April 2020

Vertec Software
Release 6.4

Zurich +41 43 444 60 00
Hamburg +49 40 30 37 36 70
Vienna +43 1 22 600 90
mail@vertec.com www.vertec.com

Table of contents

1 Introduction

These are the Release Notes for Vertec version 6.4. Please consider the following points:

1.1 Vertec first set-up

For a first set-up, use the set-up program VertecSetup-6.4.0.exe, available for downloading. Detailed information on the set-up procedure is available in the Online Knowledge Base under www.vertec.com/kb/neuinstallation.

1.2 Before upgrading to 6.4

Please read chapter 2 on Vertec 64 bits.

1.3 Vertec upgrade

To upgrade an existing Vertec system, use the set-up program as well. In the case of large customer systems with many customer-specific reports and list settings, we recommend to set-up a test installation first and test the release upgrade before doing it on the active system. More information on this subject is available in the article **Testinstallationen** in the Online Knowledge Base under www.vertec.com/kb/testinstallation.

Backwards compatibility

Backwards compatibility issues are discussed in detail in the section **Backwards compatibility** for the following features:

Article	Page
4 Business Intelligence (BI) Module, section 4.8	40
7.1 Multilevel sorting of lists	53
8.3 Native language in MLString attributes	61
14.2 Support of additional fields in Config Sets	79
14.3 MLStrings taken into account in Config Sets	80
15.5 Links to AbstractAdresseintrag eliminated	83

1.4 First start-up after the upgrade

After the data conversion, the Vertec Desktop App must be started. This first start-up after the convert is important as it is also a part of the conversion, as certain changes to data, structures, etc. are only performed as this point. It is important that this first start-up runs through flawlessly. In no case should the first start-up process after an upgrade be interrupted.

1.5 Restarting the cloud server

After the conversion and the first start-up of the Vertec Desktop App, the cloud server must be restarted. This concludes the upgrade of Vertec.

2 Vertec 64 bits

Line: Standard, Expert | Module: PSA | Apps: Full featured | Operation: On Premises | Version: 6.4

Starting with the present version, Vertec is exclusively provided as a 64-bit Application. This means:

- Starting with Vertec 6.4, a 64-bit operating system is required for the Vertec Windows server.
- Desktop App und Cloud App now only run on 64-bit operating systems.
- The Web App can also be run on 32-bit operating systems or non-Windows systems

A 32-bit Application can use at most 2 gigabytes of work memory, independently of how much physical memory is actually available. This could cause OutOfMemoryExceptions in the case of very large systems. This limit no longer exists with 64-bit systems.

In 64-bit systems, however, memory is referenced by 64-bit addresses instead of 32-bit addresses. Each pointer to a memory location therefore requires twice as much memory space. Due to its architecture, Vertec uses a large number of pointers and objects. It is therefore to be expected that memory requirements will be doubled. Depending on the size of the installation and the amount of data, the hardware might have to be upgraded.

Vertec Setup installs the Vertec 64-bit Application as follows:

- The standard location for new installations is **C:\Programme\Vertec**.
- Upgrades of existing installations will, as before, occur in the same folder as the existing installation. This means that an existing Vertec is installed as 64-bit software in the 32-bit folder "Programme (x86)". If this is not desirable, perform a new installation of Vertec and transpose the data.
- If Firebird Server is not installed, the 64-bit version of Firebird Server is installed.
- In case of an existing Firebird installation the following occurs:
  - The current Firebird version is 3.0.4. If the existing Firebird installation is older, it is upgraded to the same 64-bit version. The installation directory remains unchanged.
  - If a 32-bit Firebird installation of version 3.0.4 is present, only the 64-bit client components of Firebird are installed.
- The Python installation in subfolder **Python** is upgraded to the 64-bit version.

2.1 Conversion of a Vertec database with MS SQL-Server

The conversion of Vertec databases to MS SQL-Server is only possible with 32 Bit. Therefore there is a special 32-bit DB Convert App. It is called **Vertec.DbConvert.exe** and is available in the Vertec Installation directory.

If you have copied the database according to the instructions in https://www.vertec.com/kb/mssqlkonvert, start the DB Convert App (not the Vertec Desktop App with parameter `/DC` as previously). Everything else functions as before.

2.2 Interfaces with 64-bit Vertec

Topal accounting interface

The Topal interface will only function correctly if Topal is changed to 64 bits. A menu item is provided for this in Topal: **Tools > Wechseln nach x64**:

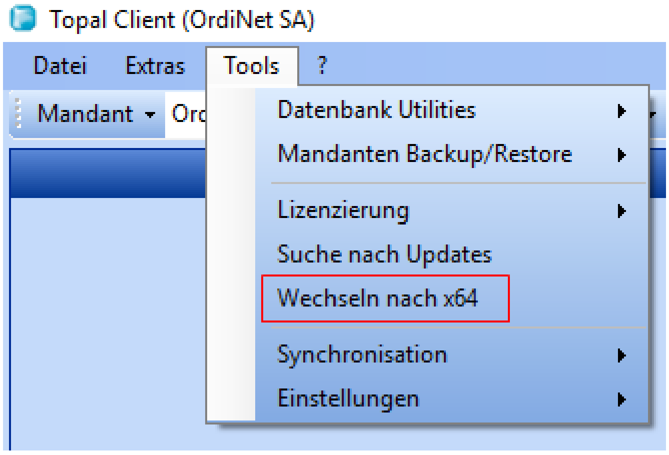


Figure 1: Change to 64-bit Topal

Sage 50 and Sage 200 interfaces

Sage 50 et Sage 200 still run in 32-bit mode. No action is required.

To use the in-process COM server, Vertec Setup registers a surrogate process (adaptation of a registry key). As a result, 32-bit Applications run with the Vertec 64-bit interfaces.

SelectLine interfaces

SelectLine runs as before, no action is required.

TAPI interface

The Vertec TAPI interface has been reimplemented for Vertec 6.4 and should continue to run normally after the upgrade.

3 Version

3.1 Highlights of version 6.4

Version 6.4 again comes with several highlights. The most interesting ones are listed here:

3.2 Pre-set features in Vertec 6.4

Benchmarking

Line: Standard, Expert | Module: Benchmarking | Apps: Desktop | Version: 6.4

The module Benchmarking is replaced by the new Business Intelligence Module (BI Module) and is no longer available in its previous form.

A detailed description of the new Business Intelligence Module is available in chapter 4.

The system setting for checking the existence of documents has been removed

Line: Standard, Expert | Module: PSA | Apps: Desktop | Version: 6.3.0.15

The system setting General > Check existence of document files is no longer available in this release. It has been removed from existing installations as well.

4 Business Intelligence (BI) Module

Line: Standard, Expert | Module: Business Intelligence | Apps: Full featured | Version: 6.4

All data values recorded in Vertec can be processed with the Business Intelligence (BI) Module. The BI Module allows a business-oriented analysis of historical data of the services provided.

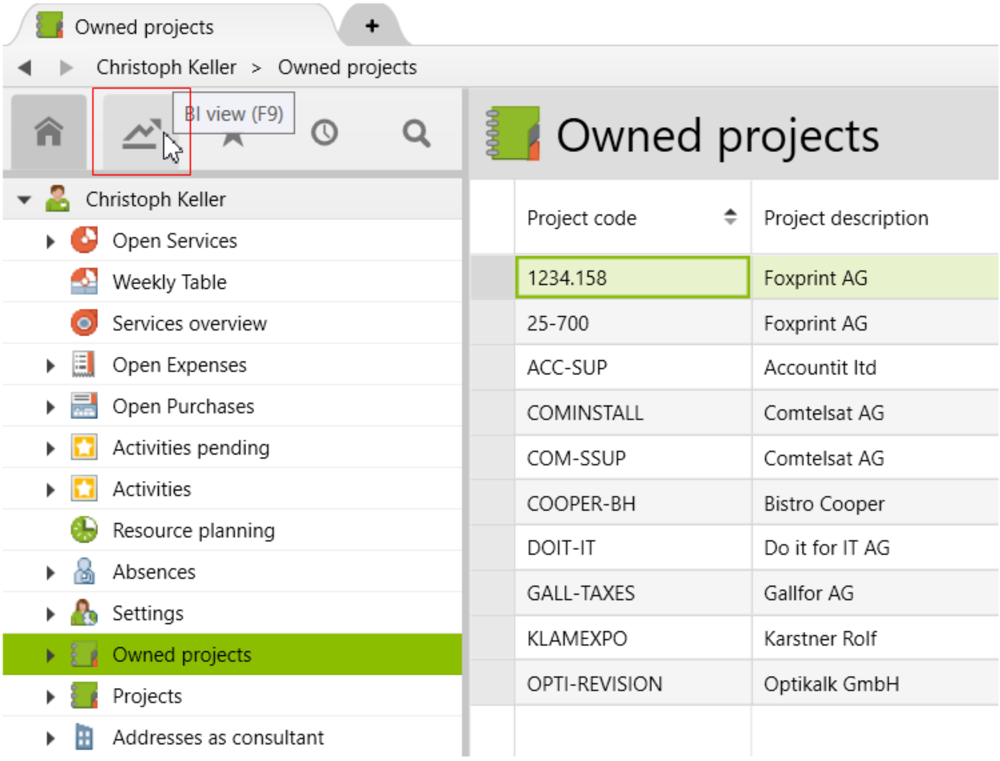
A license is required for the use of the Business Intelligence Module. The BI module replaces the previous Benchmarking. Existing Benchmarking licenses are automatically changed to BI Modul licenses.

To make it possible to efficiently display analyses of large amounts of data over long time periods, the values are precalculated (see section 4.4). When a BI analysis is performed, all predefined and precalculated values and key numbers are available.

The results presented are based on so called BI Displays. They define which values are represented where. The exact definitions and a list of provided BI Displays is given in chapter 4.2.

4.1 Open the BI view

If the Business Intelligence module has been licensed and an object or list of objects is selected for which appropriate BI Displays have been defined, the BI button appears in the navigation view at the top:



Project code	Project description
1234.158	Foxprint AG
25-700	Foxprint AG
ACC-SUP	Accountit ltd
COMINSTALL	Comtelsat AG
COM-SSUP	Comtelsat AG
COOPER-BH	Bistro Cooper
DOIT-IT	Do it for IT AG
GALL-TAXES	Gallfor AG
KLAMEXPO	Karstner Rolf
OPTI-REVISION	Optikalk GmbH

Figure 2: The button to open BI

A click on this button or using the `F9` key (please note the change of shortcuts for navigation views, chapter 8.4) opens the BI view of the selected object or list (in this example Owned projects):

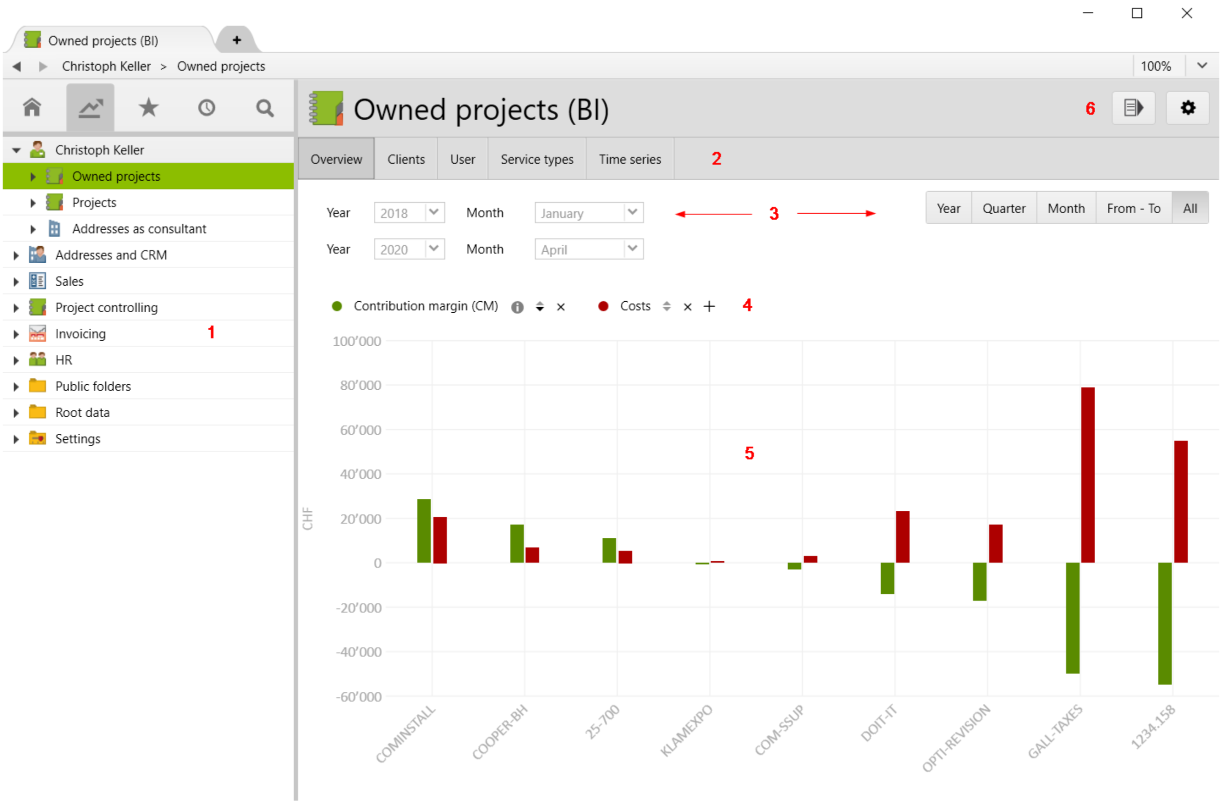
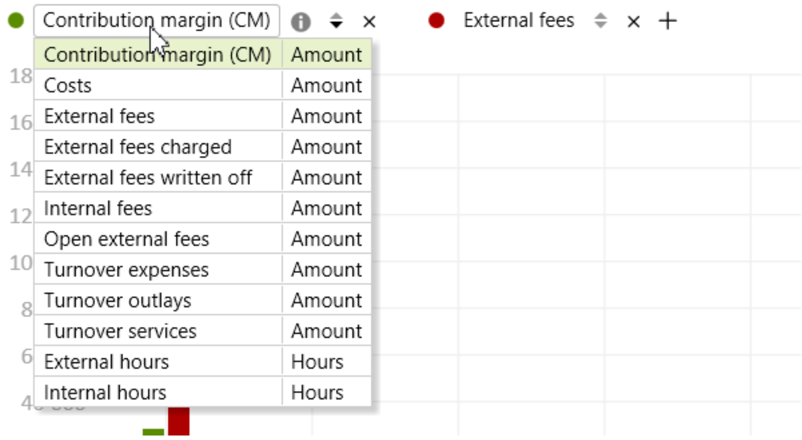


Figure 3: The BI view

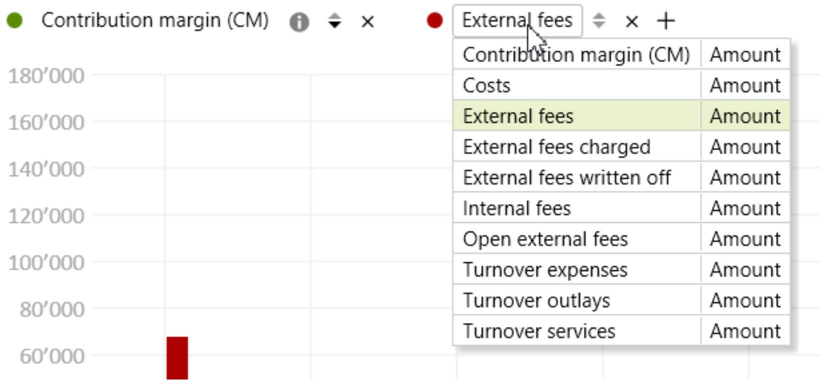
The BI view is structured as follows:

1	In BI view, only the elements for which BI evaluations are available are displayed in the tree, see 4.2.
2	The tabs at the top show the available displays for the selected Item.
3	**Year**, **Quarter**, **Month**, **From – To** and **All** are available to specify the period. The default is **All**. Depending on the defined period, the corresponding settings can be specified on the left. The available period basically depends on the data that has been precomputed. See chapter 4.4. Standard is the period starting from January of the year before last to the end of the current month.
4	Displayed measures. With the **+** button, additional measures can be added to the displayed graphic. The sorting key lets the display be sorted by a specific measure. Section 4.3 explains which measures are available. A click on a measure opens the list of available measures.



Contribution margin (CM)	Amount
Costs	Amount
External fees	Amount
External fees charged	Amount
External fees written off	Amount
Internal fees	Amount
Open external fees	Amount
Turnover expenses	Amount
Turnover outlays	Amount
Turnover services	Amount
External hours	Hours
Internal hours	Hours

With the first measure, the list includes all available measures, as shown above. As can be seen, measures with different units (Amount, Hours) are shown. Since these cannot be displayed in the same graphic, the lists displayed with the following measures only include measures with the same unit:

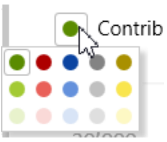


Contribution margin (CM)	Amount
Costs	Amount
External fees	Amount
External fees charged	Amount
External fees written off	Amount
Internal fees	Amount
Open external fees	Amount
Turnover expenses	Amount
Turnover outlays	Amount
Turnover services	Amount

The + button adds the next measure to the list which can then be modified if needed as shown here.

The x Button removes the measre.

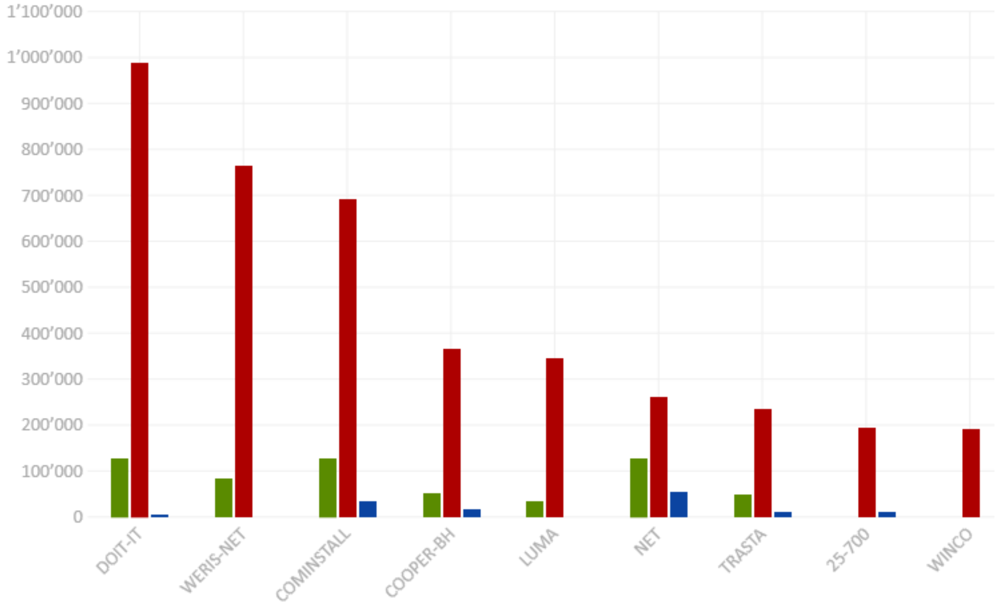
A click on a coloured point causes a colour palette to be displayed. It allows to change the colour of this u for the duration of the display:



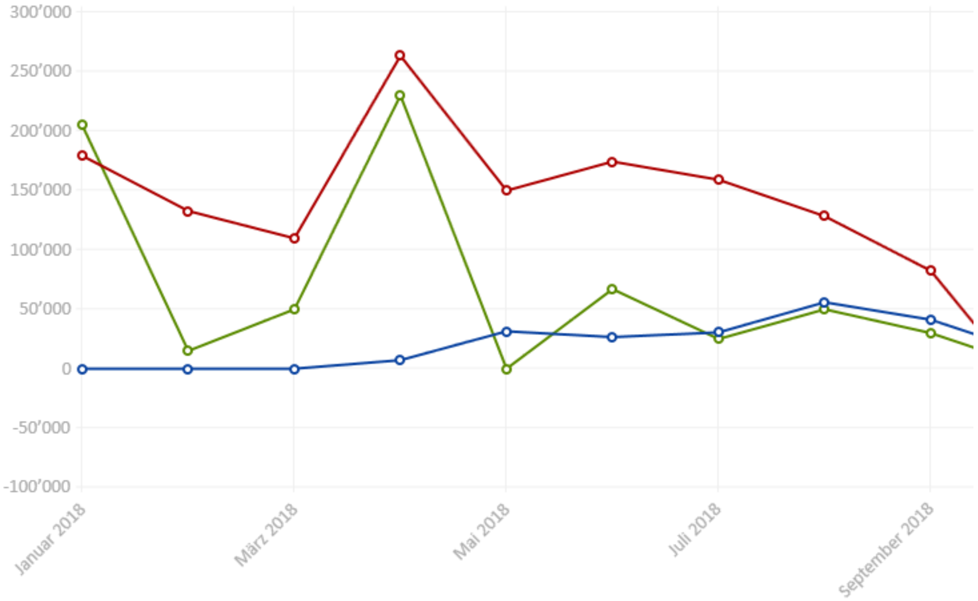
The measures selected for display in a graphic remain if a change of display occurs, provided they are available for the selected display.

5 The figure below shows the values displayed as graphic. The grouping correspond to Grouping in the BI display definitions (4.2).

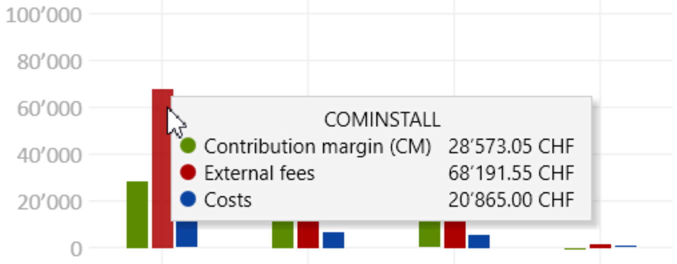
They appear either as a bar chart:



Or, in the case of time series (grouped by month, see 4.2), as a line graphic:



If the mouse pointer is set on a bar or a point, the corresponding value is displayed as a number:

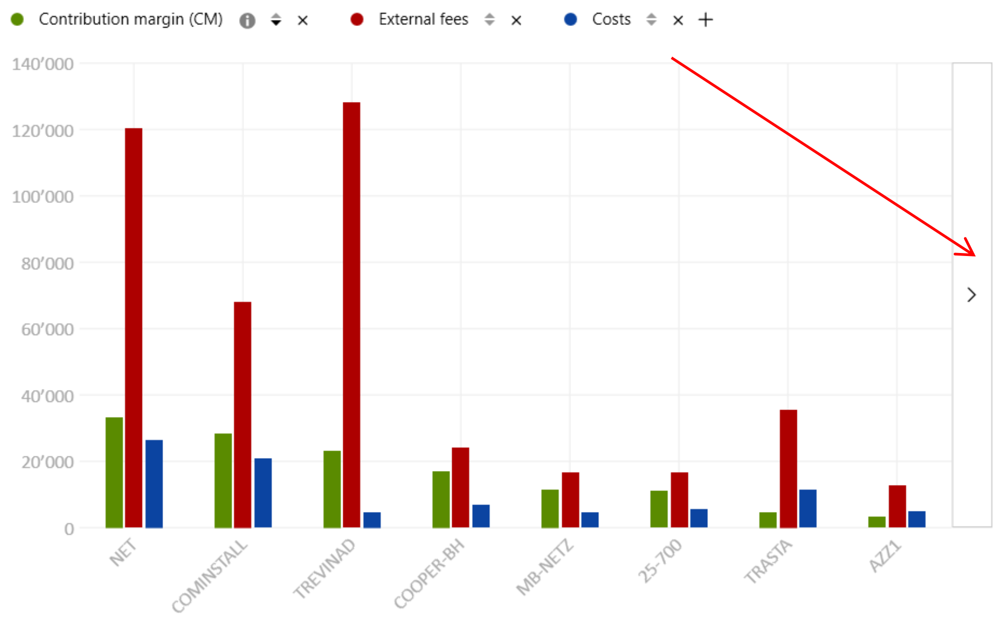


COMINSTALL

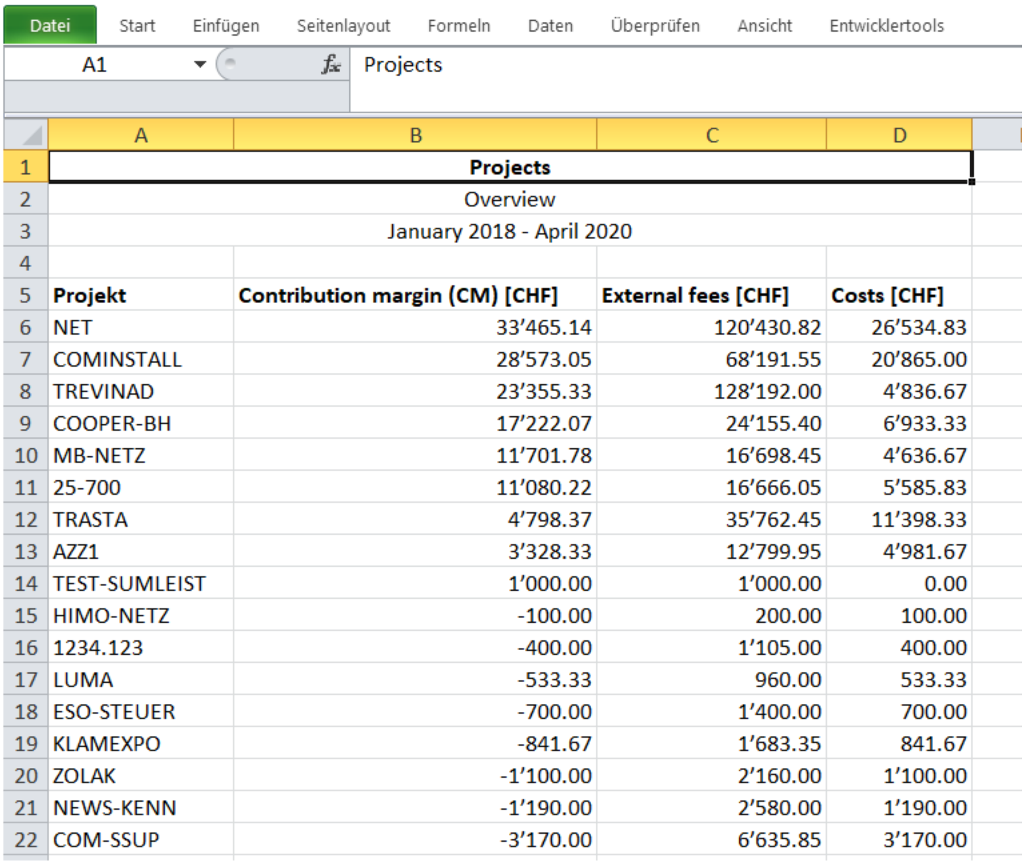
● Contribution margin (CM)	28'573.05 CHF
● External fees	68'191.55 CHF
● Costs	20'865.00 CHF

Double-clicking on a bar causes the corresponding single object to be opened in a separate tab.

If there are more bars than can be displayed in the window, a button appears at the right edge. A click on this button causes all existing bars to be displayed. Another click and only the first ones are displayed.



