



April 2020

Vertec-Software Release 6.4

Inhaltsverzeichnis

1 Einführung	5
1.1 Neuinstallation von Vertec	5
1.2 Vor dem Update auf 6.4	5
1.3 Update von Vertec	5
1.4 Das erste Aufstarten nach dem Konvert	5
2 Vertec 64 Bit	6
2.1 Konvertierung einer Vertec Datenbank mit MS SQL-Server	6
2.2 Schnittstellen mit Vertec 64 Bit	6
3 Version 6.4	8
3.1 Highlights der Version 6.4	8
3.2 Eingestellte Features in Vertec 6.4	8
4 Business Intelligence (BI) Modul	9
4.1 BI Ansicht öffnen	9
4.2 BI Anzeigen	15
4.3 BI Kennzahlen	24
4.4 Generatoren	30
4.5 Generatoren erweitern / erstellen	31
4.6 Daten berechnen	36
4.7 BI Berechtigungen	39
4.8 Rückwärtskompatibilität	40
4.9 Inbetriebnahme	45
5 Leistungs- und Spesenerfassung	46
5.1 PDF Dokumente als Spesenbelege	46
5.2 Direktes Setzen von Projekt in Suchdialog (edtProjekt)	47
5.3 Berechnung und Darstellung von Ferienwerten	48
6 Berichte	49
6.1 Rechnungen nach ZUGFeRD 2.0 Standard (XRechnung)	49
6.2 Rechnung mit Bearbeitersummen als erweiterter Office-Bericht	49
6.3 Rechnung mit Phasensummen als erweiterter Office-Bericht	49
6.4 Erweiterte Rechnungen mit QR-Code an Spezifikation 2.1 angepasst	50
6.5 Vorschuss Bericht neu mit Stichdatum	50
6.6 Report Ferienabrechnung zeigt Vortrag immer an	50

6.7 Rekapitulation Buchungen nach Währung gruppiert	51
6.8 Performance Verbesserung bei Fakturavorschlag	51
6.9 Performance Verbesserung bei Budget Übersicht	51
6.10 Expressions in Kopf- und Fusszeilen von Vertec-generierten Word-Berichten	51
6.11 Summierung über mehrere Stufen in erweiterten Excel-Berichten	52
7 Ordner und Listeneinstellungen	54
7.1 Mehrstufige Sortierung in Listen	54
7.2 Auswahl-Suchfeld für Expression-Ordner Abfragen	55
7.3 SQL Ordner setzt "Suchdialog anzeigen" wieder standardmässig	56
8 Einstellungen	58
8.1 Geplante Aufgaben	58
8.2 2 Faktor Authentisierung	60
8.3 Native Sprache in MLString Attributen	63
8.4 Veränderung bei den Shortcuts der Navigationsansichten	64
8.5 Sprache auswählen Dialog per Systemeinstellung ausblenden	65
9 Phone App Erweiterungen	66
9.1 Sprache gemäss Einstellungen des Vertec-Users	66
9.2 Umbau Wochenansicht	66
9.3 Abwesenheiten erfassen	66
9.4 Tagesdetails	67
9.5 Präsenzzeiten erfassen	68
10 Schnittstellen und Addins	69
10.1 Cloudfähige DATEV Debitoren Schnittstelle	69
10.2 Run my Accounts Debitoren Schnittstelle	69
10.3 BMD Kreditoren Schnittstelle	69
10.4 Mehrere Dokumentbibliotheken via Sharepoint-Schnittstelle anbinden	69
10.5 Aufnahme des ProCall-Add-ins in den Vertec Setup	70
11 Scripting	71
11.1 OCL Call Operatoren für Custom Businesslogik	71
11.2 Template Engine in Python	71
11.3 Eingebautes Python-Modul ziputils	72
11.4 Python Methode für File Upload	73
11.5 getwithsql Python Methode mit Objektliste und Id-Feld	74
11.6 Member basierte SQL String Suche	74

11.7 Angabe von Speicherpfad bei SendFile Mechanismus	75
11.8 executereport() auch für erweiterte Office-Berichte	75
11.9 Neue Python Module im Restricted Scripting Modus	76
11.10 Geänderte Python Script-Module automatisch neu laden	76
11.11 Unterstützung von erweiterten Berechtigungen in importieren Script-Modulen	76
11.12 Python ODBC Library pyodbc mitgeliefert	77
12 Performance	78
12.1 Optimierung von Objekt-Referenzen in OCL	78
12.2 Beschleunigung Berechtigungsprüfung auf Projekt	79
12.3 Performance-Optimierung bei der Auswahl von Tätigkeiten	79
12.4 Outlook Addin: cc: Adressen nicht mehr speichern	79
13 Customizing	80
13.1 Auch Nicht-Stichwort-Ordner in Seiten einblendbar	80
13.2 Anpassung von Code Editoren	80
13.3 Web App: Script Editor mit Syntax Highlighting	80
13.4 Tastaturkürzel für den Script Editor	80
14 Config Sets	81
14.1 Minimale Applikations-Version in Config Sets	81
14.2 Unterstützung von Zusatzfeldern in Config Sets	81
14.3 Berücksichtigung von MLStrings in Config Sets	82
15 Technik / Diverses	83
15.1 Beenden der Session nach Verbindungsabbruch verzögern	83
15.2 Instanzname in Login- und Über Vertec-Dialog anzeigen	83
15.3 EintragId für Admin User vergeben	84
15.4 CompanyName Angaben auf verrechneten Rechnungen	85
15.5 Verbindungen zu AbstractAdresseintrag aufgehoben	85

1 Einführung

Dies sind die Release-Notes zur Vertec-Version 6.4. Bitte beachten Sie folgende Punkte:

1.1 Neuinstallation von Vertec

Für eine Neuinstallation verwenden Sie das zum Download verfügbare Installationsprogramm [VertecSetup-6.4.0.exe](#). Nähere Informationen zur Installation finden Sie in unserer Online Knowledge Base unter www.vertec.com/kb/neuinstallation.

1.2 Vor dem Update auf 6.4

Bitte beachten Sie dazu das Kapitel 2 über Vertec 64 Bit.

1.3 Update von Vertec

Für ein Update einer Vertec Installation verwenden sie ebenfalls das Installationsprogramm. Bei grösseren Kundeninstallationen mit vielen kundenspezifischen Berichten und Listeneinstellungen empfehlen wir, vorgängig eine Testinstallation einzurichten und den Release-Wechsel durchzuspielen, bevor das Update auf dem aktiven System durchgeführt wird. Weitere Informationen finden Sie im Artikel [Testinstallationen](#) in unserer Online Knowledge Base unter www.vertec.com/kb/testinstallation.

Rückwärtskompatibilität

Bei den Features, bei welchen die Rückwärtskompatibilität eine Rolle spielt, wird das im entsprechenden Artikel im Abschnitt [Rückwärtskompatibilität](#) genau beschrieben:

Artikel	Seite
4 Business Intelligence (BI) Modul, Abschnitt 4.8	40
6.4 Erweiterte Rechnungen mit QR-Code an Spezifikation 2.1 angepasst	50
7.1 Mehrstufige Sortierung in Listen	54
8.3 Native Sprache in MLString Attributen	63
14.2 Unterstützung von Zusatzfeldern in Config Sets	81
14.3 Berücksichtigung von MLStrings in Config Sets	82
15.5 Verbindungen zu AbstractAdresseintrag aufgehoben	85

1.4 Das erste Aufstarten nach dem Konvert

Nach der Datenkonvertierung muss die Vertec Desktop App gestartet werden. Dieser erste Start nach dem Konvert ist wichtig und gehört ebenfalls zur Konvertierung dazu, da diverse Anpassungen an Daten und Strukturen etc. erst dann durchgeführt werden. Es ist wichtig, dass dieser erste Start reibungslos durchläuft. Keinesfalls sollte der erste Startvorgang nach einem Konvert abgebrochen werden.

Neustart des Cloud Servers

Nach der Konvertierung und dem erstmaligen Start der Vertec Desktop App muss der Cloud Server neu gestartet werden. Damit ist das Update von Vertec abgeschlossen.

2 Vertec 64 Bit

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Full featured | Betrieb: On Premises | Version: 6.4

Mit der vorliegenden Version wird Vertec nur noch als 64-bit Applikation ausgeliefert. Das bedeutet:

- Ab Vertec 6.4 wird auf dem Windows Server für Vertec ein 64-bit Betriebssystem benötigt.
- Desktop App und Cloud App laufen nur noch auf 64-bit Betriebssystemen
- Die Web App kann auch auf 32-bit Betriebssystemen oder einem nicht Windows System eingesetzt werden.

Einer 32-bit Applikation stehen maximal 2 GB Arbeitsspeicher zur Verfügung. Dabei kommt es nicht drauf an, wie viel Arbeitsspeicher physisch vorhanden ist. Das konnte bei grossen Datenmengen zu OutOfMemoryExceptions führen. Diese Beschränkung fällt mit 64 Bit weg.

64 Bit bedeutet aber auch, dass das adressierbare Memory mittels einer 64-bit Adresse referenziert wird. Jeder Pointer auf einen Memorybereich benötigt damit doppelt so viel Speicherplatz als vorher mit 32 Bit. Architekturbedingt gibt es in Vertec sehr viele Pointer und Objekte, so dass mit einem doppelt so hohen Memorybedarf gerechnet werden muss. Je nach Grösse der Installation und Datenmenge muss hier also aufgerüstet werden.

Der Vertec Setup installiert Vertec als 64-bit Applikation wie folgt:

- Der Standard Pfad für neue Installationen ist **C:\Programme\Vertec**.
- Updates von bestehenden Installationen werden wie bisher im gleichen Ordner wie die bestehende Installation durchgeführt. Das bedeutet, dass ein bestehendes Vertec als 64-bit Software im 32-bit Ordner "Programme (x86)" installiert ist. Wer das nicht möchte, kann eine Neu-Installation von Vertec durchführen und die Daten dann übernehmen.
- Falls noch kein Firebird Server installiert ist, wird die 64-bit Version von Firebird Server installiert.
- Bei bestehenden Firebird Installation wird wie folgt vorgegangen:
 - Die mitgelieferte Firebird Version ist 3.0.4. Ist die bestehende Firebird Installation älter, wird sie auf diese Version in 64 Bit upgraded. Das Installationsverzeichnis bleibt dabei gleich.
 - Besteht bereits eine 32-bit Firebird Installation der Version 3.0.4, werden nur die 64-bit Client-Komponenten von Firebird installiert.
- Die Python Installation im Unterordner **Python** wird auf 64 Bit updated.

Grundsätzlich läuft Vertec als 64-bit Applikation gleich wie vorher. Ein paar Dinge müssen jedoch beachtet werden und sind hier nachfolgend aufgelistet.

2.1 Konvertierung einer Vertec Datenbank mit MS SQL-Server

Die Konvertierung von Vertec Datenbanken auf MS SQL-Server ist nur mit 32 Bit möglich. Deshalb gibt es dafür eine spezielle 32-bit DB Convert App. Diese heisst **Vertec.DbConvert.exe** und befindet sich im Vertec Installationsverzeichnis.

