time employees are credited proportionally.
The age that is reached is considered here: [**at the end of the year**]

[**Annual hours** | **Hours to be credited** | **100**]

Annual hours	01-01-2017 to 01-01-2018
Hours to be credited	
> 01-01-2017	1248:00
Total	1248:00

28.3 Annual value based on employment hours

Stipulates that a value is to be booked, based on an employee's working hours (working week), provided that the employee's salary code is as specified.

How to use

The weekly labor-time for full-time employees equals [**Hours per week**] hours. The yearly labor-time for full time employees equals [**Hours per year**] hours. The credits below will be based upon the annual labor-time. Part-time employees will be paid out proportionally.
This rule applies for employees with a salarygroup between [**Min salary code**] and [**Max salary code**].

[**Target account** | **Target category** | **Percentage**]

- **Hours per week** is the number of hours in a full-time employee's contractual working week.
- **Hours per year** is the number of hours in a full-time employee's contractual working year.
- **Min salary code** is the lowest salary code that an employee may have for a booking to be triggered.
- **Max salary code** is the highest salary code that an employee may have for a booking to be triggered.

In the context of this rule, the booking date is the date on which the employee's employment conditions took effect, provided that this date is after 1 January of the year in question. Otherwise, the booking date is 1 January of the year in question.

Example

Employee's salary code:	C
Hours per week:	24
Employee's date of birth:	12-03-1965

The weekly labor-time for full-time employees equals [**36**] hours. The yearly labor-time for full time employees equals [**1872**] hours. The credits below will be based upon the annual labor-time. Part-time employees will be paid out proportionally.
This rule applies for employees with a salarygroup between [**A**] and [**E**].

[**Annual hours** | **Hours to be credited** | **100**]

Annual hours	01-01-2017 to 01-01-2018
Hours to be credited	
> 01-01-2017	1248:00
Total	1248:00

28.4 Correction in case of long term sick leave

Stipulates that a correction is to be made, based on an employee's working hours, if he or she takes prolonged sick leave.

How to use

The labor-time per week equals [**Hours per week**] hours for a full-time contract. The labor-time per year equals [**Hours per year**] hours for a full-time contract. During long-term sick leave a complete registration is made for the [**first/final**] [**Accrual**] days of the sick leave on the cards below. For the remaining duration of the sick leave, registration is based on the actual hours worked.
Whenever sick leave starts within [**Relapse**] days of the end of a previous sick leave periods, both sick leave periods are considered to be one continuous sick leave period.

[**Target account** | **Target category** | **Percentage**]

- Hours per week is the number of hours in a full-time employee's contractual working week.
- Hours per year is the number of hours in a full-time employee's contractual working year.
- Accrual period defines the period in respect of which no correction is to be made.
- Relapse is the maximum number of days between two periods of sick leave consistent with them being treated as one.

The rule generates a booking once a week, from the date that the accrual period elapses. The booking involves a correction value equal to the employee's annual working hours minus the number of hours worked.
Where possible, it is preferable to use the 'Carry over and apply an expression' rule.

Example

An employee who is contracted to work thirty-six hours a week begins a period of sick leave on 10 February 2016.

The labor-time per week equals [**36**] hours for a full-time contract. The labor-time per year equals [**1872**] hours for a full-time contract. During long-term sick leave a complete registration is made for the [**first**] [**182**] days of the sick leave on the cards below. For the remaining duration of the sick leave, registration is based on the actual hours worked.
Whenever sick leave starts within [**30**] days of the end of a previous sick leave periods, both sick leave periods are considered to be one continuous sick leave period.

[**Holiday** | **Correction for sick leave** | **100**]

Holiday	01-01-2017 to 01-02-2017
Correction for sick leave	
> 02-01-2017	-35:54
> 09-02-2017	-35:54
> 16-01-2017	-35:54
> 23-01-2017	-35:54
> 30-01-2017	-35:54
Total	-179:30

29 Value based on annual hours of employment

The compensation rule in the group **Value based on annual hours of employment** can be used to calculate the a value based on the average working week of an employee.

29.1 Value based on annual hours of employment

Stipulates that a value is to be booked, based on an employee's working hours (working week), provided that the employee's salary code and age are as specified.

How to use

Employees whose salary classification is between [**Min salary code**] and [**Max salary code**] at the age between [**Min age**] and [**Max age**] will have the number of extra leave hours mentioned below. The calculation is based on the age reached at [**Moment**]

[**Target account** | **Target category** | **Percentage**]

- **Min salary code** is the lowest salary code that an employee may have for a booking to be triggered.
- **Max salary code** is the highest salary code that an employee may have for a booking to be triggered.
- **Min age** is the minimum age (in years) that an employee should have reached for a booking to be made.
- **Max age** is the maximum age (in years) that an employee may have reached for a booking to be made.
- **Moment** is the reference date [at the end of the year, at the end of the month] for determining the employee's age.

In the context of this rule, the booking date is the date on which the employee's employment conditions took effect, provided that this date is after 1 January of the year in question. Otherwise, the booking date is 1 January of the year in question.
An employee's age is determined from his/her date of birth. If an employee's age is not specified, this rule will not generate a booking.

The number of hour booked is equal to the average working week for the year.

Example

Employee's salary code:	C
Hours per week:	36
Employee's date of birth:	12-03-1965

Employees whose salary classification is between [**A**] and [**E**] at the age between [**18**] and [**65**] will have the number of extra leave hours mentioned below.
The calculation is based on the age reached at [**at the end of the year**]

[**Leave** | **Current year's entitlement** | **500**]

Leave	01-01-2017 to 01-01-2018
Current year's entitlement	
> 01-01-2017	36:00 / 500% / 180:00
Total	180:00

30 Variable entry for leave request per calender day

This rule creates an account entry equal to the number of hours of a partial leave request, begin- or end time not equal to 00:00 hours.

31 Waiting day in case of sick-leave

The compensation rule in the group **Waiting day in case of a sick leave** can be used to make a correction of the annual value in case of a long sick-leave.

31.1 Waiting day in case of sick-leave

Stipulates that the working time for a sick leave day is to be booked, if the illness type is relevant in relation to waiting days.

How to use

An elimination period of [**Number**] day(s) applies to sick leave. The first day of the elimination period is the first workday that the employee does not work or leaves early due to being sick. This rule applies only to types of sickness for which the elimination period applies. If an employee marks off sick again within [**Relapse**] days after being recovered, the elimination period is determined as of the first day of the original sick leave, assuming that the elimination period applies to both types of sick leave. Otherwise the elimination period is determined as of the first day the employee does not work as a result of the second mark off.
This rule sums entries for activities within the same shift when the following properties are equal: activity type, cost center, workstation.
For this rule, [**do/do not**] take the transition from daylight savings time to standard time into account.

[**Target account** | **Target category** | **Percentage**]

- **Number** is the number of waiting days to be booked.
- **Relapse** is the maximum number of days between two periods of sick leave consistent with them being treated as one.

The properties 'Relevant in relation to waiting days' and 'Relevant in relation to continuous sick leave' are assigned in the 'Illness type maintenance' window.

Example

An employee is scheduled to work Shift A on Monday 8 and Tuesday 9 May 2017. Shift A is made up of the following activities:

Activity type	Start	End
Work	08:00	12:00
Break	12:00	12:30
Work	12:30	16:00

The employee reports sick on 8 May 2017, with the result that both shifts are replaced by sick leave shifts. The illness type of the sick leave is relevant in relation to waiting days.

An elimination period of [**1**] day(s) applies to sick leave. The first day of the elimination period is the first workday that the employee does not work or leaves early due to being sick. This rule applies only to types of sickness for which the elimination period applies. If an employee marks off sick again within [**28**] days after being recovered, the elimination period is determined as of the first day of the original sick leave, assuming that the elimination period applies to both types of sick leave. Otherwise the elimination period is determined as of the first day the employee does not work as a result of the second mark off.
This rule sums entries for activities within the same shift when the following properties are equal: activity type, cost center, workstation.
For this rule, [**do not**] take the transition from daylight savings time to standard time into account.

[**Waiting days** | **Hours** | **100**]

Waiting days	01-01-2017 to 01-01-2018
Hours	
> 08-05-2017 A	07:30
Total	07:30

32 Working pattern

In the group **Working pattern** there is available only one compensation rule which is mainly used to give an illustration of the working pattern of an employee.

32.1 Repetition

Stipulates that the hours an employee should work according to its working pattern within a specific time interval are booked per day.

How to use

Enter the hours of the working pattern between [**Start time**] and [**End time**]

[**Target account** | **Target category** | **Percentage**]

- **Start time** and **End time** are start time and end time of the interval for which the hours of the employee its working pattern trigger a booking.

In order to make use of the rule "Repetition" it is necessary to define a working pattern of an employee. This can be done by an employee in the tab **Employment conditions** within the menu **Employee**, **Employee Management**.

Example

An employee is assigned to Working pattern W for the date 01-01-2017 to 01-01-2018. Working pattern W has a duration of one week and made up the following pattern:

Day of the week	From	Until
Monday	8:00	17:00
Wednesday	8:00	15:00

Enter the hours of the working pattern between [**08:00**] and [**16:00**]

[**Hours** | **Working pattern** | **100**]

Hours	03-04-2017 to 10-04-2017
Working pattern	
> 03-04-2017	07:00
> 05-04-2017	08:00
Total	15:00

33 Expressions

Expressions are used to create formula's to indicate what needs be entered on the indicated account. This section will introduce the expressions that are available in ORTEC Workforce Scheduling (ORTEC WS) within the payroll module and explains how this expressions can be used. In ORTEC WS the expressions have been organized in six categories:

- Operators
- Functions
- Date expressions
- Employee expressions
- Value expressions
- Other expressions

The expressions are only available at the compensation rules "Transferbooking based upon an expression" and "Carry over all account categories with expression".

33.1 Operators

Operators are expressions such as ‘add’ and ‘multiply’, which define the operations to be applied to the source value. Operators have a defined order of precedence, which determines which operator is applied first when several are included in the same expression; for example ‘multiply’ always precedes ‘add’. The order of precedence is as follows:

1. Not : unary operator
2. ^ : exponent operator
3. *,/, div, mod, and : multiplication operators
4. +, -, or, xor : summation operator
5. =, <>, <, >, <=, >= : relational operator

Use brackets – ‘(‘ and ‘)’ – if it will help to avoid confusion and try to align or lay out the expression where appropriate.

Example

Suppose that VALUE equals 1 and TOTALAMOUNT equals 2.

Operator	Expression	Value
NOT	IF(NOT(VALUE=1),1,2)	2
^	4^2	16
*	3*2	6
/	3/2	1.5
DIV	3 DIV 2	1.5
MOD	3 MOD 2	1
AND	IF((VALUE=1) AND (TOTALAMOUNT=2) ,1,2)	1
+	4+5	9
-	5-4	1
OR	IF((VALUE=1) OR (TOTALAMOUNT=1),1,2)	1
XOR	IF((VALUE=1) XOR (TOTALAMOUNT=2),1,2)	2
=	IF(VALUE=1,1,2)	1
<>	IF(VALUE<>1,1,2)	2
>	IF(VALUE>2,1,2)	2
>=	IF(VALUE>=1,1,2)	1
<	IF(VALUE<2,1,2)	1
<=	IF(VALUE<=1,1,2)	1

33.2 Functions

An expression may include the following standard functions: Abs, Arctan, Ceil, Cos, Exp, Frac, Int, Ln, Max, Min, Pi, Power, Round, Sin, Sqrt ,Sqr, Sum, Trunc.