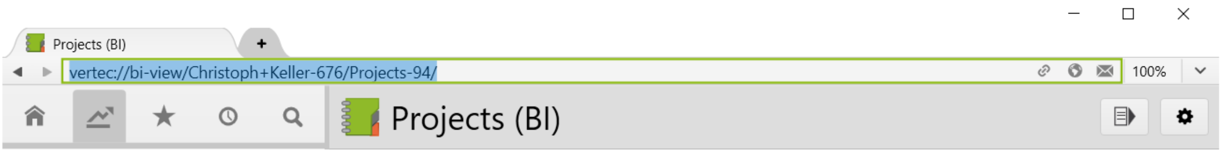
5 By clicking on the Excel Export Button, the displayed BI data is exported to Excel:



	A	B	C	D
1	**Projects**			
2	Overview			
3	January 2018 - April 2020			
4				
5	**Projekt**	**Contribution margin (CM) [CHF]**	**External fees [CHF]**	**Costs [CHF]**
6	NET	33'465.14	120'430.82	26'534.83
7	COMINSTALL	28'573.05	68'191.55	20'865.00
8	TREVINAD	23'355.33	128'192.00	4'836.67
9	COOPER-BH	17'222.07	24'155.40	6'933.33
10	MB-NETZ	11'701.78	16'698.45	4'636.67
11	25-700	11'080.22	16'666.05	5'585.83
12	TRASTA	4'798.37	35'762.45	11'398.33
13	AZZ1	3'328.33	12'799.95	4'981.67
14	TEST-SUMLEIST	1'000.00	1'000.00	0.00
15	HIMO-NETZ	-100.00	200.00	100.00
16	1234.123	-400.00	1'105.00	400.00
17	LUMA	-533.33	960.00	533.33
18	ESO-STEUER	-700.00	1'400.00	700.00
19	KLAMEXPO	-841.67	1'683.35	841.67
20	ZOLAK	-1'100.00	2'160.00	1'100.00
21	NEWS-KENN	-1'190.00	2'580.00	1'190.00
22	COM-SSUP	-3'170.00	6'635.85	3'170.00

URL support for BI views

The Vertec Breadcrumb URL is also available in the BI views:



vertec://bi-view/Christoph+Keller-676/Projects-94/

Figure 4: The Breadcrumb URL in the BI view

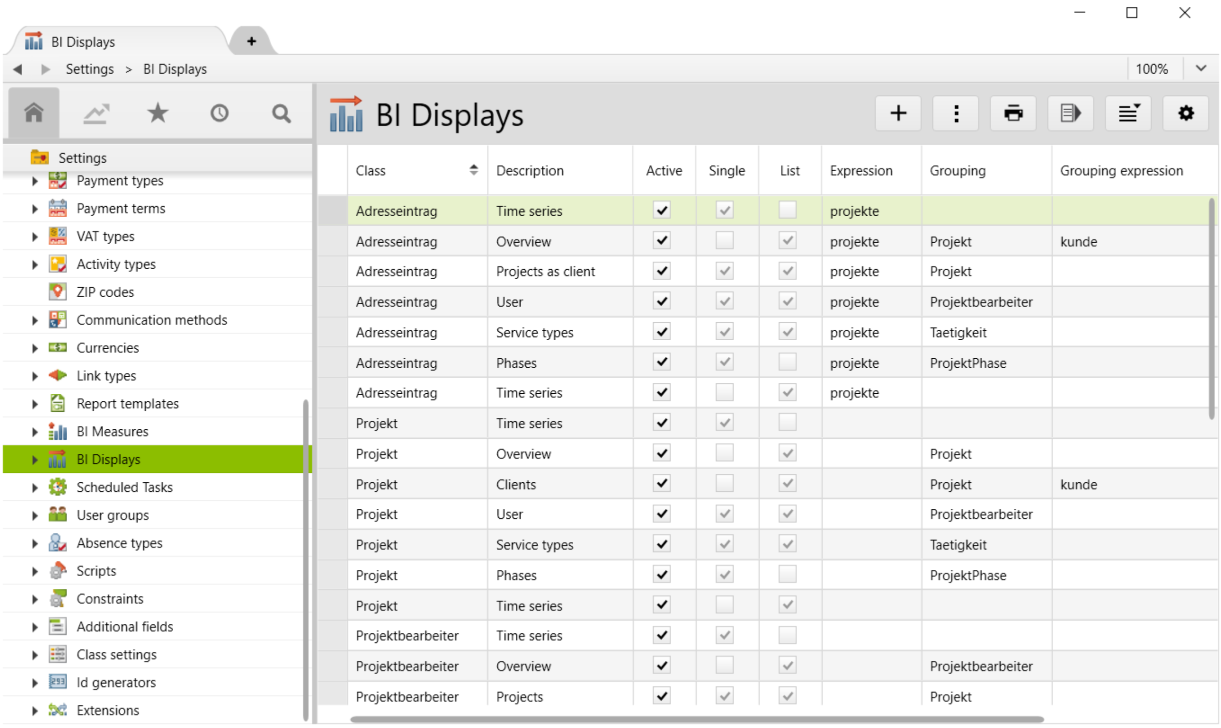
It can be copied and sent with the usual buttons. The path of a BI URL begins with **bi-view**. It represents the object selected in the tree in basic view, hence without parameters such as date interval and selected display.

If a BI path is executed without the required access rights to the view, a corresponding error message is displayed.

Additional information Additional information on Breadcrumb URLs is available in the Online Knowledge Base under www.vertec.com/kb/listennavigation#url.

4.2 BI Displays

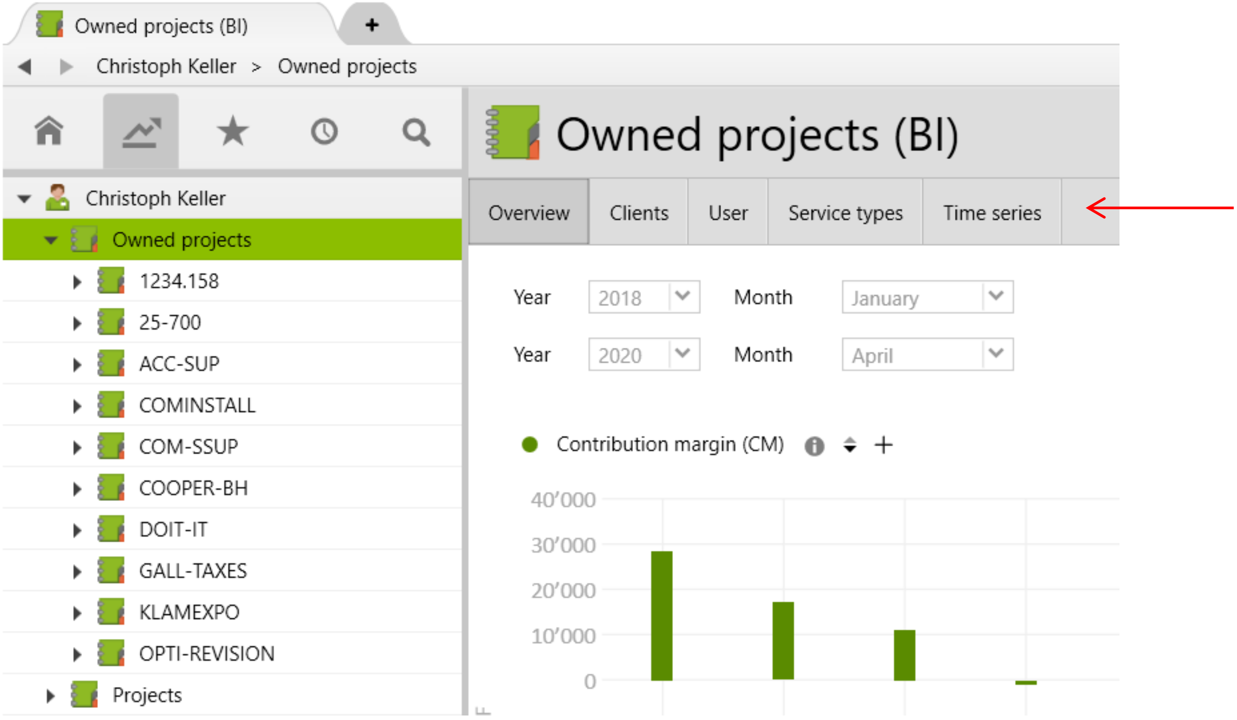
To display data in the BI module, BI Displays must be defined. They are found in the folder **Settings > BI Displays**:



Class	Description	Active	Single	List	Expression	Grouping	Grouping expression
Adresseintrag	Time series	☑	☑	□	projekte		
Adresseintrag	Overview	☑	□	☑	projekte	Projekt	kunde
Adresseintrag	Projects as client	☑	☑	☑	projekte	Projekt	
Adresseintrag	User	☑	☑	☑	projekte	Projektbearbeiter	
Adresseintrag	Service types	☑	☑	☑	projekte	Taetigkeit	
Adresseintrag	Phases	☑	☑	□	projekte	ProjektPhase	
Adresseintrag	Time series	☑	□	☑	projekte		
Projekt	Time series	☑	☑	□			
Projekt	Overview	☑	□	☑		Projekt	
Projekt	Clients	☑	□	☑		Projekt	kunde
Projekt	User	☑	☑	☑		Projektbearbeiter	
Projekt	Service types	☑	☑	☑		Taetigkeit	
Projekt	Phases	☑	☑	□		ProjektPhase	
Projekt	Time series	☑	□	☑			
Projektbearbeiter	Time series	☑	☑	□			
Projektbearbeiter	Overview	☑	□	☑		Projektbearbeiter	
Projektbearbeiter	Projects	☑	☑	☑		Projekt	

Figure 5: Display definitions in the BI module

Each of these BI Displays corresponds to a computation performed by the BI module.



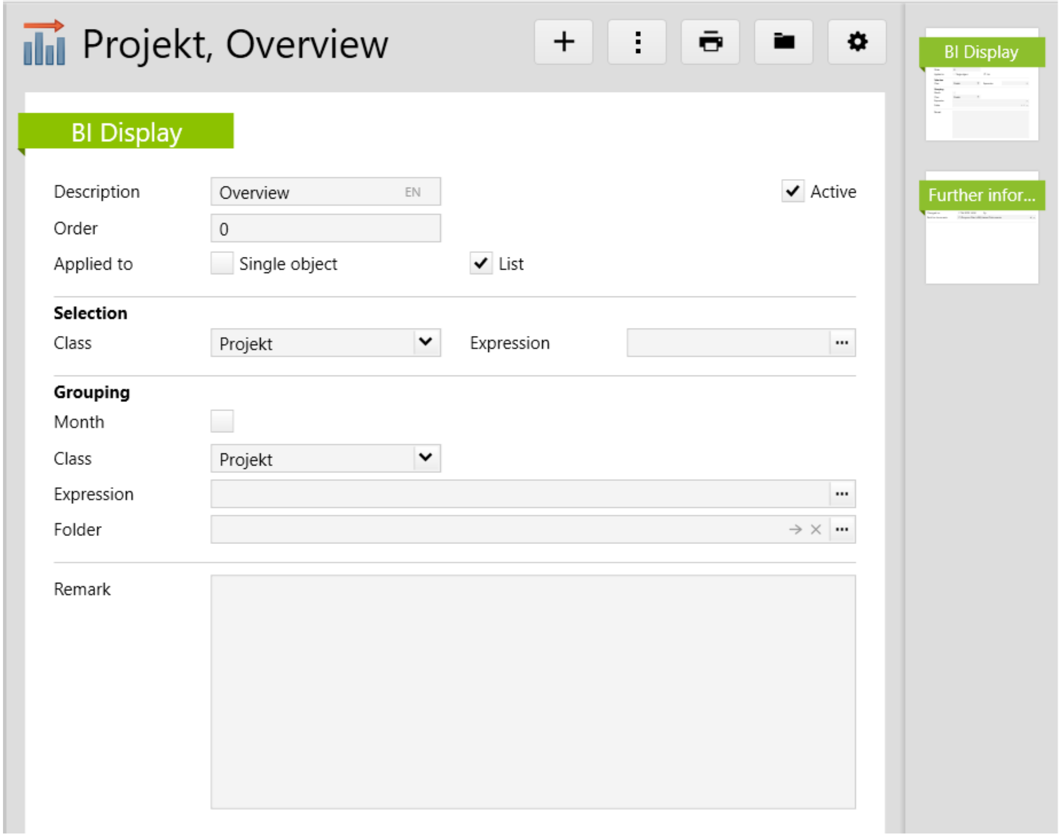
Christoph Keller
- Owned projects
  - 1234.158
  - 25-700
  - ACC-SUP
  - COMINSTALL
  - COM-SSUP
  - COOPER-BH
  - DOIT-IT
  - GALL-TAXES
  - KLAMEXPO
  - OPTI-REVISION
- Projects

Owned projects (BI)

Year 2018 Month January

Year 2020 Month April

A BI Display has the following properties:



Projekt, Overview

BI Display

Description	Overview EN		☑ Active
Order	0		
Applied to	□ Single object	☑ List	

Selection

Class	Projekt	Expression	

Grouping

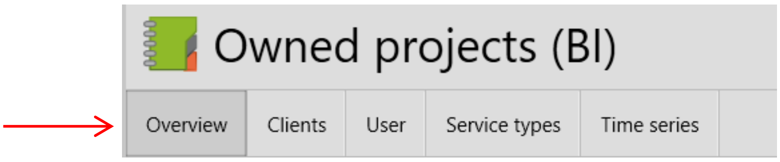
Month	□
Class	Projekt
Expression	
Folder	

Remark

Figure 6: Single view of a BI Display

Description

The Description is displayed as label on the tabs in the BI view.



This is an MLString field (see chapter 8.3).

Active

Defines whether a computation is shown in the BI view.

Order

Defines the order in which the tabs are shown in the BI view.

Applied to single object/list

Defines whether the computation is available for single objects or a list of objects.

Displays defined in Selection and Grouping for an identical class do not make sense with single objects since only one bar would then be displayed.

If computations performed on lists should be represented in a different order as chosen for single objects, they should be defined once for the single objects and once for lists (e.g. to define a standard view). If the order is irrelevant, the same view can be defined for single objects and for lists.

Selection

Selection defines, where the display is shown.

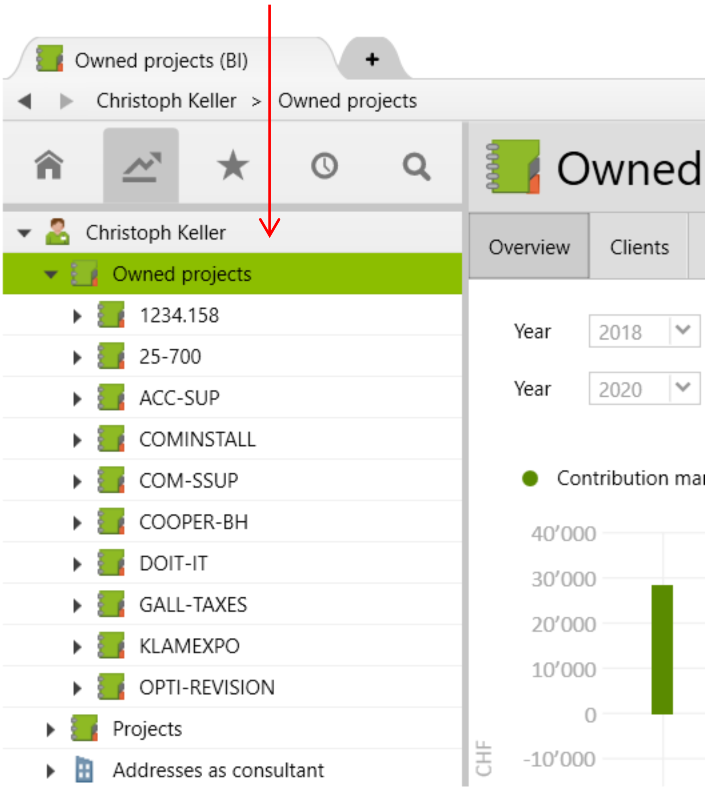


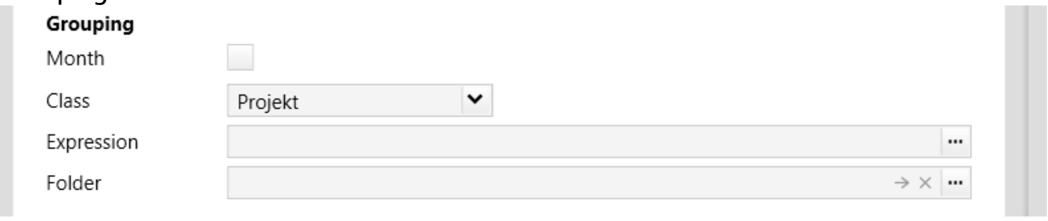
Figure 7: The computations are made available for the defined selections

Class

Defines for which class/classes the view is defined and computations are therefore available.

Expression	An expression is required if the selected class does not have any measures (see section 4.3) that can be evaluated. For instance, addresses. In this case, an OCL expression must be specified here whose result is an object (or a list of objects) that has measures. This does not refer to WHAT is displayed in the evaluation. The grouping is defined for this. It is therefore only a question of where the numbers for the display come from, not how they are displayed.

Grouping



Grouping

Month □

Class: Projekt

Expression:

Folder:

The grouping defines HOW the values should be GROUPED

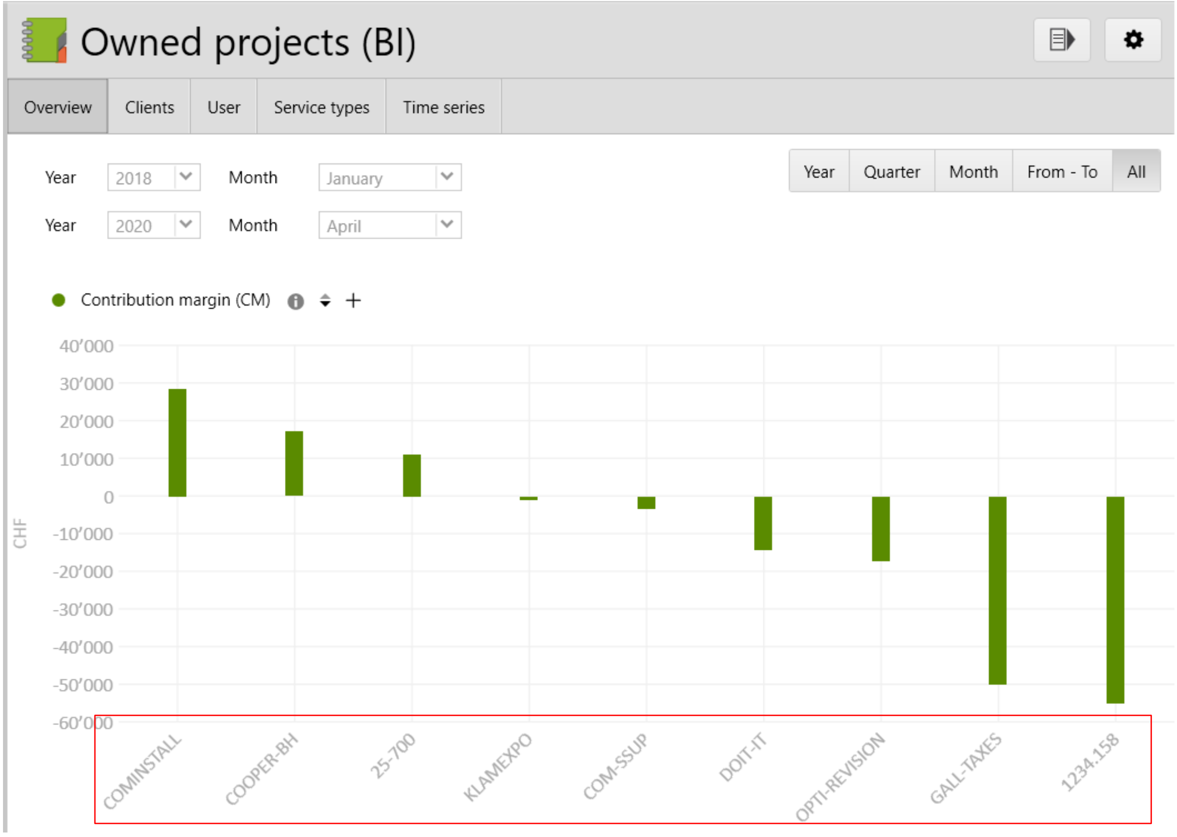
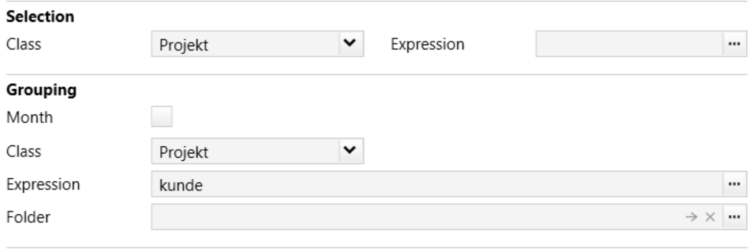


Figure 8: The following groupings are available

Month	If a computation is grouped by month, it is displayed as a time series (see 4.1). In that case the other grouping possibilities are masked out.
Class	Specifies according to which class the values must be grouped. In the example shown here, the values are grouped by project, the same class as specified for the selection. If no expression is given, the measures are simply grouped according to the objects available, Own Projects here. The class

specified here must be available as a dimension on the key figures to be displayed.

Expression This expression can be used to further refine the grouping. It is applied to the resulting objects according to Class. The results of this expression are then grouped and displayed. Here is an example of the "Client" Display. The grouping class is also a project, but in contrast to the overview shown in Figure 6, the OCL expression `kunde` is now given here:



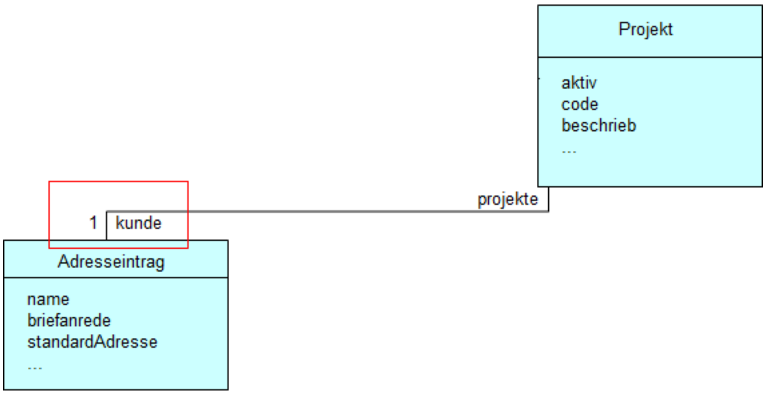
Selection

Class: Projekt
Expression:

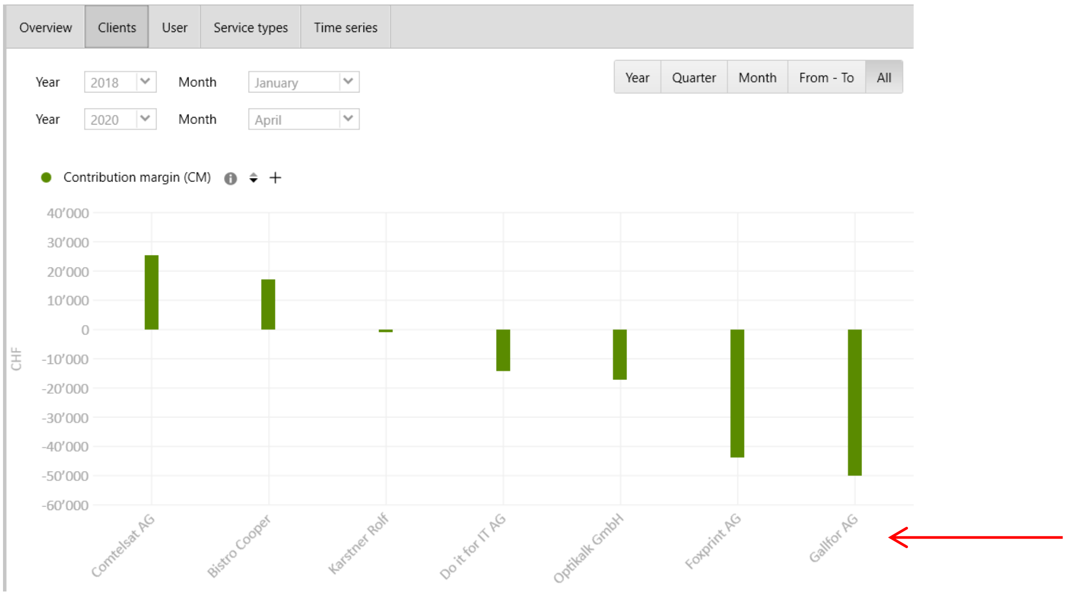
Grouping

Month: □
Class: Projekt
Expression: kunde
Folder:

It is related to the schema (extract):



In this case, the project figures are grouped by client address:

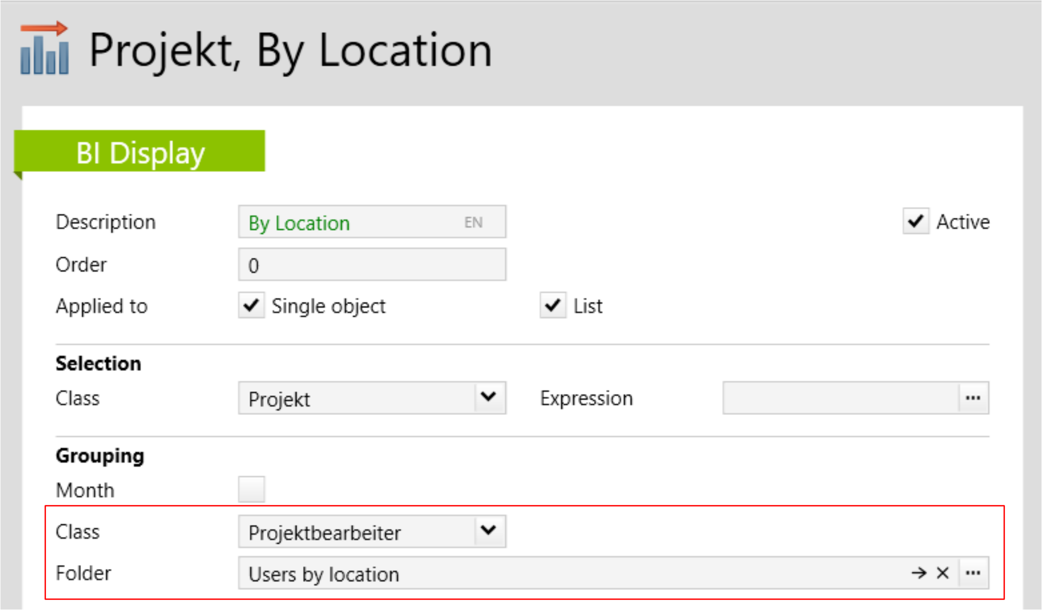


A large number of model schemas can be found in the Online Knowledge Base under www.vertec.com/support/kb/technik-und-datenmodell/modell/uml.

Folder A folder may be specified instead of an expression. In that case, the values are grouped by the objects present in this folder. In the former Benchmarking Module this possibility was called Projections (see section 4.8).

The folder or its subfolders must contain objects of the classes specified for the grouping.

In the following example, the project values in the display are grouped by location of the user. For this purpose, group the display by users (`Projektbearbeiter`) and select the keyword folder Users by location:



Projekt, By Location

BI Display

Description: By Location EN

☑ Active

Order: 0

Applied to: ☑ Single object ☑ List

Selection

Class: Projekt

Expression:

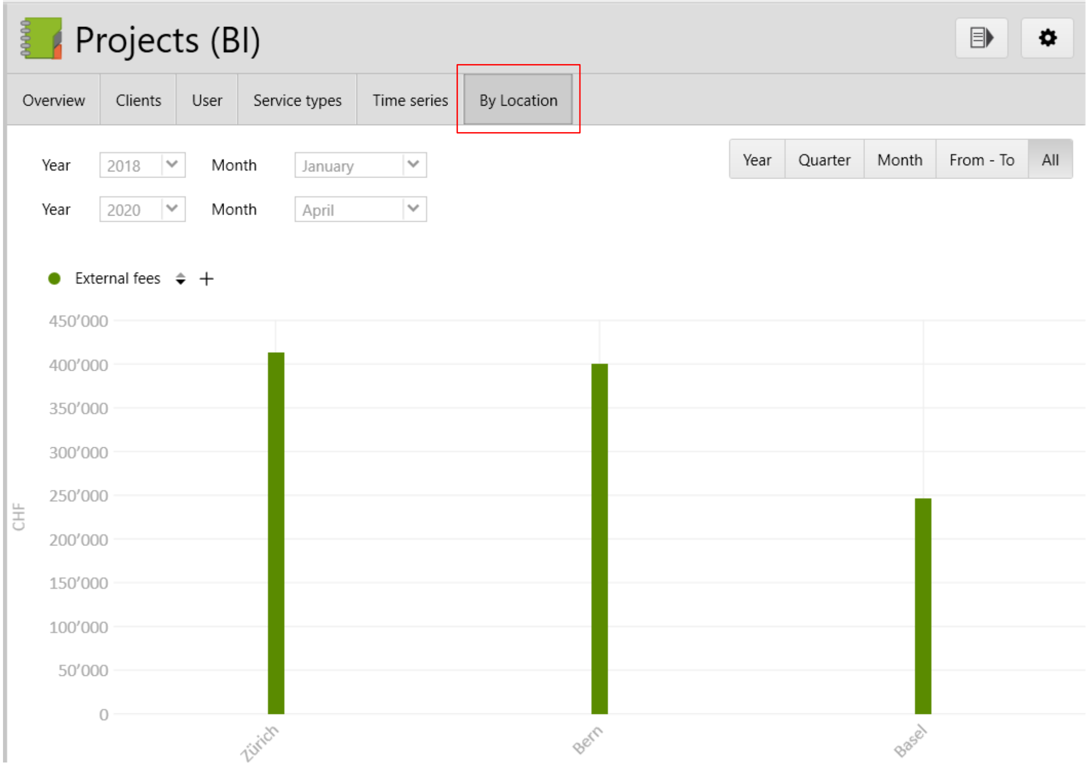
Grouping

Month: □

Class: Projektbearbeiter

Folder: Users by location

The result then includes a tab "By Location", which groups the values accordingly:



Projects (BI)

Year: 2018 Month: January

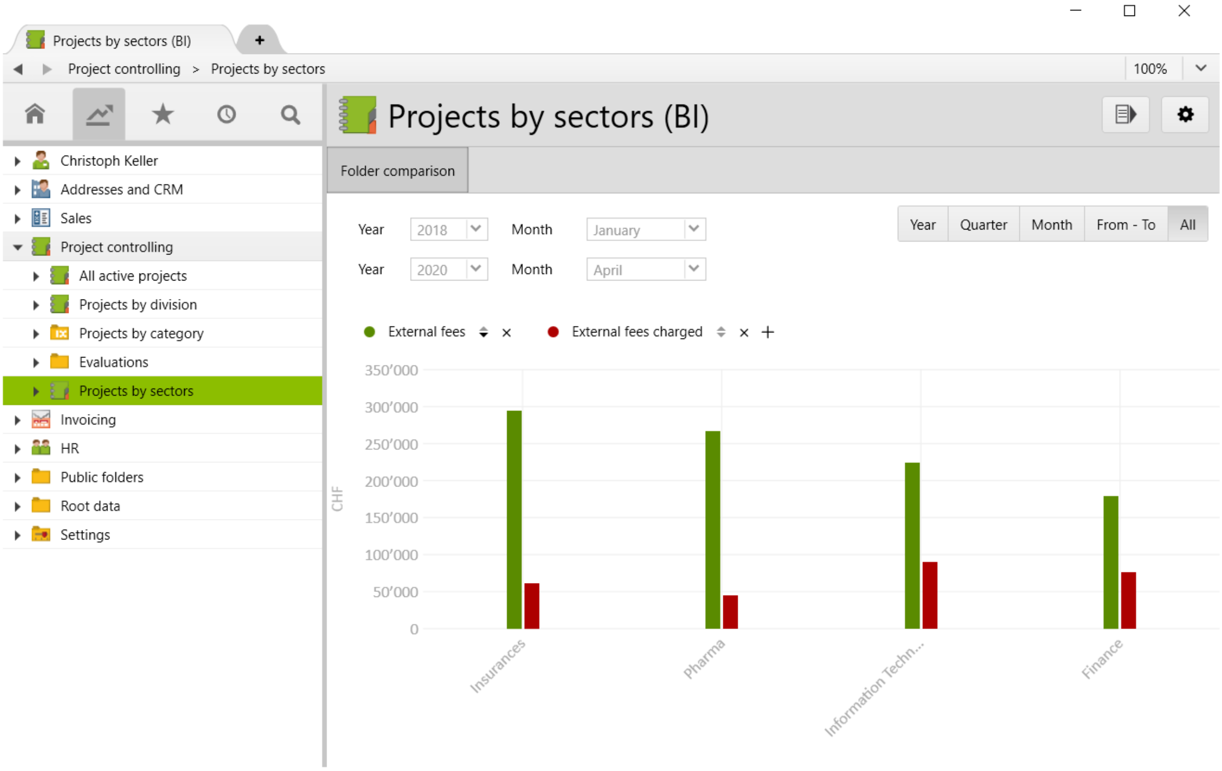
Year: 2020 Month: April

Folder comparisons

If you select a folder in the BI view that contains subfolders with BI selectable objects, a comparison view appears automatically, which displays the data grouped by the folders in which it is located. You do not need to create a BI Display for this. The first hierarchy level is taken into account.