Haben Sie die Datenbank gemäss Anleitung <https://www.vertec.com/kb/mssqlkonvert> kopiert, starten Sie die DB Convert App (und nicht wie bisher die Vertec Desktop App mit dem Parameter /DC). Alles andere funktioniert gleich wie bisher.

2.2 Schnittstellen mit Vertec 64 Bit

Topal Fibu Schnittstellen

Damit die Topal Schnittstelle läuft, muss Topal auf 64 Bit umgestellt werden. Dafür gibt es in Topal einen Menüpunkt **Tools > Wechseln nach x64**:

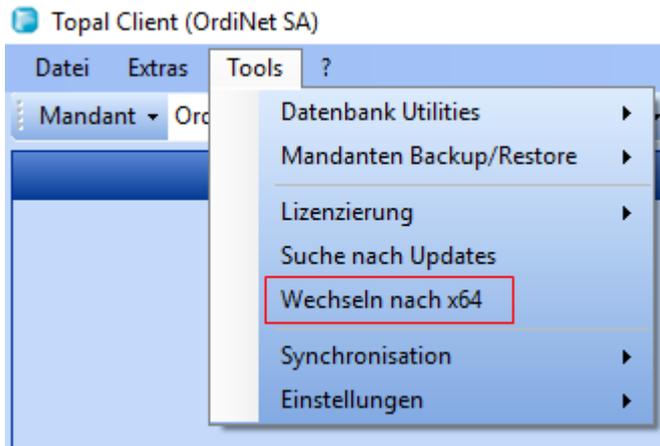


Abbildung 1: Topal auf 64 Bit umstellen

Sage 50 und Sage 200 Schnittstellen

Sage 50 und Sage 200 laufen weiter im 32-bit Modus. Sie müssen nichts machen.

Für die Verwendung des In-Process COM Servers registriert der Vertec Setup einen Surrogate Prozess (Anpassung eines Registry Keys). Dadurch laufen auch die 32-bit Applikationen mit den Vertec 64-bit Schnittstellen.

SelectLine Schnittstellen

SelectLine läuft normal weiter, Sie müssen nichts machen.

TAPI Schnittstelle

Die Vertec TAPI Schnittstelle wurde für Vertec 6.4 neu implementiert und sollte nach dem Update normal weiterlaufen.

3 Version 6.4

3.1 Highlights der Version 6.4

Die Version 6.4 wartet wieder mit vielen Highlights auf. Die Spannendsten hier in einem kurzen Überblick:

Artikel	Seite
4 Business Intelligence (BI) Modul	9
5.1 PDF Dokumente als Spesenbelege	46
6.1 Rechnungen nach ZUGFeRD 2.0 Standard (XRechnung)	49
7.1 Mehrstufige Sortierung in Listen	54
8.1 Geplante Aufgaben	58
8.2 2 Faktor Authentisierung	60
11.1 OCL Call Operatoren für Custom Businesslogik	71
12.1 Optimierung von Objekt-Referenzen in OCL	78

3.2 Eingestellte Features in Vertec 6.4

Benchmarking

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

Das Modul Benchmarking wird durch das neue Business Intelligence Modul (BI Modul) abgelöst und ist in der bisherigen Version nicht mehr verfügbar.

Eine ausführliche Beschreibung des neuen BI Moduls finden Sie im Kapitel 4.

Systemeinstellung für Existenzprüfung von Dokumenten entfernt

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Desktop | Version: 6.3.0.15

Die Systemeinstellung Allgemein > **Existenz von Dokumentfiles überprüfen** ist ab dieser Version nicht mehr vorhanden. Sie wird auch bei bestehenden Installationen entfernt.

4 Business Intelligence (BI) Modul

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

Alle Daten, die in Vertec erfasst werden, können mit dem Business Intelligence (BI) Modul ausgewertet werden. Das BI Modul ermöglicht eine betriebswirtschaftliche Auswertung aller Leistungsdaten.

Für die Verwendung ist eine Lizenzierung des **Business Intelligence Moduls** Voraussetzung. Das BI Modul löst das bisherige Benchmarking ab. Bestehende Benchmarking Lizenzen werden automatisch in BI Modul Lizenzen überführt.

Um die Auswertungen auch über längere Zeiträume und mit grossen Datenmengen performant anzeigen zu können, werden die Daten vorberechnet (siehe Kapitel 4.4). Bei der Abfrage der BI Daten stehen dann alle vordefinierten und vorberechneten Kennzahlen zur Verfügung.

Die dargestellten Auswertungen basieren auf sogenannten BI Anzeigen, welche definieren, welche Werte wo dargestellt werden können. Die genauen Definitionen sowie eine Liste der mitgelieferten BI Anzeigen finden Sie im Kapitel 4.2.

4.1 BI Ansicht öffnen

Hat man das Business Intelligence Modul lizenziert und befindet sich in Vertec auf einem Objekt oder auf einer Liste von Objekten, für welche passende BI Anzeigen definiert sind, erscheint oben in der Navigationsansicht der BI Button:

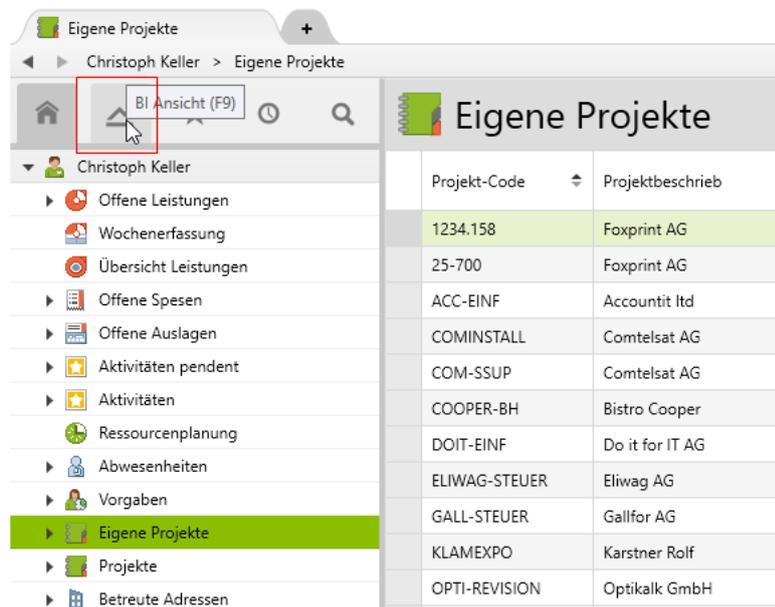


Abbildung 2: Der Button zum Öffnen des BI

Durch Klick auf diesen Button oder über die **F9** Taste (beachten Sie bitte die Veränderung in den Shortcuts der Navigationsansichten, Kapitel 8.4) öffnet sich die BI Ansicht des markierten Objekts bzw. der markierten Liste (hier im Beispiel die **Eigenen Projekte**):

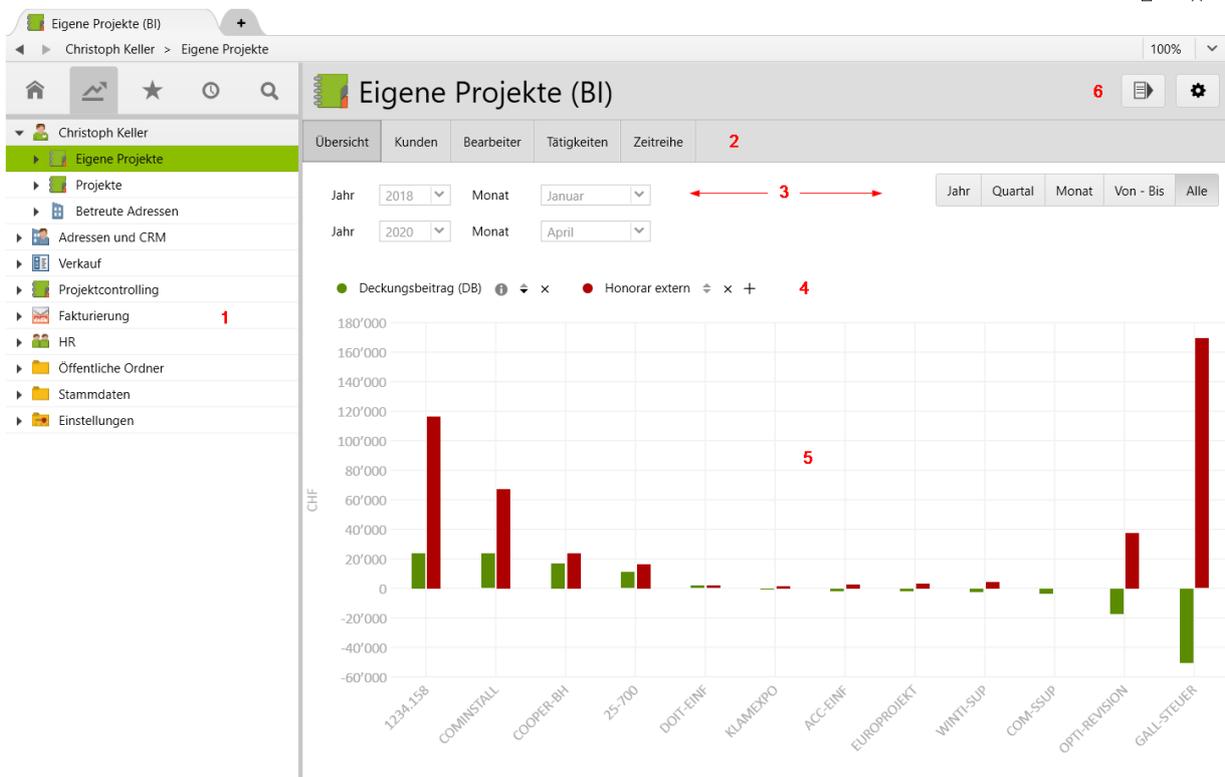
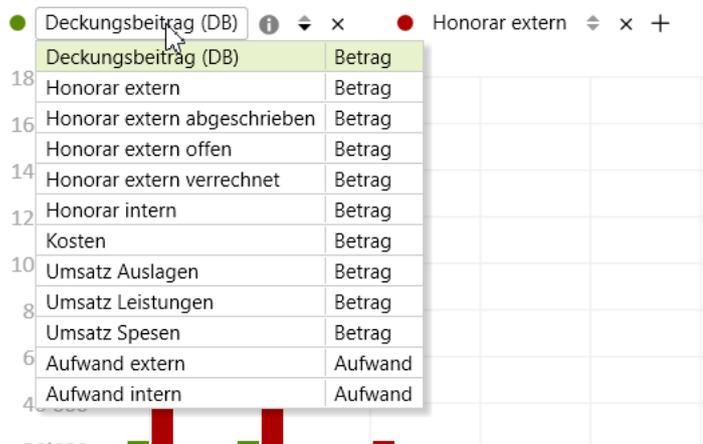