Example

Function	Expression	Value
Abs	ABS(-2.5)	2.5
Arctan	ARCTAN(2.5)	1.19
Ceil	CEIL(1.3)	2.0
Cos	COS(2.5)	-0.80
Exp	EXP(1.0)	2.72
Frac	FRAC(2.6)	0.60
Int	INT(2.6)	2.00
Ln	LN(2.72)	1.00
Max	MAX(2.6,3.0)	3.00
Min	MIN(2.6,3.0)	2.60
Pi	PI	3.14
Power	POWER(2,5)	32.00
Round	ROUND(2.6)	3.00
Sin	SIN(2.5)	0.60
Sqrt	SQRT(16)	4.00
Sqr	SQR(4)	16.00
Sum	SUM(4, 5)	9.0
Trunc	TRUNC(2.6)	2.00

33.3 Date expressions

Date expressions are used to store information related to a certain date. Date expressions are generally used as a parameter within other expressions. For example the transferring of entries for

only the months June till August.

Generally the date of an entry is the date the entry is triggered. For example if an entry is triggered when an employee performs an certain activity, the date of the booking is the date and time the employee starts to work with the activity.

BEGINDATE

This expression yields the date of the period of which the bookings are made. It is generally used as parameter within another function.

How to use

BEGINDATE

This expression does not have a parameter and is mostly used as a parameter within other expressions.

Example

Suppose that account A has the following bookings:

Account A	02-01-2017 to 09-01-2017
1A	
> 02-01-2017	08:00
> 05-01-2017	06:00
Total	14:00

In the table below you can see the value transferred to account B by different use of the expression BEGINDATE.

Expression	Value booked
VALUE('1A', BEGINDATE, BEGINDATE+1)	08:00
VALUE('1A', BEGINDATE, ENDDATE)	14:00

In this example bookings are made per **week**. The week starts on Monday 2 January 2017.

DATEOF

The expression DATEOF yields the date from a date or datetime string.

How to use

DATEOF(<date>)

- **Date** is the string for which the date is determined.

The date in the input parameter is taken without converting the timezone.

Example

In the example, the expression DATEOF yields the following results:

Expression	Value
DATEOF(TODATETIME('2017-01-01 6:00','yyyy-mm-dd h:mm',' '))	42736:00
DATEOF(TODATETIME('2017-01-01 6:00','yyyy-mm-dd h:mm',' ')-0.5)	42735:00
DATEOF(42736)	42736:00

42736 is the number representation of 01-01-2017. In the second row, we subtract 12 hours (0.5 day) from the datetime so we receive the date 31-12-2016.

DATEOFBOOKING

This expression yields the date of the booking. It is generally used as parameter within another function.

How to use

DATEOFBOOKING

This expression does not have a parameter and is mostly used as a parameter within other expressions.

Example

Suppose that account A has the following bookings:

Account A	02-01-2017 to 09-01-2017
1A	
> 02-01-2017	08:00
> 05-01-2017	06:00
Total	14:00

In the table below you can see the value transferred to account B by different use of the expression DATEOFBOOKING.

Expression	Value booked
IF(DATEOFBOOKING=STRTODATE('02-01-2017'),VALUE('1A'),0)	08:00
IF(DATEOFBOOKING=STRTODATE('05-01-2017'),VALUE('1A'),0)	06:00

In this example bookings are made per **day**, **entry** or **entry day**.

DATEOFLASTBOOKING

This expression yields the last booking date of a certain period. It is generally used as parameter within another function.

How to use

DATEOFLASTBOOKING(['<account category>', <begin time>, <end time>])

- **Account category** is the category within the last entry made is determined
- **Begin time** and **End time** indicate the interval for which the last booking is determined.

If you use no parameters within this expression the expression takes the category or categories that are target to be transferred as default and the period for which the date of the last booking is determined depends on the period for which the entries are carried over to another account.

Example

Suppose that account A has the following bookings:

Account A	02-01-2017 to 09-01-2017
1A	
> 02-01-2017	08:00
> 05-01-2017	06:00
Total	14:00

The value of Category 1A is transferred **per week** to category 1B by use of the following expression.

IF(CEIL(DATEOFLASTBOOKING)-1=STRTODATE('05-01-2017'),1,0)

In this example the entries are transferred per week, so the date of the last entry is determined per week.

Account B	02-01-2017 to 09-01-2017
1B	
> 02-01-2017	01:00
Total	01:00

DATETIMEOFSHIFTBOOKING

This expression yields the date of the shifts that corresponds to the booking. It is generally used as parameter within another function.

How to use

DATETIMEOFSHIFTBOOKING

This expression does not have a parameter and is mostly used as a parameter within other expressions.

It is necessary to transfer the entries per **entry** or **entry day** in case you wish to return the date and time the shift start that corresponds to an entry.

Example

Suppose that account A has the following bookings:

Account A	02-01-2017 to 09-01-2017
1A	
> 02-01-2017	08:00
> 05-01-2017	06:00
Total	14:00

The value of Category 1A is transferred **per entry** to category 1B by use of the following expression.

IF(CEIL(DATETIMEOFSHIFTBOOKING)-1=STRTODATE('05-01-2017'),1,0)

Account B	02-01-2017 to 08-01-2017
1B	
> 05-01-2017	01:00
Total	01:00

DAY

The expression DAY can be used to count the number of days within a certain period.

How to use

DAY

This expression does not have a parameter and is mostly used in combination with other expressions.

The period for which the number of days is counted is the **From date** given in the account till the date of the concerning entry.

Example

Suppose that account A has the following bookings:

Account A	02-01-2017 to 09-01-2017
1A	
> 02-01-2017	08:00
> 05-01-2017	06:00
Total	14:00

The value of Category 1A is transferred **per entry** to Category 1B by use of the following expression.

DAY

Account B	02-01-2017 to 09-01-2017
1B	
> 02-01-2017	02:00
> 05-01-2017	05:00
Total	07:00

The expression returns the day number of the month.

DAYOFWEEK

The expression DAYOFWEEK yields the weekday for a certain date.

How to use

DAYOFWEEK(<date>)

- **Date** is the date for which the weekday is determined.

The day of the week that is seen as the first day of the week can be changed within the settings of ORTEC WS.

Example

Suppose that in this example a week starts on a Monday and that 01-05-2017 is a Monday, than the expression DAYOFWEEK yields the following results:

Expression	Value
DAYOFWEEK(STRTODATE('01-05-2017'))	01:00
DAYOFWEEK(STRTODATE('04-05-2017'))	04:00
DAYOFWEEK(STRTODATE('09-05-2017'))	02:00

ENDDATE

This expression yields the date of the period of which the bookings are made. It is generally used as parameter within another function.

How to use

ENDDATE

In order the gain the end date of the week related to the entry it is necessary to transfer entries **per week**.

Example

Suppose that account A has the following bookings:

Account A	02-01-2017 to 09-01-2017
1A	
> 02-01-2017	08:00
> 08-01-2017	06:00
Total	14:00

In order to transfer only the entries of the last day of the week, the following expression is used by the transfer **per week**of Category 1A to Category 1B.

VALUE('1A', ENDDATE-1, ENDDATE)

Account B	02-01-2017 to 09-01-2017
1B	
> 02-01-2017	06:00
Total	06:00

ENDDATEOFDEPARTMENTPERIOD

The expression ENDDATEOFDEPARTMENTPERIOD yields the date a department period ends.

How to use

ENDDATEOFDEPARTMENTPERIOD('<PeriodType>', [<date>])

This expression has a parameter to determine the department period type.

Example

Suppose that account A has the following bookings:

Account A	02-01-2017 to 30-01-2017
1A	
> 02-01-2017	08:00
> 29-01-2017	06:00
Total	14:00

The period type 'Four-weeks Calendar' is defined as department period and has the following periods:

Name	Four-weeks Calendar
2017 01-04	02-01-2017 - 30-01-2017
2017 05-08	30-01-2017 - 27-02-2017
2017 09-12	27-02-2017 - 27-03-2017

In order to transfer only the entries of the last day of the department period, the following expression is used by the transfer **per day**of Category 1A to Category 1B.

IF(DATEOFBOOKING=ENDDATEOFDEPARTMENTPERIOD('Four-weeks Calendar'),VALUE('1A'),0)

Account B	02-01-2017 to 30-01-2017
1B	
> 29-01-2017	06:00
Total	06:00

ENDDATEOFPREVIOUSDEPARTMENTPERIOD

The expression ENDDATEOFPREVIOUSDEPARTMENTPERIOD yields the date the previous department period ends.

How to use

ENDDATEOFPREVIOUSDEPARTMENTPERIOD('<PeriodType>', [<date>])

This expression has a parameter to determine the department period type.

Example

Suppose that account A has the following bookings:

Account A	30-01-2017 to 28-02-2017
1A	
> 30-01-2017	08:00
> 27-02-2017	06:00
Total	14:00

The period type 'Four-weeks calendar' is defined as department period and has the following periods:

Name	Four-weeks Calendar
2017 01-04	02-01-2017 - 30-01-2017
2017 05-08	30-01-2017 - 27-02-2017
2017 09-12	27-02-2017 - 27-03-2017

In order to transfer all entries of in the department period, the following expression is used by the transfer **per period type Accounting Calendar**of Category 1A to Category 1B.

VALUE('1A', ENDDATEOFPREVIOUSDEPARTMENTPERIOD('Four-weeks Calendar'), ENDDATEOFDEPARTMENTPERIOD('Four-weeks Calendar'))

Account B	30-01-2017 to 28-02-2017
1B	
> 30-01-2017	14:00
Total	14:00

FIRSTDAYOFDEPARTMENTPERIOD

The expression FIRSTDAYOFDEPARTMENTPERIOD yields the date a department period starts.

How to use

FIRSTDAYOFDEPARTMENTPERIOD('<PeriodType>', [<date>])

This expression has a parameter to determine the department period type.

Example

Suppose that account A has the following bookings:

Account A	02-01-2017 to 30-01-2017
1A	
> 02-01-2017	08:00
> 29-01-2017	06:00
Total	14:00

The period type 'Four-weeks Calendar' is defined as department period and has the following periods:

Name	Four-weeks Calendar
2017 01-04	02-01-2017 - 30-01-2017
2017 05-08	30-01-2017 - 27-02-2017
2017 09-12	27-02-2017 - 27-03-2017

In order to transfer only the entries of the first day of the department periods, the following expression is used by the transfer **per day**of Category 1A to Category 1B.

IF(DATEOFBOOKING=FIRSTDAYOFDEPARTMENTPERIOD('Four-weeks Calendar'),VALUE('1A'),0)

Account B	02-01-2017 to 30-01-2017
1B	
> 02-01-2017	08:00
Total	08:00

FIRSTDAYOFLASTMONTH

Determines the day of the first day of the month previous to the month of the booking.

How to use

FIRSTDAYOFLASTMONTH(['<date>'])

- **Date** is the date of which the first day of the last month should be determined.

Generally the expression is used without the date parameter. In that case the expression determines the day of the first day of the month previous to the month of the booking.

Example

In the example, the booking is made on 02-05-2017.

Expression	Value booked
DATEOFBOOKING-FIRSTDAYOFLASTMONTH	31:00

FIRSTDAYOFLASTYEAR

Determines the day-number of the first day of the year previous to the year of the booking.

How to use

FIRSTDAYOFLASTYEAR(['<date>'])

- **Date** is the date for which the first day of the last year should be determined.

Generally the expression is used without the date parameter. In that case the expression determines the day-number of the first day of the year previous to the year of the booking.

Example

In the example, the booking is made on 02-01-2005.

Expression	Value booked
DATEOFBOOKING-FIRSTDAYOFLASTYEAR	366:00

FIRSTDAYOFLASTSALARYPERIOD

The expression FIRSTDAYOFLASTSALARYPERIOD yields the date a salary period starts.

How to use

FIRSTDAYOFLASTSALARYPERIOD

Example

The salary periods defined are:

Name	Period
01 2017	02-01-2017 - 30-01-2017
02 2017	30-01-2017 - 27-02-2017
03 2017	27-02-2017 - 03-04-2017

The following expression is used by the transfer **per day**.In the example, the booking is made on 01-02-2017.