The following conditions must be met for a folder comparison to take place:

– The property Enable Subfolders must be set.
– The folder must have at least one subfolder.
– All subfolders must have the same class.



Year 2018 Month January
Year 2020 Month April

Figure 9: Projects by sectors are displayed in the folder comparison

WHAT is displayed, the values shown, is determined by the BI Measures (see section 4.3) and is not specified with the display definition.

A large number of predefined BI Displays is provided by default in Vertec:

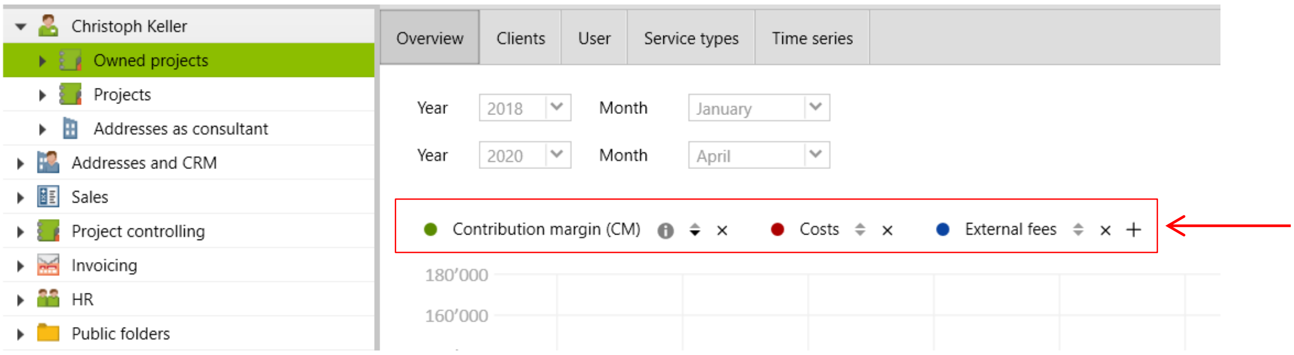
Class	Description	Expression	Grouping / Expression	Description
Adresseintrag	Time series	`projekte`	Month	On single objects: sum of project amounts for this client, per month.
Adresseintrag	Overview	`projekte`	Projekt / `kunde`	Sum of project amounts per client.
Adresseintrag	Projects as clients	`projekte`	Projekt	Amounts per project for a single client or all clients in the list. Corresponds to Project Overview.

Class	Description	Expression	Grouping / Expression	Description
Adresseintrag	User	`projekte`	Projektbearbeiter	Amounts per project collaborator on projects for a single client or all clients in the list.
Adresseintrag	Service types	`projekte`	Taetigkeit	Amounts per service type for the projects of a single client or all clients in the list.
Adresseintrag	Phases	`projekte`	Projektphase	Amounts per project phase for projects of a single client or all clients in the list.
Adresseintrag	Time series	`projekte`	Month	On lists: amount totals for all clients in the list, per month.
Projekt	Time series		Month	On single objects: amount totals for this project, per month.
Projekt	Overview		Projekt	Amounts per project.
Projekt	Clients		Projekt / `kunde`	Amounts grouped by client.
Projekt	Collaborator		Projektbearbeiter	Amounts grouped by collaborator.
Projekt	Service types		Taetigkeit	Amounts grouped by service types.
Projekt	Phases		Projektphase	Amounts grouped by phases.
Projekt	Time series		Month	On lists: amount totals for all projects of the list, per month.
Projektbearbeiter	Time series		Month	On single objects: total of amounts for this user, per month.
Projektbearbeiter	Overview		Projektbearbeiter	Amounts by user.
Projektbearbeiter	Projects		Projekt	User amounts grouped by projects.
Projektbearbeiter	Clients		Projekt / `kunde`	User amounts grouped by client.
Projektbearbeiter	for Project Managers		Projekt / `projektleiter`	User amounts grouped by project manager.
Projektbearbeiter	For Persons in charge		Project / `hB`	User amounts grouped by persons in charge.
Projektbearbeiter	Service types		Taetigkeit	User amounts grouped by service types.

Class	Description	Expression	Grouping / Expression	Description
Projektbearbeiter	Owned projects	`eigprojekte`	Projekt	Amounts for the project manager's own projects, grouped by projects.
Projektbearbeiter	User owned projects	`eigprojekte`	Projektbearbeiter	Amounts for the project manager's own projects, grouped by users.
Projektbearbeiter	Time series		Month	On lists: amount totals for all collaborators in the list, per month.
Projektphase	Time series		Month	On single objects: amount totals for this project phase, per month
Projektphase	Overview		Projektphase	Amounts per project phase.
Projektphase	Service type		Taetigkeit	Amounts for project phases, grouped by activity.
Projektphase	Project		Projekt	Amounts for project phases, grouped by project.
Projektphase	User		Projektbearbeiter	Amounts for project phases, grouped by user.
Projektphase	Time series		Month	On lists: Amount totals for all project phases of the list, per month.
Taetigkeit	Time series		Month	On single objects: amount totals for this activity, per month.
Taetigkeit	Overview		Taetigkeit	Amounts per service type.
Taetigkeit	User		Projektbearbeiter	Amounts per service type, grouped by user
Taetigkeit	Project		Projekt	Amounts for service types, grouped by project.
Taetigkeit	Clients		Projekt / `kunde`	Amounts for service types, grouped by client.
Taetigkeit	Time series		Month	On lists: amount totals for all service types in the list, per month.

4.3 BI Measures

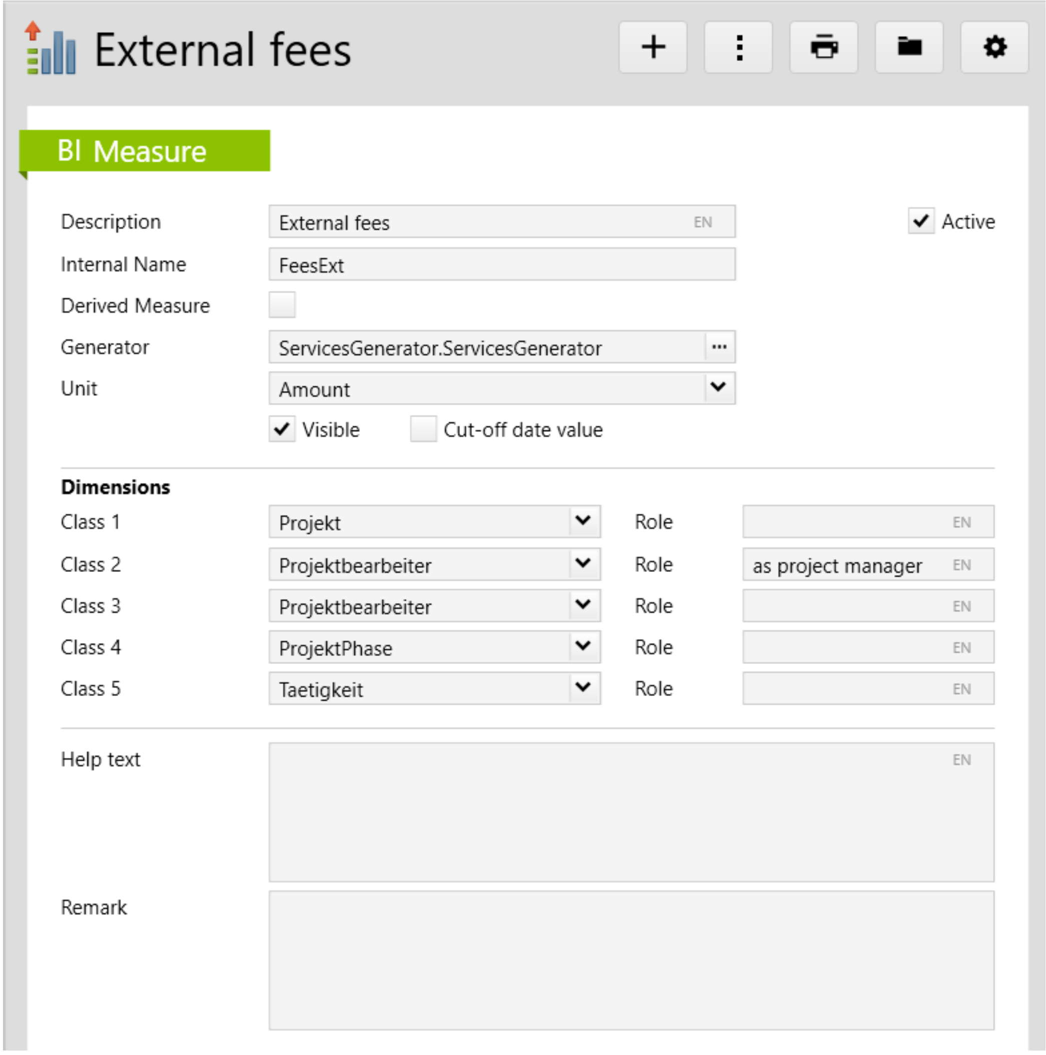
The various types of available values are called BI Measures:



Year 2018 Month January

Year 2020 Month April

The measures are defined in the folder Settings > BI Measures. Each measure has the following attributes:



External fees

BI Measure

Description: External fees (EN)

☑ Active

Internal Name: FeesExt

Derived Measure: □

Generator: ServicesGenerator.ServicesGenerator

Unit: Amount

☑ Visible □ Cut-off date value

Dimensions

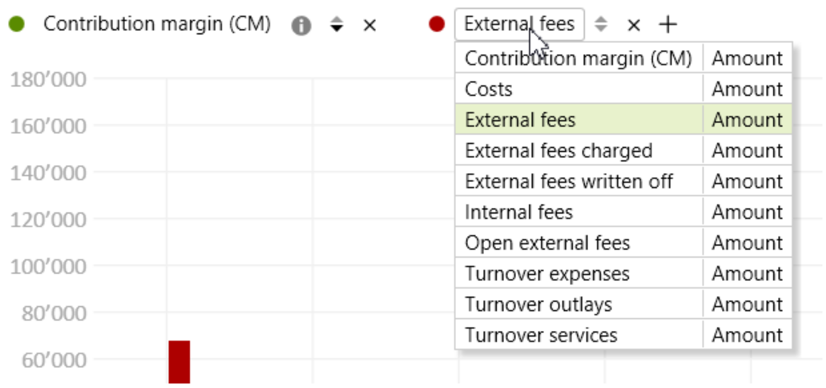
Class 1	Projekt	Role	EN
Class 2	Projektbearbeiter	Role	as project manager EN
Class 3	Projektbearbeiter	Role	EN
Class 4	ProjektPhase	Role	EN
Class 5	Taetigkeit	Role	EN

Help text: EN

Remark:

Figure 10: Definition of a BI measure

Description — Description as MLString (see 8.3). The description is displayed in the BI view and should therefore be unambiguous and meaningful to clearly describe what is displayed.



Contribution margin (CM)	Amount
Costs	Amount
External fees	Amount
External fees charged	Amount
External fees written off	Amount
Internal fees	Amount
Open external fees	Amount
Turnover expenses	Amount
Turnover outlays	Amount
Turnover services	Amount

Active

Only active measures are precalculated and stored. If a measure provided by default with the program is not needed, it can be deactivated.

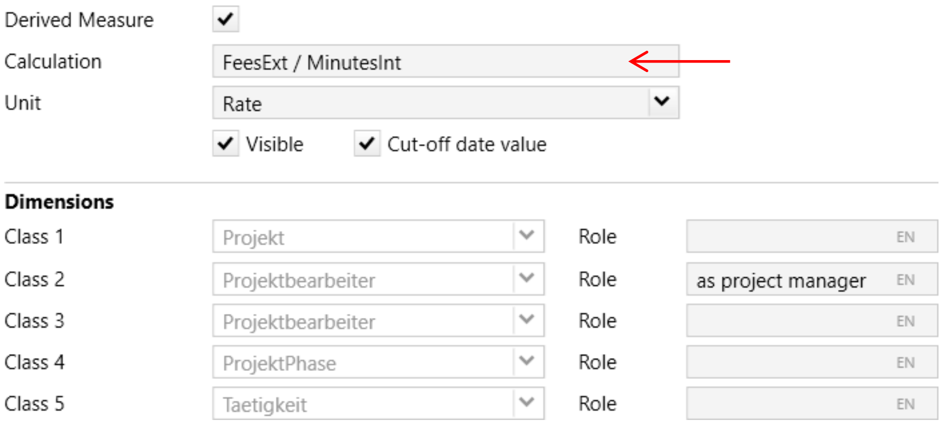
If, however, a measure should be calculated but not displayed, deactivate it with the checkbox Visible. See below.

Internal name

The internal name must be unique since the measure is identified with this name in the generator code. More on this topic in section 4.4.

Derived Measure

If a measure is defined as derived, its calculation is not performed with generator code, but from other measures. In this case, a field for the calculation is displayed instead of the field Generator:



Derived Measure ☑

Calculation: FeesExt / MinutesInt

Unit: Rate

☑ Visible ☑ Cut-off date value

Dimensions

Class 1	Projekt	Role	
Class 2	Projektbearbeiter	Role	as project manager
Class 3	Projektbearbeiter	Role	
Class 4	ProjektPhase	Role	
Class 5	Taetigkeit	Role	

The calculation is defined using the internal names of the corresponding measures. In this example, an hourly rate is calculated: (external fees / internal hours).

The dimensions are fixed by the underlying measures and cannot be separately defined here.

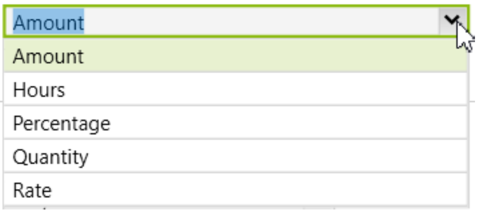
For the time being, only divisions are supported.

Generator

The computation of the measures occurs by means of generators (4.4), except for derived measures (see attribute Derived Measure).

Units

The following units are available:



- Amount: numerical values, displayed according to country settings
- Hours: number of minutes, displayed according to system setting Display Minutes
- Percentage: percentage values displayed as %. The values are interpreted as quotients based on 1. 1 is therefore displayed as 100%.
- Quantity: a numerical value with 2 decimals.
- Rate: displayed according to country settings.

Visible

Defines whether a measure is visible in the BI display. Particularly useful if a measure is used to compute another measure (see Derived Measure above) but should not be displayed.

Measures provided by default but not needed at all can be deactivated (see checkbox Active above).

Cut-off date value

Each measure has a time aspect. It is either related to the selected period or is a cut-off date value. Cut-off dates always apply to the end of a month.

Cut-off date values only appear in the following BI views:

- Computation for 1 month (date interval in the BI view, 4.1)
- Presentation as a time series (the BI view is grouped by month, 4.2)

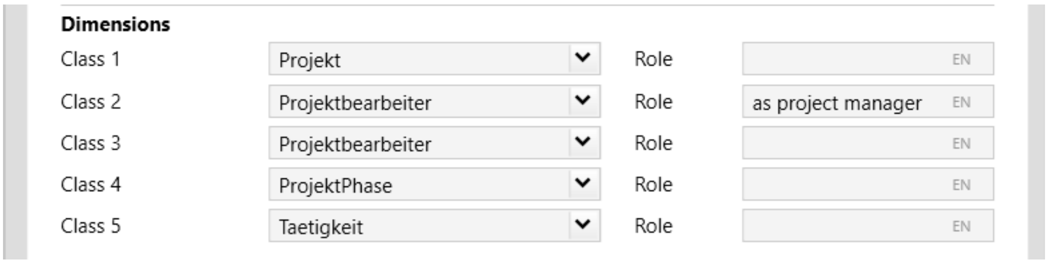
Dimensions 1 – 5,

Roles

Defines, on which classes (dimensions) the measures are available. They must be computed with the specified generator.

Additionally, a role can be specified for each dimension. This role allows to indicate the context of the measure for this dimension.

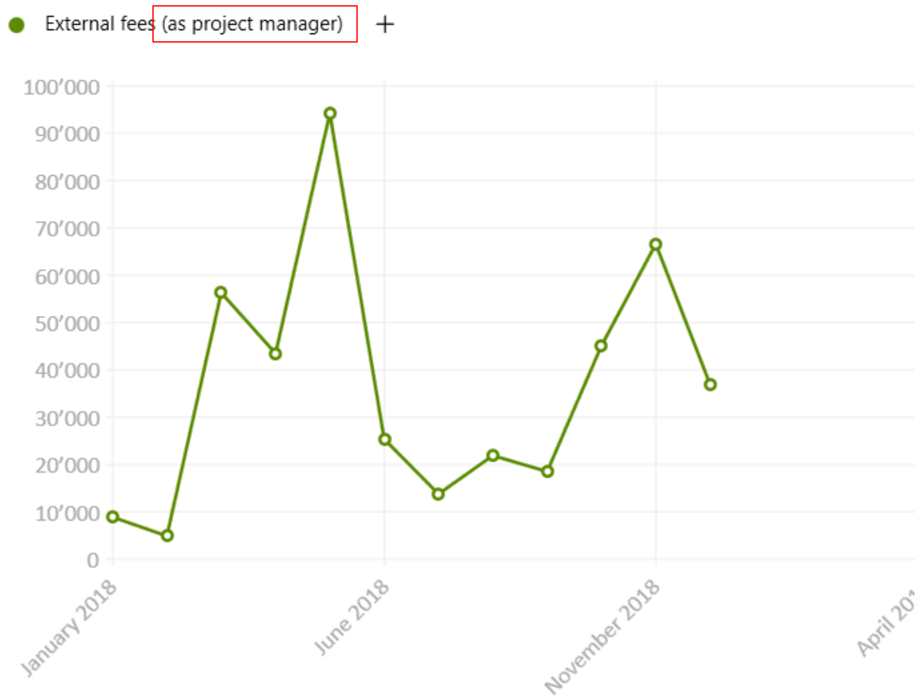
Example: If, on a list of users, "External fees" is displayed, the context is not clear: is it as user or as project manager? This role can be specified:



Dimensions

Class 1	Projekt	Role	EN
Class 2	Projektbearbeiter	Role	as project manager EN
Class 3	Projektbearbeiter	Role	EN
Class 4	ProjektPhase	Role	EN
Class 5	Taetigkeit	Role	EN

The role will then be displayed in the BI view:



Help text

A help text may be entered here, for instance to explain to the user how the measure is computed.

If a help text has been entered, a small i-symbol appears in the view. By placing the mouse pointer on this symbol, the help text appears as a tooltip:

The display is cut off after 19 lines. When the text is entered care must be taken that the text fits into the display.

By default, Vertec provides the following measures:

Description / Int. name	Unit	Meaning	Available on

Commenced Word
`FeesCommenced`
	Amount	External fees for the services of productive projects, which are not on an invoice at the end of the month or whose invoice was not settled until later.
		`if not leistsum.rechnung or leistsum.rechnung.valutadatum > enddate:`
		`leistsum.wertExtOffen + leistsum.wertExtVerrechnet`
			– Project

External hours `MinutesExt`	Hours	External hours `leistsum.minutenExtOffen + leistsum.minutenExtVerrechnet`	– Project – User – Project phase – Service type

Description / Int. name	Unit	Meaning	Available on
Internal hours `MinutesInt`	Hours	Internal Hours `leistsum.minutenIntOffen + leistsum.minutenIntVerrechnet`	– Project – User – Project phase – Service type

Stock of advances
`Advances`
	Amount	Net amount of all advances - advances paid on invoices by the end of the month (booking date).
		`advances += advance.nettoAmount`
		`if invoice.verrechnet and invoice.valutadatum <= enddate:`
		`advances -= invoice.vorschussEffektiv`
			– Project

Contribution margin (CM) `MarginContributed`	Amount	External fees charged - Costs `leistsum.wertExtVerrechnet - leistsum.wertKostenVerrechnet`	– Project – User – Project phase – Service type
Vacation balance `VacationBalance`	Hours	Vacation balance by the end of the month `->getFerienSaldo(enddate)`	– User
Delimited vacation balance `VacationBalanceDelimited`	Hours	Delimited vacation balance by the end of the month `->getFerienSaldoAbgegrenzt(enddate)`	– User
External fees `FeesExt`	Amount	External fees `leistsum.wertExtOffen + leistsum.wertExtVerrechnet`	– Project – User (as project manager) – User – Project phase – Service type
External fees written off `FeesExtWrittenoff`	Amount	Internal fees charged - External fees charged `leistsum.wertIntVerrechnet - leistsum.wertExtVerrechnet`	– Project – User (as project manager) – User – Project phase – Service type
Open external fees `FeesExtOpen`	Amount	Open external fees `leistsum.wertExtOffen`	– Project – User (as project manager) – User – Project phase – Service type
External fees charged `FeesExtCharged`	Amount	External fees charged `leistsum.wertExtVerrechnet`	– Project – User (as project manager) – User – Project phase – Service type

Description / Int. name	Unit	Meaning	Available on
Internal fees `FeesInt`	Amount	Internal fees `leistsum.wertIntOffen + leistsum.wertIntVerrechnet`	– Project – User (as project manager) – User – Project phase – Service type
Costs `Costs`	Amount	Costs `leistsum.wertKostenOffen + leistsum.wertKostenVerrechnet`	– Project – User (as project manager) – User – Project phase – Service type
Working time `WorkingTime`	Hours	Working Time `->getArbeitszeit(startdate, enddate)`	– User
Standard hours `SetTime`	Hours	Set time `->getSollzeit(startdate, enddate)`	– User
Rate chargeable `RateChargeable`	Rate	External fees / Internal hours (Calculation: see the corresponding measures)	– User
Overtime balance `OvertimeBalance`	Hours	Overtime balance by the end of the month `->getUeberzeitsaldo(enddate)`	– User
Turnover outlays `TurnoverOutlays`	Amount	Purchases charged (booking date of the invoice) `Invoice.auslagenExt`	– Project – User Only available if module Purchases is licensed
Turnover services `TurnoverServices`	Amount	Total after reduction charged (booking date of the invoice) `Invoice.leistWertExtNachRabatt`	– Project – User
Turnover expenses `TurnoverExpenses`	Amount	Expenses charged (booking date of the invoice) `Invoice.spesenExt`	– Project – User
Full time equivalents `Fte`	Percentage	Level of capacity `->getBeschaeftigungsgrad(enddate)`	– User

In addition, the following measures are provided with status Inactive. To use them, a license for the module Budget & Phases, Expert line, as well as an exact budget planification are prerequisites.

Description / Int. name	Unit	Meaning	Available on
Order backlog fees `FeesOrderbacklog`	Amount	Fees budget on phases - accumulated external fees. Only accepted and not yet concluded	– Project – User – Project phase

Description / Int. name	Unit	Meaning	Available on

		phases.
		`bdgvalue('planWertExt',enddate,-1) -`
		`leistSum.wertExtOffen + leistSum.wertExtVerrechnet`	

Order intake fees
`FeesOrderintake`
	Amount	Fees budget on newly accepted phases or changes to this budget value during this month.
		`bdgvalue('planWertExt',enddate,-1) -`
		`bdgvalue('planWertExt',(startdate-1 day), -1)`
			– Project
			– User
			– Project phase

Remaining budget hours `MinutesBudgetRest`	Hours	Total budget hours - Cumulated hours (Calculation: see the corresponding measures)	– Project – User – Project phase
Total budget hours `MinutesBudgetTotal`	Hours	Budget hours of accepted and not yet concluded phases `bdgvalue('planMinutenInt', enddate, -1)`	– Project – User – Project phase
Cumulated hours `MinutesCumulated`	Hours	Accumulated internal hours on accepted and not yet concluded phases `leistSum.minutenIntOffen + leistSum.minutenIntVerrechnet`	– Project – User – Project phase
Percentage of completion `PercentageOfCompletion`	Percentage	Cumulated hours / Total budget hours (Calculation: see the corresponding measures)	– Project – User – Project phase

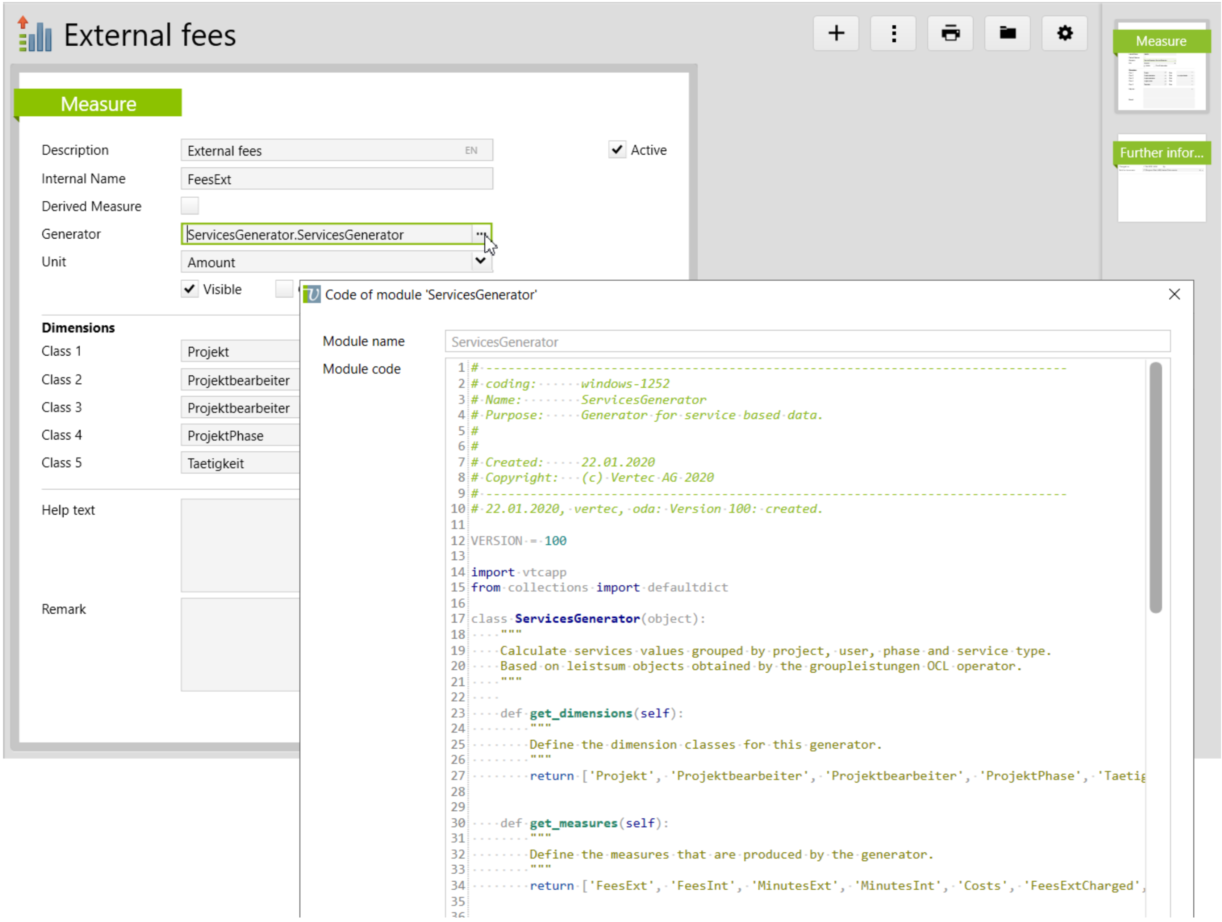
4.4 Generators

To make the BI data available for analysis, a data store called data warehouse, or OLAP cube, is set up. When the axes of the desired analysis are selected, a corresponding section is cut out of this cube

The data warehouse consists of a huge data table (cube) with precalculated values. This ensures an efficient access to the BI data independently of the business logic and a better performance displaying the results even with large data volumes.