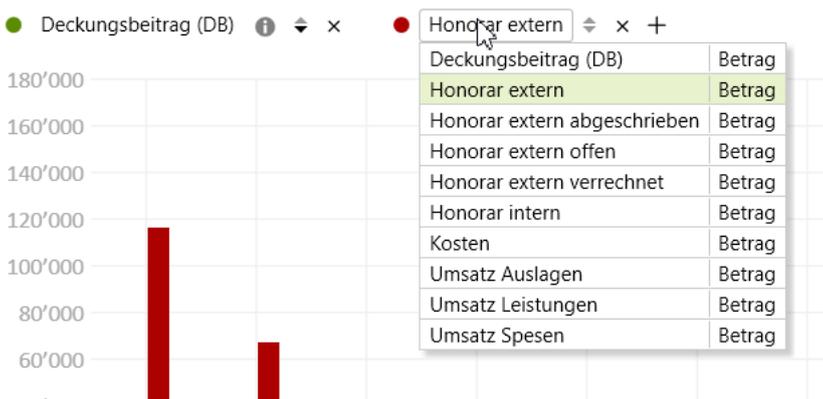
Abbildung 3: Die BI Ansicht

Die BI Ansicht ist wie folgt aufgebaut:

- 1 In der BI Ansicht werden im Baum nur diejenigen Elemente angezeigt, für welche BI Auswertungen zur Verfügung stehen, siehe 4.2.
- 2 Oben als Tabs sehen Sie die Anzeigen, die für das selektierte Element zur Verfügung stehen.
- 3 Als Datumsbereiche stehen die Varianten **Jahr**, **Quartal**, **Monat**, **Von – Bis** und **Alle** zur Verfügung. Standard ist **Alle**. Je nach gewähltem Bereich können auf der linken Seite die entsprechenden Einstellungen gemacht werden.
Welcher Zeitraum grundsätzlich zur Verfügung steht, hängt davon ab, welche Daten berechnet wurden. Siehe dazu das Kapitel 4.4. Standardmässig ist es das Intervall von Januar des vorletzten Jahres bis und mit aktuellem Monat.
- 4 Angezeigte Kennzahlen. Über den + Button können weitere Kennzahlen in der Grafik eingeblendet werden. Mit dem Sortierpfeil kann die Ansicht nach einer bestimmten Kennzahl sortiert werden.
Welche Kennzahlen zur Verfügung stehen, ist im Abschnitt 4.3 erklärt.
Durch Klick auf eine Kennzahl öffnet sich die Liste der verfügbaren Kennzahlen:



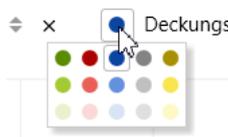
Auf der ersten Kennzahl enthält die Liste alle verfügbaren Kennzahlen, wie hier dargestellt. Man sieht, dass es Kennzahlen mit unterschiedlichen Einheiten hat (Betrag und Aufwand). Da diese nicht in der gleichen Grafik dargestellt werden können, zeigten die Listen auf den Kennzahlen weiter hinten dann nur noch die verfügbaren Kennzahlen der gleichen Einheit:



Der + Button fügt jeweils die nächste Kennzahl in der Liste an, welche dann nach Bedarf wie hier dargestellt geändert werden kann.

Der x Button entfernt die Kennzahl wieder.

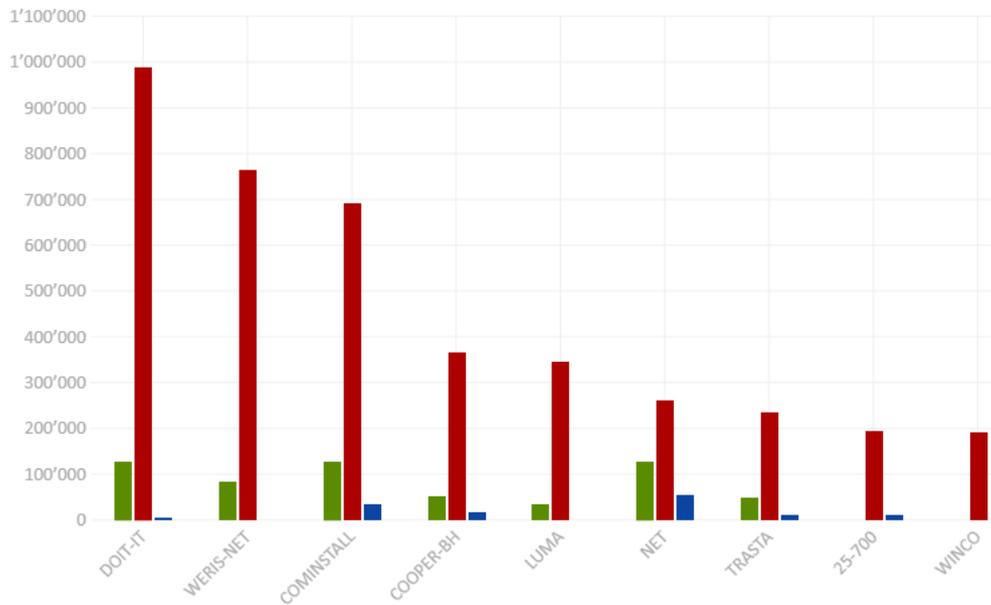
Durch einen Klick auf den Farbpunkt erscheint eine Farbpalette, wo Sie die Farbe dieser Kennzahl für die Dauer der Anzeige wechseln können:



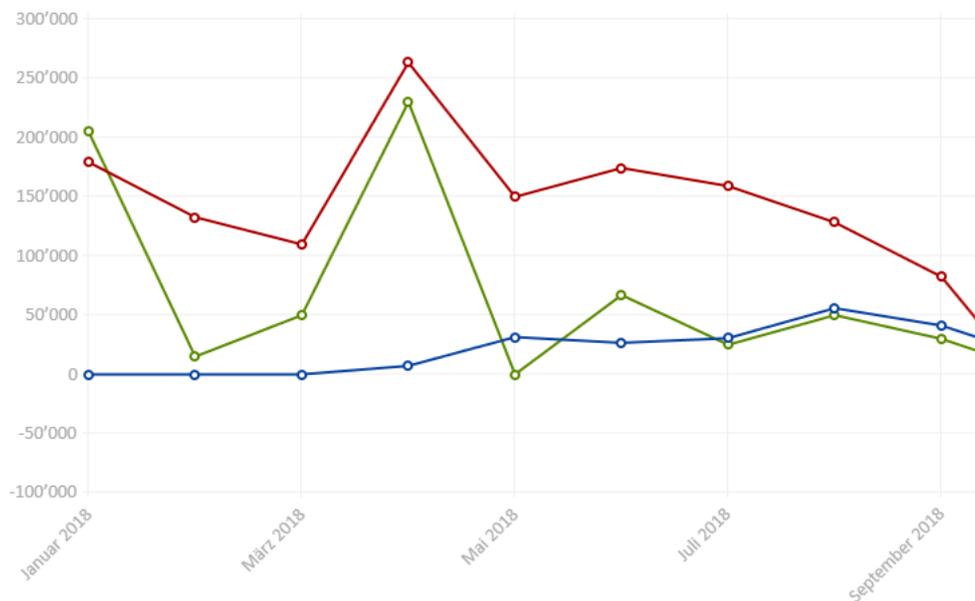
Die für die Anzeige in der Grafik ausgewählten Kennzahlen bleiben bei einem Wechsel zwischen verschiedenen Anzeigen bestehen, sofern sie auf der gewählten Anzeige zur Verfügung stehen.

5 Hier sehen Sie die Werte in der Grafik dargestellt. Die Gruppierung entspricht der **Gruppierung** in den BI Anzeige-Definitionen (4.2).

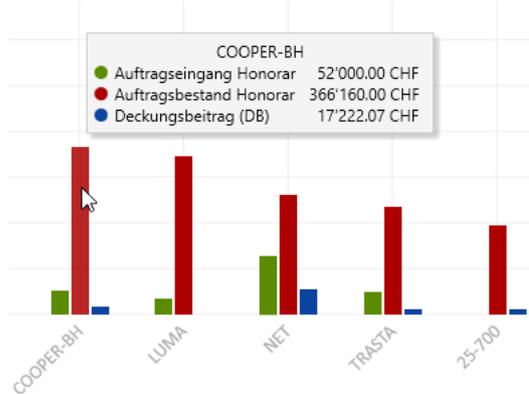
Diese erscheinen entweder als Balkendiagramm:



oder, im Falle von Zeitreihen (Gruppierung nach Monat, siehe 4.2), als Liniengrafik:

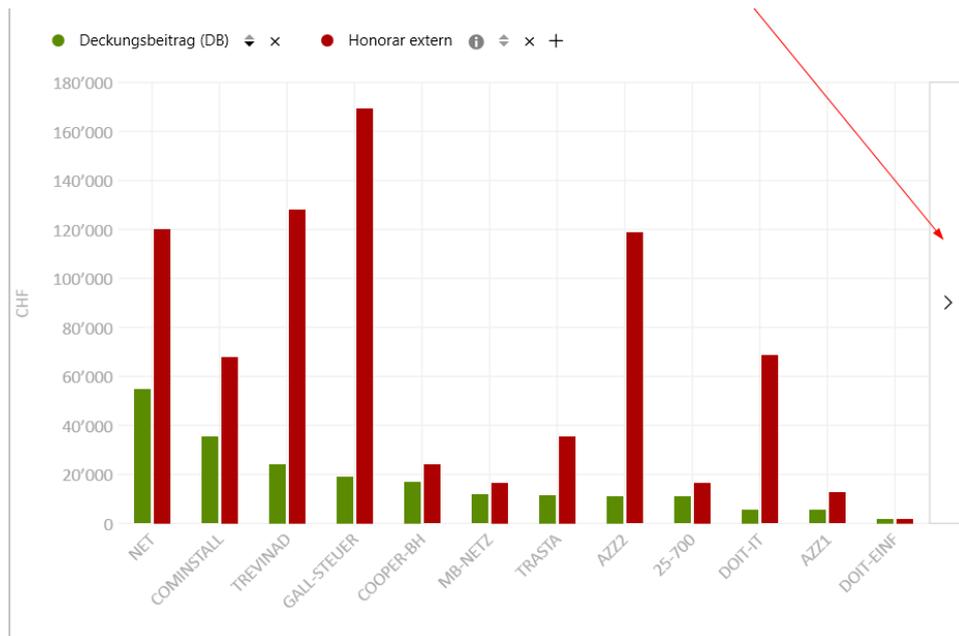


Wenn Sie den Mauszeiger auf einen Balken bzw. einen Punkt setzen, werden die effektiven Werte als Zahlen angezeigt:



Durch Doppelklick auf einen Balken wird das entsprechende Einzelobjekt in einem separaten Tab geöffnet.