Expression	Value booked
DATEOFBOOKING-FIRSTDAYOFLASTSALARYPERIOD	30:00

FIRSTDAYOFLASTYEAR

Determines the day-number of the first day of the year prior to the year of the booking.

How to use

FIRSTDAYOFLASTYEAR(['<date>'])

- **Date** is the date for which the first day of the month should be determined.

Generally the expression is used without the date parameter.

Example

In the example, the booking is made on 02-05-2017.

Expression	Value booked
DATEOFBOOKING-FIRSTDAYOFLASTYEAR	487:00

FIRSTDAYOFMONTH

Determines the day-number of the first day of month period prior to the month of the booking.

How to use

FIRSTDAYOFMONTH(['<date>'])

- **Date** is the date for which the first day of the month should be determined.

Generally the expression is used without the date parameter.

Example

In the example, the booking is made on 02-05-2017.

Expression	Value booked
DATEOFBOOKING-FIRSTDAYOFMONTH	01:00

FIRSTDAYOFNEXTDEPARTMENTPERIOD

The expression FIRSTDAYOFNEXTDEPARTMENTPERIOD yields the date a department period starts.

How to use

FIRSTDAYOFNEXTDEPARTMENTPERIOD('<PeriodType>', [<date>])

This expression has a parameter to determine the department period type.

Example

Suppose that account A has the following bookings:

Account A	02-01-2017 to 30-01-2017
1A	
> 02-01-2017	08:00
> 29-01-2017	06:00
Total	14:00

The period type 'Four-weeks Calendar' is defined as department period and has the following periods:

Name	Four-weeks Calendar
2017 01-04	02-01-2017 - 30-01-2017
2017 05-08	30-01-2017 - 27-02-2017
2017 09-12	27-02-2017 - 27-03-2017

In order to transfer all the entries on the first day of the department periods, the following expression is used by the transfer **per department period Four-weeks Calendar**of Category 1A to Category 1B.

VALUE('1A', FIRSTDAYOFDEPARTMENTPERIOD('Four-weeks Calendar'), FIRSTDAYOFNEXTDEPARTMENTPERIOD('Four-weeks Calendar'))

Account B	02-01-2017 to 30-01-2017
1B	
> 02-01-2017	14:00
Total	14:00

FIRSTDAYOFNEXTMONTH

Determines the day-number of the first day of the month following the month of the booking.

How to use

FIRSTDAYOFNEXTMONTH(['<date>'])

- **Date** is the date for which the first day of the following month should be determined.

Generally the expression is used without the date parameter. In that case the expression determines the day-number of the first day of the month following the month of the booking.

Example

In the example, the booking is made on 02-05-2017.

Expression	Value booked
FIRSTDAYOFNEXTMONTH-DATEOFBOOKING	30:00

FIRSTDAYOFNEXTQUARTER

Determines the day-number of the first day of the quarter following the quarter of the booking.

How to use

FIRSTDAYOFNEXTQUARTER(['<date>'])

- **Date** is the date for which the first day of the following quarter should be determined.

Generally the expression is used without the date parameter. In that case the expression determines the day-number of the first day of the quarter following the quarter of the booking.

Example

In the example, the booking is made on 02-05-2017.

Expression	Value booked
FIRSTDAYOFNEXTQUARTER-DATEOFBOOKING	60:00

FIRSTDAYOFNEXTSALARYPERIOD

The expression FIRSTDAYOFNEXTSALARYPERIOD yields the date the next salary period starts.

How to use

FIRSTDAYOFNEXTSALARYPERIOD

Example

Suppose that account A has the following bookings:

Account A	02-01-2017 to 30-01-2017
1A	
> 01-01-2017	08:00
> 29-01-2017	06:00
Total	14:00

The salary periods defined are:

Name	Period
01 2017	02-01-2017 - 30-01-2017
02 2017	30-01-2017 - 27-02-2017
03 2017	27-02-2017 - 03-04-2017

In order to transfer all entries of the current salary period, the following expression is used by the transfer **per salary period**of Category 1A to Category 1B.

VALUE('1A',FIRSTDAYOFSALARYPERIOD,FIRSTDAYOFNEXTSALARYPERIOD)

Account B	02-01-2017 to 30-01-2017
1B	
> 02-01-2017	14:00
Total	14:00

FIRSTDAYOFNEXTYEAR

Determines the day-number of the first day of the year following the year of the booking.

How to use

FIRSTDAYOFNEXTYEAR(['<date>'])

- **Date** is the date for which the first day of the following year should be determined.

Generally the expression is used without the date parameter. In that case the expression determines the day-number of the first day of the year following the year of the booking.

Example

In the example, the booking is made on 01-01-2005 (which is the first day of the booking period).

Expression	Value booked
FIRSTDAYOFNEXTYEAR-DATEOFBOOKING	365:00

FIRSTDAYOFPREVIOUSQUARTER

Determines the day-number of the first day of the quarter before the quarter of the booking.

How to use

FIRSTDAYOFPREVIOUSQUARTER(['<date>'])

- **Date** is the date for which the first day of the previous quarter should be determined.

Generally the expression is used without the date parameter. In that case the expression determines the day-number of the first day of the quarter before the quarter of the booking.

Example

In the example, the booking is made on 02-05-2017.

Expression	Value booked
DATEOFBOOKING-FIRSTDAYOFPREVIOUSQUARTER	121

FIRSTDAYOFQUARTER

Determines the day-number of the first day of the quarter of the booking.

How to use

FIRSTDAYOFQUARTER(['<date>'])

- **Date** is the date for which the first day of the quarter should be determined.

Generally the expression is used without the date parameter. In that case the expression determines the day-number of the first day of the quarter of the booking.

Example

In the example, the booking is made on 02-05-2017.

Expression	Value booked
DATEOFBOOKING-FIRSTDAYOFQUARTER	31

FIRSTDAYOFSALARYPERIOD

The expression FIRSTDAYOFSALARYPERIOD yields the date the salary period starts.

How to use

FIRSTDAYOFSALARYPERIOD

Example

Suppose that account A has the following bookings:

Account A	02-01-2017 to 30-01-2017
1A	
> 02-01-2017	08:00
> 29-01-2017	06:00
Total	14:00

The salary periods defined are:

Name	Period
01 2017	02-01-2017 - 30-01-2017
02 2017	30-01-2017 - 27-02-2017
03 2017	27-02-2017 - 03-04-2017

In order to transfer all entries of the current salary period, the following expression is used by the transfer **per salary period**of Category 1A to Category 1B.

VALUE('1A',FIRSTDAYOFSALARYPERIOD,FIRSTDAYOFNEXTSALARYPERIOD)

Account B	02-01-2017 to 30-01-2017
1B	
> 02-01-2017	14:00
Total	14:00

FIRSTDAYOFWEEK

Determines the day-number of the first day of the week of the booking.

How to use

FIRSTDAYOFWEEK(['<date>'])

- **Date** is the date for which the first day of the week should be determined.

Generally the expression is used without the date parameter. In that case the expression determines the day-number of the first day of the week of the booking.

Example

In the example, the booking is made on 02-04-2017.

Expression	Value booked
DATEOFBOOKING-FIRSTDAYOFWEEK	6:00

FIRSTDAYOFYEAR

Determines the day-number of the first day of the year within which the booking date falls.

How to use

FIRSTDAYOFYEAR['<date>']

- **Date** is the date for which the first day of the week should be determined.

Generally the expression is used without the date parameter. In that case the expression determines the day-number of the first day of the year of the booking.

Example

In the example, the booking is made on 02-05-2017.

Expression	Value booked
DATEOFBOOKING-FIRSTDAYOFYEAR	121

ISDAYNUMBER

Returns a boolean to verify if a certain date is on a Monday, Tuesday,... or Sunday.

How to use

ISDAYNUMBER(<date>, '<daynumbers>')

- **Date** is the date for which the daynumber is checked.
- **Daynumber** is the number that corresponds to the day of the week for the given date. It is possible to set multiple daynumbers, separated by a ' ' (space) character.

In case Monday is the first day of the week, the daynumbers are Monday = 1, Tuesday = 2, Wednesday = 3, Thursday = 4, Friday = 5, Saturday = 6, Sunday = 7.

Example

In the example, the booking is made on 04-04-2017, which is on a Tuesday. Further the first day of the week is set on Monday.

Expression	Value booked
IF(ISDAYNUMBER(STRTODATE('04-04-2017'),'1'),1,0)	0:00
IF(ISDAYNUMBER(STRTODATE('04-04-2017'),'2'),1,0)	1:00
IF(ISDAYNUMBER(STRTODATE('04-04-2017'),'2 5'),1,0)	1:00

ISINLAST7DAYS

Returns a boolean to verify if a certain date is within 7 days of today.

How to use

ISINLAST7DAYS(<date>)

- **Date** is the date for which the expression check if it is within 7 days before today.

Example

In the example, we assume that Account A has exactly one booking per day for the month July 2017. In case the it is today 29-07-2017, the transferring of Account A to Account B **per entry** with use of the expression below gives the following result:

IF(ISINLAST7DAYS(DATEOFBOOKING),1,0)

Account B	01-07-2017 to 01-08-2017
1B	
> 22-07-2017	01:00
> 23-07-2017	01:00
> 24-07-2017	01:00
> 25-07-2017	01:00
> 26-07-2017	01:00
> 27-07-2017	01:00
> 28-07-2017	01:00
> 29-07-2017	01:00
Total	08:00

ISINNEXT7DAYS

Returns a boolean to verify if a certain date is within 7 days after today.

How to use

ISINNEXT7DAYS(<date>)

- **Date** is the date for which the expression check if it is within 7 days after today.

Example

In the example, we assume that Account A has exactly one booking per day for the months July and August 2017. In case the it is today 29-07-2017, the transferring of Account A to Account B **per entry** with use of the expression below gives the following result:

IF(ISINNEXT7DAYS(DATEOFBOOKING),1,0)

Account B	01-07-2017 to 01-09-2017
1B	
> 29-07-2017	01:00
> 30-07-2017	01:00
> 31-07-2017	01:00
> 01-08-2017	01:00
> 02-08-2017	01:00
> 03-08-2017	01:00
> 04-08-2017	01:00
> 05-08-2017	01:00
Total	08:00

ISINNEXTDAYSONDAYNUMBER

Returns a boolean to verify if a certain date is within the indicated days after today including only the days that contains the indicated daynumber.

How to use

ISINNEXTDAYSONDAYNUMBER(<date>, <number>, '<daynumbers>')

- **Date** is the date for which the expression check if it is within the indicated number of days after today.
- **Number** is the number of days that has the indicated daynumber for which the expression checked if it is equal to the given date.
- **Daynumbers** are the numbers corresponding to the days of the week that should be included within the next number of days.

Example

In the example, we assume that Account A has exactly one booking per day for the months July and August 2017. In case the it is today 29-07-2017 (Saturday), the transferring of Account A to Account B **per entry** with use of the expression below gives the following result:

IF(ISINNEXTDAYSONDAYNUMBER(DATEOFBOOKING, 7, '1 2'),1,0)

Account B	01-07-2017 to 01-09-2017
1B	
> 31-07-2017	01:00
> 01-08-2017	01:00
> 07-08-2017	01:00
> 08-08-2017	01:00
> 14-08-2017	01:00
> 15-08-2017	01:00
> 21-08-2017	01:00
Total	07:00

ISLASTMONTH

Returns a boolean to verify if the given date is within last month.

How to use

ISLASTMONTH(<date>)

- **Date** is the date for which the expression check if it is within the previous month.

The previous month depends on today.

Example

In the example, we assume that today is 29-07-2017.

Expression	Value booked
IF(ISLASTMONTH(STRTODATE('04-05-2017')),1,0)	0:00
IF(ISLASTMONTH(STRTODATE('08-06-2017')),1,0)	1:00
IF(ISLASTMONTH(STRTODATE('03-07-2017')),1,0)	0:00

ISLASTWEEK

Returns a boolean to verify if the given date is within last week.