The precomputation of the values is performed in a time-independent process with support of the full business logic by periodic batch jobs (scheduled tasks). In this way, the computation can, if necessary, occur for certain parts (single months, for example) in separate process, even be performed in parallel. See section 4.6.

The computation logic is defined in Python scripts called «generators». These generators are stored with the measures (4.3). To inspect the code in a generator, click on the button with the three points next to the Generator field of the measure:



External fees

Measure

Description: External fees EN

☑ Active

Internal Name: FeesExt

Derived Measure: □

Generator: ServicesGenerator.ServicesGenerator

Unit: Amount

☑ Visible

Dimensions

Class 1: Projekt

Class 2: Projektbearbeiter

Class 3: Projektbearbeiter

Class 4: ProjektPhase

Class 5: Taetigkeit

Help text

Remark

Code of module 'ServicesGenerator'

Module name: ServicesGenerator

Module code:

```
# -----------------------------------------------------------------------------
# coding:      windows-1252
# Name:        ServicesGenerator
# Purpose:     Generator for service based data.
#
#
# Created:     22.01.2020
# Copyright:   (c) Vertec AG 2020
# -----------------------------------------------------------------------------
# 22.01.2020, vertec, oda: Version 100: created.

VERSION = 100

import vtcapp
from collections import defaultdict

class ServicesGenerator(object):
    """
    Calculate services values grouped by project, user, phase and service type.
    Based on leistsum objects obtained by the groupleistungen OCL operator.
    """
    
    def get_dimensions(self):
        """
        Define the dimension classes for this generator.
        """
        return ['Projekt', 'Projektbearbeiter', 'Projektbearbeiter', 'ProjektPhase', 'Taetig


    def get_measures(self):
        """
        Define the measures that are produced by the generator.
        """
        return ['FeesExt', 'FeesInt', 'MinutesExt', 'MinutesInt', 'Costs', 'FeesExtCharged',

```

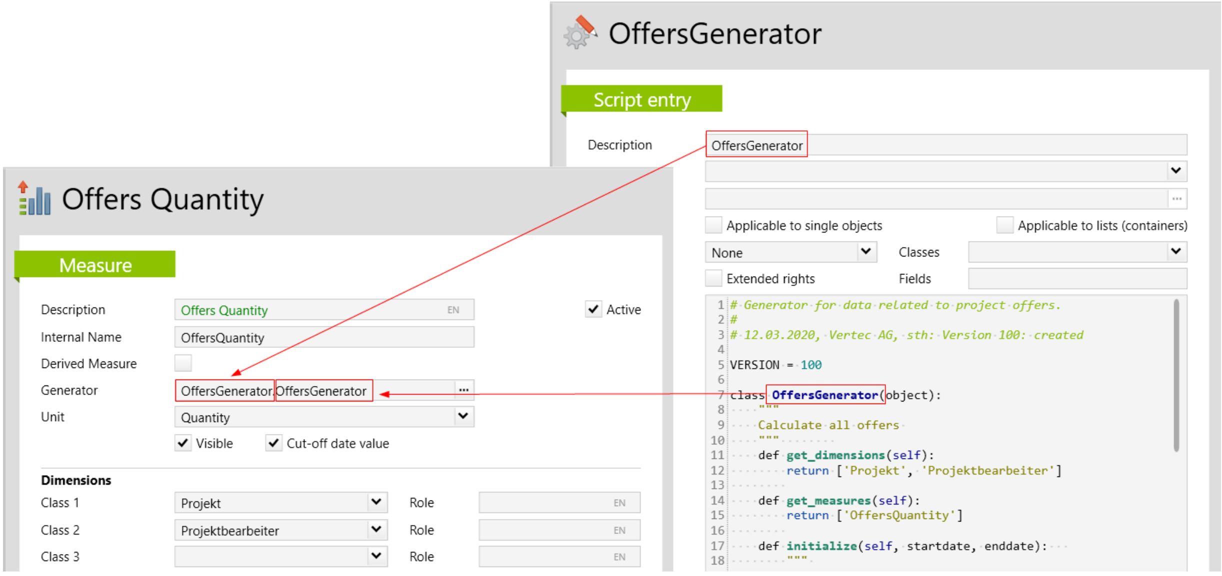
Figure 11: Python code of a generator

The generator holds the computation for the measure to which it is related. The internal name of the measure must therefore correspond to the values returned by the generator.

4.5 Create/extend generators

Existing generators can be extended and new generators can be created. For this, create a new script entry in Settings > Scripts and specify the generator code as Python script.

The name of the script entry is the module name. The generator ID consists of this name and the class name of the generator. These names can be specified as generator names for all BI measures computed by the generator.



OffersGenerator

Script entry

Description: OffersGenerator

☐ Applicable to single objects ☐ Applicable to lists (containers)

None | Classes

☐ Extended rights | Fields

```
# Generator for data related to project offers.
#
# 12.03.2020, Vertec AG, sth: Version 100: created

VERSION = 100

class OffersGenerator(object):
    """
    Calculate all offers
    """
    def get_dimensions(self):
        return ['Projekt', 'Projektbearbeiter']

    def get_measures(self):
        return ['OffersQuantity']

    def initialize(self, startdate, enddate):
        """
```

Offers Quantity

Measure

Description	Offers Quantity	☑ Active
Internal Name	OffersQuantity	
Derived Measure	☐	
Generator	OffersGenerator OffersGenerator	
Unit	Quantity	
	☑ Visible ☑ Cut-off date value	

Dimensions

Class 1	Projekt	Role	
Class 2	Projektbearbeiter	Role	
Class 3		Role	

Figure 12 The name of the script entry (module name) plus the name of the generator in the script are specified for the measure

Module names must be unique. An existing generator can therefore not be «overwritten» by developing a new module with the same name. Create instead a new generator for the corresponding measures.

A generator is declared as follows:

```
class <Generatorname>(object):
```

The generator object can at all times be accessed with `self`. Within the generator, the following methods are available:

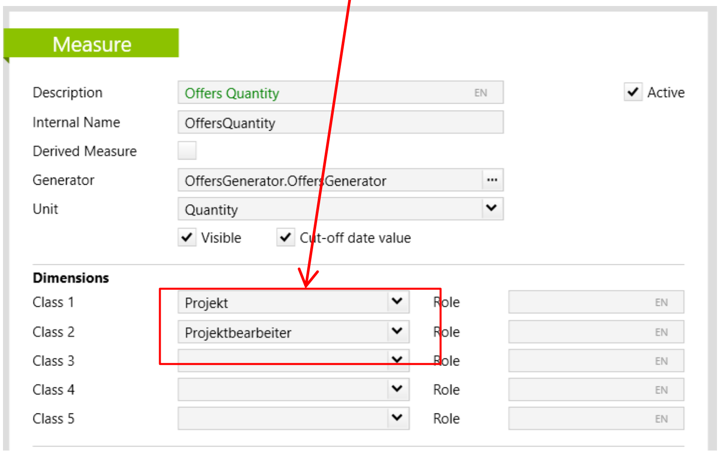
Method — Description

get_dimensions(self)

Specify for which dimensions the generator computes values:

```
def get_dimensions(self):
    return ['Projekt', 'Projektbearbeiter']
```

The same number of dimensions in the same order must be specified in the key values as are returned here:



Measure

Description	Offers Quantity	☑ Active
Internal Name	OffersQuantity	
Derived Measure	☐	
Generator	OffersGenerator.OffersGenerator	
Unit	Quantity	
	☑ Visible ☑ Cut-off date value	

Dimensions

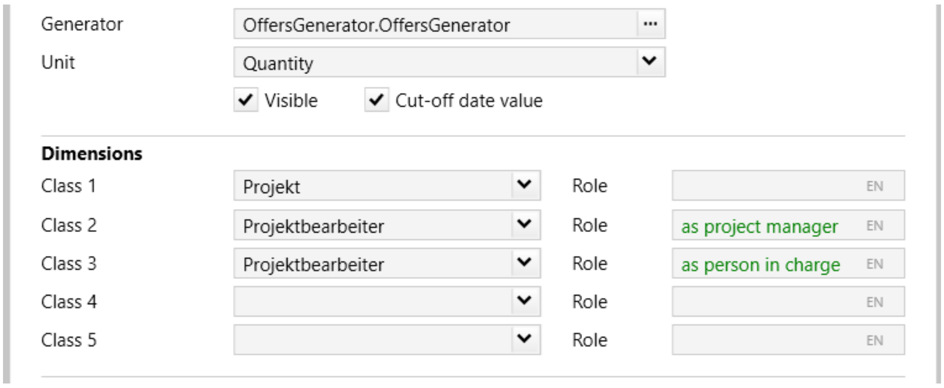
Class 1	Projekt	Role	
Class 2	Projektbearbeiter	Role	
Class 3		Role	
Class 4		Role	
Class 5		Role	

If, for example, one wishes to display values for the project manager and the person in charge of each project, the dimension Projektbearbeiter must be specified twice.

Method Description

```
def get_dimensions(self):
    return ['Projekt','Projektbearbeiter',
                'Projektbearbeiter']
```

And, on the measure:



Generator OffersGenerator.OffersGenerator

Unit Quantity

☑ Visible ☑ Cut-off date value

Dimensions

Class 1	Projekt	Role	
Class 2	Projektbearbeiter	Role	as project manager
Class 3	Projektbearbeiter	Role	as person in charge
Class 4		Role	
Class 5		Role	

get_measures(self)

Specify with this method which measures are computed. The returned values must correspond to the internal names of the measures.

```
def get_measures(self):
    return ['OffersQuantity']
```

The generator can then be applied to all the measures which are returned here.

initialize(self, startdate, enddate)

This is the first method executed. The variables `startdate` and `enddate` define the date limits of the period for the computation
(see section 4.6 below).

This allows to preload the lists of the used objects, optimized for performance. All the relevant data for the full period is now available. The values are then computed for each month with the method `generate()`.

generate(self, startdate, enddate)

The method generate() is executed once for each month of the specified period.

The variables `startdate` and `enddate` always specify the first and last days of the month.

The data preloaded with initialize() can now be filtered and computed for each month.

Returned value

Returned values are a tuple of dimensions and a tuple of measures (measures) in the exact same order as declared above.

The returned value is not obtained with return() (as normally in Python) but with `yield()`:

```
yield((project, project.projektleiter), (OffersQuantity,))
```

With the use of yield(), the returned result is not a single value (in which case all the data would have to be painfully included in an array). Instead, each data row is returned until the program code comes to an end. In this way, the data for various projects can for instance be continually delivered with a loop over projects (see example code below).

Method	Description
	Currencies If more than one currency is used in Vertec, it must be taken into account for the computation. For this purpose, it is optionally possible to return a currency value with each returned data row (example from the provided InvoiceGenerator): `yield ((project), turnover_tuple, project.waehrung)` In this case, Vertec takes care by itself of the correct conversion to the main currency. Values are always displayed in the main currency.
finalize(self)	The finalize method is executed at the end. It allows for instance to remove no longer used objects from memory.

Code example

```
# Generator for data related to project offers.
#
# 12.03.2020, Vertec AG, sth: Version 100: created

VERSION = 100

class OffersGenerator(object):
    """
    Calculate all offers
    """
    def get_dimensions(self):
        return ['Projekt', 'Projektbearbeiter']

    def get_measures(self):
        return ['OffersQuantity']

    def initialize(self, startdate, enddate):
        """
        Preload all projects that could have offers in given date range
        """
        # collect all corresponding offers
        offers = vtcapp.getwithsql("Offerte", "datum<=%s AND datum>=%s" %
(vtcapp.sqldateliteral(enddate), vtcapp.sqldateliteral(startdate)), "")
        if offers:
            # collect all corresponding projects
            self.projects = vtcapp.getwithsql("Projekt", "bold_id IN (SELECT
projekt from offerte where bold_id in (%s))" % offers.idstring(), "")
        else:
            self.projects = []

    def generate(self, startdate, enddate):
        """
        Calculate measures for given period. Uses the stored projects from the
initialization step.
        """
        for project in self.projects:
            # calculate quantity for current month
            OffersQuantity = project.evalocl("offerten->select((datum<=%s) and
(datum>=%s))->size" % (vtcapp.ocldate(enddate), vtcapp.ocldate(startdate)))

            yield((project, project.projektleiter), (OffersQuantity,))

    def finalize(self):
        pass
```

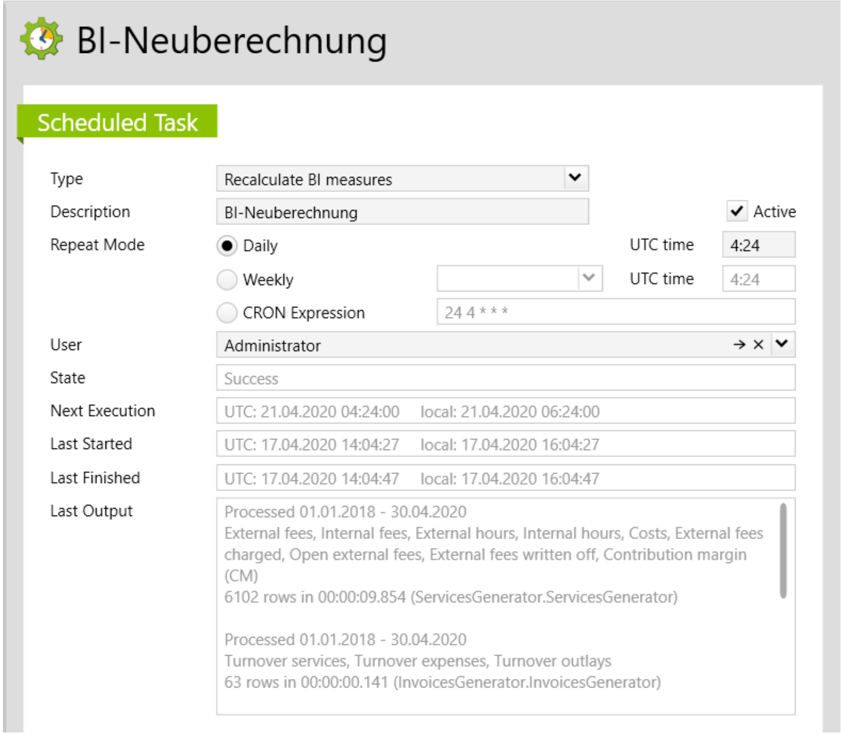
Vertec provides the following standard generators:

Generator name	Used data	Computed measures
ServicesGenerator.ServicesGenerator	Leistungssummen (global operator	– Costs

Generator name	Used data	Computed measures
	`groupLeistungen`) grouped by project, phase, user and service type.	– External fees – External fees charged – Open external fees – External fees written off – Internal fees – Contribution margin (CM) – External hours – Internal hours
ProjectGenerator.ProjectGenerator	Leistungssummen (global operator `groupLeistungen`) grouped by project and invoice. Only uses productive projects.	– Stock of advances – Commenced work
UsersGenerator.UsersGenerator	Users with entry date and leaving date not before the end date, respectively with leaving date after the start date.	– Full time equivalents – Overtime balance – Standard hours – Vacation balance – Delimited vacation balance – Working time
InvoicesGenerator.InvoicesGenerator	Booked invoices with value date within the computed date interval.	– Turnover expenses – Turnover outlays – Turnover services
PhasesGenerator.PhasesGenerator	Top-level project phases which have a Date accepted and whose Date accepted is before the end date , and which have not been concluded or only after the start date: – Budget changes of these phases – Leistungssummen for these phases (`groupLeistungenPWG`)	– Order backlog fees – Order intake fees – Remaining budget hours – Total budget hours – Cumulated hours

4.6 Processing values

The measures are regularly processed over night by one or several scheduled tasks. Scheduled tasks are found in the folder Settings > Scheduled Tasks. Additional information on scheduled tasks in section 8.1.



BI-Neuberechnung

Scheduled Task

Type: Recalculate BI measures

Description: BI-Neuberechnung

☑ Active

Repeat Mode:
● Daily — UTC time 4:24
○ Weekly — UTC time 4:24
○ CRON Expression — 24 4 * * *

User: Administrator

State: Success

Next Execution: UTC: 21.04.2020 04:24:00 local: 21.04.2020 06:24:00

Last Started: UTC: 17.04.2020 14:04:27 local: 17.04.2020 16:04:27

Last Finished: UTC: 17.04.2020 14:04:47 local: 17.04.2020 16:04:47

Last Output:

```
Processed 01.01.2018 - 30.04.2020
External fees, Internal fees, External hours, Internal hours, Costs, External fees
charged, Open external fees, External fees written off, Contribution margin
(CM)
6102 rows in 00:00:09.854 (ServicesGenerator.ServicesGenerator)

Processed 01.01.2018 - 30.04.2020
Turnover services, Turnover expenses, Turnover outlays
63 rows in 00:00:00.141 (InvoicesGenerator.InvoicesGenerator)
```

Figure 13: A scheduled task to recalculate the BI measures

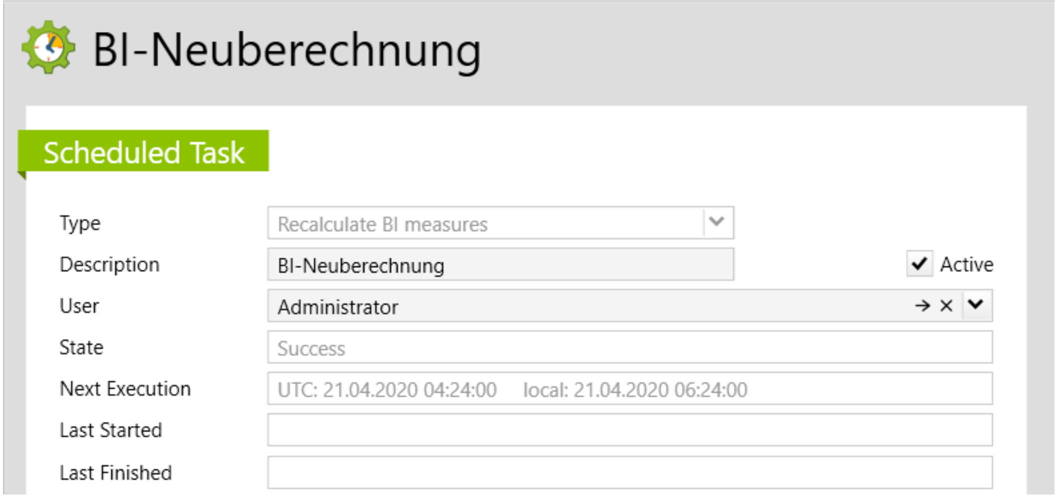
With this procedure, all generators specified for the active measures are activated and the values are precalculated for the date interval from January of the year before last to the end of the current month.

The state of the task indicates whether the computation was successful or failed. The last output displays the message returned by the computation.

It is not necessary for this purpose to have a running Vertec App, the only prerequisite is a running cloud server. For the execution of the batch calculation, a user is specified for this scheduled task. This user must have Administrator access rights to execute the computation.

Cloud Abo Version

In the Cloud Abo, the time cannot be changed on the interface. The computation is always performed during the night. At what time exactly is displayed under Next execution



BI-Neuberechnung

Scheduled Task

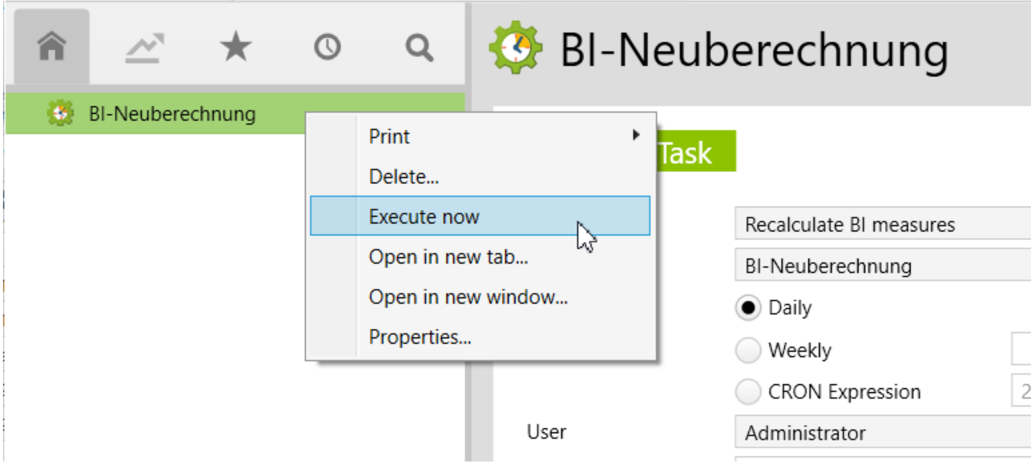
Type	Recalculate BI measures
Description	BI-Neuberechnung ☑ Active
User	Administrator
State	Success
Next Execution	UTC: 21.04.2020 04:24:00 local: 21.04.2020 06:24:00
Last Started	
Last Finished	

Figure 14: Specification of the execution time occurs automatically in Cloud Abo

Start the computation manually

A BI computation can also be started manually. Two variants are available:

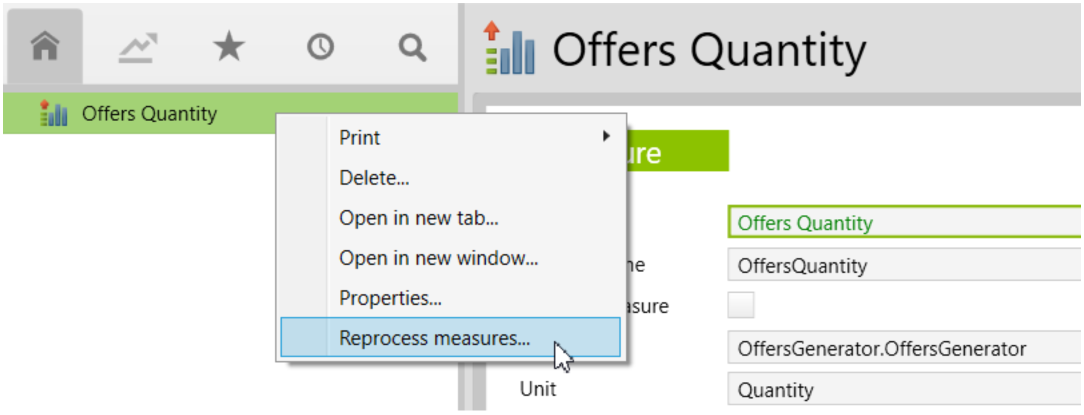
- Execute immediately a scheduled task: select `Execute now` in the context menu:



  This starts the execution of all generators on all active measures.

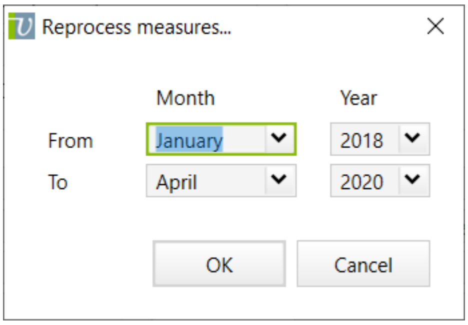
  Careful: The calculation is started directly in the running Vertec App and not in a separate Task Runner process like the scheduled execution. This should not be done during normal working hours as the computation may take a very long time and cause Vertec to be blocked.

- Triggering the calculation of a generator: the calculation may be started manually on a BI measure by selecting `Reprocess measures…` in the context menu:



Offers Quantity
OffersQuantity
OffersGenerator.OffersGenerator
Quantity

In this case, the generator attached to the measure is started and the corresponding measures are recomputed. The date interval can be set as well:



From: January 2018
To: April 2020

A message with the computed measures is returned at the end.

Python methods to compute the BI data

Two new Python methods are available to compute the BI data.

vtcapp.standardprocessbi()	Starts the computation for all generators on all active measures. Requires Administrator access rights. Corresponds to the execution of the scheduled task provided by default, see above.
vtcapp.processbi(from: String, to: String, generator: String)	Starts the computation for the generator specified in the parameter. Requires Administrator access rights. **From and To dates must be specified as strings with format** "Year-Month". The generator must be specified in the same way as for measures: "<Modulname>.<Generatorname>". `vtcapp.processbi("2018-01", "2020-02", "OffersGenerator.OffersGenerator")` Corresponds to the computation of the BI data for one measure, see above.

4.7 BI Authorizations

A new BI Data authorization has been defined to access the BI data.

To see the results of the computation, the user must own this access right for the selected class (see chapter 4.2 about BI displays).

The following standard access rights have been defined:

- A project manager has access rights to the BI data of his/her own projects.
- A project supervisor has general access rights to the BI data.
- A team manager has access rights to the BI data of all members of his/her team.

A new OCL-Operator `hasBiDataRight` has been defined which tells in each case whether the logged-in user owns the necessary BI data access rights.

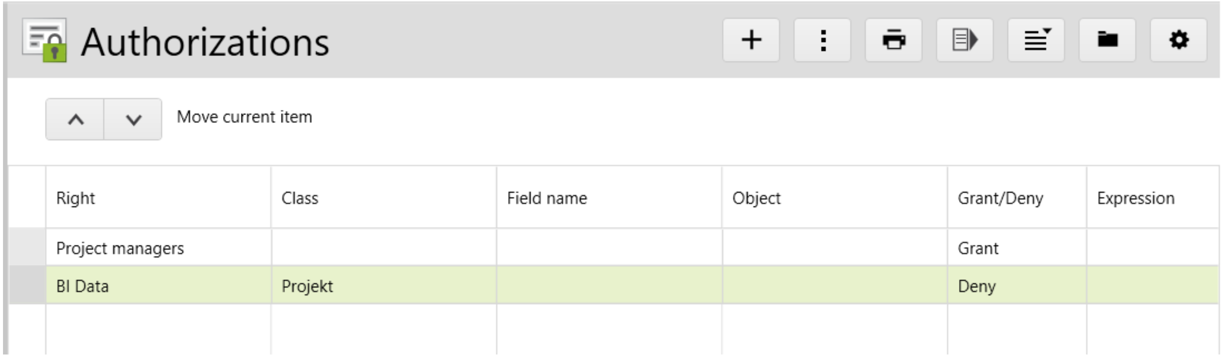
The following happens if the required rights are missing:

- Users without BI access rights to projects will, if attempting a computation on projects or project lists, see a special display with the message ("No access to BI data for projects").
- The access to a specific object can be controlled with object access control. If a user does not have BI data access rights to an object, the error "Access to an object of type 'Project' has been denied" is displayed.
- Users with BI data access rights to projects, but not to team members, will nevertheless see the results if computations are performed on projects grouped by team members. Only the original dimension (selected in the BI panel) is relevant for the access rights check, not the dimension displayed (group by in the BI panel).
- For users without any BI access rights, the button for the BI view is hidden. This only happens if the user does not, with absolute certainty, have any access rights to BI data. In complicated situations, the button for the BI view will appear even if in fact the user is not allowed to see any BI calculation.

Revoking the BI Data right of a user with implicit authorization

If you still do not want to grant the BI right to a user with implicit authorization (for example, project manager, see above), you can revoke this right as follows:

- Insert an additional line on the corresponding group (for example, for project manager)
- Selecting the BI Data right
- Important: You must specify the class for which the BI Data right is to be denied. Without this specification, the line has no effect.
- If you want to revoke all permissions for BI Data, you have to enter `UserEintrag` as class.



Right	Class	Field name	Object	Grant/Deny	Expression
Project managers				Grant	
BI Data	Projekt			Deny	

Figure 15: Project managers are denied the BI Data right on projects

4.8 Backwards compatibility

The Business Intelligence module is a totally new product. It replaces the old Benchmarking module, but does not offer backwards compatibility in the usual sense.

Here is a list of differences with the old Benchmarking.

Computation periods

The Benchmarking computation periods no longer exist. This functionality is split into two parts:

1. The values for the period ranging from the year before last to the present year are precalculated every night and stored in the data base. See section 4.6.
2. In the BI view, the period can be freely specified (within the precalculated period) for each computation, see section 4.1.
3. All values are always presented at least for a full month.

Computations

All list-computations previously available in Benchmarking are included in the new BI module and provided by default.

The aggregation of detailed views no longer exists.

Projections: see at the end of this section.

Folder comparisons

The Bi module supports only one hierarchy level for folder comparisons. They appear automatically on all BI selectable objects, see Section 4.2.

Graphics vs. numbers

The existing BI Module displays the results of computations in graphical form. The actual values can be seen by moving the mouse pointer to a specific bar or line point.

The full display can be exported to Excel, where the values will be fully visible.

On this subject see section 4.1.

Authorizations

The old authorization "Benchmarking" no longer exists. The new BI data authorization is fully explained in section 4.7. Rights are not transferred with the update, meaning that users who previously owned Benchmarking rights do not automatically own the new BI data access right. Access rights must therefore be newly attributed after the update. Please take note of the fact that certain BI access rights are implicitly given to project managers, project supervisors and team leaders. More information in section 4.7.

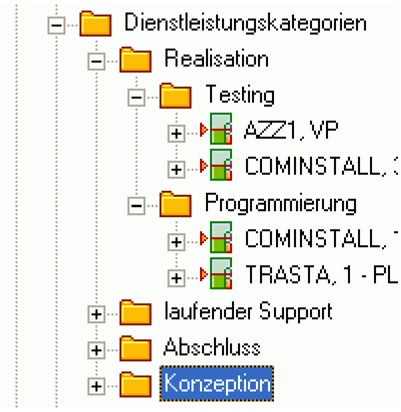
Projections

The projections existing in Benchmarking must be redefined in the new BI module (grouping by folders in the BI view, see 4.2).