Sind mehr Balken vorhanden, als im Fenster Platz haben, erscheint am rechten Rand ein Button. Durch Klick auf diesen Button werden im Fenster alle vorhandenen Balken angezeigt, durch erneuten Klick wieder nur die ersten.



6 Durch Klick auf den Excel Export Button werden die aktuell dargestellten BI-Daten ins Excel übertragen:

200416 BI.xlsx [Schreibgeschützt] - Microsoft Excel

Start Einfügen Seitenlayout Formeln Daten Überprüfen Ansicht Entwicklertools

A1 Projekte

	A	B	C	D
1	Projekte			
2	Übersicht			
3	Januar 2018 - April 2020			
4				
5	Projekt	Deckungsbeitrag (DB) [CHF]	Honorar extern [CHF]	
6	NET	33'465.14	120'430.82	
7	1234.158	23'998.71	116'493.55	
8	COMINSTALL	23'664.40	67'566.55	
9	TREVINAD	23'355.33	128'192.00	
10	COOPER-BH	17'222.07	24'155.40	
11	MB-NETZ	11'701.78	16'698.45	
12	25-700	11'080.22	16'666.05	
13	TRASTA	4'798.37	35'762.45	
14	AZZ1	3'328.33	12'799.95	
15	DOIT-EINF	2'000.00	2'000.00	
16	TEST-SUMLEIST	1'000.00	1'000.00	
17	HIMO-NETZ	-100.00	200.00	
18	1234.123	-400.00	1'105.00	
19	LUMA	-533.33	960.00	
20	ESO-STEUER	-700.00	1'400.00	
21	KLAMEXPO	-841.67	1'683.35	

URL Unterstützung für BI Ansichten

Die Vertec Breadcrumb URL ist auch in den BI Ansichten verfügbar:

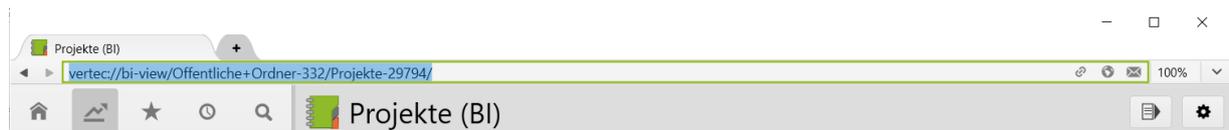


Abbildung 4: Die Breadcrumb URL in der BI Ansicht

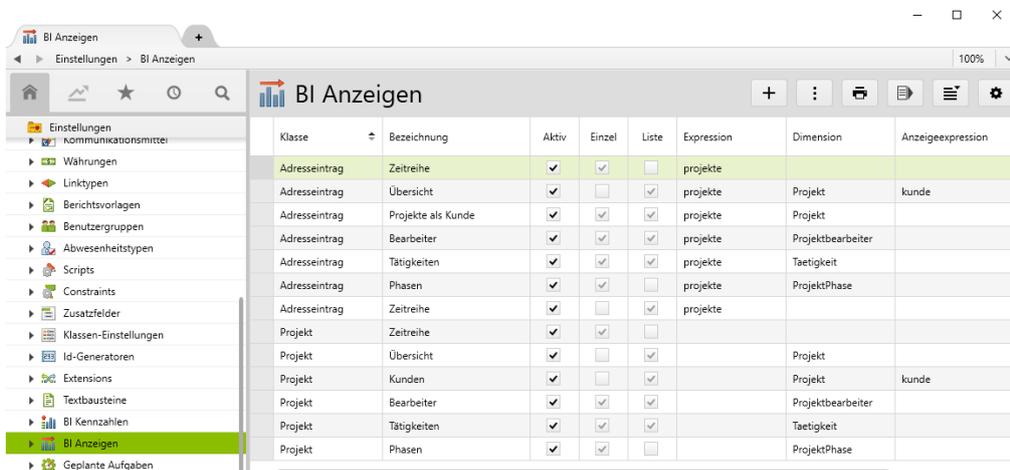
Sie kann mit den gewohnten Buttons kopiert und versendet werden. Der Pfad einer BI URL beginnt mit **bi-view**. Sie repräsentiert das im Baum selektierte Objekt in der Grundansicht, also ohne Parameter wie Datumsbereich und selektierte Auswertung.

Wird ein BI Pfad ausgeführt, ohne dass eine Berechtigung für die Ansicht besteht, erscheint eine entsprechende Fehlermeldung.

Weitere Informationen Weitere Informationen zum Thema Breadcrumb URLs finden Sie in der Online Knowledge Base unter www.vertec.com/kb/listennavigation#url.

4.2 BI Anzeigen

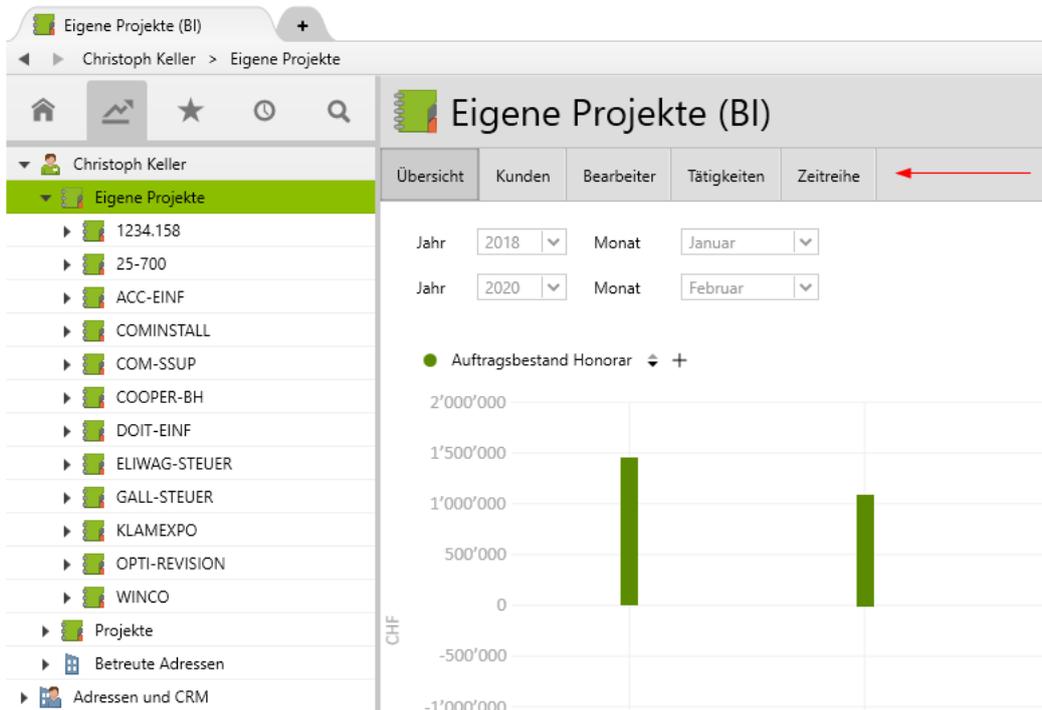
Um die Daten in der BI Ansicht darstellen zu können, müssen sogenannte BI Anzeigen definiert werden. Diese finden Sie im Ordner Einstellungen > BI Anzeigen:



Klasse	Bezeichnung	Aktiv	Einzel	Liste	Expression	Dimension	Anzeigeexpression
Adresseintrag	Zeitreihe	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	projekte		
Adresseintrag	Übersicht	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	projekte	Projekt	kunde
Adresseintrag	Projekte als Kunde	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	projekte	Projekt	
Adresseintrag	Bearbeiter	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	projekte	Projektbearbeiter	
Adresseintrag	Tätigkeiten	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	projekte	Taetigkeit	
Adresseintrag	Phasen	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	projekte	ProjektPhase	
Adresseintrag	Zeitreihe	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	projekte		
Projekt	Zeitreihe	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>			
Projekt	Übersicht	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		Projekt	
Projekt	Kunden	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		Projekt	kunde
Projekt	Bearbeiter	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Projektbearbeiter	
Projekt	Tätigkeiten	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Taetigkeit	
Projekt	Phasen	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		ProjektPhase	

Abbildung 5: Anzeigen-Definitionen für das BI Modul

Jede dieser BI Anzeigen repräsentiert im BI Modul eine Auswertung.



Eine BI Anzeige hat folgende Eigenschaften:

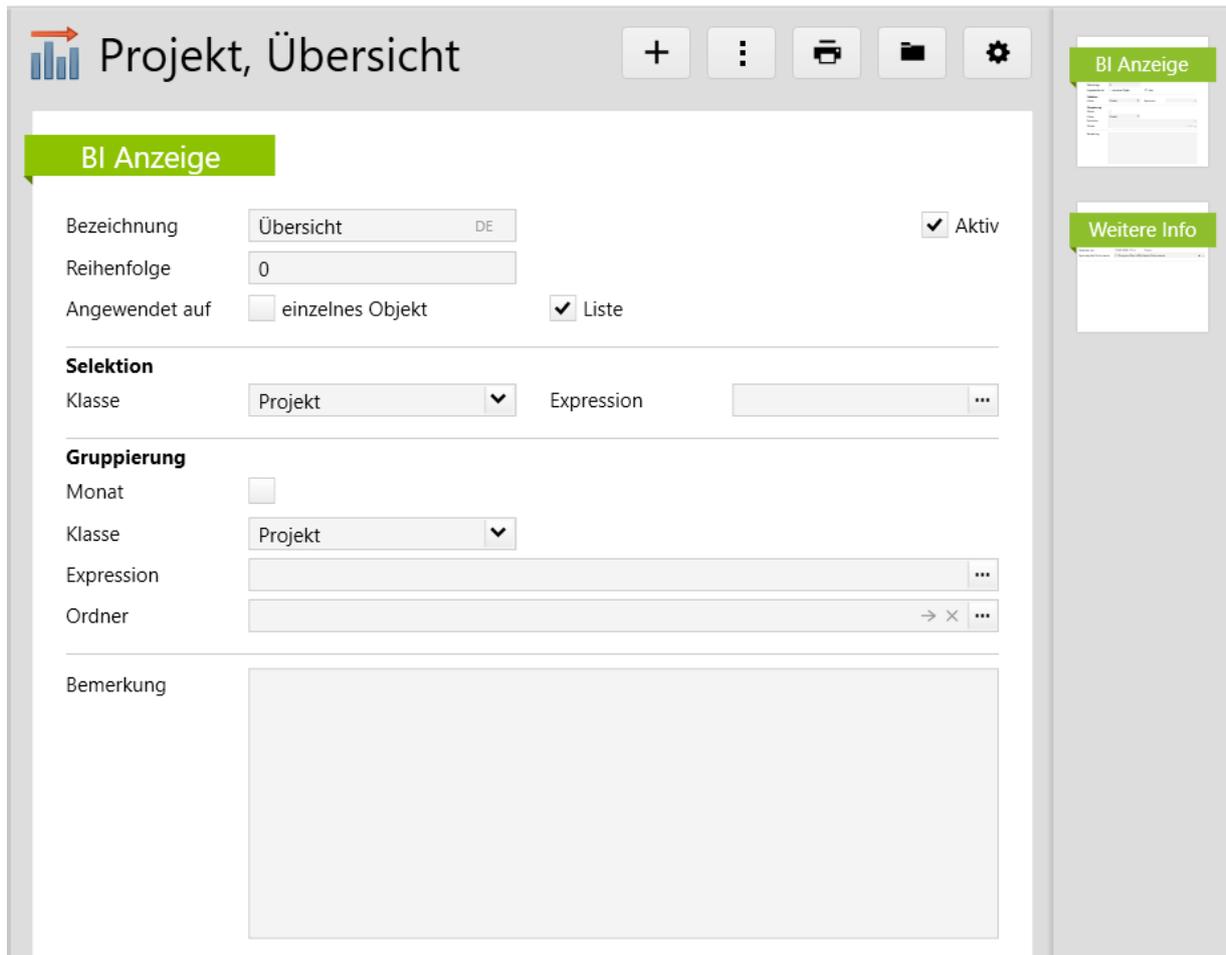


Abbildung 6: Einzelansicht einer BI Anzeige

Bezeichnung Die Bezeichnung der Auswertung wird als Beschriftung der Tabs in der BI Ansicht angezeigt.