How to use

ISLASTWEEK(<date>)

- **Date** is the date for which the expression check if it is within the previous week.

  The previous week depends on today.

Example

In the example, we assume that today is 29-07-2017 (Saturday).

Expression	Value booked
IF(ISLASTWEEK(STRTODATE('20-07-2017')),1,0)	1:00
IF(ISLASTWEEK(STRTODATE('27-07-2017')),1,0)	0:00
IF(ISLASTWEEK(STRTODATE('03-08-2017')),1,0)	0:00

ISNEXTMONTH

Returns a boolean to verify if the given date is within the next month.

How to use

ISNEXTMONTH(<date>)

- **Date** is the date for which the expression check if it is within the following month.

  The next month depends on today.

Example

In the example, we assume that today is 29-07-2017.

Expression	Value booked
IF(ISNEXTMONTH(STRTODATE('04-06-2017')),1,0)	0:00
IF(ISNEXTMONTH(STRTODATE('08-07-2017')),1,0)	0:00
IF(ISNEXTMONTH(STRTODATE('03-08-2017')),1,0)	1:00

ISNEXTWEEK

Returns a boolean to verify if the given date is within the next week.

How to use

ISNEXTWEEK(<date>)

- **Date** is the date for which the expression check if it is within the following week.

  The next week depends on today.

Example

In the example, we assume that today is 29-07-2017 (Saturday).

Expression	Value booked
IF(ISNEXTWEEK(STRTODATE('20-07-2017')),1,0)	0:00
IF(ISNEXTWEEK(STRTODATE('30-07-2017')),1,0)	0:00
IF(ISNEXTWEEK(STRTODATE('04-08-2017')),1,0)	1:00

ISTHISMONTH

Returns a boolean to verify if the given date is within the current month.

How to use

ISTHISMONTH(<date>)

- **Date** is the date for which the expression check if it is within the current month.

> The current month depends on today.

Example

In the example, we assume that today is 29-07-2017.

Expression	Value booked
IF(ISTHISMONTH(STRTODATE('20-06-2017')),1,0)	0:00
IF(ISTHISMONTH(STRTODATE('15-07-2017')),1,0)	1:00
IF(ISTHISMONTH(STRTODATE('08-08-2017')),1,0)	0:00

ISTHISWEEK

Returns a boolean to verify if the given date is within the current week.

How to use

ISTHISWEEK(<date>)

- **Date** is the date for which the expression check if it is within the current week.

> The current week depends on today.

Example

In the example, we assume that today is 29-07-2013 (Saturday).

Expression	Value booked
IF(ISTHISWEEK(STRTODATE('20-07-2017')),1,0)	0:00
IF(ISTHISWEEK(STRTODATE('30-07-2017')),1,0)	1:00
IF(ISTHISWEEK(STRTODATE('08-08-2017')),1,0)	0:00

ISTODAY

Returns a boolean to verify if the given date is today.

How to use

ISTODAY(<date>)

- **Date** is the date for which the expression check if it is today.

Example

In the example, we assume that today is 29-07-2017.

Expression	Value booked
IF(ISTODAY(STRTODATE('28-07-2017')),1,0)	0:00
IF(ISTODAY(STRTODATE('29-07-2017')),1,0)	1:00
IF(ISTODAY(STRTODATE('30-07-2017')),1,0)	0:00

ISTOMORROW

Returns a boolean to verify if the given date is tomorrow.

How to use

ISTOMORROW(<date>)

- **Date** is the date for which the expression check if it is the day following today.

Example

In the example, we assume that today is 29-07-2017.

Expression	Value booked
IF(ISTOMORROW(STRTODATE('29-07-2017')),1,0)	0:00
IF(ISTOMORROW(STRTODATE('30-07-2017')),1,0)	1:00

ISYESTERDAY

Returns a boolean to verify if the given date is yesterday.

How to use

ISYESTERDAY(<date>)

- **Date** is the date for which the expression check if it is the day before today.

Example

In the example, we assume that today is 29-07-2017.

Expression	Value booked
IF(ISYESTERDAY(STRTODATE('28-07-2017')),1,0)	1:00
IF(ISYESTERDAY(STRTODATE('29-07-2017')),1,0)	0:00

MONTH

Returns the month of the year, for instance on February the 15th this expression results the value 2.

How to use

MONTH

The expression MONTH has no parameter and is generally used as a parameter within other expressions.

The Month numbers are: January = 1, February = 2, March = 3, April = 4, May = 5, June = 6, July = 7, August = 8, September = 9, October = 10, November = 11 and December = 12.

Example

In the example, we assume that account A has the following bookings.

Account A	01-01-2017 to 01-01-2018
1A	
> 24-03-2017	08:00
> 18-07-2017	08:00
Total	16:00

Now you wish to count the entries made in the months July and August for 1.5 times.

To do this you can transfer the entries of category 1A **per entry** to Account B, category 1B, by use of the following expression.

IF((MONTH = 6) or (MONTH = 7),1.5*VALUE('1A'),VALUE('1A'))

Account B	01-01-2017 to 01-01-2018
1B	
> 24-03-2017	08:00
> 18-07-2017	12:00
Total	20:00

MONTHOFYEAR

Returns the number related to the month of the given date, for instance if the given date is April the 16th, this expression returns the value 4.

How to use

MONTHOFYEAR(<date>)

- **Date** is the date for which the expression returns the month number.

This expression looks like the expression MONTH. The main difference is that this expression needs a date as parameter.

The Month numbers are: January = 1, February = 2, March = 3, April = 4, May = 5, June = 6, July = 7, August = 8, September = 9, October = 10, November = 11 and December = 12.

Example

In the example, we assume that account A has the following bookings.

Account A	01-06-2017 to 01-08-2017
1A	
> 24-06-2017	08:00
> 18-07-2017	08:00
Total	16:00

Now the entries of category 1A are transferred **per entry** to Account B, category 1B, by use of the following expression.

IF(MONTHOFYEAR(DATEOFBOOKING)=7,VALUE('A1')*0.5,VALUE('A1'))

Account B	01-06-2017 to 01-08-2017
1A	
> 24-06-2017	08:00
> 18-07-2017	04:00
Total	12:00

NOW

The expression NOW returns the current date and time within ORTEC WS.

How to use

NOW

The expression NOW has no parameter and is mostly used in combination with other expressions.

Example

Assume that the current time is 20:00 and the current date is 30-07-2017.

Expression	Value (in dec.)
NOW	41485.83
NOW - TODAY	0.83

In ORTEC WS the time is represent as a factor of 1/24. This means that in ORTEC WS the value in decimals of one day (or in other words 24 hours) is shown as 1.00 and 20 hours is shown as 20/24 = 0.83.

SHIFTDATE

Indicates the start date of the shift.

How to use

SHIFTDATE

This expression does not have a parameter and is mostly used as a parameter within other expressions.

Example

In the example below the shift has start date 02-01-2017.

Expression	Value booked
SHIFTDATE-FIRSTDAYOFYEAR	1

RAWDATEOF

The expression RAWDATEOF yields the date from a date or datetime string and includes a correction of the timezone to UTC.

How to use

RAWDATEOF(<date>)

- **Date** is the string for which the date is determined.

The date in the input parameter is converted from the server timezone to UTC.

Example

In the example the server timezone is UTC+1. The expression RAWDATEOF yields the following results:

Expression	Value
RAWDATEOF(TODATETIME('2017-01-01 1:00','yyyy-mm-dd h:mm','))	42736:00
RAWDATEOF(TODATETIME('2017-01-01 0:00','yyyy-mm-dd h:mm','))	42735:00
RAWDATEOF(TODATETIME('2017-01-01 7:00','yyyy-mm-dd h:mm',')-0.25)	42736:00

42736 is the number representation of 01-01-2017.

RAWTIMEOF

The expression RAWTIMEOF yields the time from a datetime string and includes a correction of the timezone to UTC.

How to use

RAWTIMEOF(<datetime>)

- **Datetime** is the string for which the time is determined.

The datetime in the input parameter is converted from the server timezone to UTC.

Example

In the example the server timezone is UTC+1. The expression RAWTIMEOF yields the following results:

Expression	Value
RAWTIMEOF(TODATETIME('2017-01-01 6:00', 'yyyy-mm-dd h:mm', ' '))	0:13
RAWTIMEOF(TODATETIME('2017-01-01 7:00', 'yyyy-mm-dd h:mm', ' '))	0:15
RAWTIMEOF(TODATETIME('2017-01-01 13:00', 'yyyy-mm-dd h:mm', ' '))	0:30

The time in decimals is shown as 1/24 day. 7:00 is first converted to 6:00 UTC, which is 0:15 day or 0.25 in decimal values.

STRTODATE

Can be used to specify a date.

How to use

STRTODATE('dd-MM-yyyy')

In ORTEC WS a date is not indicated in the form 'dd-MM-yyyy' but is indicated as a number.

Example

In the example below a booking of value 08:00 is booked in category A1 on 01-01-2017.

Expression	Value booked
STRTODATE('01-01-2017')	42736:00
IF(DATEOFBOOKING=STRTODATE('01-01-2017'),VALUE('A1'),0)	08:00

TIMEOF

The expression TIMEOF yields the time from a datetime string.

How to use

RAWTIMEOF(<datetime>)

- **Datetime** is the string for which the time is determined.

The datetime in the input parameter is taken without converting the timezone.

Example

In the example, the expression TIMEOF yields the following results:

Expression	Value
TIMEOF(TODATETIME('2017-01-01 6:00', 'yyyy-mm-dd h:mm', ' '))	0:15
TIMEOF(TODATETIME('2017-01-01 6:00', 'yyyy-mm-dd h:mm', ' ')-0.5)	0:45
TIMEOF(TODATETIME('2017-01-01 12:00', 'yyyy-mm-dd h:mm', ' '))	0:30

The time in decimals is shown as 1/24 day. 6:00 is 0:15 day or 0.25 in decimal values.

TODATE

Can be used to specify a date for a chosen format.

How to use

TODATE('<date string>', '<format string>')

- **Date string** is the date string which the expression should specify as a number.
- **Date format** is the format of the indicated date string.

> In ORTEC WS a date is not indicated in the form of for example 'dd-mm-yyyy' but is indicated as a number.
>
> The date formats you can choose from are: 'dd-mm-yyyy', 'mm-dd-yyyy' or 'yyyy-mm-dd'. Instead of '-' also '/' can be used.

> **Example**
> In the example below TODATE is used to specify the date of March the 1th of 20173 in the three different date formats.

Expression	Value
TODATE('01-03-20173','dd-mm-yyyy')	427951334:00
TODATE('03-01-20137','mm-dd-yyyy')	427951334:00
TODATE('20173-03-01','yyyy-mm-dd')	427951334:00

TODATETIME

Can be used to specify a date and time for a chosen format.

How to use

TODATETIME('<date time string>', '<date time format string>', '<date time separator>')

- **Date time string** is the datetime string which the expression should specify as a number.
- **Date time format string** is the format of the indicated datetime string.
- **Date time separator** is the separator between the date and the time in the string

> In ORTEC WS a date is not indicated in the form of for example 'dd-mm-yyyy' but is indicated as a number.
>
> The date formats you can choose from are: 'dd-mm-yyyy', 'mm-dd-yyyy' or 'yyyy-mm-dd'. Instead of '-' also '/' can be used.
> The time formats you can choose form are: 'h:mm', 'hh:mm' or 'hh:mm:ss'.

> This expressions takes the server timezone into account and converts this back to UTC.

Example

In the example below TODATETIME is used to specify the date and time of March the 1th of 20176:00:00 in the three different datetime formats. The server timezone is UTC+0.

Expression	Value
TODATE('01-03-2017 6:00','dd-mm-yyyy h:mm',' ')	42795:15
TODATE('03-01-2017+06:00','mm-dd-yyyy+hh:mm','+')	42795:15
TODATE('2017-03-01 06:00:00','yyyy-mm-dd hh:mm:ss',' ')	42795:15

TODAY

Can be used to specify a date.