Here an example of a projection in Benchmarking and how to specify it correctly in the BI module.

Projection in Benchmarking

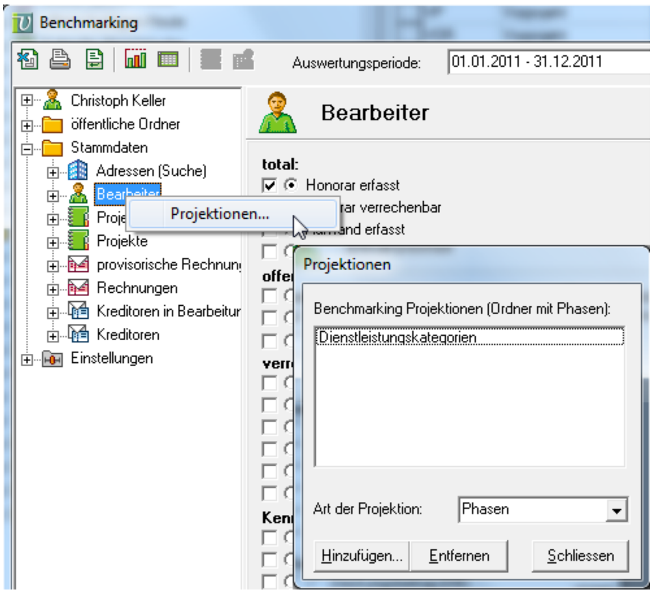
The example shows a folder structure Dienstleistungskategorien.



- Dienstleistungskategorien
  - Realisation
    - Testing
      - AZZ1, VP
      - COMINSTALL, :
    - Programmierung
      - COMINSTALL, ·
      - TRASTA, 1 - PL
  - laufender Support
  - Abschluss
  - Konzeption

The top folder `Dienstleistungskategorien` specifies the projection name, the directly subordinated subfolders define the subdivision of the values. The hierarchy of folders may include several layers and the real objects are found at the lowest level. In benchmarking, these had to be projects, users or project phases, now objects of all classes for which key figures are available can be assigned (see 4.3).

For the folder Dienstleistungskategorien to be displayed as a projection in Benchmarking, it must be assigned to any folder, for instance the folder Users. The assignment occurs in the Benchmarking window by right clicking on the folder:



Benchmarking

Auswertungsperiode: 01.01.2011 - 31.12.2011

- Christoph Keller
- öffentliche Ordner
- Stammdaten
  - Adressen (Suche)
  - Bearbeiter
  - Proje
  - Projekte
  - provisorische Rechnung
  - Rechnungen
  - Kreditoren in Bearbeitur
  - Kreditoren
- Einstellungen

Projektionen...

Bearbeiter

total:
☑ ● Honorar erfasst
ar verrechenbar
and erfasst

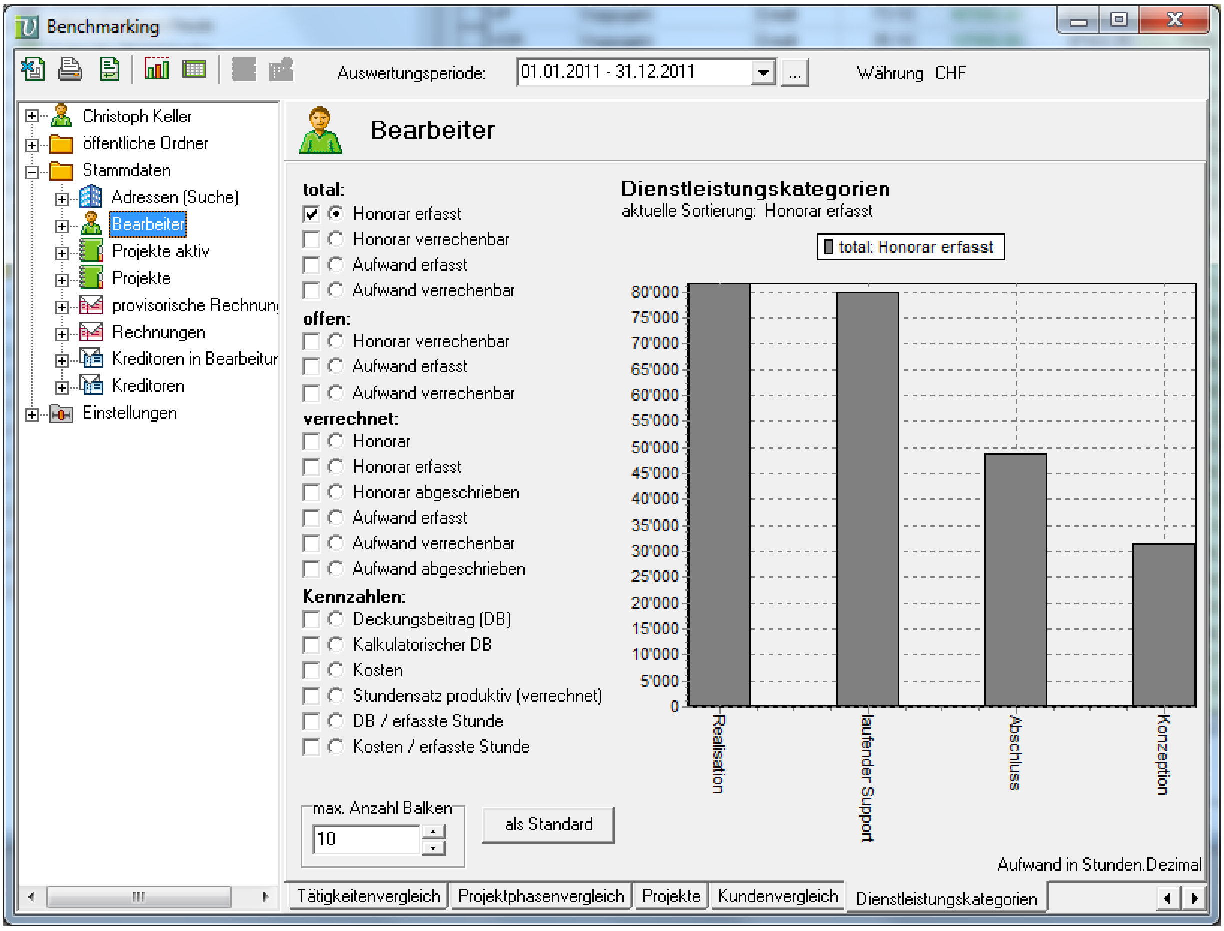
Projektionen

Benchmarking Projektionen (Ordner mit Phasen):

Dienstleistungskategorien

Art der Projektion: Phasen

If the folder `Users` is then selected in the Benchmarking window, a new computation named Dienstleistungskategorien will be visible, in which all values of services performed by collaborators will be grouped by service categories.



Benchmarking

Auswertungsperiode: 01.01.2011 - 31.12.2011

Währung CHF

- Christoph Keller
- öffentliche Ordner
- Stammdaten
  - Adressen (Suche)
  - Bearbeiter
  - Projekte aktiv
  - Projekte
  - provisorische Rechnun
  - Rechnungen
  - Kreditoren in Bearbeitur
  - Kreditoren
- Einstellungen

Bearbeiter

total:

☑ ● Honorar erfasst

☐ ○ Honorar verrechenbar

☐ ○ Aufwand erfasst

☐ ○ Aufwand verrechenbar

offen:

☐ ○ Honorar verrechenbar

☐ ○ Aufwand erfasst

☐ ○ Aufwand verrechenbar

verrechnet:

☐ ○ Honorar

☐ ○ Honorar erfasst

☐ ○ Honorar abgeschrieben

☐ ○ Aufwand erfasst

☐ ○ Aufwand verrechenbar

☐ ○ Aufwand abgeschrieben

Kennzahlen:

☐ ○ Deckungsbeitrag (DB)

☐ ○ Kalkulatorischer DB

☐ ○ Kosten

☐ ○ Stundensatz produktiv (verrechnet)

☐ ○ DB / erfasste Stunde

☐ ○ Kosten / erfasste Stunde

max. Anzahl Balken: 10

als Standard

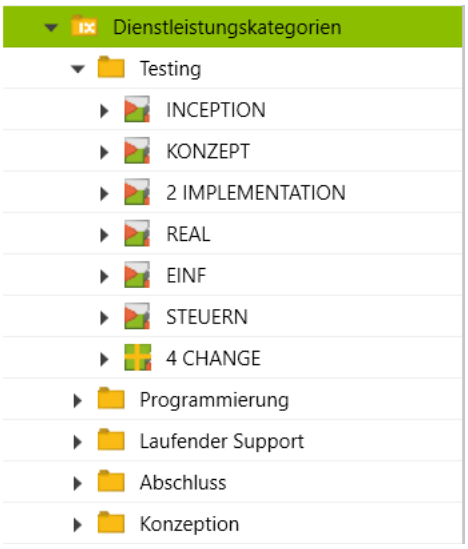
Dienstleistungskategorien

aktuelle Sortierung: Honorar erfasst

Figure 16: Benchmarking projection

"Projection" in the BI module

We now define the same projection in the new BI module. For this, we use the same folder structure. Only one hierarchy level is supported from now on. It is therefore necessary to adapt the folder structure accordingly:



- Dienstleistungskategorien
  - Testing
    - INCEPTION
    - KONZEPT
    - 2 IMPLEMENTATION
    - REAL
    - EINF
    - STEUERN
    - 4 CHANGE
  - Programmierung
  - Laufender Support
  - Abschluss
  - Konzeption

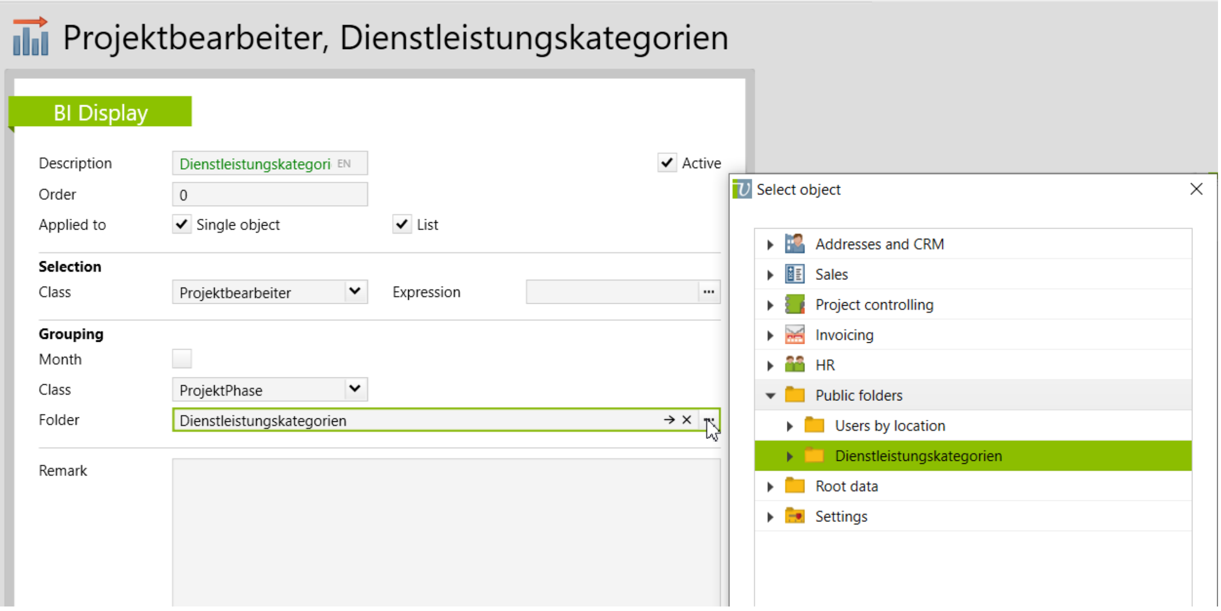
Figure 17: Folder structure with one hierarchy level

Additional information To move the folders, the script Ordner verschieben in the Online Knowledge Base was used: www.vertec.com/kb/changeparentordner

We now define a new BI view (see 4.2).

Since we wish to display the projection on users, we use `Projektbearbeiter` as selection.

For the grouping, we select the class which we have attributed in the folder structure. In this case, project phases.



Projektbearbeiter, Dienstleistungskategorien

BI Display

Description: Dienstleistungskategori EN
☑ Active
Order: 0
Applied to: ☑ Single object ☑ List

Selection
Class: Projektbearbeiter
Expression:

Grouping
Month: □
Class: ProjektPhase
Folder: Dienstleistungskategorien

Remark

Figure 18: Definition of the BI Display for folder projections

If, in the BI view, a collaborator or a list of collaborators is selected, the view `Dienstleistungskategorien` will appear, in which all values for services provided by the collaborators will appear, grouped by service categories:

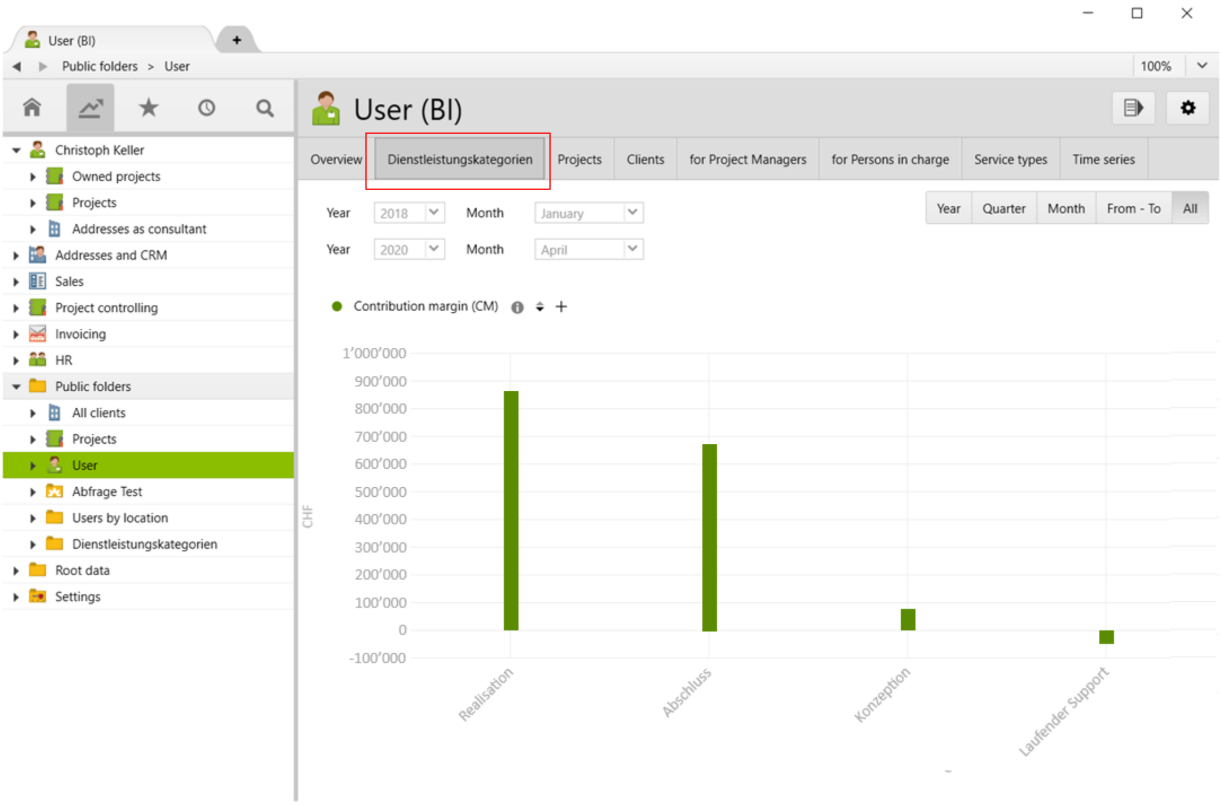


Figure 19: User values grouped by service categories[1]

As can be seen, the new BI Module is more powerful than Benchmarking:

- Every computed class can be projected, not only project phases.
- The projection is automatically available for all objects of the selected class, in our example not only for the team members folder, but for all team member lists as well as single team members.
- All selected measures (see 4.3) are automatically also available for the new computation.

4.9 Implementation

When Vertec 6.4 is started for the first time, the BI data is not yet computed (see chapter 4.6). No values are therefore available in BI view.

The values will be automatically computed during the night. Wait for the next day to see the results of the computation.

Another possibility is to start the computation manually. The way to do it is described in section 4.6 under Start the computation manually. Be however aware that the computation will require a large number of resources if big data volumes are involved and could possibly block the normal use of Vertec during this time.

[1] Note: Figure 19 is based on different data than Figure 16 , the values cannot be compared.

5 Services and Expenses

5.1 PDF documents used as document image

Line: Standard, Expert | Module: PSA | Apps: All | Version: 6.4

PDF receipts can now be loaded as pictures of expenses. The procedure is the same as for pictures: click with the right mouse button on page Document image and select `Load image…`.

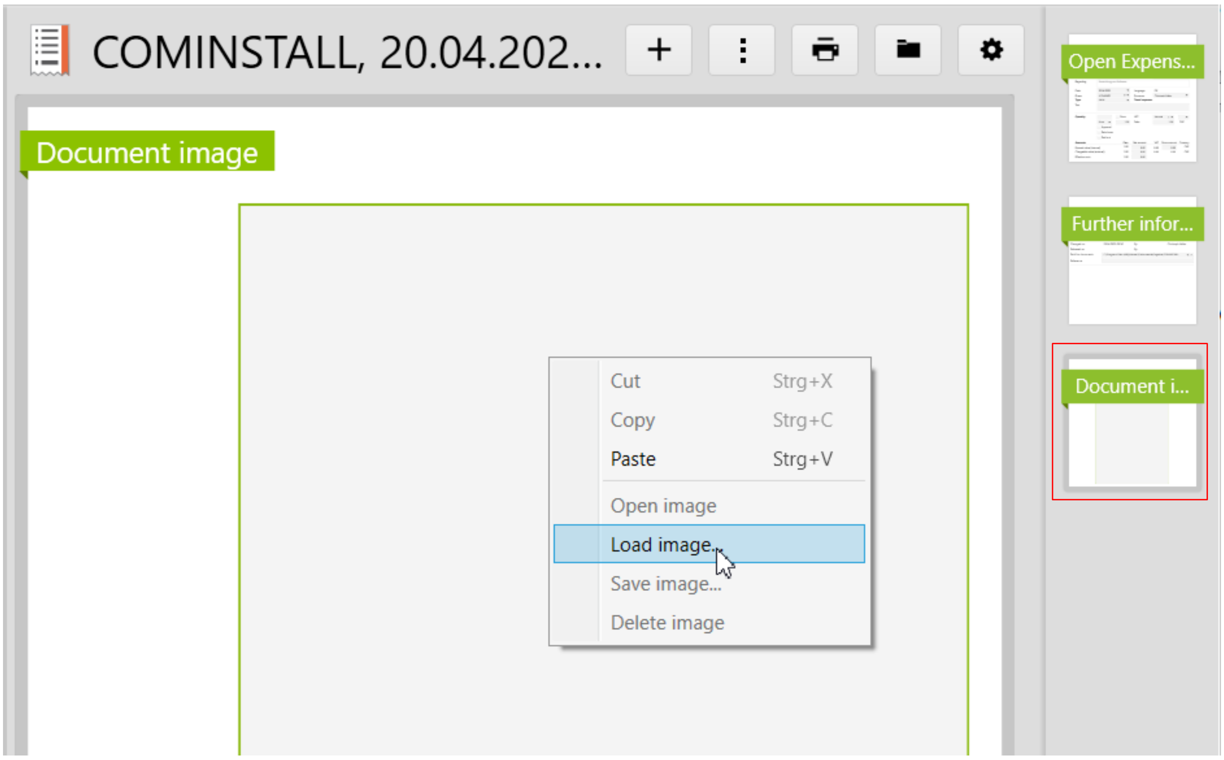


Figure 20: Load a document image

In addition to graphic files, a PDF document can now be selected:

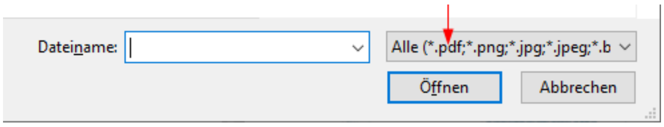


Figure 21: PDFs can now be selected

A preview of the first page is displayed in the picture. An additional click with the right mouse button opens the document.

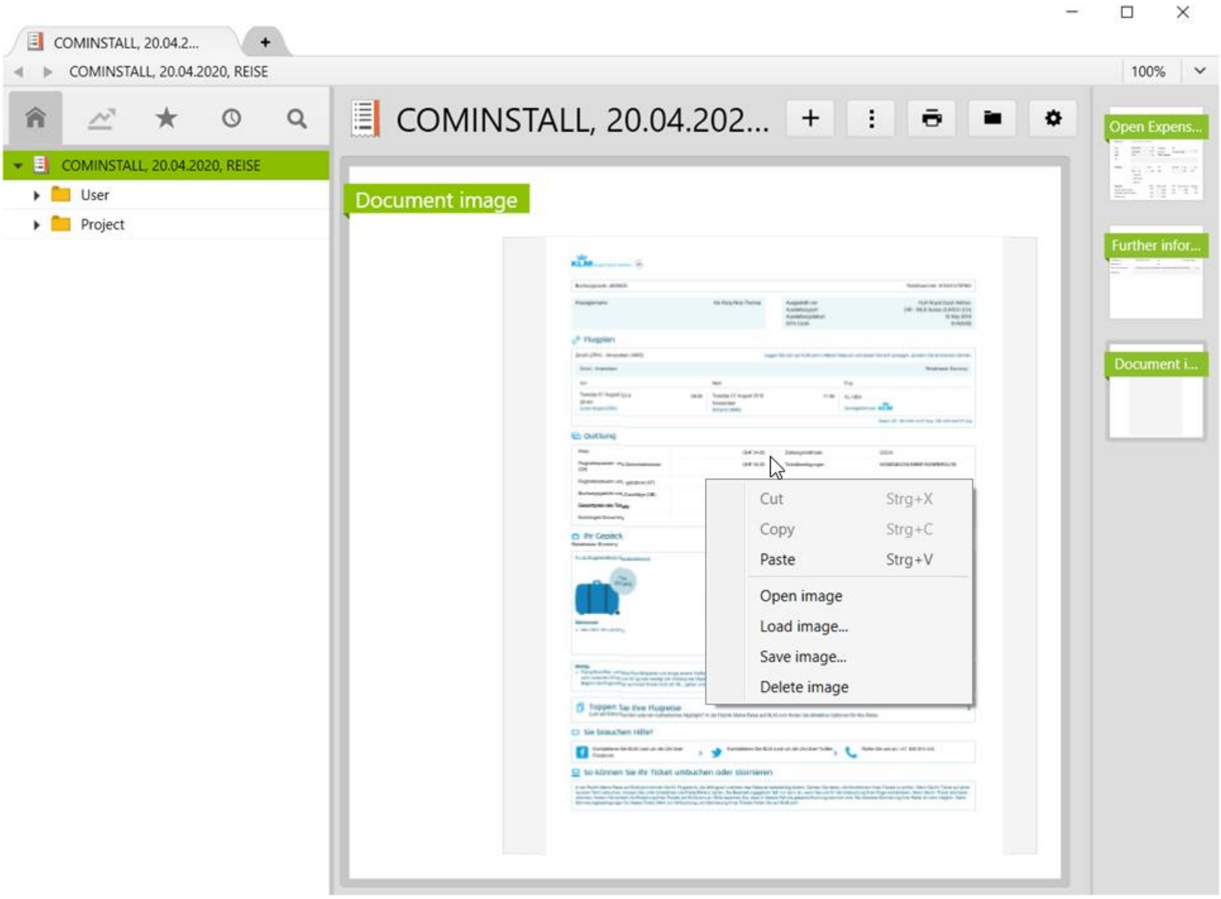


Figure 22: a PDF is stored as an expense picture

This functionality is available as follows:

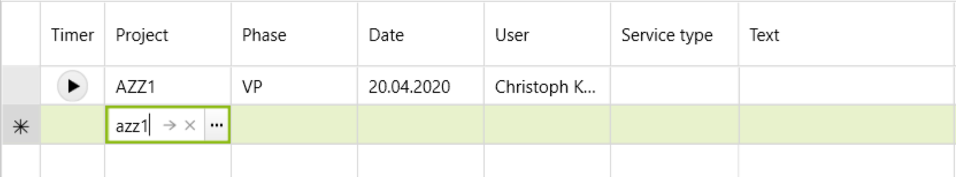
- Upload and download PDF receipts: in the Desktop-, Cloud- and Web App
- Preview of the first page: in the Desktop-, Cloud-, Web- and Phone App
- Use the preview as image in extended word reports
- Input of PDF receipts in the Phone App will be available in a version after 6.4.

5.2 Direct specification of the project in the search dialog (edtProjekt)

Line: Standard, Expert | Module: PSA | Apps: Full featured | Version: 6.3.0.4

The project search dialog when inputting a service (configurable as control element edtProjekt in the column setting) has been extended as follows: if a unique project code is entered, the project is immediately selected and the search dialog does not appear.

Input the code, click on TAB and the project is inserted



	Timer	Project	Phase	Date	User	Service type	Text
	▶	AZZ1	VP	20.04.2020	Christoph K...		
✳		azz1					

Figure 23: The unique code is directly selected with TAB.

This will improve the performance particularly for clients with numerous projects.

5.3 Computation and presentation of vacation data

Line: Standard, Expert | Module: PSA | Apps: Full featured | Version: 6.3.0.17

The following changes have been made to the computation and display of vacation data:

- If the entry date of an employee happened to fall within the current vacation period (normally the current year) a vacation credit carried forward was not considered (the credit was always 0). Vacation credits are now taken into account independently of the entry date.
- If a vacation credit is input during the vacation period (but not on the first day of the period), it replaces the computation of the vacation credit, since it directly defines the balance at a certain date. The display of the vacation credit then no longer shows 0.00 but remains empty (the presentation is changed, not the computation).
- The end date of a vacation credit is now displayed as the leaving date.
- Values and presentation of vacation tables are now identical in the folders Settings and Services overview.

Additional information Additional information on this subject is available in the Online Knowledge Base under www.vertec.com/kb/faq_sollzeit/#ferienberechnung.

6 Reports

6.1 Invoices according to the ZUGFeRD 2.0 standard (XRechnung)

Line: Standard, Expert | Module: PSA | Apps: Full featured | Version: 6.3.0.12

Starting with Vertec version 6.3.0.12, invoices following the ZUGFeRD 2.0 standard can be produced. For this purpose, the XML metadata is generated by the code in the extended Office report and passed to the reporting mechanism which integrates the metadata into the PDF.

A method is available for this in extended Office reports

```
metadata_zugferd(context)
```

The method returns a tuple consisting of:

- **profilename**: name of the ZUGFeRD embedded profile (see below). If empty, a schema definition will be provided in the 3rd argument. Otherwise, the report will produce an error at execution time
- **data**: ZUGFeRD XML data. The XML document presenting the ZUGFeRD data in the invoice.
- **schema**: optional XMP metadata structure. If the profile is not embedded (see above), an XML data string can optionally be passed here defining the schema of the profile to be used.

The following presently existing ZUGFeRD 2.0 profiles are supported and provided as embedded profiles:

- EXTENDED
- EN 16931 (COMFORT, XRechnung)
- BASIC
- BASIC WL
- MINIMUM

The provided standard report Invoice with list includes an example the code for the EN 16931 profile. Clients who wish to generate their invoices according the ZUGFeRD 2.0 standard or as XRechnung should adapt this code accordingly.

Additional information Additional information on the subject Vertec and ZUGFeRD is available in the Online Knowledge Base under www.vertec.com/kb/zugferd.

6.2 Invoice with user sums as extended Office report

Line: Standard, Expert | Module: PSA | Apps: Full featured | Version: 6.3.0.17

The standard report Invoice with user sums is now provided as an extended Office report.

In new Vertec installations, only this version of the report is provided and registered as active.

in existing installations, the extended Office report is provided and registered as inactive. If you wish to use it, activate it in the report models.

6.3 Invoice with phase sums as extended Office report

Line: Standard, Expert | Module: Budget & Phases | Apps: Full featured | Version: 6.3.0.17

The standard report Invoice with phase sums is now provided as an extended Office report

In new Vertec installations, only this version of the report is provided and registered as active.

In existing installations, the extended Office report is provided and registered as inactive. If you wish to use it, activate it in the report models.

6.4 Extended invoices with QR code adapted to the current specification

Line: Standard, Expert | Module: PSA | Apps: Full featured | Version: 6.3.0.17

In all provided extended Office reports for invoices, the page with the QR code has been adapted to the current specification 2.1. The following reports are concerned:

- Invoice with list
- Invoice with phase sums (see 6.3)
- Invoice with user sums (see 6.2)

Important notice In the case of client-specific report templates, normal Word as well as extended, the QR page must be manually adapted. The exact procedure is described in our Online Knowledge Base under www.vertec.com/kb/rechnungsvorlagen-fuer-qr-code-bereit-machen.