Es handelt sich hier um ein MLString Feld, siehe Kapitel 8.3.

Aktiv Definiert, ob eine Auswertung in der BI Ansicht angezeigt wird.

Reihenfolge Definiert, in welcher Reihenfolge die Tabs in der BI Ansicht dargestellt werden.

Angewendet auf einzelnes Objekt / Liste Definiert, ob die Auswertung auf einzelnen Objekten und / oder auf Listen von Objekten zur Verfügung stehen.

Anzeigen, welche unter Selektion und Gruppierung dieselbe Klasse definiert haben, machen auf einzelnen Objekten wenig Sinn, da sie dann nur einen einzelnen Balken für sich selbst anzeigen. Sollen gewisse Auswertungen auf Listen in einer anderen Reihenfolge als auf einzelnen Objekten dargestellt werden, müssen sie jeweils einmal für einzelne Objekte und einmal für Listen definiert werden:

Klasse	Bezeichnung	Aktiv	Einzel	Liste	Expression	Gruppierung	Gruppierungsexpression	Monate	Reihenfoi
Adresseintrag	Zeitreihe	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	projekte			<input checked="" type="checkbox"/>	0
Adresseintrag	Übersicht	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	projekte	Projekt	kunde	<input type="checkbox"/>	0
Adresseintrag	Projekte als Kunde	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	projekte	Projekt		<input type="checkbox"/>	10
Adresseintrag	Bearbeiter	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	projekte	Projektbearbeiter		<input type="checkbox"/>	20
Adresseintrag	Tätigkeiten	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	projekte	Taetigkeit		<input type="checkbox"/>	30
Adresseintrag	Phasen	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	projekte	ProjektPhase		<input type="checkbox"/>	40
Adresseintrag	Zeitreihe	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	projekte			<input checked="" type="checkbox"/>	50

Spielt die Reihenfolge keine Rolle, kann die gleiche Anzeige sowohl für einzelne Objekte als auch für Listen verwendet werden.

Selektion

Selektion

Klasse Expression

Die Selektion definiert, **WO** die Anzeige dargestellt wird.

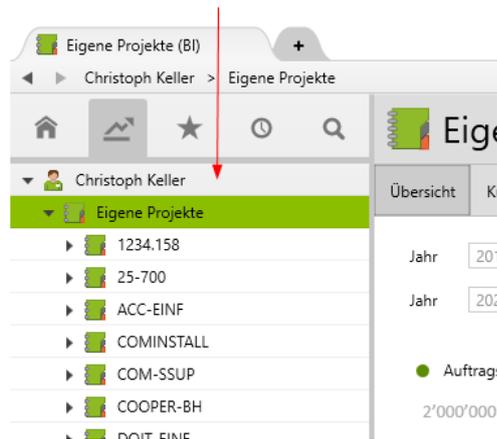


Abbildung 7: Die Auswertungen werden für die definierten Selektionen zur Verfügung gestellt

Klasse Definiert, für welche Klasse(n) diese Anzeige definiert ist, also wo Auswertungen zur Verfügung stehen.

Expression Eine Expression wird dann benötigt, wenn die selektierte Klasse gar nicht über Kennzahlen verfügt, welche ausgewertet werden können (siehe dazu Kapitel 4.3). Dies ist zum Beispiel bei Adressen so. In diesem Fall muss hier eine OCL Expression angegeben werden, deren Ergebnis ein Objekt (oder eine Liste von Objekten) ist, welches Kennzahlen hat.

Dies bezieht sich nicht darauf, **WAS** in der Auswertung angezeigt wird. Dafür wird die **Gruppierung** definiert. Es geht also nur darum, woher die Zahlen für die Anzeige kommen, nicht wie sie dargestellt werden.

Gruppierung

Gruppierung

Monat

Klasse

Expression

Ordner

Die Gruppierung definiert, **WONACH** die Zahlen dieser Anzeige **GRUPPIERT** werden

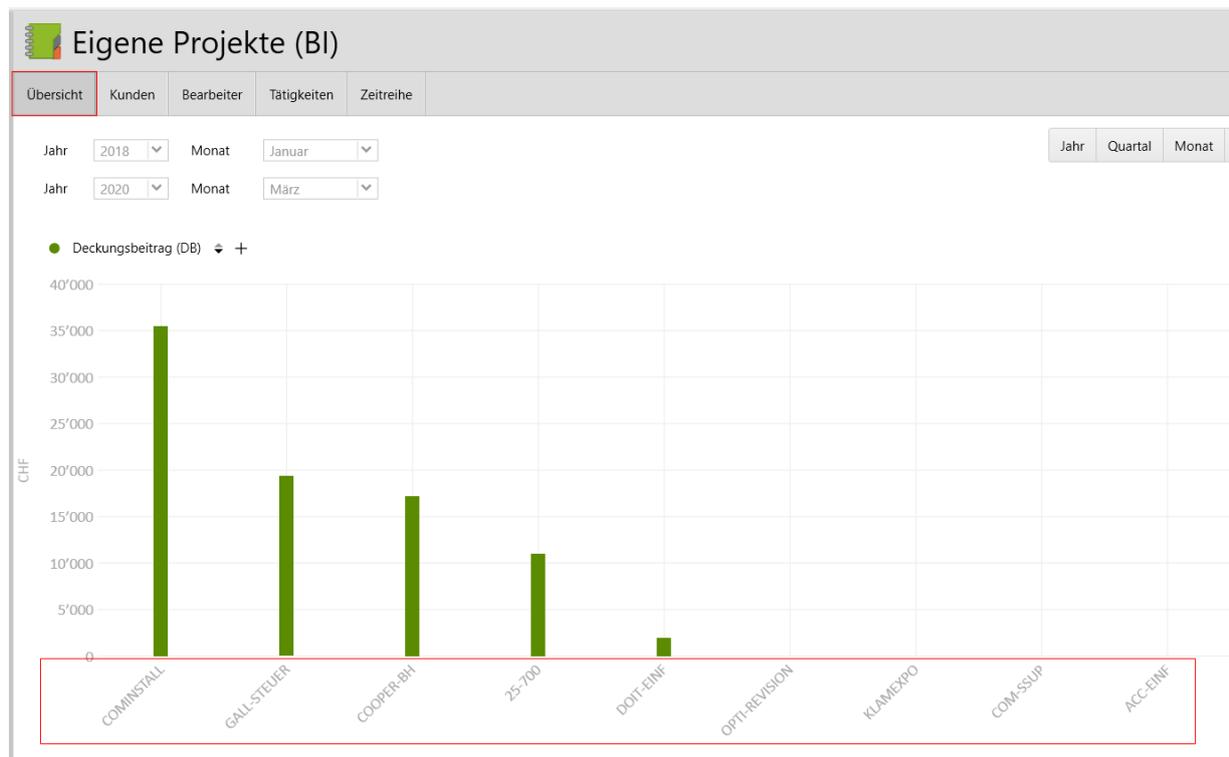


Abbildung 8: In unserem Beispiel stehen die Zahlen gruppiert nach Projekt zur Verfügung

Monat Wird eine Auswertung nach Monat gruppiert, wird sie als Zeitreihe dargestellt (siehe 4.1). In diesem Fall werden die folgenden Gruppierungsmöglichkeiten ausgeblendet.

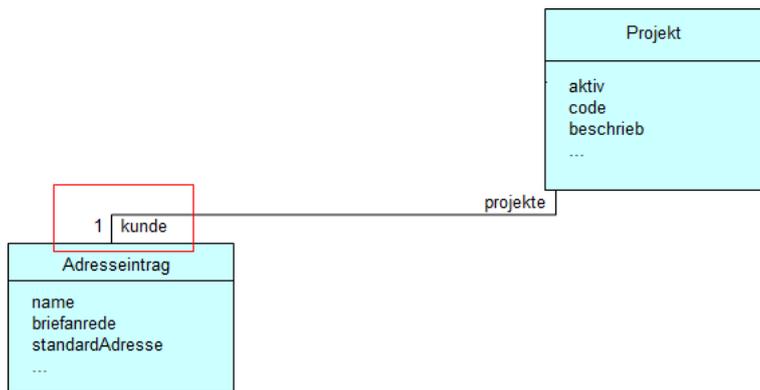
Klasse Hier wird angegeben, nach welchen Klassen die Zahlen gruppiert dargestellt werden. Für die Übersicht hier im Beispiel werden die Zahlen nach Projekt gruppiert, also nach der gleichen Klasse, die auch in der Selektion angegeben ist. Wird dann keine Expression angegeben, werden die Kennzahlen einfach nach den zur Verfügung stehenden Objekten gruppiert, im Beispiel oben nach den eigenen Projekten. Die hier angegebene Klasse muss auf den darzustellenden Kennzahlen als Dimension verfügbar sein.

Expression Mit dieser Expression kann die Gruppierung zusätzlich verfeinert werden. Sie wird auf die Ergebnisobjekte gemäss „Klasse“ angewendet. Gruppiert und dargestellt werden dann die Ergebnisse diese Expression. Hier als Beispiel die Auswertung "Kunden". Die Gruppierungsklasse ist ebenfalls Projekt, anders aber als bei der Übersicht wird hier nun als OCL-Expression **kunde** angegeben.