How to use

TODAY

This expression does not have a parameter and is mostly used as a parameter within other expressions.

Example

In the example below, the booking date is 02-05-2017 and the current date is 25-08-2017.

Expression	Value booked
TODAY-DATEOFBOOKING	115
TODAY-FIRSTDAYOFYEAR	236

TOTIME

Can be used to specify a time for a chosen format.

How to use

TOTIME('<time string>', '<format string>')

- **Time string** is the time string which the expression should specify as a number.
- **Format string** is the format of the indicated time string.

In ORTEC WS a time is not indicated in the form of for example 'hh:mm:ss' but is indicated as a number. One hour is 1/24.

The time formats you can choose form are: 'h:mm', 'hh:mm' or 'hh:mm:ss'.

Example

In the example below TOTIME is used to specify the time 6:00:00 in the three different time formats.

Expression	Value
TOTIME('6:00', 'h:mm')	0:15
TOTIME('06:00', 'hh:mm')	0:15
TOTIME('06:00:00', 'hh:mm:ss')	0:15

WEEKNUMBER

Returns the week number of the year for the indicated date.

How to use

WEEKNUMBER(<date>)

- **Date** is the date for which the expression determines the week number.

The ISO week number format is used to determine the week number.

Example

In the example the expression WEEKNUMBER is used for different dates within the year 2017.

Expression	Value
WEEKNUMBER(STRTODATE('01-01-2017'))	52:00
WEEKNUMBER(STRTODATE('24-06-2017'))	25:00
WEEKNUMBER(STRTODATE('31-12-2017'))	52:00

In the year 2017 the day 02-01-2017is the day in week 1.

YEAR

Returns the year of a certain date. For instance on January 15th of 2017 this expression returns the value 2017.

How to use

YEAR

This expression does not have a parameter and is mostly used as a parameter within other expressions.

Example

In the example, we assume that account A has the following bookings.

Account A	01-01-2017 to 01-01-2019
1A	
> 24-06-2017	08:00
> 18-07-2018	08:00
Total	16:00

Now you wish to count the entries made after the year 2012 for 1.5 times.

To do this you can transfer the entries of category 1A **per entry** to Account B, category 1B, by use of the following expression.

IF(YEAR>2017,1.5*VALUE('1A'),VALUE('1A'))

Account B	01-01-2017 to 01-01-2019
1B	
> 24-06-2017	08:00
> 18-07-2018	12:00
Total	20:00

YEARSBETWEEN

Gives the number of years between the booking date and the set number of days in the past.

How to use

YEARSBETWEEN(<Number of days>)

- **Number of days** is the number of days in the past for which the number of years is determined from the booking date.

Example

In the example, the booking date is 02-05-2017. 2016 is a leap year.

Expression	Value booked
YEARSBETWEEN(365)	1
YEARSBETWEEN(730)	1
YEARSBETWEEN(731)	2

33.4 Employee expressions

Employee expressions are expressions that return information related to an employee. Examples are the age of an employee, the salary code of an employee or the departments an employee is available to work at on a certain date.

AVAILABILITYFORALLDEPARTMENTS

Yields the total number of hours an employee is on average contracted to work for all department for a given time interval.

How to use

AVAILABILITYFORALLDEPARTMENTS([<From_Date>, <To_Date>])

- **From date** and **To date** identify the period for which the departments' availability should be calculated.

In case the **From data** and **To date** are the same date (so the interval length is zero) the availability for all departments is calculated for one day, namely for the **From date** (**To date**).

Example

For the period 01-03-2017 to 01-04-2017 (one month) an employee is contracted to work 20 hours at department A. For the period 01-04-2017 to 01-05-2017 (one month) the employee is contracted to work 20 hours at department A and 10 hours at department B.

Expression	Value booked
AVAILABILITYFORALLDEPARTMENTS(STRTODATE('01-03-2017'), STRTODATE('01-04-2017'))	20
AVAILABILITYFORALLDEPARTMENTS(STRTODATE('01-04-2017'), STRTODATE('01-05-2017'))	30
AVAILABILITYFORALLDEPARTMENTS(STRTODATE('01-03-2017'), STRTODATE('01-05-2017'))	24.918

The value of the last expression is calculated as (20 * 31 (days in March) + 30 * 30 (days in April)) / 61 (days in March and April) = 24.918

AVAILABILITYFORDEPARTMENT

Yields the number of hours an employee is on average contracted to work for adepartment the employee is assigned to as it longest for a given time interval. If an employee is assigned to multiple departments at the same time, the highest availability of the departments is taken.

How to use

AVAILABILITYFORDEPARTMENT([<From_Date>, <To_Date>])

- **From date** and **To date** identify the period for which the departments' availability should be calculated.

In case the **From data** and **To date** are the same date (so the interval length is zero) the availability of a department is calculated for one day, namely for the **From date** (**To date**).

Example

For the period 01-03-2017 to 01-04-2017 (one month) an employee is contracted to work 20 hours at department A. For the period 01-04-2017 to 01-05-2017 (one month) the employee is contracted to work 10 hours at department A and 10 hours at department B.

Expression	Value booked
AVAILABILITYFORDEPARTMENT(STRTODATE('01-03-2017'), STRTODATE('01-04-2017'))	20
AVAILABILITYFORDEPARTMENT(STRTODATE('01-04-2017'), STRTODATE('01-05-2017'))	10
AVAILABILITYFORDEPARTMENT(STRTODATE('01-04-2017'), STRTODATE('01-05-2017'))	15.082

The value of the last expression is calculated as (20 * 31 (days in March) + 10 * 30 (days in April)) / 61 (days in March and April) =15.082.

CONTRACTDAYSININTERVAL

Yields the number of days on which the employee is contracted to work at least a minimum number of hours for a given time interval.

How to use

CONTRACTDAYSININTERVAL(<begin time>, <end time>, <min_contractduration>, <bool_CountWeekendDays>, <bool_CountHolidays>)

- **Begin time** and **End time** are the first and last day of the time interval for which the number of contract days is counted.
- **Min contractduration** is the minimum number of hours an employee is contracted to work before a day is counted as a contract day.
- **Bool_CountWeekendDays** boolean that takes the value 'true' or 'false'. If it takes the value 'true' all days of the week are included, if 'false' only the weekdays are included in the calculation.
- **Bool_CountHolidays** boolean that takes the value 'true' or 'false'. If it takes the value 'true' all public holidays are included, if 'false' the public holidays are not included in the calculation.

Example

CONTRACTDAYSININTERVAL(FIRSTDAYOFYEAR, DATEOFBOOKING, 30, false, false)
Days on which the employee is contracted to work as part of a contractual working week of thirty or more hours count as contract days. Weekend days and holiday days do not count as contract days. If the employee's contractual working week falls below thirty hours, his/her contract days are no longer counted.

CONTRACTLENGTH

Yields the length of the employee's contract in days on the booking date.

How to use

CONTRACTLENGTH('<Before date as YYYY-MM-DDThh:mm:ss>')

- **Before date as YYYY-MM-DDThh:mm:ss** is a date in the specified format. The result of the function is the number of days that the employee has been under contract, up to the specified date.

It is not necessary to specify a source account when using this function.

Example

Suppose that the employee entered service on 01-01-2016 (2016 was a leap year).

Expression	Value booked
CONTRACTLENGTH('2017-01-01')	366
CONTRACTLENGTH('2017-01-02')	367
IF(CONTRACTLENGTH('2017-01-01')>0,1,2)	1

CONTRACTLENGTH may also be defined relative to a calculated date:
CONTRACTLENGTH(DATEOFBOOKING) yields the number of days that an employee has been in service on the date that the booking is made.

To ascertain the number of years that an employee has been in service, calculated from the date of the booking, the following can be used:
YEARSBETWEEN(CONTRACTLENGTH(DATEOFBOOKING))

EMPLOYEE

Identifies the employee to whom a booking relates.

Can be used as a parameter within other functions.

EMPLOYEEAGE

Returns the age of an employee on a certain date.

How to use

EMPLOYEEAGE([<Date>])

- **Date** is the date for which the age of an employee is returned.

Can be used as a parameter within other functions.

Example

The management of an organization wish to give two additional vacation days to all employees who are older than 50.
The following formula can be used to add those two additional days:

IF(EMPLOYEEAGE(STRTODATE('01-01-2010')) >= 50, 2, 0)

EMPLOYEECONTRACTHOURS

Statement of the length of the employee's contractual working week in hours.

How to use

EMPLOYEECONTRACTHOURS([<From_Date>, <To_Date>])

- **From Date** and **To Date** identify the period for which the average contractual working week should be calculated.

It is not necessary to specify a source account when using this function.

Example

Suppose that the employee is contracted to work 36 hours a week. Note that, in the example, bookings are made on a weekly basis.

Expression	Value booked
EMPLOYEECONTRACTHOURS	36:00

EMPLOYEEDATEOFBIRTH

Returns the date of birth of an employee.

How to use

EMPLOYEEDATEOFBIRTH

Can be used as a parameter within other functions.

Example

The expression allows you to enter entitlements to all employees born before a certain date. Another option is to add entitlements, like an additional personal day off, on the employee's birth date.

The following formula adds two vacation days to all employees born before 01-01-1960:

IF(EMPLOYEEDATEOFBIRTH < STRTODATE('01-01-1960'),2,0)

EMPLOYEEMAXCONTRACTHOURS

Statement of the average length an employee is maximally contracted to work for a defined period.

How to use

EMPLOYEEMAXCONTRACTHOURS([<From_Date>, <To_Date>])

- **From Date** and **To Date** identify the period for which the average maximally working week should be calculated.

It is not necessary to specify a source account when using this function.

Example

Suppose that the employee is contracted to work maximal 45 hours a week. Note that, in the example, bookings are made on a weekly basis.

Expression	Value booked
EMPLOYEECONTRACTHOURS	45:00

EMPLOYEEMINCONTRACTHOURS

Statement of the average length an employee should minimal work according to its contract for a defined period.

How to use

EMPLOYEEMINCONTRACTHOURS([<From_Date>, <To_Date>])

- **From Date** and **To Date** identify the period for which the average minimally working week should be calculated.

It is not necessary to specify a source account when using this function.

Example

Suppose that the employee is contracted to work minimal 20 hours a week. Note that, in the example, bookings are made on a weekly basis.

Expression	Value booked
EMPLOYEECONTRACTHOURS	20:00

EMPLOYEEPROPERTY

Can be used to specify the value of an employee attribute.

How to use

EMPLOYEEPROPERTY('<Property_Name>', [<Date>])

- **Property name** is the name of the employee attribute.
- **Date** is the date from which the attribute value is effective.

It is not necessary to specify a source account when using this function.

Example

Suppose that the value of the employee's 'Distance to work' attribute is 24 and that the value of his/her 'Lease car' attribute is 'No'. Bookings are made on a weekly basis.

Expression	Value booked
EMPLOYEEPROPERTY('Distance to work')	24
IF(EMPLOYEEPROPERTY('Lease car'=TRUE,1,2)	1

EMPLOYEESALARYGROUP

Retuns the salarygroup of an employee for a defined date.

How to use

EMPLOYEESALARYGROUP([<Date>])

- **Date** is the date for which the salarygroup of the employee should be specified.

It is not necessary to specify a source account when using this function.

Example

Suppose that the an employee falls within the salary group F for the period 01-01-2017 to 01-01-2018.

Expression	Value booked
EMPLOYEESALARYGROUP(STRTODATE('01-03-2017'))	'F'
IF(EMPLOYEESALARYGROUP(STRTODATE('01-03-2017'))='F',1,0)	1

FIRSTDAYOFILLNESS

Determines the day-number of the start of a period of sick leave. Takes account of the possibility that two separately reported periods of sick leave should be treated as one. Relates to illness types that can form part of a period of continuous sick leave.