6.5 Advance payments report now with cut-off date

Line: Standard, Expert | Module: PSA | Apps: Full featured | Version: 6.3.0.17

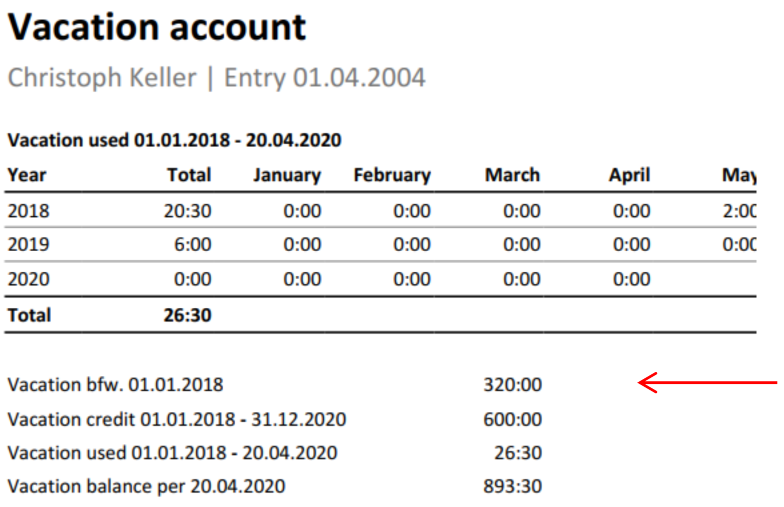
The extended Office report **Advance payments** has been redeveloped with following changes:

- Introduction of a cut-off date. The default value is the last day of the preceding month. This leads to the following changes:
  - Only advance payments with a non-null balance (or had a balance left at the cut-off date) are displayed. Otherwise the list with zero values would become longer and longer.
  - Advance payments used on an invoice are only displayed if booked by the cut-off date (otherwise there is no way to find when the advance has been used). The column **Booked** therefore disappears.
- The cut-off date appears on the report as subtitle.
- Lists are grouped by currency.

6.6 Vacation Accounts report always displays the vacation bfw.

Line: Standard, Expert | Module: PSA | Apps: Full featured | Version: 6.3.0.13

In the extended Office report Vacation Accounts, the vacation brought forward was only displayed if entered manually. Now it is always displayed at the beginning of the period:



Vacation account

Christoph Keller | Entry 01.04.2004

Vacation used 01.01.2018 - 20.04.2020

Year	Total	January	February	March	April	May
2018	20:30	0:00	0:00	0:00	0:00	2:0
2019	6:00	0:00	0:00	0:00	0:00	0:0
2020	0:00	0:00	0:00	0:00	0:00	
Total	**26:30**					

Vacation bfw. 01.01.2018	320:00
Vacation credit 01.01.2018 - 31.12.2020	600:00
Vacation used 01.01.2018 - 20.04.2020	26:30
Vacation balance per 20.04.2020	893:30

- The Vacation bfw. is calculated with `->getFerienvortrag(startDate)`.
- The Vacation credit is calculated as: `Vacation Balance + Vacation used - Vacation bfw`.

6.7 Recapitulation of bookings grouped by currency

Line: Expert | Module: PSA | Apps: Full featured | Version: 6.3.0.17

In the extended Office report Recapitulation of bookings, the entries are now grouped by currency and the currency is displayed with the total.

6.8 Performance improvement for Invoice proposal

Line: Standard, Expert | Module: PSA | Apps: Full featured | Version: 6.3.0.17

The extended Office report Invoice proposal has been completely redeveloped for performance reasons with following changes:

- Only a "Date to" is available, no longer a "Date from". If no "Date to" is specified, an error message is displayed and the execution is aborted.
- Fixed-price phases without budgets are no longer taken into account.
- The project currency is printed next to the project title.

6.9 Performance improvement for the Budget overview report

Line: Standard, Expert | Module: PSA | Apps: Full featured | Version: 6.3.0.17

The extended Office report Budget overview has been completely redeveloped for performance reasons and now runs with higher performance when large data volumes are involved.

In addition, this report is now grouped by currency.

6.10 Expressions in headers and footers of Vertec-generated Word reports

Line: Standard, Expert | Module: PSA | Apps: Full featured | Version: 6.3.0.4

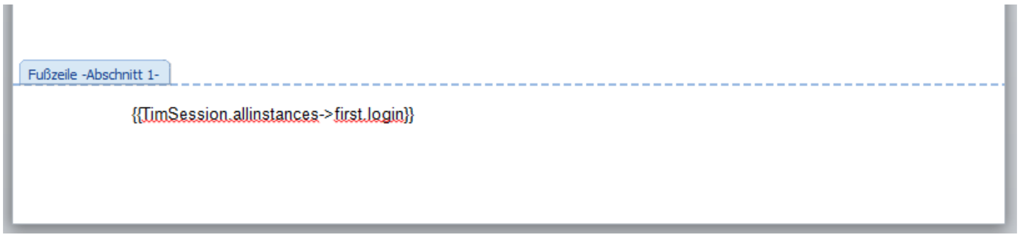
It is now possible to insert expressions in headers and footers of Word reports generated with Vertec.

As in extended Office reports, expressions are specified in double curly brackets, interpreted as OCL expressions at execution time and filled with the corresponding data.

Example

```
{{TimSession.allinstances->first.login}}
```

Inserts the name of the logged-in Vertec user into the header or footer.



Fußzeile -Abschnitt 1-

{{TimSession.allinstances->first.login}}

Figure 24: The name of the logged in user is inserted in the footer

6.11 Sum over several hierarchy levels in extended Excel reports

Line: Standard, Expert | Module: PSA | Apps: Full featured | Version: 6.3.0.7

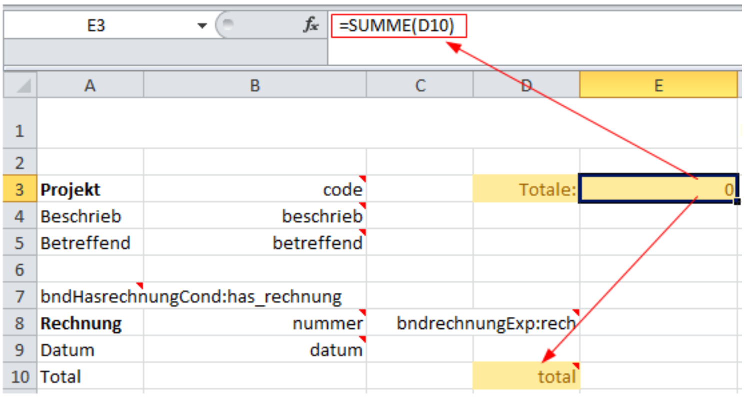
Summing cells in Excel reports via band sums produces the sum of all lines of a column within a band.

In many cases, it is desirable to sum single cells of a band distant from the sum by several hierarchy levels.

This is now possible. When summing a single cell, a sum of all copies of this cell is now performed.

Example

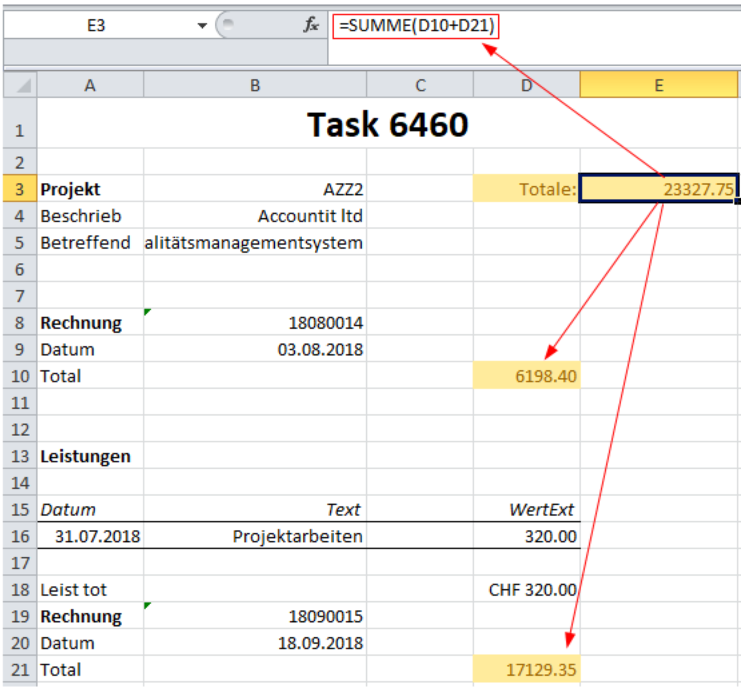
When the sum is specified on a single cell, a sum of all copies of this cell is now performed at execution time.



E3 =SUMME(D10)

	A	B	C	D	E
1					
2					
3	**Projekt**	code		Totale:	0
4	Beschrieb	beschrieb			
5	Betreffend	betreffend			
6					
7	bndHasrechnungCond:has_rechnung				
8	**Rechnung**	nummer	bndrechnungExp:rech		
9	Datum	datum			
10	Total			total	

Figure 25: The sum refers to the cell Total.



E3 =SUMME(D10+D21)

	A	B	C	D	E
1		Task 6460			
2					
3	**Projekt**	AZZ2		Totale:	23327.75
4	Beschrieb	Accountit ltd			
5	Betreffend	alitätsmanagementsystem			
6					
7					
8	**Rechnung**	18080014			
9	Datum	03.08.2018			
10	Total			6198.40	
11					
12					
13	**Leistungen**				
14					
15	*Datum*	*Text*		*WertExt*	
16	31.07.2018	Projektarbeiten		320.00	
17					
18	Leist tot			CHF 320.00	
19	**Rechnung**	18090015			
20	Datum	18.09.2018			
21	Total			17129.35	

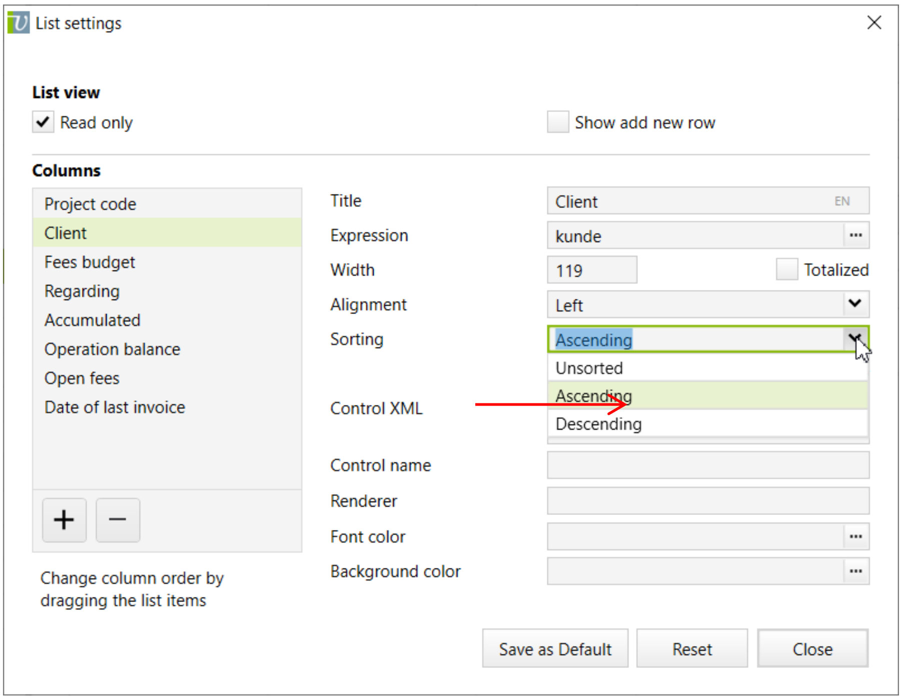
Figure 26: At execution time, both cells are added.

7 Folders and List Settings

7.1 Multilevel sorting of lists

Line: Standard, Expert | Module: PSA | Apps: Full featured | Version: 6.4

Vertec lists now can be sorted by more than one column. The sorting specification occurs in the list settings:



List settings

List view

☑ Read only

□ Show add new row

Columns

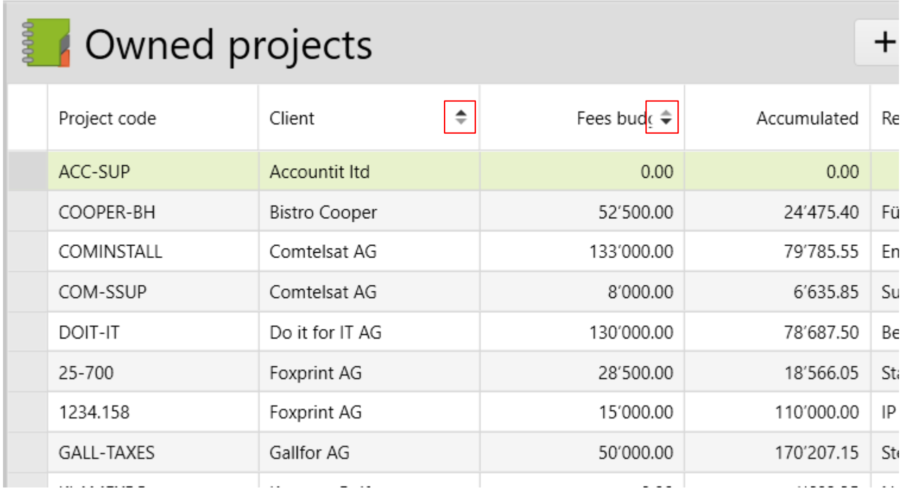
- Project code
- Client
- Fees budget
- Regarding
- Accumulated
- Operation balance
- Open fees
- Date of last invoice

Title	Client
Expression	kunde
Width	119 □ Totalized
Alignment	Left
Sorting	Ascending
Control XML	
Control name	
Renderer	
Font color	
Background color	

Change column order by dragging the list items

Figure 27: The sorting specification for the various columns occurs in the list settings

– The sorting follows the sequence of the column definitions, from left to right.
– All columns involved in the sorting appear sorted in the title.



Owned projects

Project code	Client	Fees bud	Accumulated	Re
ACC-SUP	Accountit ltd	0.00	0.00	
COOPER-BH	Bistro Cooper	52'500.00	24'475.40	Fü
COMINSTALL	Comtelsat AG	133'000.00	79'785.55	En
COM-SSUP	Comtelsat AG	8'000.00	6'635.85	Su
DOIT-IT	Do it for IT AG	130'000.00	78'687.50	Be
25-700	Foxprint AG	28'500.00	18'566.05	St
1234.158	Foxprint AG	15'000.00	110'000.00	IP
GALL-TAXES	Gallfor AG	50'000.00	170'207.15	St

Figure 28: The list is sorted by ascending order of Client and, within the client, by descending order of Budget.

If one clicks on a sorting arrow in the list, the list is exclusively sorted by the selected column. The query modified in this way remains active until the list is displayed again. The sorting then occurs again according to the list settings.

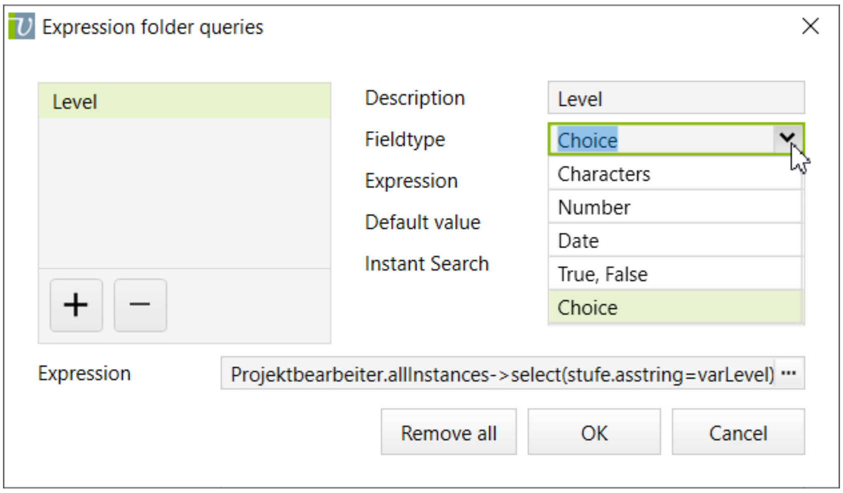
Backwards compatibility
Existing list specifications remain unchanged.

Compared to earlier versions, a change occurs if the sorting of the columns is changed in the list specifications. Until now, setting the sorting on a column has reset the sorting on other columns. This is no longer so. If the list should no longer be sorted by a specific column, the column must explicitly be set to `Unsorted`.

7.2 Choice search fields for queries on expression folders

Line: Standard, Expert | Module: PSA | Apps: Full featured | Version: 6.4

The field type Choice available for SQL queries since version 6.3 is now available as well with expression folders.



Expression folder queries

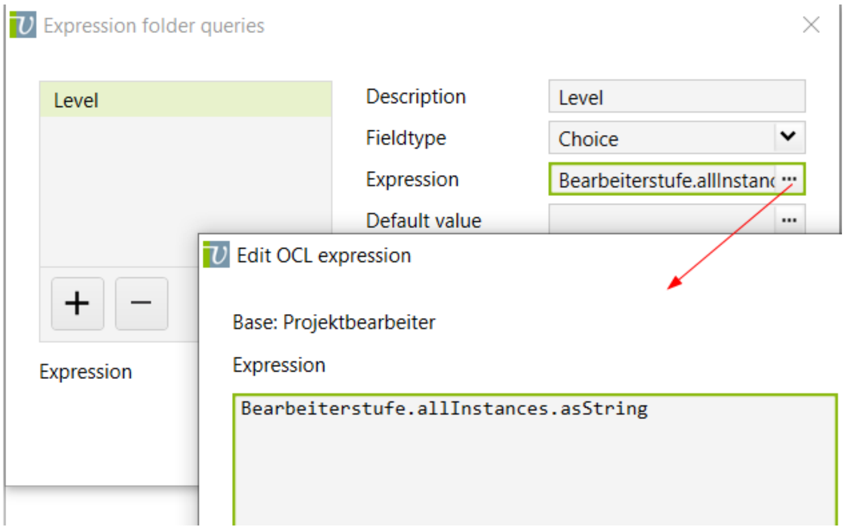
Level

Description: Level
Fieldtype: Choice
Expression
Default value
Instant Search

Expression: Projektbearbeiter.allInstances->select(stufe.asstring=varLevel)

Figure 29: Selection of field type

As soon as the field type is set, a field for the expression is immediately displayed below it.



Expression folder queries

Level

Description: Level
Fieldtype: Choice
Expression: Bearbeiterstufe.allInstanc
Default value

Expression

Edit OCL expression

Base: Projektbearbeiter

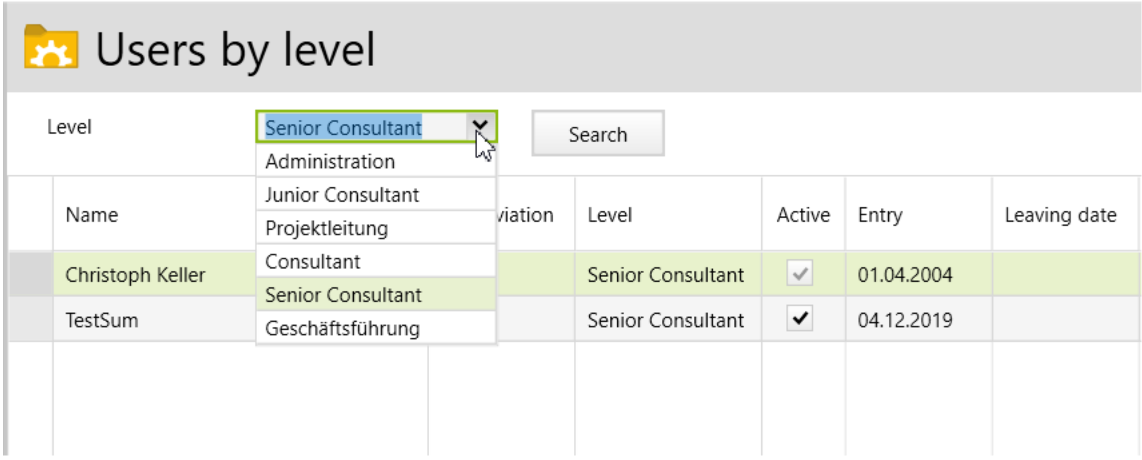
Expression

```
Bearbeiterstufe.allInstances.asString
```

Figure 30: The expression for the selection

As a result, the expression must return a list of strings.

A list box is then displayed, allowing to select the desired value.



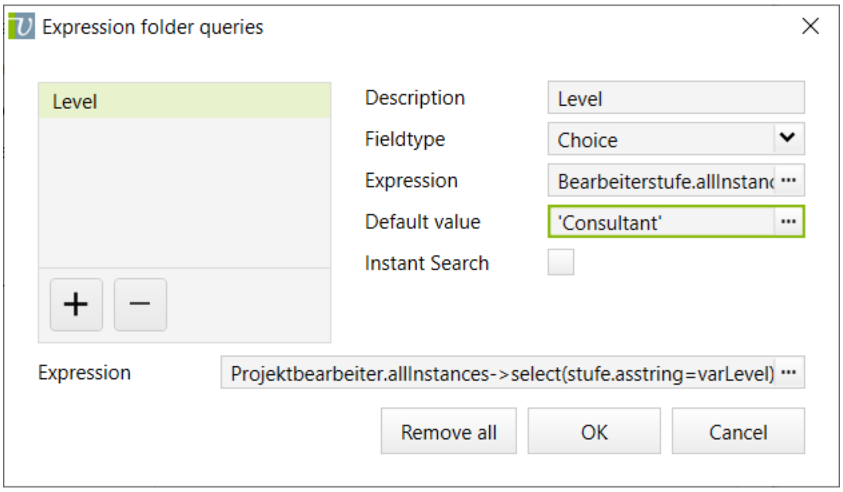
Users by level

Level: Senior Consultant

Name	viation	Level	Active	Entry	Leaving date
Christoph Keller		Senior Consultant	✓	01.04.2004	
TestSum		Senior Consultant	✓	04.12.2019	

Figure 31: The desired value can be selected from the list box

If a default value is defined, it is displayed by default.



Expression folder queries

Level

Description: Level

Fieldtype: Choice

Expression: Bearbeiterstufe.allInstanc

Default value: 'Consultant'

Instant Search: □

Expression: Projektbearbeiter.allInstances->select(stufe.asstring=varLevel)

Figure 32: A default value is defined

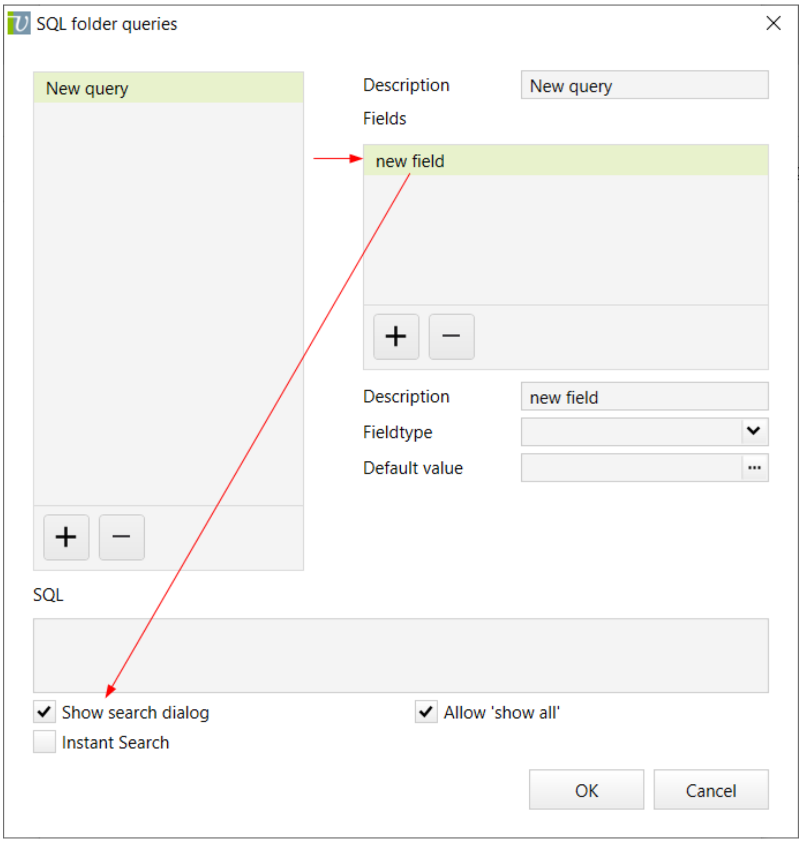
If the specified default value is not in the list box or no default value exists, the first entry from the selection list is displayed.

7.3 SQL folder sets "Show search dialog" again by default

Line: Standard, Expert | Module: PSA | Apps: Full featured | Version: 6.3.0.4

Starting with Vertec 6.2, the option Show search dialog was no longer set by default when the first query field was added to an SQL folder. This has been reversed.

In SQL folders, the option Show search dialog is now set by default as soon as a query field is added. If this is not desired, the option must be manually deactivated.



SQL folder queries

New query

Description: New query

Fields: new field

Description: new field

Fieldtype:

Default value:

SQL

☑ Show search dialog

☑ Allow 'show all'

□ Instant Search

Figure 33: As soon as a query field is added, the option "Show search dialog" is automatically activated

8 Settings

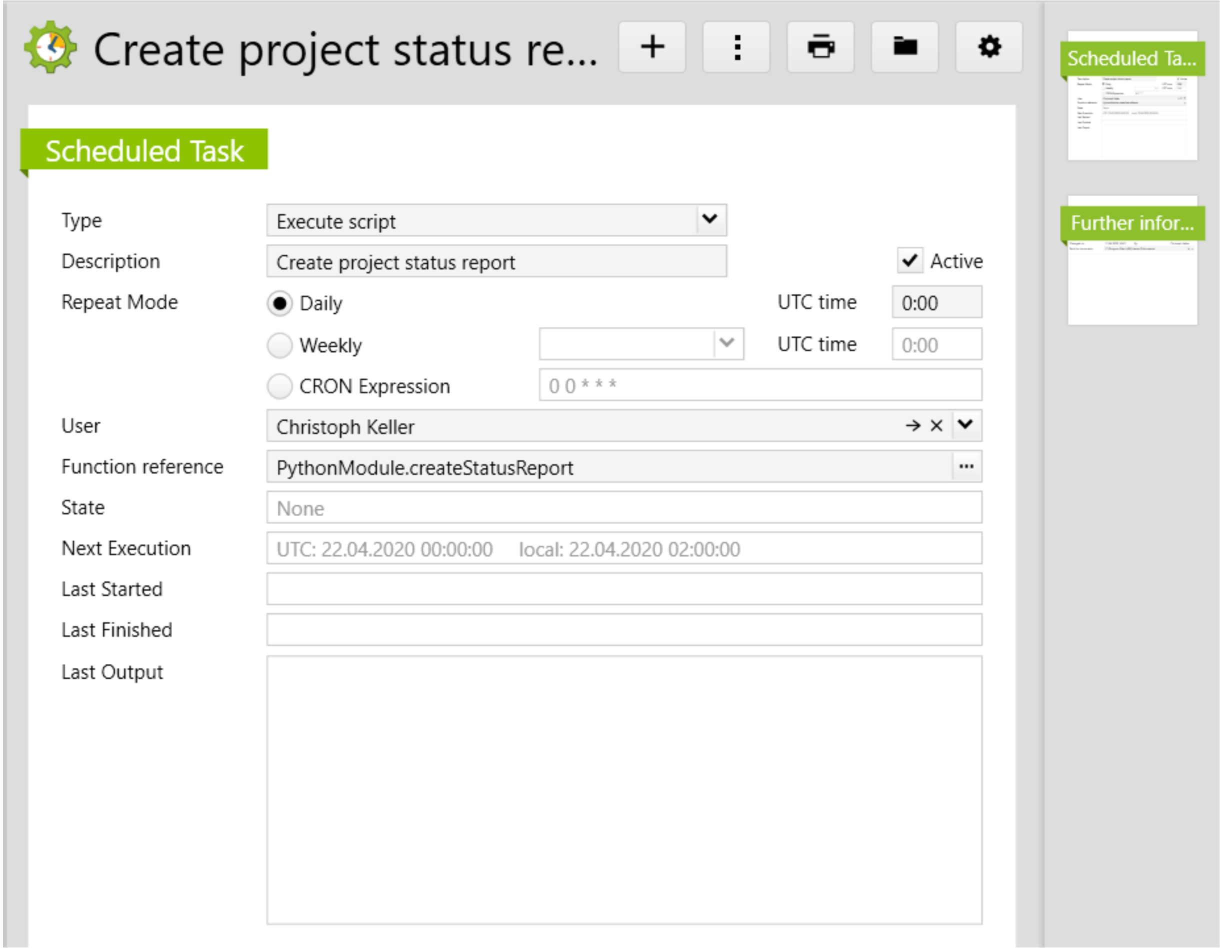
8.1 Scheduled Tasks

Line: Standard, Expert | Module: PSA | Apps: Full featured | Version: 6.4

The BI recalculation as described in section 4.6 occurs in Vertec by means of a scheduled task.

These scheduled tasks can also be used for other tasks.

New tasks can be specified in the folder Settings > Scheduled Tasks.