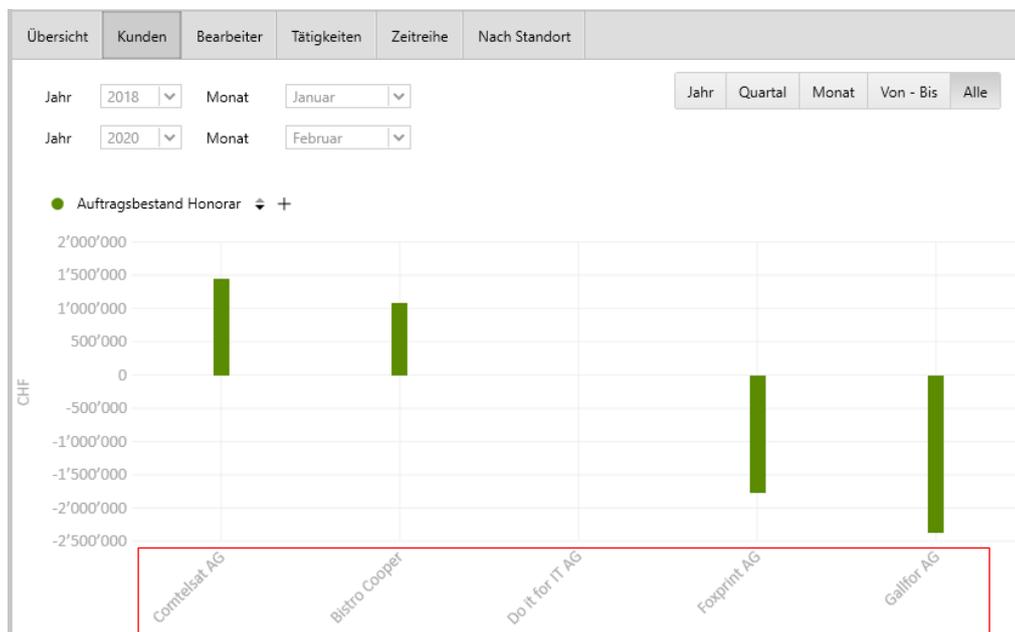
Selektion
 Klasse: Expression:

Gruppierung
 Monat:
 Klasse:
 Expression: ←
 Ordner:

Das bezieht sich auf das Modell (Ausschnitt):



das heisst, dass in diesem Fall die Projekt-Zahlen nach Kundenadresse gruppiert dargestellt werden.

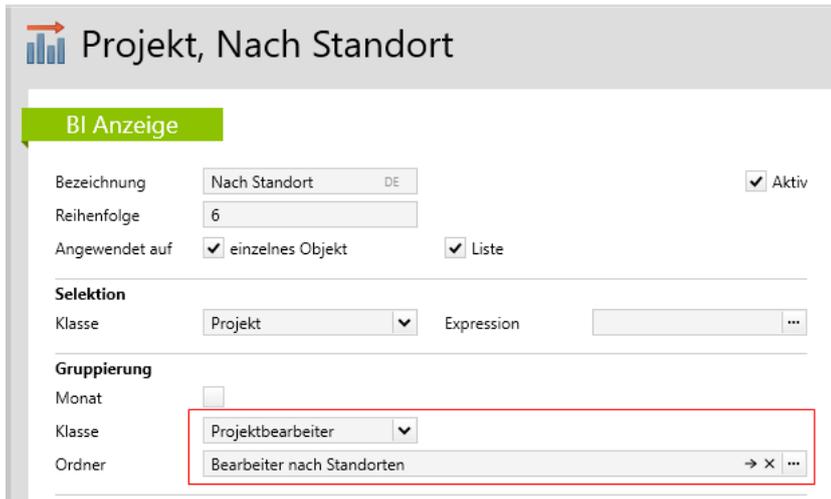


Eine Vielzahl von Modell-Grafiken finden Sie in der Online Knowledge Base unter www.vertec.com/support/kb/technik-und-datenmodell/modell/uml.

Ordner Statt einer Expression kann auch ein Ordner angegeben. In diesem Fall werden die Zahlen nach Unterordnern des angegebenen Ordners gruppiert. Im bisherigen Benchmarking Modul hiess diese Möglichkeit **Projektionen** (siehe Abschnitt 4.8).

Der Ordner bzw. seine Unterordner müssen Objekte der Klassen enthalten, die als Gruppierung angegeben wird.

Hier im folgenden Beispiel möchten wir die Projektzahlen nach Bearbeiterstandort gruppiert darstellen. Dafür gruppieren wir die Anzeige nach Projektbearbeitern und wählen als Ordner den Stichwortordner "Bearbeiter nach Standorten" aus:



Projekt, Nach Standort

BI Anzeige

Bezeichnung: Nach Standort DE Aktiv

Reihenfolge: 6

Angewendet auf: einzelnes Objekt Liste

Selektion

Klasse: Projekt Expression:

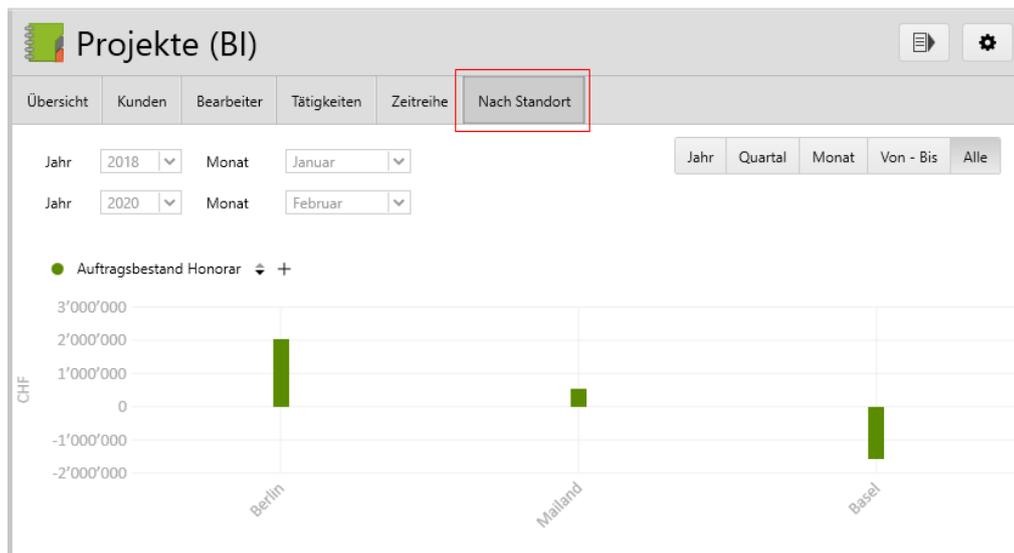
Gruppierung

Monat:

Klasse: Projektbearbeiter

Ordner: Bearbeiter nach Standorten → X ...

In der Auswertung erscheint dann ein Tab "Nach Standort", welche die Zahlen entsprechend gruppiert:



Ordnervergleiche

Wird in der BI Ansicht ein Ordner selektiert, welcher Subordner mit BI-selektierbaren Objekten enthält, dann erscheint darauf automatisch eine Vergleichsansicht, welche die Daten gruppiert nach den Ordnern anzeigt, in denen sie sich befinden. Es muss dafür keine BI Anzeige erstellt werden. Berücksichtigt wird dabei die erste Hierarchieebene.

Folgende Bedingungen müssen erfüllt sein, damit ein Ordnervergleich stattfindet:

- Die Eigenschaft **Subordner erlauben** muss gesetzt sein.
- Der Ordner muss mindestens einen Subordner haben.
- Alle Subordner müssen die gleiche Klasse haben.

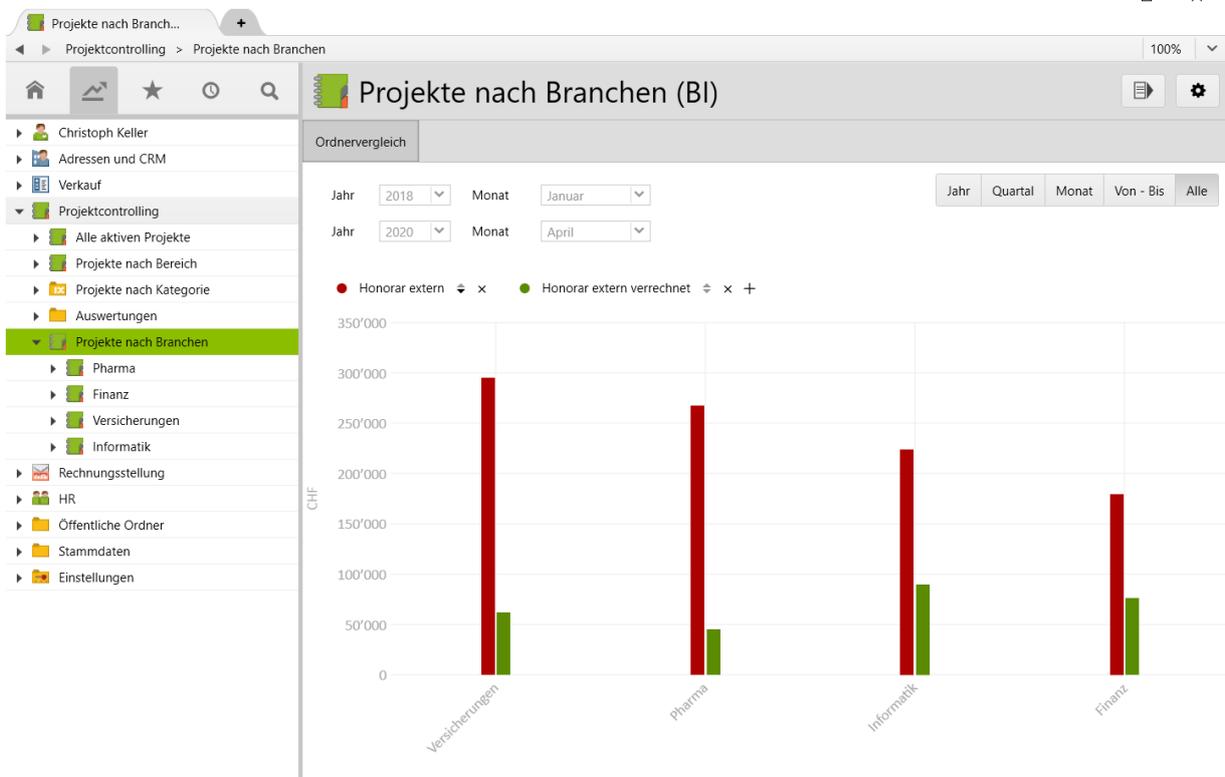


Abbildung 9: Projekte nach Branchen werden im Ordnervergleich angezeigt

WAS angezeigt wird, also welche Zahlen, definiert sich über die BI Kennzahlen (siehe Abschnitt 4.3). Dies muss bei der Definition der Anzeige nicht angegeben werden.

Eine Vielzahl von vordefinierten BI Anzeigen wird von Vertec standardmässig mitgeliefert.