How to use

FIRSTDAYOFILLNESS(<Number of days>, [<Date>, <Employee>])

- **Number of days** is a statement of the maximum number of days between a recovery notice and a subsequent period of sick leave that is consistent with the two periods being treated as one.
- **Date** is the date within the employee's sick-leave period that is to be checked. If no date is specified, the booking date is used.
- **Employee** is the employee whose first sick-leave day is defined. If no employee is specified, the definition is taken as relating to the employee to whom the booking relates.

Example

Suppose that, in the example given, the employee was on sick leave for the following periods:

- From 30-12-2016 to 04-01-2017,
- From 11-04-2017 to 14-04-2017 and
- From 30-04-2017 to 16-05-2017.

In each case, the illness type was one that can form part of a period of continuous sick leave. The booking date is 02-05-2017.

Expression	Value booked
IF(FIRSTDAYOFILLNESS(30)>0,BOOKINGDATE-FIRSTDAYOFILLNESS(30),0)	21
IF(FIRSTDAYOFILLNESS(30,'2017-01-01')>0,BOOKINGDATE-FIRSTDAYOFILLNESS(30,'2017-01-01'),0)	123

HASILLNESSPROPERTY

Returns the value 'True' if there is an overlapping sick leave with the given illness name and 'False' if not.

How to use

HASILLNESSPROPERTY('<Property_Name>', ['<Property_Value>'])

- **Property name** is the name of the sick-leave property to which the expression relates e.g. 'Illness type'.
- **Property value** is the value of the sick-leave property that will result in the expression yielding the result ‘TRUE’, e.g. ‘maternity leave’.

Only the property name 'IllnessType' can be used.

Example

Suppose that an employee was on sick leave from 13-07-2017 to 15-07-2017 and that the illness type was ‘maternity leave’. The expression containing HASILLNESSPROPERTY specifies the period as ‘per booking’.
HASILLNESSPROPERTY(‘IllnessType’, ‘maternity leave’) will then yield ‘TRUE’ for the bookings on 13-07-2005 and 14-07-2005, but ‘FALSE’ for all other bookings.

The following expression ensures that the value ‘8.0’ is carried over for each booking that relates to a day on which the employee was on sick leave and the illness type was ‘maternity leave’:
IF(HASILLNESSPROPERTY(‘IllnessType’, ‘maternity leave’), 8.0, 0.0)

HASILLNESSWITHPROPERTY

Returns the value 'True' if there is an overlapping sick leave that has illness property with a given value and returns 'False' if not.

How to use

HASILLNESSWITHPROPERTY(<Date>, '<Property_Name>', '<Property_Value>')

- **Date** is the date for which the illness property should be checked.
- **Property_name** is the name of the illness property to which the expression relates.

- **Property_value** is the value of the illness property that will result in the expression yielding the result 'TRUE'.

Example

Suppose that, in the example given, the employee was on sick leave for the following periods:

- From 30-12-2016 to 04-01-2017,
- From 11-04-2017 to 14-04-2017 and
- From 30-04-2017 to 16-05-2017.

The illness 'sick leave' has an illness property 'ShortSick' with default value 'No'. This value is set to 'Yes' when the illness spans less then one week. The first and second periods are set to the value 'Yes'. The expression containing HASILLNESSPROPERTY specifies the period as 'per day'.

IF(HASILLNESSWITHPROPERTY(DATEOFBOOKING, 'ShortSick', 'True'),1,0)
The expression results to the value 1 on:

- 30-12-2016
- 01-01-2017
- 02-01-2017
- 03-01-2017
- 11-04-2017
- 12-04-2017
- 13-04-2017

ILLNESSPERCENTAGE

Determines the average illness percentage in a given period.

How to use

ILLNESSPERCENTAGE([<begin time>, <end time>])

- **Begin time** and **End time** are the begin and end time of the time interval for which the average illness percentage is calculated.

The parameters **Begin time** and **End time** are both optional.

Example

Carry over of a four-week salary period.

Over the four weeks in question, the relevant employee's sick-leave pattern is as follows:

- Week 1: sick, no WT/LW (work therapy/limited working)
- Week 2: not sick
- Week 3: sick, WT/LW (60%)
- Week 4: sick, WT/LW (30%)

Result of the expression: 52.5

Calculation: (7*100 + 7*0 + 7*40 + 7*70)/28 = 52.5

LENGTHILLNESS

Determines the length in days of a sick-leave period, counting from the last day of the booking period.

How to use

LENGTHILLNESS(<Max number of days between illnesses>)

- **Max number of days between illnesses** is a statement of the maximum number of days between a recovery notice and a subsequent period of sick leave that is consistent with the two periods being treated as one.

Example

Suppose that, in the example given, the employee was on sick leave for the following periods:

- From 30-12-2016 to 04-01-2017,
- From 11-04-2017 to 14-04-2017 and
- From 30-04-2017 to 16-05-2017.

In each case, the illness type was one that can form part of a period of continuous sick leave. The booking date is 02-05-2017 and the period is a day.

Expression	Value booked
LENGTHILLNESS(30)	22

33.5 Value expressions

The value expressions can be included to specify how the bookings of the source account should be carried over.

In order to explain the amount expressions with an example this document will make use of Account 1. Account 1, for which the categories 1A and 1B are defined, includes the following bookings:

Account 1	01-05-2017 to 08-05-2017
1A	
> 01-05-2017 A	10:30 / 100% / 10:30
> 05-05-2017 A	-5:00 / -50% / 02:30
1B	
> 03-05-2017 A	10:00 / 140% / 14:00
Total	27:00

AVERAGEWEIGHT

Inclusion of this function results in the average percentage of the bookings for the relevant period being carried over.

How to use

AVERAGEWEIGHT(['<account category>'])

- **Account category** is the name of the category, the average percentage of whose bookings is to be carried over.

  If no category is specified, the average of the bookings under all categories is carried forward.

Example

Expression	Value booked
AVERAGEWEIGHT	0.63
AVERAGEWEIGHT('1A')	0.25

The value of the first expression is calculated as (100% - 50% + 140%) / 3 = 0.63 (or 63%)
The value of the second expression is calculated as (100% - 50%) / 2 = 0.25 (or 25%0

TOTALAMOUNT

TOTALAMOUNT yields the sum of the amount hours booked for the booking period.

How to use

TOTALAMOUNT(['<account category>', <begin time>, <end time>])

- **Account category** is the name of the category for which the aggregate bookings are to be carried over.

  If no category is specified, the average of the bookings under all categories is carried forward.

- **Begin time** and **End time** define the period in respect of which the bookings are to be aggregated.

  If no start and end times are specified, the bookings for the entire booking period are aggregated.

Example

Expression	Value booked
TOTALAMOUNT	15:30
TOTALAMOUNT('1A')	05:30
TOTALAMOUNT('1A','01-05-2017','04-05-2017')	10:30

TOTALRELATIVEAMOUNT

TOTALRELATIVEAMOUNT yields the aggregate value of the hours booked (the hours multiplied by a percentage) for the booking period.

How to use

TOTALRELATIVEAMOUNT(['<account category>', <begin time>, <end time>])

- **Account category** is the name of the category for which the aggregated bookings are to be carried over.

If no category is specified, the average of the bookings under all categories is carried forward.

- **Begin time** and **End time** define the period in respect of which the bookings are to be aggregated.

  If no start and end times are specified, the bookings for the entire booking period are aggregated.

Example

Expression	Value booked
TOTALRELATIVEAMOUNT	27:00
TOTALRELATIVEAMOUNT('1A')	13:00
TOTALRELATIVEAMOUNT('1A','01-05-2017','04-05-2017')	10:30

TOTALWEIGHT

TOTALWEIGHT yields the sum of the booking percentages for the booking period.

How to use

TOTALWEIGHT(['<account category>'])

- **Account category** is the name of the category whose booking percentages are to be aggregated.

  If no category is specified, the booking percentages under all categories are aggregated.

Example

Expression	Value booked
TOTALWEIGHT	1,90
TOTALWEIGHT('1A')	0,50

VALUE

VALUE yields the aggregate value of the hours booked (the hours multiplied by a percentage) for the booking period.

How to use

TOTALWEIGHT(['<account category>', <begin time>, <end time>])

- **Account category** is the name of the category whose booking percentages are to be aggregated.

  If no category is specified, the booking percentages under all categories are aggregated.

- **Begin time** and **End time** define the period in respect of which the bookings are to be aggregated.

If no start and end times are specified, the bookings for the entire booking period are aggregated.

Example

Expression	Value booked
VALUE	27:00
VALUE('1A')	13:00
VALUE('1A','01-05-2017','04-05-2017')	10:30

VALUEFORSHIFTSBETWEEN

VALUEFORSHIFTSBETWEEN yields the aggregate value of the hours booked (the hours multiplied by a percentage) for the shifts that start or end within a certain period.

How to use

VALUEFORSHIFTSBETWEEN(['<account category>', <begin time>, <end time>])

- **Account category** is the name of the category whose booking percentages are to be aggregated.

  If no category is specified, the booking percentages under all categories are aggregated.

- **Begin time** and **End time** define the period in which a shifts should end in order to count the booking related to this shift.

  If no end time is specified, the shift should start within the defined period in order to count the booking related to this shift.

  The **Begin time** and **End time** should be expressed in dates.

Example

Assume that the booking of '10:30' on account category 1A on 01-05-2005 is obtained from shift A and B.

Shift A is defined as:

Activity	Start time	End time
Work	07:00	12:00

Shift B is defined as:

Activity	Start time	End time
Work	20:00	01:30

Shift A and B did both start on the date 01-05-2017, only shift B end a day later.

Expression	Value booked
VALUEFORSHIFTSBETWEEN('1A','01-05-2017','01-05-2017')	05:00
VALUEFORSHIFTSBETWEEN('1A','01-05-2017','02-05-2017')	10:30
VALUEFORSHIFTSBETWEEN('1A','01-05-2017')	10:30

33.6 Other expressions

In the other expressions you can find the remaining expressions. These expressions mainly concerns expressions that are useful to filter on specifications of bookings.

ACTIVITYCOSTCENTERHASPARENTWITHNAME
The expression ACTIVITYCOSTCENTERHASPARENTWITHNAME can be used to verify if an employee's activity has a cost center with a certain parent cost center.

How to use

ACTIVITYCOSTCENTERHASPARENTWITHNAME(['<cost center>'])

Example
Assume that department A has the following bookings.

Hours	01-01-2017 to 01-02-2017
Working hours	
Cost center A.1	
> 01-01-2017	120
Cost center B.1	
> 01-01-2017	95
Total	215

Cost center A.1 has a parent cost center named 'Cost center A'. Cost center B.1 has a parent cost center named 'Cost center B'. The management is now interested in the number of hours made by employees on parent cost center A. Bookings are carried over **per entry**.

IF(ACTIVITYCOSTCENTERHASPARENTWITHNAME('Cost center A'),VALUE('Working hours'),0)

Cost center A	01-01-2017 to 01-02-2017
Working hours	
Cost center A.1	
> 01-01-2017	120
Total	120

ACTIVITYCOSTCENTERNAME
The expression ACTIVITYCOSTCENTERNAME can be used to verify if an employee's activity has a certain cost center.

How to use

ACTIVITYCOSTCENTERNAME

This expression does not have a parameter.

Example

Assume that department A has the following bookings.

Hours	01-01-2017 to 01-02-2017
Working hours	
Cost center A.1	
> 01-01-2017	120
Cost center B.1	
> 01-01-2017	95
Total	215

The management is now interested in the number of hours made by employees on cost center B.1. Bookings are carried over **per entry**.

IF(ACTIVITYCOSTCENTERNAME='Cost center B.1',VALUE('Working hours'),0)

Cost center B.1	01-01-2017 to 01-02-2017
Working hours	
Cost center B.1	
> 01-01-2017	95
Total	95

ACTIVITYFUNCTIONNAME

The expression ACTIVITYFUNCTIONNAME can be used to verify if an employee has a certain job function.