Scheduled Task

Type	Execute script		
Description	Create project status report	☑ Active	
Repeat Mode	● Daily	UTC time	0:00
	○ Weekly	UTC time	0:00
	○ CRON Expression	0 0 * * *	
User	Christoph Keller		
Function reference	PythonModule.createStatusReport		
State	None		
Next Execution	UTC: 22.04.2020 00:00:00 local: 22.04.2020 02:00:00		
Last Started			
Last Finished			
Last Output			

Figure 34: A scheduled task

Type	In addition to the type `Recalculate BI measures` (see section 4.6) there is the type `Execute script`. It allows to execute a script registered in Vertec at a specific time.
Description	Description of the script
Active	Only active scheduled tasks are executed
Repeat Mode	The choice is **Daily** or **Weekly**. The time is specified in coordinated universal time mode (UTC). Alternatively, a **CRON Expression** can be specified. Information on CRON expressions can be found in the internet, for instance in Wikipedia:

	https://en.wikipedia.org/wiki/Cron. In the background, each of these indications leads to a CRON expression which determines the next execution time. Your input can be checked in the field Next Execution (see below). It displays the time of the next execution, in UTC as well as in local time.
User	The scheduled task is executed in the name of this user. Therefore, a user must be specified with sufficient rights to execute the script.
Function reference	The name of the script entry followed by the function to be executed within the script, separated by a point. It must be a function without arguments. <Script entry>.<Function name> Once this field is specified, the code can be displayed by clicking on the three-point button. The button Open in script editor in the popup directly opens the script entry in the script editor where It can be edited.
State	Indicates if the execution was successful or not. If the execution was not successful, an error message is displayed in Last Output (see below).
Next Execution	UTC and local time of the next execution. The time of the next execution is determined by the CRON Expression (see field Repeat Mode above) and the actual time. If, however, the cloud server is not running at this moment, the execution cannot take place. As soon as the server runs again, it determines the time of the next execution. To avoid the simultaneous execution of many tasks when the cloud server is started, past "missed" executions are not made up.
Last Started	Start time of the last execution.
Last Finished	End time of the last execution.
Last Output	Output of the last execution. If the execution was not successful, an error message is displayed here.

The application which executes scheduled tasks is called **Vertec.TaskRunner.exe** and is found in the Vertec installation directory.

Immediate execution If immediate execution is desired, there is a menu item Execute now in the context menu of the scheduled task. This procedure however does not immediately start the scheduled task: only the script is executed. Exactly as if the script or the called method were executed in the script editor. Moreover, this happens in the name of the logged-in user, not the user specified in the scheduled task.

Parameter in the Vertec.ini file

The following new parameters now exist in the Vertec.ini file in the section [CloudServer]:

Task Scheduler	Specifies if task scheduling is active or not. If not, no scheduled tasks are executed. True is the default if nothing is specified. If a Cloud Server Section should not

	execute scheduled tasks, specify `Task Scheduler = False` When operating multiple cloud servers, only one of them may take over the task of running tasks. For all others this parameter must be set to `False`.
Task Scheduler Polling Minutes	Determines the interval in minutes after which a check for waiting scheduled tasks occurs Natural number, default is 1.
Task Scheduler Concurrent Tasks	Determines how many scheduled tasks can maximally be executed in parallel. Natural number, default is 1.

BI Task Scheduler Range Start
BI Task Scheduler Range End
	Determine the period in the night during which the BI data can be computed. A time is then randomly determined by the cloud server within this period to start the execution.
	These parameters are especially important for the Vertec Cloud Subscription (see section 4.6), but can of course also be used elsewhere.
	`BI Task Scheduler Range Start=23:00`
	`BI Task Scheduler Range End=05:00`
	The random determination is as follows. The number of minutes between Range Start and Range End is computed. A random number between 0 and the number of minutes is then chosen and added to Range Start. This yields the CRON expression to determine the moment of the next execution (see the description above of the field Next Execution).

8.2 Two-factor authentication

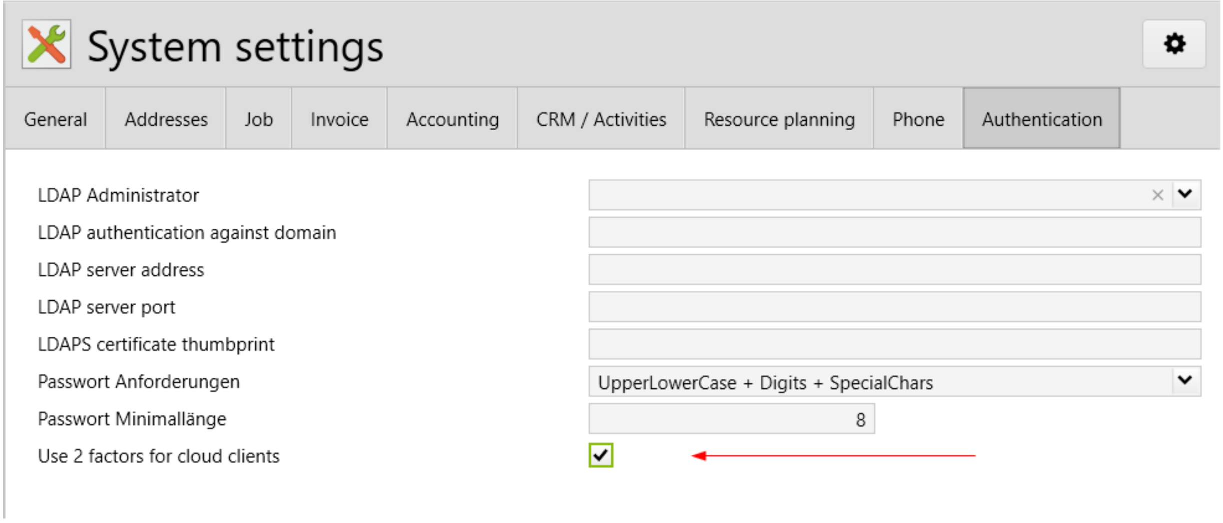
Line: Standard, Expert | Module: PSA | Apps: Cloud Clients | Version: 6.3.0.12

Starting with version 6.3.0.12, Vertec supports for the login to cloud clients (Cloud App, Web App, Phone App) a second factor by means of an authenticator app. Google Authenticator, for instance.

The authenticator app must be a "soft-token" app which generates a time based one time-password. With this system, no communication between the authenticator app and Vertec is necessary, just once must a common secret be exchanged. Based on this secret, the authenticator app and Vertec can independently generate the same codes and compare them.

A precondition is that the times on the mobile device and on the Vertec server agree.

Set up in Vertec
In System settings > Authentication there is a setting Use 2 factors for cloud clients:



System settings

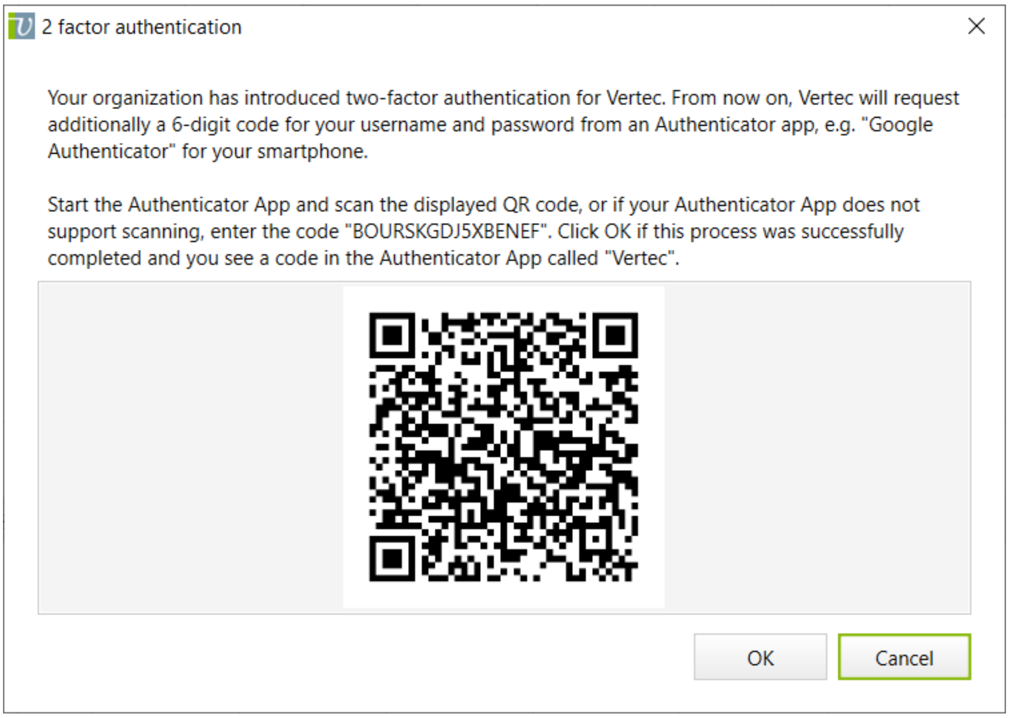
LDAP Administrator
LDAP authentication against domain
LDAP server address
LDAP server port
LDAPS certificate thumbprint
Passwort Anforderungen: UpperLowerCase + Digits + SpecialChars
Passwort Minimallänge: 8
Use 2 factors for cloud clients: ☑

Figure 35 Activation of the two-factor authentication in the Vertec system settings

If this setting is activated, the following happens:

- When the user logs into Vertec with user name and password for the first time after the activation of the system setting, a dialog to install two-factor authentication (2FA) takes place. The user can only leave this page by aborting the process or by bringing the process to a successful completion. In both cases, the user is logged out and must then log in again.
- If the process is aborted, the secret is not stored. When the next login occurs, the set-up dialogue occurs again.
- If the process is completed, the secret is stored. When the next login occurs, the user has to enter the corresponding authenticator pin.
- No more actions in Vertec are possible until 2FA has been installed for this user and he logs in again.

The installation dialogue is as follows:



2 factor authentication

Your organization has introduced two-factor authentication for Vertec. From now on, Vertec will request additionally a 6-digit code for your username and password from an Authenticator app, e.g. "Google Authenticator" for your smartphone.

Start the Authenticator App and scan the displayed QR code, or if your Authenticator App does not support scanning, enter the code "BOURSKGDJ5XBENEF". Click OK if this process was successfully completed and you see a code in the Authenticator App called "Vertec".

Figure 36: Dialogue to install 2 factor authentication

Start the Authenticator app and scan the displayed code or input the given code. As soon as a code with designation "Vertec" appears, click on `OK`. Both systems are now coupled. From this moment on, when logging in to Vertec, enter as second factor the code generated by the Authenticator app.

Important notice Two-factor authentication applies to the Phone App as well, but installation of the authentication as described here must occur in Cloud App or Web App. Only these apps provide the installation dialogue.

Each user must therefore **first log in to Cloud App or Web App** and install two-factor authentication there, before 2FA can be used in Phone App.

Reset a secret
In emergency cases like a loss of smartphone or of data, the administrator can reset the secret of a user. For this purpose, the administrator can log in and, for the concerned collaborator, start the dialogue via the Context menu or Menu actions > **Edit 2nd factor for authentication** and click on the button `Delete secret`.

8.3 Native language in MLString attributes

Line: Standard, Expert | Module: PSA | Apps: Full featured | Version: 6.4

In many places in Vertec so-called multi-language strings (MLStrings) are used:

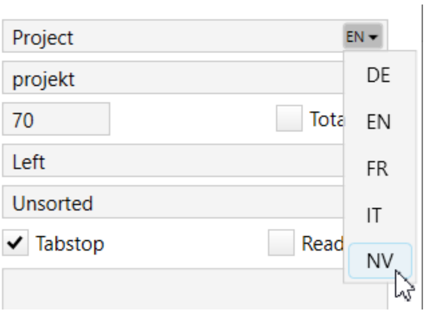
- Titles in list settings
- Texts related to activities, expense and purchase types
- Description of terms of payments
- Description of supplementary fields
- Description of absence types
- In the new Business Intelligence module (see chapter 4).

For every language, a term can be entered. It will be displayed in a different language on the interface or in reports for clients, for instance.

This has been a problem if the same language had to be dynamically translated into the jargon (Project – Mandate); in such cases, the desired text had to be specified.

From now on, there is additionally a "native" language, in which the term to be translated can be specified. This term will then be automatically translated by the translation system into the profession-specific expression.

For instance, in the list settings:



Project

projekt

70 □ Tota

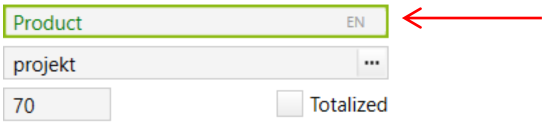
Left

Unsorted

☑ Tabstop □ Read

The following applies:

- If the MLString does not have text in the actual language when a term is displayed on the interface, the native text is used and translated by the translation system. In this case, the term is displayed in black.
- If the MLString does have text in the actual language when a term is displayed on the interface, this text is used and displayed in green.



Product

projekt

70 □ Totalized

If the term in the actual language is erased, the translated term is displayed (in black) again.

- If the native term is not found by the translation system, it will be displayed non translated in all languages for which it has not been overwritten.
- If there is no native term and the term has not been overwritten in the actual language, the text remains empty.
- Terms overwritten in the actual language always have precedence over translated terms.

Backwards compatibility

At the time of the first start-up after a convert (see 1.4), Vertec checks the existing MLString attributes. If only an EN or DE text is defined for an attribute, this term is stored in native language NV:

- If an EN text is defined, it is stored as NV text
- If no EN text, but a DE text is defined, the latter is stored as NV text
- If neither an EN nor a DE text is defined, no NV text is stored.

Since texts previously set are not changed and have precedence over native terms, nothing is changed for existing installations. The change will only become noticeable if a switch to another interface language occurs for which values have not been set before.

8.4 Change to shortcuts in navigation views

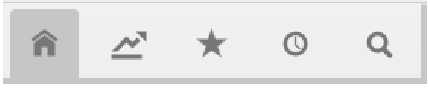
Line: Standard, Expert | Module: PSA | Apps: Full featured | Version: 6.4

Because of the new BI module (see chapter 4), some changes have been made to the shortcuts in the navigation views.

The Default view is opened with shortcut `F8` (instead of F9 in the old system).

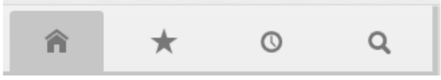
The new BI view is opened with shortcut `F9`.

New



F8	**Default view**
F9	**BI view**
F10	Favorites
F11	History
F6 / F12	Search

Before



F9	Default view
F10	Favorites
F11	History
F6 / F12	Search

8.5 Hide Change language dialog via system settings

Line: Standard, Expert | Module: PSA | Apps: Full featured | Version: 6.3.0.12

In System settings > General there is a new setting Show language selection. It causes the menu item `Change language...` to be displayed in the Settings-Menu, allowing the user to select the language.

PropertyName: **AllowLanguageSelection**. BooleanProperty. Default: Yes.

9 Extensions of the Phone App

9.1 Language according to the Vertec user's settings

Line: Standard, Expert | Module: PSA | Apps: Phone | Version: 6.3.0.12

The Phone App no longer uses the language set up on the device, but the language defined for the Vertec user.

9.2 Changes in the weekly view

Line: Standard, Expert | Module: PSA | Apps: Phone | Version: 6.3.0.15

The menu selectors for Services, Expenses, Activities and Absences (new) are now displayed in a single row at the bottom of the screen:

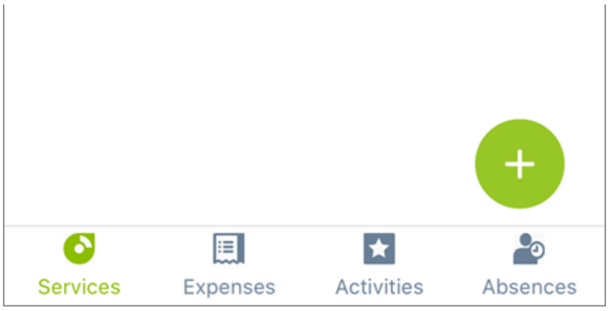


Figure 37: New menu selection in the Phone App

The main menu of the Phone App newly only offers the following options:

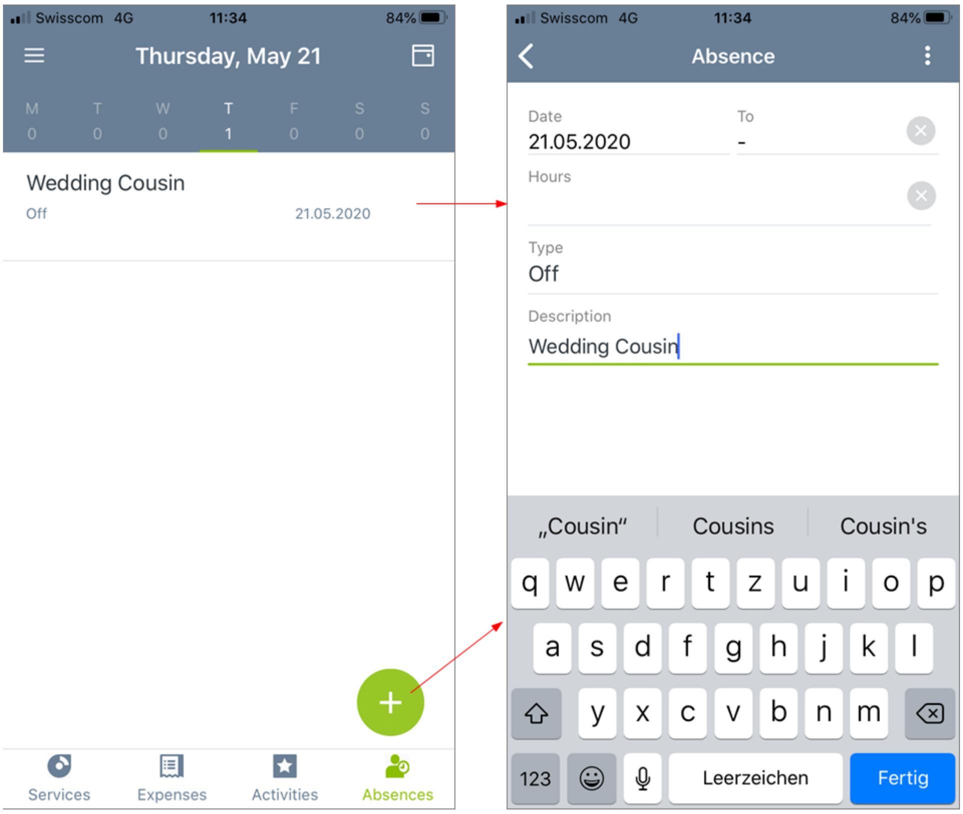
- Weekly overview
- Addresses
- Activities pending
- Settings

9.3 Input an absence

Line: Standard, Expert | Module: PSA | Apps: Phone | Version: 6.3.0.15

Absences can now be directly input in the Phone App.

Clicking on the green + Button generates a new absence.



Wedding Cousin
Off 21.05.2020

Date: 21.05.2020
To: -
Hours
Type: Off
Description: Wedding Cousin

Figure 38: Input an absence in the Phone App

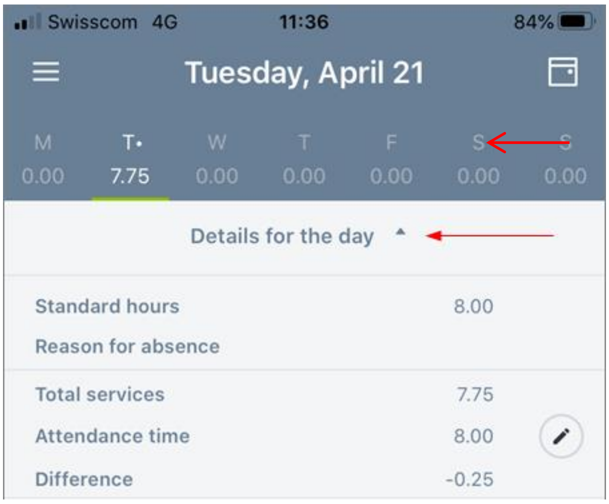
A click on Type displays the list of absence types available in Vertec.

Absences registered for the day are displayed in the daily details (see 9.4).

9.4 Daily details

Line: Standard, Expert | Module: PSA | Apps: Phone | Version: 6.3.0.15

For each day, the following daily details may be shown or hidden:



Standard hours	8.00
Reason for absence	
Total services	7.75
Attendance time	8.00
Difference	-0.25

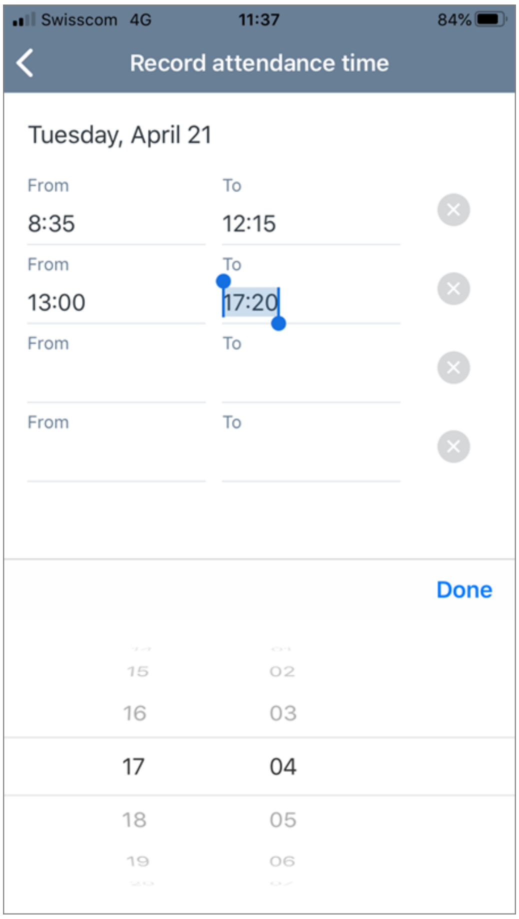
- Standard hours. Not shown, if the system setting Project/Mandate > Display Std hours overwiew on services list is set to `No` or if the user has no overtime balance provided.
- Reasons for absences, if existing for the day (see 9.3).

- Total services for the day
- Attendance time: Attendance time can now be entered here by clicking on the pen symbol (see 9.5).
- Difference (between services and attendance time). Both are displayed only if the system setting Project/Mandate > Display attendance time list for service records has been set to `Yes`.

9.5 Record attendance time

Line: Standard, Expert | Module: PSA | Apps: Phone | Version: 6.3.0.15

A click on the pen symbol behind Attendance time opens the form to record attendance time:



Record attendance time

Tuesday, April 21

From	To
8:35	12:15
13:00	17:20

Figure 39: Input presence hours for the day

10 Interfaces and Addins

10.1 Cloud capable DATEV accounts receivable interface

Line: Expert | Module: PSA | Apps: Full featured | Version: 6.3.0.6

The new DATEV accounts receivable interface makes it possible to transfer debtor data und bookings from Vertec to DATEV.

The interface is cloud enabled and implements list-based bookings. Since a file is created for each booking operation, booking can be performed for lists of transactions instead of single objects.

Detailed information related to the DATEV accounts receivable interface is available Online Knowledge Base under www.vertec.com/kb/datev-debitoren-schnittstelle.

10.2 Run my Accounts accounts receivable interface

Line: Standard, Expert | Module: PSA | Apps: Full featured | Version: 6.3.0.9

Run my Accounts (www.runmyaccounts.ch) is an online bookkeeping service. With the list-based and cloud capable Vertec Run my Accounts interface, debtor transactions can be booked and reversed.

The interface is cloud capable and implements list-based booking. Since a file is created for each booking operation, booking can be performed for lists of transactions instead of single objects.

Detailed information A detailed description of how to install and use the Run my Accounts accounts receivable interface is available in the Online Knowledge Base under www.vertec.com/kb/rma-debitoren-schnittstelle.

10.3 BMD accounts payable interface

Line: Standard, Expert | Module: PSA | Apps: Full featured | Version: 6.4

The BMD accounts payable interface generates CSV files. It allows to book and reverse invoices from suppliers in BMD.

The interface is cloud capable and implements list-based booking. Since a file is created at the time of each booking operation, booking can occur for lists of transactions instead of single objects.

Detailed information A detailed description of how to install and use the BMD accounts payable interface is available in **the** Online Knowledge Base under www.vertec.com/kb/bmd-kreditoren-schnittstelle.

10.4 Address several document libraries via the Sharepoint extension

Line: Standard, Expert | Module: PSA | Apps: Full featured | Version: 6.3.0.18

With the Vertec Sharepoint Extension, various document libraries can now be addressed if required.

The different libraries are distinguished via their file path in Vertec. This means that the **Library URL** setting in Vertec no longer necessarily represents the complete URL to a Document Library. Only together with the file path it must be a valid Sharepoint URL for a specific Document Library.

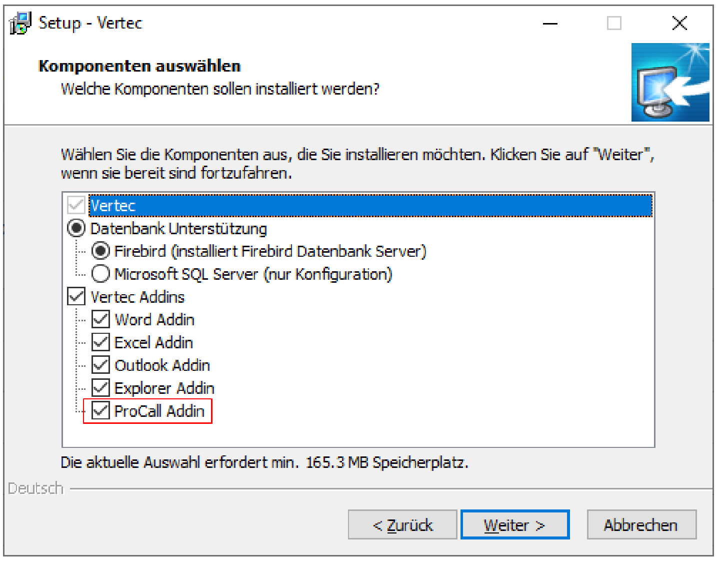
If you want to use several Document Libraries and store documents per class, you have to consider the following:

- In System settings > Document path, the root DMS path must be stored, for instance: `sharepoint:\MultiSite2\Shared Documents`. This is the standard path used for all objects without a given individual path (see next point). The standard path must be a self-contained valid path.
- In folder Settings > Class settings, an absolute SharePoint path must be stored for all classes differing from the standard directory. For addresses, for instance, `sharepoint:MultiSite1\Shared Documents\Contacts\%if oclistypeof(Kontakt) then oclastype(Kontakt).firma.name else name endif%`.

10.5 Integration of the ProCall Addin in Setup

Line: Standard, Expert | Module: PSA | Apps: Desktop, Cloud | Version: 6.3.0.13

The ProCall Addin has been integrated into Vertec Setup and, like other Addins, can now be selected in the Setup dialogue:



Setup - Vertec

Komponenten auswählen

Welche Komponenten sollen installiert werden?

Wählen Sie die Komponenten aus, die Sie installieren möchten. Klicken Sie auf "Weiter", wenn sie bereit sind fortzufahren.

- ☑ Vertec
- ● Datenbank Unterstützung
  - ● Firebird (installiert Firebird Datenbank Server)
  - ○ Microsoft SQL Server (nur Konfiguration)
- ☑ Vertec Addins
  - ☑ Word Addin
  - ☑ Excel Addin
  - ☑ Outlook Addin
  - ☑ Explorer Addin
  - ☑ ProCall Addin

Die aktuelle Auswahl erfordert min. 165.3 MB Speicherplatz.

Figure 40: Selection of the ProCall Addin in Vertec Setup

ProCall will then be automatically installed by Setup.

The ProCall Addin is also integrated in the Cloud App installer:

- Install Cloud App via Portalpage
- Allow Extended installations
- The ProCall Addin is copied into the Cloud App installation

Important notice Before the installation of the ProCall Addin, ProCall must be terminated. Additional information in www.vertec.com/kb/procall-addin.

11 Scripting

11.1 OCL call operators for custom business logic

Line: Expert | Module: PSA | Apps: Full featured | Version: 6.3.0.2

With OCL call operators, specific Python functions may be called from OCL code. This makes it possible to write extensions to the business logic in Python and use them in different places instead of the provided business logic, in scripts for instance or to change the display of lists or reports.

Since OCL is static and strongly typed, it requires a separate OCL operator for each combination of returned value and arguments. The following are available:

- callString (-> string)
- callCurr (-> currency)
- callCurrDate (date -> currency)
- callCurrDateDate (date, date -> currency)
- callStringString (string-> string)

OCL operators are applied to BusinessclassesRoot and all have as first argument the name of the implementing Python function. This name is specified in the form `<module>.<function name>`.

Example
A script module with name customlogic is set up. It includes a function with the signature

```
def percentofcompletion(project, eff_date)
```

The call in OCL would be as follows:

`proj->callCurrDate('customlogic.percentofcompletion', somedate)`

The result of the OCL Expression is a currency value.

The name of the used call operator includes the type of the returned value and of the arguments (returned value currency "Curr", argument "Date").

11.2 Template engine in Python

Line: Expert | Module: PSA | Apps: Full featured | Version: 6.3.0.9

There’s a new generally usable template engine in Python. It is based on the Jinja2 template engine (http://jinja.pocoo.org).

To make it easily usable in Vertec, the method **rendertemplate(templateString, data)** has been made available on the vtcapp module.

This method is also usable in restricted scripting mode (cloud subscription) and returns a Unicode string.

- The **templateString** argument can be a string or Unicode String and can include Jinja2-specific markups.
- To pass data to the function through additional arguments, the following possibilities exist:
  - Python dictionary with string values as keys. Defines the variables available in the template
  - Any number of keyword arguments which define the variables available in the template.
  - No additional argument. In that case the template is processed without data.