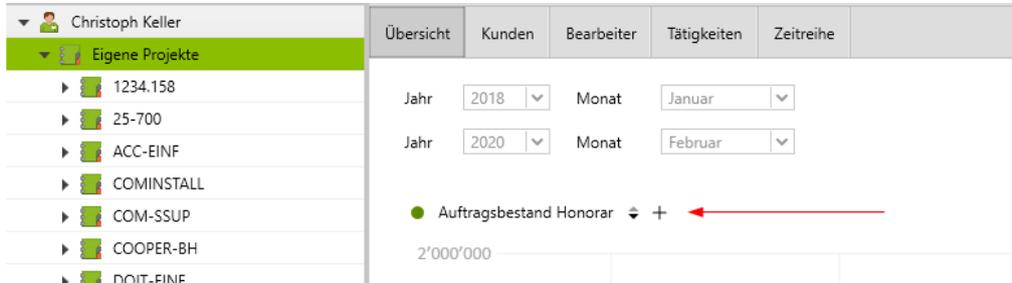
Klasse	Bezeichnung	Expression	Gruppierung / Expression	Bedeutung
Adresseintrag	Zeitreihe	projekte	Monat	Auf einzelnen Objekten: Für diesen Kunden summierte Projektwerte, pro Monat.
Adresseintrag	Übersicht	projekte	Projekt / kunde	Pro Kunde summierte Projektwerte.
Adresseintrag	Projekte als Kunde	projekte	Projekt	Werte pro Projekt des einzelnen Kunden bzw. der Kunden in der Liste. Entspricht der Projekt-Übersicht.
Adresseintrag	Bearbeiter	projekte	Projektbearbeiter	Werte pro Projektbearbeiter auf Projekten des einzelnen Kunden bzw. der Kunden in der Liste.
Adresseintrag	Tätigkeiten	projekte	Taetigkeit	Werte pro Tätigkeit auf Projekten des einzelnen Kunden bzw. der Kunden in

Klasse	Bezeichnung	Expression	Gruppierung / Expression	Bedeutung
				der Liste.
Adresseintrag	Phasen	projekte	ProjektPhase	Werte pro Projektphase von Projekten des einzelnen Kunden bzw. der Kunden in der Liste.
Adresseintrag	Zeitreihe	projekte	Monat	Auf Listen: Gesamthafte Werte über alle Kunden der Liste, pro Monat.
Projekt	Zeitreihe		Monat	Auf einzelnen Objekten: Für dieses Projekt summierte Werte, pro Monat.
Projekt	Übersicht		Projekt	Werte pro Projekt
Projekt	Kunden		Projekt / kunde	Werte gruppiert nach Kunden
Projekt	Bearbeiter		Projektbearbeiter	Werte gruppiert nach Bearbeitern
Projekt	Tätigkeiten		Taetigkeit	Werte gruppiert nach Tätigkeiten
Projekt	Phasen		ProjektPhase	Werte gruppiert nach Phasen
Projekt	Zeitreihe		Monat	Auf Listen: Gesamthafte Werte über alle Projekte der Liste, pro Monat.
Projektbearbeiter	Zeitreihe		Monat	Auf einzelnen Objekten: Für diesen Bearbeiter summierte Werte, pro Monat.
Projektbearbeiter	Übersicht		Projektbearbeiter	Werte pro Projektbearbeiter
Projektbearbeiter	Projekte		Projekt	Bearbeiter-Werte gruppiert nach Projekten
Projektbearbeiter	Kunden		Projekt / kunde	Bearbeiter-Werte gruppiert nach Kunden
Projektbearbeiter	für Projektleiter		Projekt / projektleiter	Bearbeiter-Werte gruppiert nach Projektleiter
Projektbearbeiter	für Hauptbearbeiter		Projekt / hB	Bearbeiter-Werte gruppiert nach Hauptbearbeiter
Projektbearbeiter	Tätigkeiten		Taetigkeit	Bearbeiter-Werte gruppiert nach Tätigkeiten
Projektbearbeiter	Eigene Projekte	eigprojekte	Projekt	Zahlen der eigenen Projekte des Bearbeiters, nach Projekten gruppiert

Klasse	Bezeichnung	Expression	Gruppierung / Expression	Bedeutung
Projektbearbeiter	Bearbeiter eigene Projekte	eigprojekte	Projektbearbeiter	Zahlen der eigenen Projekte des Bearbeiters, nach Bearbeiter gruppiert
Projektbearbeiter	Zeitreihe		Monat	Auf Listen: Gesamthafte Werte über alle Bearbeiter der Liste pro Monat
ProjektPhase	Zeitreihe		Monat	Auf einzelnen Objekten: Für diese Projektphase summierte Werte, pro Monat
ProjektPhase	Übersicht		ProjektPhase	Werte pro Projektphase
ProjektPhase	Tätigkeit		Taetigkeit	Projektphasen-Werte, gruppiert nach Tätigkeit
ProjektPhase	Projekt		Projekt	Projektphasen-Werte, gruppiert nach Projekt
ProjektPhase	Bearbeiter		Projektbearbeiter	Projektphasen-Werte, gruppiert nach Bearbeiter
ProjektPhase	Zeitreihe		Monat	Auf Listen: Gesamthafte Werte über alle Projektphasen der Liste pro Monat
Taetigkeit	Zeitreihe		Monat	Auf einzelnen Objekten: Für diese Tätigkeit summierte Werte, pro Monat
Taetigkeit	Übersicht		Taetigkeit	Werte pro Tätigkeit
Taetigkeit	Bearbeiter		Projektbearbeiter	Tätigkeits-Werte, gruppiert nach Bearbeiter
Taetigkeit	Projekt		Projekt	Tätigkeits-Werte, gruppiert nach Projekt
Taetigkeit	Kunden		Projekt / kunde	Tätigkeits-Werte, gruppiert nach Kunden
Taetigkeit	Zeitreihe		Monat	Auf Listen: Gesamthafte Werte über alle Tätigkeiten der Liste pro Monat

4.3 BI Kennzahlen

Die verschiedenen Arten von Werten, welche verfügbar sind, bezeichnen wir als BI Kennzahlen:



Die Kennzahlen werden im Ordner Einstellungen > BI Kennzahlen konfiguriert. Jede Kennzahl hat folgende Eigenschaften:

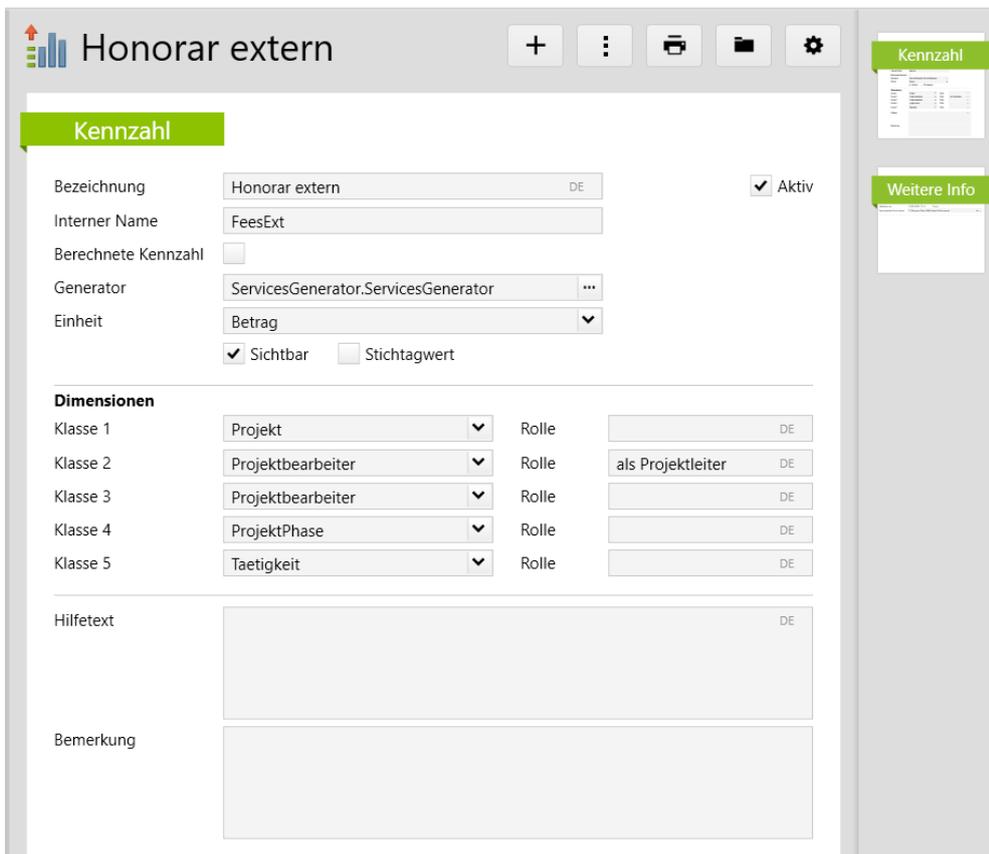
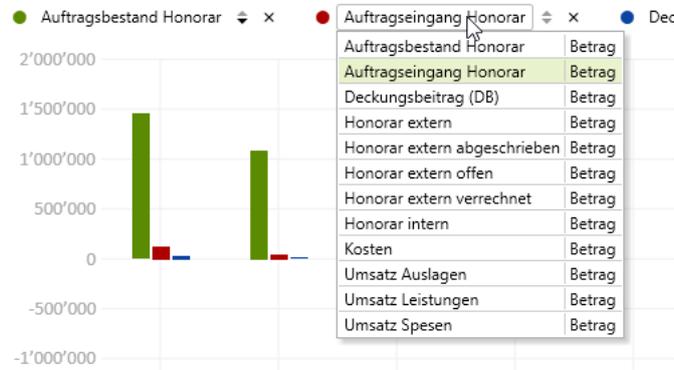


Abbildung 10: Definition einer BI Kennzahl

Bezeichnung Die Bezeichnung wird in der BI Ansicht angezeigt und sollte deshalb möglichst eindeutig und aussagekräftig sein, damit dann klar ist, was angezeigt wird:



Es handelt sich hier um ein MLString Feld, siehe Kapitel 8.3.

Aktiv Nur aktive Kennzahlen werden vorberechnet gespeichert. Wenn Sie eine standardmässig mitgelieferte Kennzahl also gar nie benötigen, können Sie sie deaktivieren.

Möchten Sie eine Kennzahl hingegen berechnen, aber nicht anzeigen (zum Beispiel weil Sie sie nur für eine Berechnung - siehe **Berechnete Kennzahl** weiter unten - nicht aber zur Anzeige benötigen), deaktivieren Sie stattdessen die Checkbox **Sichtbar** weiter unten.

Interner Name Der interne Name muss eindeutig sein, da die Kennzahl vom Generator-Code über diesen Namen identifiziert wird. Mehr dazu im Abschnitt 4.4.

Berechnete Kennzahl Wird eine Kennzahl als berechnete Kennzahl markiert, erfolgt die Berechnung nicht über den Generator-Code, sondern über andere Kennzahlen. In diesem Fall wird nicht das Feld Generator angezeigt, sondern ein Feld für die Berechnung:

Berechnete Kennzahl

Berechnung ←

Einheit

Sichtbar Stichtagwert

Dimensionen

Klasse	Wert	Rolle	DE
Klasse 1	Projekt	Rolle	DE
Klasse 2	Projektbearbeiter	Rolle	als Projektleiter DE
Klasse 3	Projektbearbeiter	Rolle	DE
Klasse 4	ProjektPhase	Rolle	DE
Klasse 5	Taetigkeit	Rolle	DE

Hier wird über den internen Namen der entsprechenden Kennzahlen die Berechnung definiert. Im Beispiel einen Stundensatz verrechenbar (Honorar extern / Minuten intern).