How to use

ACTIVITYFUNCTIONNAME

This expression does not have a paramater.

Example

Assume that department A has the following bookings.

Hours	01-01-2017 to 01-02-2017
Working hours	
Betty	
> 01-01-2017	120
John	
> 01-01-2017	95
Total	215

Employee Betty has the function Manager. The management is now interested in the number of hours made by employees with the function Manager.

IF(ACTIVITYFUNCTIONNAME='Manager',VALUE('Working hours'),0)

Hours	01-01-2017 to 01-02-2017
Working hours managers	
Betty	
> 01-01-2017	120
Total	120

ACTIVITYPROPERTY

The expression ACTIVITYPROPERTY can be used to verify if an employee's activity has a certain property.

How to use

ACTIVITYPROPERTY('<Property_Name>')

- **Property name** is the name of the activity property on which you wish to filter.

An activity property can be created in the menu Maintenance, Organization and tab Activity properties of the main organization. After this the created activity property can be used in any department.

Example

Shift A, as scheduled for Saturday 8 April 2017, is made up of the following activities and values for the activity property 'Outside'. The activity property 'Outside' is of type 'Yes/No' and has default value No.

Activity type	Start	End	Outside
Work	08:00	12:00	Yes
Break	12:00	12:30	No
Work	12:30	17:00	No

The hours of Shift A are booked on the account 'Hours' category 'Working hours'. We use this expression to transfer the hours to the category 'Working hours outside'. Bookings are carries over **per entry**.

IF(ACTIVITYPROPERTY('Outside')=TRUE,VALUE('Working hours'),0)

Hours	01-01-2017 to 01-01-2018
Working hours outside	
> 08-09-2017	4:00
Total	4:00

ALL_CATEGORIES

The expression ALL_CATEGORIES can be used as parameter to ensure that only the categories that contain one or more bookings are carried over.

The expression ALL_CATEGORIES is only available in the compensation rule "Carry over all account categories with expression".

How to use

ALL_CATEGORIES

Example

Suppose that Account A has the following bookings:

Account A	01-01-2017 to 01-01-2018
1A	
> 01-01-2017	8:00
> 01-06-2017	5:00
1B	
> 01-07-2017	2:00
Total	15:00

Normally by use of the compensation rule "Carry over all account categories with expression" to carry over the total amount of bookings of Account A to Account B gives the following result.

TOTALAMOUNT('1A')

Account B	01-01-2017 to 01-01-2018
1A	
> 01-01-2017	13:00
1B	
> 01-01-2017	13:00
Total	26:00

By using the ALL_CATEGORIES as a parameter in the amount expression results in:

TOTALAMOUNT(ALL_CATEGORIES)

Account B	01-01-2017 to 01-01-2018
1A	
> 01-01-2017	13:00
1B	
> 01-01-2017	2:00
Total	15:00

In the example the value is booked **per year**.

BOOKEDBYREMUNERATION

Indicates whether at least one of the account category values has been generated by application of the named compensation rule.

How to use

BOOKEDBYREMUNERATION ('<name>')

- **Name** is the name of the compensation rule for which you wish to check if a value on the account category is generated by.

Example

Assume that the booking on an account category is generated by the compensation rule "Allowances" and not by the compensation rule "Overtime".

Expression	Value
IF(BOOKEDBYREMUNERATION('Allowances'=TRUE,1,0)	1
IF(BOOKEDBYREMUNERATION('Overtime'=TRUE,1,0)	0

BOOKINGPHYSICALLOCATIONHASPARENTWITHNAME

Transfer all the bookings of the same workstation given that the workstations are within the indicated location.

How to use

BOOKINGPHYSICALLOCATIONHASPARENTWITHNAME(['<location>'])

- **Location** is the parent location in which the workstation must be in order to trigger a booking.

Example

Assume that account A has the following bookings:

Account A	01-04-2017 to 08-04-2017		
Category 1A		Location	Workstation
> 01-04-2017	8:00	Building A	Room X
> 02-04-2017	8:00	Building B	Room X
> 05-04-2017	8:00	Building A	Room X
> 06-04-2017	8:00	Building A	Room Y
Total	32:00		

As can be seen in Account A, the workstation Room X is available in the location Building A and building B.

In the example below you only transfer the bookings of the shifts that were performed in Room X at Building A.

IF(BOOKINGPHYSICALLOCATIONHASPARENTWITHNAME('Building A')='Room X',VALUE('Category 1A'),0)

Account B	01-04-2017 to 08-04-2017		
Category 1B		Location	Workstation
> 01-04-2017	8:00	Building A	Room X
> 05-04-2017	8:00	Building A	Room X
Total	16:00		

In the example bookings are transferred from account A to account B **per day** or **per entry**.

BOOKINGPHYSICALLOCATIONNAME

Transfer the bookings of all workstations within the indicated location.

How to use

BOOKINGPHYSICALLOCATIONNAME

This expression does not have a parameter.

Example

Assume that account A has the following bookings:

Account A	01-04-2017 to 08-04-2017		
Category 1A		Location	Workstation
> 01-04-2017	8:00	Building A	Room X
> 02-04-2017	8:00	Building B	Room X
> 05-04-2017	8:00	Building A	Room X
> 06-04-2017	8:00	Building A	Room Y
Total	32:00		

As can be seen in Account A, the available workstations at Building A are Room X and Room Y.

In the example you only want to transfer the bookings of shifts that were performed at Building A.

IF(BOOKINGPHYSICALLOCATIONNAME='Building A',VALUE('Category 1A'),0)

Account B	01-04-2017 to 08-04-2017		
Category 1B		Location	Workstation
> 01-04-2017	8:00	Building A	Room X
> 05-04-2017	8:00	Building A	Room X
> 06-04-2017	8:00	Building A	Room Y
Total	24:00		

In the example bookings are transferred from account A to account B **per day** or **per entry**.

BOOKINGWORKSTATIONNAME

Transfer the bookings of the indicated workstations within all locations.

How to use

BOOKINGWORKSTATIONNAME

This expression does not have a parameter.

Example

Assume that account A has the following bookings:

Account A	01-04-2017 to 08-04-2017		
Category 1A		Location	Workstation
> 01-04-2017	8:00	Building A	Room X
> 02-04-2017	8:00	Building B	Room X
> 05-04-2017	8:00	Building A	Room X
> 06-04-2017	8:00	Building A	Room Y
Total	32:00		

As can be seen in Account A, Room X is available at Building A and Building B.

In the example below you only want to transfer the bookings of shifts that were performed at Room X.

IF(BOOKINGWORKSTATIONNAME='Room X',VALUE('Category 1A'),0)

Account B	01-04-2017 to 08-04-2017		
Category 1B		Location	Workstation
> 01-04-2017	8:00	Building A	Room X
> 02-04-2017	8:00	Building B	Room X
> 05-04-2017	8:00	Building A	Room X
Total	24:00		

In the example bookings are transferred from account A to account B **per day** or **per entry**.

COSTCENTERPROPERTY

The expression COSTCENTERPROPERTY returns the value of the indicated cost center property.

How to use

COSTCENTERPROPERTY('<Property_Name>', [<Date>])

- **Property name** is the indicated cost center property.

  A cost center property can be created in the menu Maintenance, Cost centers and tab Cost center properties definition.

- **Date** is the date for which the indicated cost center property must be valid for a cost center in order to trigger a booking.

Example

Suppose that the value of the shift's 'Cost center property 1' attribute is 24 and that the value of his/her 'Cost center property 2' attribute is 'Yes'. Bookings are made on a weekly basis.

Expression	Value booked
COSTCENTERPROPERTY('Cost center property 1')	24
IF(COSTCENTERPROPERTY('Cost center property 2'=TRUE,1,0))	1

DEPARTMENTPROPERTY

The expression DEPARTMENTPROPERTY returns the value of the indicated location property.

How to use

DEPARTMENTPROPERTY('<Property_Name>', [<Date>])

- **Property name** is the indicated department property.

  > A department property can be created in the menu Maintenance, Organization and tab Department properties of the main organization. After this the created department property can be used in any department.

- **Date** is the date for which the indicated department property must be valid for a department in order to trigger a booking.

Example

Suppose that the value of the shift's 'Department property 1' attribute is 24 and that the value of his/her 'Department property 2' attribute is 'Yes'. Bookings are made on a weekly basis.

Expression	Value booked
DEPARTMENTPROPERTY('Department property 1')	24
IF(DEPARTMENTPROPERTY('Department property 2'=TRUE,1,0))	1

DUTYPROPERTY

DUTYPROPERTY can be used to specify the value of a shift's duty property.

How to use

DUTYPROPERTY('<Property_Name>')

- **Property name** is the name of the duty property on which you wish to filter.

Example

Suppose that the value of the shift's 'Duty property 1' attribute is 24 and that the value of his/her 'Duty property 2' attribute is 'Yes'. Bookings are made on a weekly basis.

Expression	Value booked
DUTYPROPERTY('Duty property 1')	24
IF(DUTYPROPERTY('Duty property 2'=TRUE,1,0))	1

FILTERAPPROVED

Indicates whether at least one of the bookings in the specified period has been approved. The result of the function is TRUE or FALSE.

How to use

FILTERAPPROVED('<account category>')

- **Account category** is the name of the category whose bookings are to be checked to ascertain whether any have been approved.

  > If no category is specified, all categories are taken into account.

Example

Suppose that, in the example given, the booking made under category 1A on 02-05-2017 has been approved, but that the other bookings have not.

Expression	Value
IF(FILTERAPPROVED,1,2)	1
IF(FILTERAPPROVED('1A'),1,2)	1
IF(FILTERAPPROVED('2A'),1,2)	2

In order to carry over only the values of approved bookings, work on a 'per entry' basis and use IF (FILTERAPPROVED,VALUE,0.0)

FILTERDAY

Indicates whether at least one of the bookings in the specified period was made on one of the days of the week specified in the parameters.

How to use

FILTERDAY(bool_mo, bool_tu, bool_we, bool_th, bool_fr, bool_sa, bool_su, bool_ho, ['<account category>'])

- **bool_mo**, **bool_tu**,..., **bool_ho** indicates for each day of the week (Monday to Sunday and public holiday) whether that day's bookings are to be carried over.
- **Account category** is the name of the category whose bookings are to be checked to ascertain whether any were made on the specified day(s) of the week.

If no category is specified, all categories are taken into account.

Example

Expression	Value
IF(FILTERDAY(TRUE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE),1,2)	1
IF(FILTERDAY(TRUE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,'2A'),1,2)	2

In order to carry over only the values of bookings made on the specified days, work on a 'per day' or 'per entry' basis and use IF(FILTERDAY(...),VALUE,0.0).

FILTERTIMEFORTIMETYPE

Indicates whether at least one of the bookings in the specified period is of the specified time-for-time type.

How to use

FILTERTIMEFORTIMETYPE('<Time for time type>', '<account category>')

- **Time for time type** indicates the time-for-time type of the bookings to be carried over.
- **Account category** is the name of the category whose bookings are to be checked to ascertain whether any of those made in the relevant booking period were of the specified type.

If no category is specified, all categories are taken into account.

A type can be specified only if it is defined as an option in the source account to which the rule relates.

Example

Suppose that, in the example given, the booking made under category 1A on 02-05-2017 was of the type 'No time for time', while the other bookings were of the type 'Time for time'.

Expression	Value
IF(FILTERTIMEFORTIMETYPE('No time for time'),1,2)	1
IF(FILTERTIMEFORTIMETYPE('No time for time','2A'),1,2)	2

HASAPPROVED

Indicates whether at least one of the bookings made in the specified period is labeled as approved. Result of the function is TRUE or FALSE.

How to use

HASAPPROVED('<account category>')

- **Account category** is the name of the category whose bookings are to be checked to ascertain whether any of those made in the booking period have been approved.