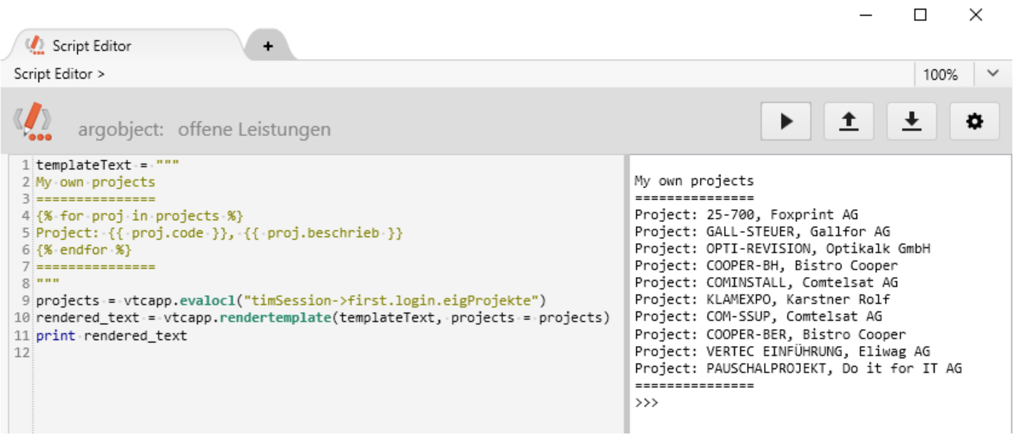
The most important structures are as follows:

- A block is enclosed in `{% ... %}`. It contains a control statement or a variable assignment.

  - `{% if proj.code == "ABC" %} ... {% endif %}`. An if statement is used in a block for the conditional output of a template part. An if must be terminated with an endif. An optional `{% else %}` is possible within.
  - `{% for proj in projects %} ... {% endfor` A for statement is used for the repetition (iteration) of a template part. Must be terminated with endfor.
- An expression is delimited by `{{ ... }}`. The expression will be evaluated within the context of the template and mixed into the output of the template.

Example code to output of a project list

```
templateText = """
My own projects
===============
{% for proj in projects %}
Project: {{ proj.code }}, {{ proj.beschrieb }}
{% endfor %}
===============
"""
projects = vtcapp.evalocl("timSession->first.login.eigProjekte")
rendered_text = vtcapp.rendertemplate(templateText, projects = projects)
print rendered_text
```



```
templateText = """
My own projects
===============
{% for proj in projects %}
Project: {{ proj.code }}, {{ proj.beschrieb }}
{% endfor %}
===============
"""
projects = vtcapp.evalocl("timSession->first.login.eigProjekte")
rendered_text = vtcapp.rendertemplate(templateText, projects = projects)
print rendered_text
```

```
My own projects
===============
Project: 25-700, Foxprint AG
Project: GALL-STEUER, Gallfor AG
Project: OPTI-REVISION, Optikalk GmbH
Project: COOPER-BH, Bistro Cooper
Project: COMINSTALL, Comtelsat AG
Project: KLAMEXPO, Karstner Rolf
Project: COM-SSUP, Comtelsat AG
Project: COOPER-BER, Bistro Cooper
Project: VERTEC EINFÜHRUNG, Eliwag AG
Project: PAUSCHALPROJEKT, Do it for IT AG
===============
>>>
```

Figure 41: The script of the example executed in the script editor

Additional information Additional information on this topic is found in the Online Knowledge Base under www.vertec.com/kb/pythoninterfaces#rendertemplate.

11.3 Python module ziputils available

Line: Expert | Module: PSA | Apps: Full featured | Version: 6.3.0.2

A Vertec module named ziputils is available as of version 6.3.0.2. It allows to generate ZIP files in Python. It is also available in restricted scripting mode and contains the method:

createzip(contentlist: tuples): bytestream

The method generates a ZIP file from a list of tuples (filename, content) and returns a byte stream which can be downloaded/stored as filename.zip.

Example

```
import ziputils

zip = ziputils.createzip([('file.txt', 'Hello, this is a text')])
vtcapp.sendfile(zip, 'ziptest.zip', True)
```

11.4 Python method for file upload

Line: Expert | Module: PSA | Apps: Full featured | Version: 6.3.0.4 / 6.3.0.12

With the new Python method for file upload, the user can select a local file in the client application. This file is then uploaded to the server and is available for further processing as a binary stream value returned by the Python method.

```
requestfilefromclient(title: string, path: string, filter: string
                             [, abspath: string]): (filename, file)
```

- **title**: title of the dialogue
- **path**: path of the directory to be opened in the dialogue. If the path does not exist, the desktop is displayed.
- **filter**: filter expression to narrow the selection of files in the dialogue. Consists of filter pairs which must each include a name and an expression. `"Office Documents|*.docx;*.xlsx|Text|*.txt|Xml|*.xml"`
  The character | must be used as a separation symbol between pairs and, in each pair between name and expression.
  In the case of several file extensions, they must be separated by semicolons.
- **abspath**: absolute path of the file to be uploaded. This parameter is optional; If not specified, no dialogue is displayed.

The method returns a tuple consisting of the absolute filename and the content of the file.

The maximum size of the file to be uploaded is 50 megabytes.

There exists a blacklist of directories from which no files can be requested, for example the Windows SYSTEMROOT (typically C:\Windows). In this case, an error message is displayed.

Examples

```
vtcapp.requestfilefromclient("Hello world", r"C:\MyDirectory", "Office Documents|*.docx;*.xlsx|Text|*.txt|Xml|*.xml")
```

Opens a dialogue with the title "Hello world" in the directory C:\MyDirectory. It allows to upload Office-, TXT- or XML files.

A simpler syntax is supported as well:

```
vtcapp.requestfilefromclient("Hello world", r"C:\MyDirectory", "*.txt")
```

Opens a dialogue with the title "Hello world" in the directory C:\MyDirectory. It allows to upload TXT files.

```
vtcapp.requestfilefromclient("", "", "", r"C:\MyDirectory\MyFile.txt")
```

Uploads the file without dialogue.

Restrictions with the Web App

The file dialogue cannot display a user-defined dialogue title nor can it be set on a starting directory. The first two parameters `title` and `path` are therefore ignored.

If a client-side path is specified in the parameter `abspath`, an error message is displayed.

11.5 getwithsql Python method with object list and Id field

Line: Standard, Expert | Module: PSA | Apps: Full featured | Version 6.3.0.8

SQL queries, which are based on lists of objects, have been implemented with a "...in (id1, id2, id3,...)" construction in getwithsql. With Firebird databases this type of query leads to error messages if more than 1500 id's are checked. To circumvent this and simplify SQL queries with references to an object list, the getwithsql method has been extended by 2 optional arguments for an object list and an id field.

Call before

```
getwithsql(class: string, where: string, order: string)
```

Call now

```
getwithsql(class: string, where: string, order: string, [idfilterfield: string, objectlist: list])
```

– **idfilterfield**: name of the member used to filter the list.
– **objectlist**: List of objects to be searched for the field specified in idfilterfield.
The new parameters are optional.

Additional information and examples related to this subject are available in the Online Knowledge Base under www.vertec.com/kb/pythoninterfaces.

11.6 Member based SQL string search

Line: Standard, Expert | Module: PSA | Apps: Full featured | Version 6.3.0.8

A new method is available to select single members using SQL. In such a search, it is possible to sufficiently secure the search parameter in the code and to check as well in advance the access right to the class. For this reason, the use of the method is also permitted without explicit SQL rights. The syntax is as follows:

GetWithSqlByMember(class, member, expression, order [optional]): object-list

The expression must be a comparison string (without delimiter). It may include %-place holders. String delimiters within the string are not taken into account (escaped)..

If the expression includes wild card characters, the comparison occurs using the SQL "LIKE" operator, otherwise using the "=" operator.

The comparison is case insensitive, meaning that no difference is made between upper-case and lower-case characters.

When the function is called, the access right of the user to the member used in the selection is checked. The query will be executed only if the user has a class-wide access right to the member, otherwise an error message is displayed.

Example:

```
vtcapp.getwithsqlbymember("Projekt", "Code", "ABC", "")
```

Finds all projects with project code "ABC", "aBc" oder "abc".

```
vtcapp.getwithsqlbymember("Projekt", "Code", "AB%", "")
```

Finds all projects with project code beginning with "AB", "ab", etc.

Modification of the ProCall Addin

The ProCall Addin has been converted to the new method and the standard user group has been given reading rights to the field `normalizedZiel` of the class KommMittel.

Starting with Vertec version 6.3.0.8, no SQL right is therefore required to use the ProCall Addin.

11.7 Specifying the search path with the SendFile mechanism

Line: Standard, Expert | Module: PSA | Apps: Full featured | Version 6.3.0.8

The Python method **sendfile()** allows to store on the client a file from the server. This functionality has been available since Vertec 6.2.

Up to now, only the specification of the file name was supported. And, with the Desktop- and Cloud App, the file was always stored in the user’s Temp directory.

A full storage path can now be specified. If possible, the file will be stored at this location on the client. The following rules apply:

- If the target path does not exist, it is created without asking for confirmation.
- If the storage (or the creation of the path) fails due to missing access rights, an error message is displayed.
- Storage with a path specification works identically in the Desktop- and Cloud App. In the Web App, the specified path is ignored, only the filename is taken into account for the download.

For this reason, the method now has a return value `True` or `False`:

- True, if the file has been stored,
- False, if the file has not been stored,
- False, if the operation has been interrupted due to user input (click on Cancel).
- The Web App always returns True as it is not possible in this case to check if the file has been stored. When the Web App is used, we recommend turning on the automatic download in the browser.

Additional information and examples on this subject are available in the Online Knowledge Base under www.vertec.com/kb/pythoninterfaces.

11.8 executereport() available for extended Office reports

Line: Standard, Expert | Module: PSA | Apps: Full featured | Version: 6.3.0.4

The Python method `executereport()` can now also be used for extended Office reports.

The method expects a report definition as argument (report registered in Vertec).

Additional information Additional information related to this Python method is available in the Online Knowledge Base under www.vertec.com/kb/pythoninterfaces.

11.9 New Python module in restricted scripting mode

Line: Expert | Module: PSA | Apps: Full featured | Version 6.3.0.2/6.3.0.16

The following Python modules are now available in restricted scripting mode:

As of version 6.3.0.2

- ziputils (see section 11.11)

As of version 6.3.0.16

- array
- bisect
- calendar
- collections
- decimal
- difflib
- fractions
- functools
- hashlib
- heapq
- hmac
- HTMLParser
- itertools
- locale
- operator
- random
- StringIO
- stringprep
- textwrap
- time
- traceback
- unicodedata

As well as:

- Python `email` package with all its submodules
- Built-in `help` function

Additional information Additional information related to the subject **Restrict Scripting** is available in the Online Knowledge Base under www.vertec.com/kb/restrict-scripting.

11.10 Automatically reload a modified Python script module

Line: Expert | Module: PSA | Apps: Full featured | Version: 6.4

When script entries with Python code are imported as Python modules, changes in the code were only taken into account when Vertec was restarted.

In case of a code modification in a script entry, a module already loaded will now be deleted from intermediate memory on the basis of its designation. When the next import of the module occurs, it is reloaded with the changed Python code and the new code will therefore be used.

Nothing has changed with extensions. If the extension code in a script entry has been changed, the extension must still be reloaded. The module however will already be up to date.

11.11 Support of extended rights in imported script modules

Line: Expert | Module: PSA | Apps: Full featured | Version: 6.3.0.14

A script can be given extended rights. Up to now, the management of these extended rights based on the fact that, at some point, a script text is executed in a specific context. When executing code from imported modules via OCL call operators (see 11.1) or via scripts, this was not possible, since the code is then executed in a different context.

This mechanism has been extended in such a way that the marking of script entries now also functions with extended rights if the script entry is called by an OCL Call Operator or imported as module in another Script.

Additional information Additional information on extended rights is found in the Online Knowledge Base under www.vertec.com/kb/erweiterte-berechtigungen.

11.12 Python ODBC library pyodbc is provided

Line: Expert | Module: PSA | Apps: Desktop | Version: 6.4

Starting with Vertec 6.4, we provide the pyodbc package in the Python folder in the Vertec installation directory. It is called **pyodbc.pyd** and provides database accesses with ODBC, for instance for migrations.

With `import pyodbc` in Vertec Python scripts, the whole functionality becomes available (only in non-restricted mode, without restrict scripting).

Additional information A detailed documentation of the Python ODBC library pyodbc is available from the editor at https://github.com/mkleehammer/pyodbc.

12 Performance

12.1 Optimizing object references in OCL

Line: Standard, Expert | Module: PSA | Apps: Full featured | Version: 6.4.

OCL evaluates expressions strictly from left to right and uses the result of the first expression part as a source for the next expression part. Simple expressions like

`projekt` (respectively `Projekt.allInstances`)

consist of one expression part only and do not give rise to performance problems.

If this expression is now extended with a persistent member, for instance "aktiv":

`projekt->select(aktiv)`

the duration will be the same since projects are already loaded with all their members.

If, however, a selection is performed on a sublist, for instance

`projekt->select(phasen->size>0)`

(all projects with at least one phase) the execution becomes much slower. The reason is that the evaluation is performed from left to right as described above.

1. Load all projects
2. Evaluate the expression `phasen->size>0`. This means that for each single project in the list, the phases of this project must be individually loaded, to determine if it becomes part of the list.
3. Continue with the next project to the end of the list.

This is where the new optimization steps in. Since it is clear in our example that all phases of projects in the list must be loaded, the complete list of phases is now loaded into memory at once. This means that there is only a one-time database access. Processing will then be much faster because all phases are already loaded.

This optimization occurs automatically: as of now, expressions are processed in this optimized way whenever OCL expressions with sublists are executed directly, for instance as expression in Python or in an expression folder.

Optimization of object references in Vertec lists

The optimization has also been implemented for column expressions in Vertec lists. Indeed, sublists or link objects are often accessed there, for instance `projektleiter` on projects, or phases.

This optimization is used in lists:

1. For all columns for which all the data is required anyway, in any case when summing or sorting is performed.
2. For all objects which are loaded by default when a list is called: all the represented objects plus one page.
3. For a total number of objects up to approx. 300, since the probability that the user will scroll is large. In that case, a slowdown may occur when the list is built for the first time. In the case of more than 300 entries. the building of the list will be much slower, even though the user may not wish to see all entries.

It is not used:

– If the expression is too complex.
– In the case of sublists of lists, subphases for instance.
– If none of the conditions mentioned above apply.

12.2 Speed up of access-right check on projects

Line: Standard, Expert | Module: PSA | Apps: Full featured | Version: 6.3.0.4

The order in which the access right to a project is checked has been modified. This leads to a performance increase when opening the project combo box to input services for a very large number of projects.

12.3 Performance optimization when choosing service type

Line: Standard, Expert | Module: Budget & Phases | Apps: Full featured | Version: 6.3.0.6

When a type is selected while inputting services, the system must check, on the base of assignments related to the entry, if the type is allowed for the current phase and user.

This check has been optimized by distinguishing if a project phase has already been selected or not. If a phase has been selected, the check is performed for this single phase, not for all phases. Hence a performance improvement.

Additional information Additional information on the subject of phase assignments in the Online Knowledge Base under www.vertec.com/kb/projektphasen.

12.4 Outlook Addin: cc: addresses are no longer stored

Line: Standard, Expert | Module: PSA | Apps: Desktop, Cloud | Version: 6.3.0.6

Performance optimization for the **Outlook Addin**: E-Mail addresses only appearing in the cc: section of an e-mail are no longer stored as addressees in the activity.

13 Customizing

13.1 Non-keyword folders can also be inserted in pages

Line: Expert | Module: PSA | Apps: Full featured | Version: 6.3.0.12

Through customizing, single keywords can through the element `KeywordGroup` be placed in other pages.

In this case, a group with the designation of the keyword folder is represented as either a checkbox or a combo box depending on its type.

Starting with Vertec 6.3.0.12, folders not marked as keyword folders can also be placed in pages in this way.

Additional information Additional information on placing keywords in pages can be found in the Online Knowledge Base under www.vertec.com/kb/seiten#stichwort.

13.2 Modification of code editors

Line: Standard, Expert | Module: PSA | Apps: Full featured | Version: 6.3.0.3

In the code editors, a monospaced font is now used (instead of a proportional font). The font type is **`Consolas`**, already used in the Python console (F3).

This concerns the following editors:

- Code for registered scripts (script entries)
- Code in the script editor
- Report definitions of extended Office reports
- XML code boxes in list and class settings
- OCL expression editor

In addition, a horizontal scroll bar appears in the Python console (F3) and in the code window of the Script editor.

13.3 Web App: Script editor with syntax highlighting

Line: Standard, Expert | Module: PSA | Apps: Web | Version 6.3.0.8

The syntax highlighting introduced with Vertec 6.2 in code editors is now available in Web App as well.

13.4 Keyboard shortcuts for the Script Editor

Line: Standard, Expert | Module: PSA | Apps: Full featured | Version 6.3.0.8

The following shortcuts have been introduced in the Script Editor:

- `Ctrl+E` (Execute script)
- `Ctrl+Z` (Undo)

14 Config Sets

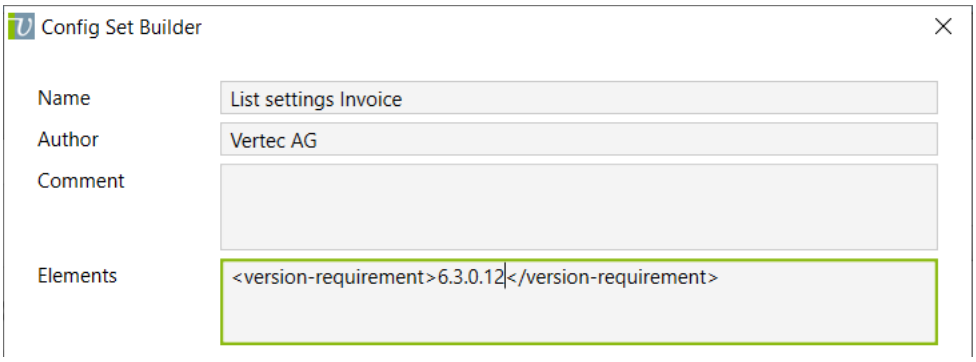
14.1 Minimal Vertec version for Config Sets

Line: Standard, Expert | Module: PSA | Apps: Desktop, Cloud | Version: 6.4

Optionally, the youngest Vertec version required for a Config Set can now be specified. As for other existing requirements, it is implemented by a manual XML-Element. The syntax is as follows:

```
<version-requirement>6.3.0.12</version-requirement>
```

This is specified in the field **Elements** in the Config Set Builder:



Config Set Builder

Name: List settings Invoice

Author: Vertec AG

Comment:

Elements: <version-requirement>6.3.0.12</version-requirement>

Figure 42: This Config Set requires at least the 6.3.0.12 version of Vertec.

- If a Config Set with a version requirement is imported and the Vertec version is older than the specified one, an error occurs and the Config Set cannot be imported.
- If a Config Set with version requirement is imported into a version of Vertec in which the version requirement feature is not implemented, an error occurs as well and the Config Set cannot be imported. Hence Vertec 6.4 or later is required for Config Sets with this feature.

14.2 Support of additional fields in Config Sets

Line: Standard, Expert | Module: PSA | Apps: Desktop, Cloud | Version: 6.4

If an object that has additional fields is added to a Config Set, the field values and the corresponding additional field definitions are automatically added. These do not have to have an entry ID like other objects, but are uniquely identified by class and name.

If the additional fields are on an additional class, the reference is also entered directly. Here it must be checked that an additional class with the same number does not already exist in the target system, because the class is simply copied into the additional fields and would in that case be incorrectly linked.

If the class settings of the additional class should be transferred as well, they must be separately added to the Config Set.

Backwards compatibility
Config sets created in version 6.4 or later can no longer be imported in older Vertec versions if they include objects with additional fields.

In these cases, we recommend to specify a minimal version (see section 14.1).

14.3 MLStrings taken into account in Config Sets

Line: Standard, Expert | Module: PSA | Apps: Desktop, Cloud | Version: 6.4

From now on, for MLString attributes (see section 8.3) not only the text in the current language is transmitted, but all specified languages including the native language introduced with this version.

Backwards compatibility

- If one attribute only is given in the actual language, the attribute is passed as a normal string. In this case, the import works with all versions.
- In every other case (also if only a native term is given) the specified languages are passed as `mltext` attributes. This element only exists from version 6.4 on, which means:
  - Old Config Sets can also be imported with newer Vertec versions.
  - Config sets created with Vertec 6.4 or later cannot be imported in older Vertec versions if more than one attribute is specified in actual language in the MLString attributes.

15 Technical aspects / Miscellaneous

15.1 Shutdown of a session delayed in case of a disconnection

Line: Standard, Expert | Module: PSA | Apps: Cloud Clients | Version: 6.3.0.18

In Vertec, the session of a cloud client is terminated on the server if a disconnection occurs, even if the session has not been regularly terminated. Example: the browser on which Web App is running is closed.

The reason for this is to avoid having no longer used hanging processes on the server.

The shutdown of the session has now been delayed by 5 minutes to allow certain processes to complete their task (a running event script, for instance).

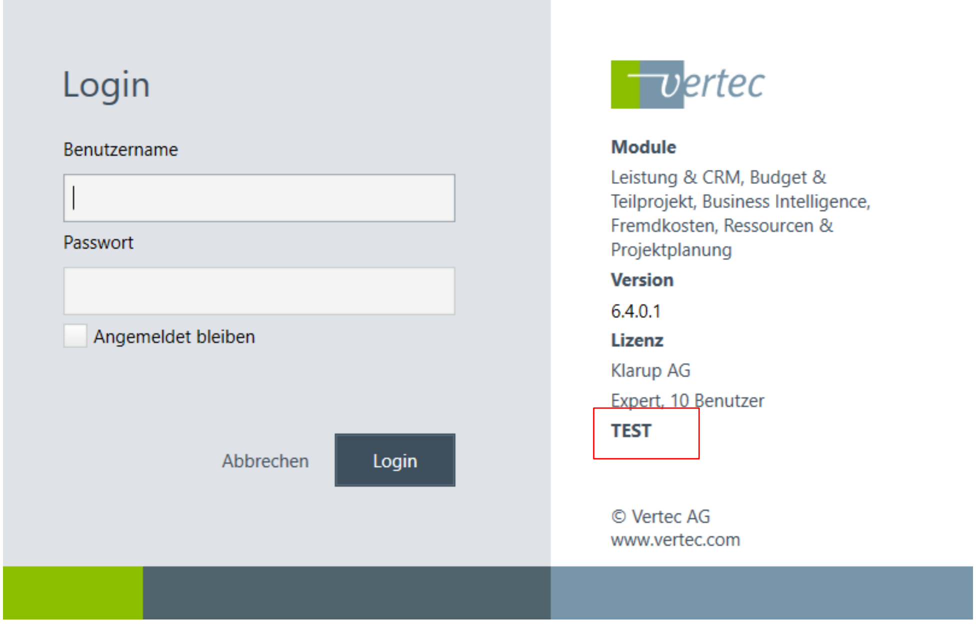
This happens in the background; the user is not aware of this change.

15.2 Display instance name in Login and About Vertec dialogues

Line: Standard, Expert | Module: PSA | Apps: Desktop | Version: 6.4

A database section in the Vertec.ini file may include a value named `InstanceName`. This value is displayed on the taskbar icon of the Vertec Desktop App.

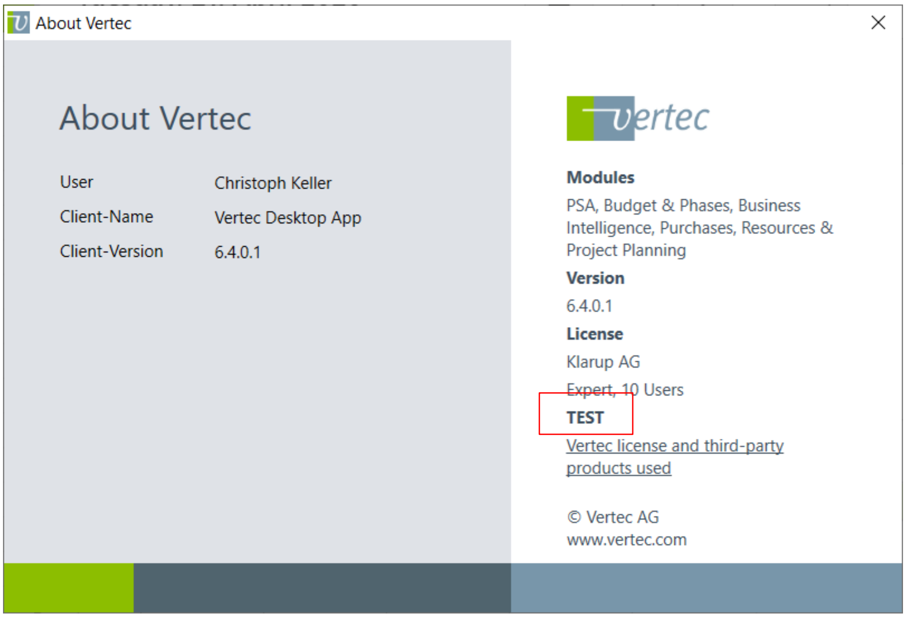
This instance name is now also displayed in the Login and About Vertec dialogues:



Module
Leistung & CRM, Budget & Teilprojekt, Business Intelligence, Fremdkosten, Ressourcen & Projektplanung
Version
6.4.0.1
Lizenz
Klarup AG
Expert, 10 Benutzer
TEST

© Vertec AG
www.vertec.com

Figure 43: The instance name is displayed in the login dialogue



About Vertec

About Vertec

User Christoph Keller

Client-Name Vertec Desktop App

Client-Version 6.4.0.1

Modules

PSA, Budget & Phases, Business Intelligence, Purchases, Resources & Project Planning

Version

6.4.0.1

License

Klarup AG

Expert, 10 Users

TEST

Vertec license and third-party products used

© Vertec AG
www.vertec.com

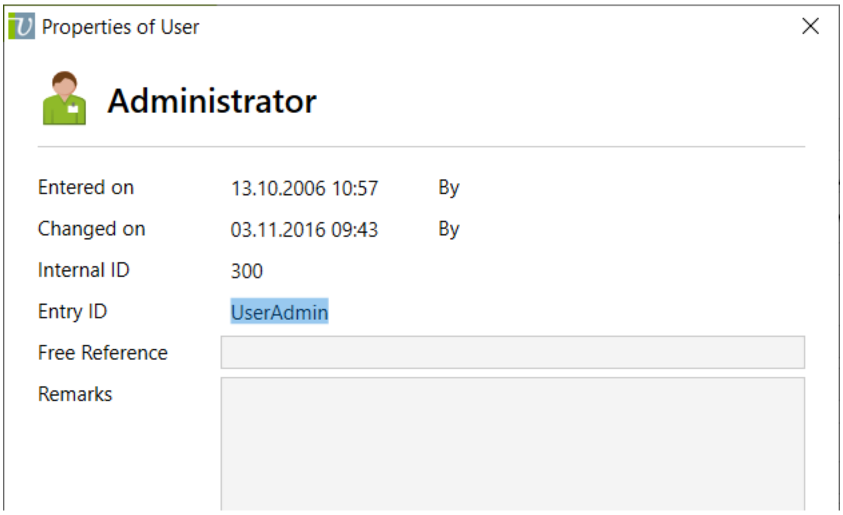
Figure 44: The instance name is displayed in the About Vertec dialogue

Additional information Additional information on this subject is available in the Online Knowledge Base under www.vertec.com/kb/ini#testinstallationen.

15.3 Entry ID attributed to user Administrator

Line: Standard, Expert | Module: PSA | Apps: Full featured | Version: 6.3.0.14

The Administrator, a user provided by default, now has an Entry ID. The important reason is that he may be referenced in the future:



Properties of User

Administrator

Entered on 13.10.2006 10:57 By

Changed on 03.11.2016 09:43 By

Internal ID 300

Entry ID UserAdmin

Free Reference

Remarks

Figure 45: Entry ID for the administrator

This value is also set for the administrator in existing databases if no value has been entered there before.

15.4 CompanyName details on booked invoices

Line: Standard, Expert | Module: PSA | Apps: Full featured | Version: 6.3.0.17

The field **CompanyName** and related fields on the invoice have been introduced in Vertec 6.3. When an invoice is booked, the actual values of these fields are stored in persistent fields with the prefix `verr`. When booked invoices are displayed, these fixed values are accessed.

In invoices which have been booked before the upgrade to version 6.3, this fixed company information is left empty (all fields starting with `verrCompany`). The consequence is for instance that if old invoices are displayed again, the address is missing. To avoid this problem, a distinction between following situations is made for the computation of the derived `CompanyXXX` members:

- If all `verrCompanyXXX` members are empty, the actually valid not-fixed value (via the payment type or the system settings) is used.
- If at least one `verrCompanyXXX` member is not empty, all verr values are used (as before).

15.5 Links to AbstractAdresseintrag eliminated

Line: Standard, Expert | Module: PSA, Purchases | Apps: Full featured | Version: 6.4

Various associations in the model directly pointed to AbstractAdresseintrag, the base class of Adresseintrag. This had as a consequence that a cast was required when accessing the client of a project for instance:

`argobject.eval("kunde.name")` did not work. A cast to the Adresseintrag was then necessary: `argobject.eval("kunde.oclastype(Adresseintrag).name")`.

The model has now been changed in such a way that the associations directly point to the class Adresseintrag. The following links have been modified:

- Projekt <> Gegenparteien
- Projekt <> Kunde
- Auslage <> Lieferant
- Auslage <> xRechnungsadresse
- Auslage <> xLieferadresse
- Kreditor <> Lieferant

Backwards compatibility
Since the expressions with casts still function and the class AbstractAdresseintrag remains in the model as base class of Adresseintrag, nothing is changed for existing installations.