  If no category is specified, all categories are taken into account.

Example

Suppose that, in the example given, the booking made under category 1A on 02-05-2017 has been approved, but that the other bookings have not.

Expression	Value
IF(HASAPPROVED,1,2)	1
IF(HASAPPROVED('1A'),1,2)	1
IF(HASAPPROVED('2A'),1,2)	2

In order to carry over only the values of approved bookings, work on a 'per entry' basis and use IF (HASAPPROVED,VALUE,0.0).

HASCALLOUT

Indicates whether at least one of the bookings in the specified period has the check box 'Call out' enabled.

How to use

HASCALLOUT('<account category>')

- **Account category** is the name of the category whose bookings are to be checked to ascertain whether any have the check box 'Call out' enabled.

  If no category is specified, all categories are taken into account.

Example

Suppose that, in the example given, the booking made under category 1A on 02-05-2017 is marked as Call out, but that the other bookings have not.

Expression	Value
IF(HASCALLOUT,1,2)	1
IF(HASCALLOUT('2A'),1,2)	2

In order to carry over only the values of bookings made on the specified days, work on a ‘per day’ or ‘per entry’ basis and use IF(HASCALLOUT,VALUE,0.0).

HASCOSTCENTER

Indicates whether at least one of the bookings in the specified period has a cost center assigned.

How to use

HASCOSTCENTER

Bookings can only be carried over per entry, per entry day or per shift.

Example

Shift A, as scheduled for Saturday 8 April 2017, is made up of the following activities and cost centers.

Activity type	Start	End	Cost center
Work	08:00	12:00	(None)
Break	12:00	12:30	(None)
Work	12:30	17:00	Cost center A

The hours of Shift A are booked on the account 'Hours' category 'Working hours'. We use this expression to transfer the hours to the category 'Costs'. Bookings are carries over **per entry**.

IF(HASCOSTCENTER,VALUE('Working hours'),0)

Hours	03-04-2017 to 10-04-2017
Costs	
> 08-04-2017	4:30
Total	4:30

HASDAY

Indicates whether at least one of the bookings in the specified period was made on one of the days of the week specified in the parameters.

How to use

HASDAY(bool_mo, bool_tu, bool_we, bool_th, bool_fr, bool_sa, bool_su, bool_ho, ['<account category>'])

- **bool_mo**, **bool_tu**,..., **bool_ho** indicates for each day of the week (Monday to Sunday and public holiday) whether that day’s bookings are to be carried over.
- **Account category** is the name of the category whose bookings are to be checked to

ascertain whether any were made on the specified day(s) of the week

> If no category is specified, all categories are taken into account.

Example

Suppose that, in the example given, we want to book 1 on Mondays when there is a booking made. A booking under category 1A on Monday 01-05-2017 has been made.

Expression	Value
IF(HASDAY(TRUE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE),1,2)	1
IF(HASDAY(TRUE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,'2A'),1,2)	2

> In order to carry over only the values of bookings made on the specified days, work on a 'per day' or 'per entry' basis and use IF(HASDAY(…),VALUE,0.0).

HASTIMEFORTIMETYPE

Indicates whether at least one of the bookings in the specified period is of the specified time-for-time type.

How to use

HASTIMEFORTIMETYPE(<Time for time type>,'<account category>')

- **Time for time type** indicates the time-for-time type of the bookings to be carried over.
- **Account category** is the name of the category whose bookings are to be checked to ascertain whether any of those made in the relevant booking period were of the specified type. If no category is specified, all categories are taken into account.

> If no category is specified, all categories are taken into account.
>
> A type can be specified only if it is defined as an option in the source account to which the rule relates.

Example

Suppose that, in the example given, the booking made under category 1A on 01-05-2017 was of the type 'No time for time', while the other bookings were of the type 'Time for time'.

Expression	Value
IF(HASTIMEFORTIMETYPE('No time for time'),1,2)	1
IF(HASTIMEFORTIMETYPE('No time for time','2A'),1,2)	2

> In order to carry over only the values of bookings of the specified type, work on a 'per day' or 'per entry' basis and use IF(HASTIMEFORTIMETYPE (…),VALUE,0.0).

ISBROKENSHIFT

Determines by means of parameters whether the shift to which a source account booking relates constitutes a broken shift.

How to use

ISBROKENSHIFT(<Hours>, <'Treatment1;Treatment2;...'>)

- **Hours** is the length of the continuous period (expressed as a number of hours) within a shift that needs to be devoted to activities of the kinds specified in the second parameter for the shift to be deemed broken.
- **Treatment1**;**Treatment2**,… are the activity kinds that constitute shift breaks.
- This function can be used only if the carryover is made on a 'per entry day' or 'per shift' basis.

Example

Shift A, as scheduled for Saturday 8 April 2017, is made up of the following activities:

Activity type	Start	End
Work	08:00	12:00
Break	12:00	17:30
Work	17:30	20:00

The activity type 'Work' belongs to the activity kind 'Work'. The activity type 'Break' belongs to the activity kind 'Break'.

Expression	Value
IF(ISBROKENSHIFT(4.0,PAUSE),1,2)	1

The specified treatments should be as in typeconsts.

ISFLEXPOOLDEPARTMENT

The expression ISFLEXPOOLDEPARTMENT can be used to identify shifts originating from a department that is assigned the property 'Flexpool'.

How to use

ISFLEXPOOLDEPARTMENT

This expression does not have a parameter.

Example

Assume that an employee works in the Flexpool Department. On 01-04-2017 he/she works a 8 hour shift that is subcontracted to the Flexpool department and on 02-04-2013 he/she works a 8 hour shift that is the Flexpool's own shift.

Hours	01-04-2017 to 08-04-2017
Hours worked	
> 01-04-2017	8:00
> 02-04-2017	8:00
Total	16:00

In this example the planner wants to book shifts of the Flexpool department.

IF(ISFLEXPOOLDEPARTMENT =TRUE,VALUE('Hours worked'),0)

Hours	01-04-2017 to 08-04-2017
Hours Flexpool	
> 02-04-2017	8:00
Total	8:00

In the example bookings are transferred from account A to account B **per day** or **per entry**.

Example

In case the planner wants to exclude shifts of the Flexpool department, the planner can use:

IF(ISFLEXPOOLDEPARTMENT=FALSE,VALUE('Hours worked'),0)

Hours	01-04-2017 to 08-04-2017
Hours non-Flexpool	
> 01-04-2017	8:00
Total	8:00

In the example bookings are transferred from account A to account B **per day** or **per entry**.

ISFLEXPOOLSHIFT

The expression ISFLEXPOOLSHIFT can be used to make a distinction in flexpool shifts and regular shifts in case an employee works in both its own department and the Flexpool.

How to use

ISFLEXPOOLSHIFT

This expression does not have a parameter.

Example

Assume that an employee works on 01-04-2017 8 hours for its own department and on 02-04-2017 8 hours for Flexpool.

Hours	01-04-2017 to 08-04-2017
Hours worked	
> 01-04-2017	8:00
> 02-04-2017	8:00
Total	16:00

In this example the planner wants to book only the hours the employee works for Flexpool

IF(ISFLEXPOOLSHIFT=TRUE,VALUE('Hours worked'),0)

Hours	01-04-2017 to 08-04-2017
Hours Flexpool	
> 02-04-2017	8:00
Total	8:00

In the example bookings are transferred from account A to account B **per day** or **per entry**.

Example

In case the planner wants to book only the hours made in regular shifts, the planner can use:

IF(ISFLEXPOOLSHIFT=FALSE,VALUE('Hours worked'),0)

Hours	01-04-2017 to 08-04-2017
Hours non-Flexpool	
> 01-04-2017	8:00
Total	8:00

In the example bookings are transferred from account A to account B **per day** or **per entry**.

ISHOLIDAYWITHPROPERTY

The expression ISHOLIDAYWITHPROPERTY can be used to filter on entries that have a certain holiday property and a certain property value.

How to use

ISHOLIDAYWITHPROPERTY(<Date>, '<Property_Name>', '<Property_Value>')

- **Date** is the date for which the indicated holiday property should be valid in order to trigger a booking.
- **Property name** is the name of the holiday property you wish to filter on.
- **Property value** is the value that should correspond with the indicated property in order to add the account entry.

A holiday property can be created in the window **Maintenance**, **Public holiday** and tab **Definition holiday properties**.

In the tab **Definition holiday properties** you will identify the default value of a certain property. In the tab **Values holiday properties** you can change the default value to the required value in the row **Value** per public holiday.

If you create a holiday property it is necessary to identify a **type** [Text, Integer, Decimal number, Date, Text, Yes/No, Time]. In case you want to make use of **Decimal number** it is better to use the type **Text** instead. This because if you choose the type **Decimal number** you will identify the value to for example the number 2.4. Now by use of the expression ISHOLIDAYWITHPROPERTY you should not set the parameter '<Property_Value>' to 2.4 but to 2,4. This inconsistent use of a point or a comma in a decimal number does not play a role in case you choose the type **Text**.

Example

Generally employees prefer to work on the public holiday Good Friday (14-04-2017) than on the public holiday New Year's Eve (31-12-2017). The company therefore make use of the property 'Importance' so they are able to give an different allowance for working on a holiday with an high importance.

Assume that these holidays have the following definitions:

Public holiday	Property	Type	Value
Good Friday (14-04-2017)	Importance	Text	Low
New Year's Eve (31-12-2017)	Importance	Text	High

Assume that an employee works 8:00 on the public holiday Good Friday (14-04-2017) and New Year's Eve (31-12-2017).

Hours	01-01-2017 to 01-01-2018
Hours worked on public holiday	
> 14-04-2017	8:00
> 31-12-2017	8:00
Total	16:00

Now for working on holidays that have the value High for the property Importance, the company wish to give an extra allowance of 0.5 times the hours the employee works on this day.

IF(ISHOLIDAYWITHPROPERTY(DATETIMEOFSHIFTBOOKING,'Importance','High'),0.5*VALUE('Hours worked on public holiday'),0)

Allowances	01-01-2017 to 01-01-2018
Extra	
31-12-2017	4:00
Total	4:00

In the example bookings are transferred from **Hours worked on public holiday** to **Extra** per **day** or **booking day**.

NOFWEEKDAYS

Determines the number of weekdays (Monday to Friday) in the booking period.

How to use

NOFWEEKDAYS(<Count holidays (true/false)>, [<Begindate>, <Endate>])

- **Count holidays (true/false)** specifies whether public holidays should or should not be counted.
- **Begin date** and **End date** define the period for which the number of weekdays is counted.

> If the **Begin date** and **End date** are not defined the number of weekdays are counted all dates within the account horizon.

Example

In the example below, Friday 14-04-2017 is a public holiday.

Expression	Value
NOFWEEKDAYS(TRUE,'10-04-2017','17-04-2017')	5
NOFWEEKDAYS(FALSE,'10-04-2017','17-04-2017')	4

PHYSICALLOCATIONPROPERTY

The expression PHYSICALLOCATIONPROPERTY returns the value of the indicated location property.

How to use

PHYSICALLOCATIONPROPERTY('<Property_Name>', [<Date>])

- **Property name** is the indicated location property.

> A location property can be created in the menu Maintenance, Locations and workstations and tab Definition location properties.

- **Date** is the date for which the indicated location property should be valid in order to trigger a booking.

Example

Suppose that the value of the shift's 'Location property 1' attribute is 24 and that the value of his/her 'Location property 2' attribute is 'Yes'. Bookings are made on a weekly basis.

Expression	Value booked
PHYSICALLOCATIONPROPERTY('Location property 1')	24
IF(PHYSICALLOCATIONPROPERTY('Location property 2'=TRUE,1,0))	1

Contact information

For further information contact ORTEC, either through your existing ORTEC representative or by using the appropriate contact details listed on www.ortec.com

Our website offers case studies, white papers, brochures, demos and much more.