Die Dimensionen ergeben sich aus den zugrunde liegenden Kennzahlen und können hier nicht separat definiert werden.

Aktuell werden nur Divisionen unterstützt.

Generator Die Berechnung der Kennzahlen erfolgt über Generatoren (4.4), ausser es handelt sich um eine berechnete Kennzahl (siehe Attribut **Berechnete Kennzahl**).

Einheit Folgende Einheiten stehen zur Verfügung:

Betrag
Betrag
Aufwand
Prozent
Menge
Stundensatz

- Betrag: Zahlenwerte, dargestellt gemäss Ländereinstellungen
- Aufwand: Minutenwerte, dargestellt gemäss Systemeinstellung **Anzeige Minuten**
- Prozent: Prozentzahlen, dargestellt mit %. Die Werte werden als 1 basierte Quotienten interpretiert, ein Wert von 1 also als 100% dargestellt.
- Menge: Anzahl, mit zwei Nachkommastellen
- Stundensatz, Darstellung gemäss Ländereinstellungen

Sichtbar Steuert, ob eine Kennzahl in der BI Ansicht angezeigt werden kann. Dies ist vor allem dann sinnvoll, wenn Sie eine Kennzahl für die Berechnung einer anderen Kennzahl (siehe **Berechnete Kennzahl** weiter oben) benötigen, sie selbst jedoch nicht anzeigen möchten.

Standardmässig mitgelieferte Kennzahlen, die Sie gar nicht benötigen, können Sie auch deaktivieren (siehe Checkbox **Aktiv** weiter oben).

Stichtagwert Jede Kennzahl hat eine zeitliche Charakteristik: Sie bezieht sich entweder auf die ausgewählte Periode oder ist ein Stichtagwert. Diese gelten immer per Ende Monat.

Stichtagwerte werden nur in folgenden BI Ansichten dargestellt:

- Auswertung für 1 Monat (Datumsbereich in der BI Ansicht, 4.1)
- Darstellung als Zeitreihe (Gruppierung nach Monat in der BI Anzeige, 4.2)

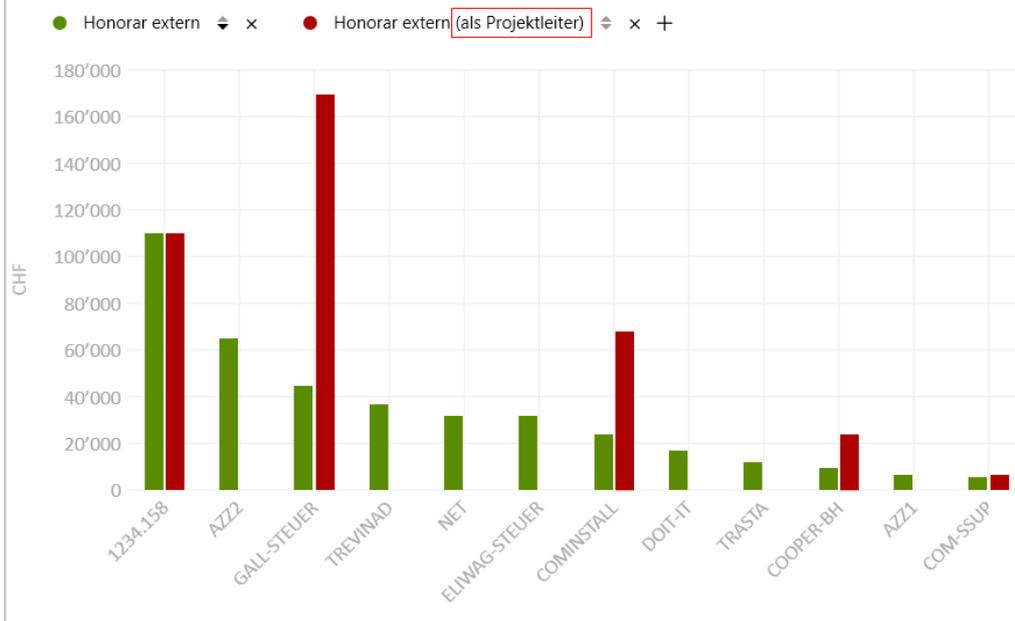
Dimensionen Gibt an, auf welchen Klassen (Dimensionen) die Kennzahlen zur Verfügung stehen. Diese müssen im angegebenen Generator berechnet werden.

Rollen Zusätzlich kann für jede Dimension auch eine Rollenbezeichnung angegeben werden. Mit der Rollenbezeichnung kann der Kontext der Kennzahl für diese Dimension angegeben werden.

Bei der Berechnung des Auftragseingangs beispielsweise wird auch der Projektleiter des Projekts zurückgegeben. Sehen wir nun auf einer Bearbeiterliste einfach den "Auftragseingang", dann ist der Kontext nicht klar: Ist das der Verkäufer, sind das seine eigenen Leistungen? Hier kann beispielsweise die Rolle angegeben werden:

Dimensionen			
Klasse 1	Projekt	Rolle	DE
Klasse 2	Projektbearbeiter	Rolle	als Projektleiter DE
Klasse 3	Projektbearbeiter	Rolle	DE
Klasse 4	ProjektPhase	Rolle	DE
Klasse 5	Taetigkeit	Rolle	DE

Die Rolle wird dann bei der Kennzahl in der BI Ansicht angezeigt:



Hilfetext Hier kann ein Hilfetext eingegeben werden, beispielsweise um dem Anwender zu erläutern, wie sich die Kennzahl berechnet.

Ist bei einer Kennzahl ein Hilfetext angegeben, erscheint in der Ansicht ein kleines i-Symbol. Platziert man den Mauszeiger darauf, erscheint der Hilfetext in als Tooltip:



Die Anzeige wird nach 19 Zeilen abgeschnitten. Bei der Erfassung muss darauf geachtet werden, dass der Text in die Anzeige passt.

Vertec liefert standardmässig folgende Kennzahlen mit:

Bezeichnung / Int. Name	Bedeutung	Verfügbar auf
Angefangene Arbeiten <i>FeesCommenced</i>	Honorar extern der Leistungen von produktiven Projekten, welche sich per Ende des Monats nicht auf einer Rechnung befinden oder deren Rechnung erst nachher verrechnet wurde. <pre>if not leistsum.rechnung or leistsum.rechnung.valutadatum > enddate: leistsum.wertExtOffen + leistsum.wertExtVerrechnet</pre>	– Projekt
Arbeitszeit <i>WorkingTime</i>	Arbeitszeit <pre>->getArbeitszeit(Startdatum, Enddatum)</pre>	– Projektbearbeiter
Aufwand extern <i>MinutesExt</i>	Aufwand extern <pre>leistsum.minutenExtOffen + leistsum.minutenExtVerrechnet</pre>	– Projekt – Projektbearbeiter – Projektphase – Tätigkeit

Bezeichnung / Int. Name	Bedeutung	Verfügbar auf
Aufwand intern <u>MinutesInt</u>	Aufwand intern <code>leistsum.minutenIntOffen + leistsum.minutenIntVerrechnet</code>	– Projekt – Projektbearbeiter – Projektphase – Tätigkeit
Bestand Vorschüsse <u>Advances</u>	Nettobetrag aller Vorschüsse – Geleistete Vorschüsse der bis Ende des Monats verrechneten Rechnungen (Valutadatum). <code>advances += advance.nettobetrag if invoice.verrechnet and invoice.valutadatum <= enddate: advances -= invoice.vorschussEffektiv</code>	– Projekt
Deckungsbeitrag (DB) <u>MarginContributed</u>	Honorar extern verrechnet – Kosten verrechnet <code>leistsum.wertExtVerrechnet - leistsum.wertKostenVerrechnet</code>	– Projekt – Projektbearbeiter – Projektphase – Tätigkeit
Feriensaldo <u>VacationBalance</u>	Feriensaldo per Ende des Monats <code>->getFerienSaldo (Enddatum)</code>	– Projektbearbeiter
Feriensaldo abgegrenzt <u>VacationBalanceDelimited</u>	Abgegrenzter Feriensaldo per Ende des Monats <code>->getFerienSaldoAbgegrenzt (Enddatum)</code>	– Projektbearbeiter
Honorar extern <u>FeesExt</u>	Honorar extern <code>leistsum.wertExtOffen + leistsum.wertExtVerrechnet</code>	– Projekt – Projektbearbeiter (als Projektleiter) – Projektbearbeiter – Projektphase – Tätigkeit
Honorar extern abgeschrieben <u>FeesExtWrittenoff</u>	Honorar intern verrechnet – Honorar extern verrechnet <code>leistsum.wertIntVerrechnet - leistsum.wertExtVerrechnet</code>	– Projekt – Projektbearbeiter (als Projektleiter) – Projektbearbeiter – Projektphase – Tätigkeit
Honorar extern offen <u>FeesExtOpen</u>	Honorar extern offen <code>leistsum.wertExtOffen</code>	– Projekt – Projektbearbeiter (als Projektleiter) – Projektbearbeiter – Projektphase – Tätigkeit
Honorar extern verrechnet <u>FeesExtCharged</u>	Honorar extern verrechnet <code>leistsum.wertExtVerrechnet</code>	– Projekt – Projektbearbeiter (als Projektleiter) – Projektbearbeiter – Projektphase – Tätigkeit

Bezeichnung / Int. Name	Bedeutung	Verfügbar auf
Honorar intern FeesInt	Honorar intern <code>leistsum.wertIntOffen + leistsum.wertIntVerrechnet</code>	– Projekt – Projektbearbeiter (als Projektleiter) – Projektbearbeiter – Projektphase – Tätigkeit
Kosten Costs	Kosten <code>leistsum.wertKostenOffen + leistsum.wertKostenVerrechnet</code>	– Projekt – Projektbearbeiter (als Projektleiter) – Projektbearbeiter – Projektphase – Tätigkeit
Sollzeit SetTime	Sollzeit <code>->getSollzeit (Startdatum, Enddatum)</code>	– Projektbearbeiter
Stundensatz verrechenbar RateChargeable	Honorar extern / Aufwand intern (Berechnung siehe bei den Kennzahlen dieses Namens)	– Projektbearbeiter
Überzeitsaldo OvertimeBalance	Überzeitsaldo per Ende des Monats <code>->getUeberzeitsaldo (Enddatum)</code>	– Projektbearbeiter
Umsatz Auslagen TurnoverOutlays	Auslagen verrechnet (Valutadatum der Rechnung) <code>Rechnung.auslagenExt</code>	– Projekt – Projektbearbeiter Nur verfügbar, wenn Modul Fremdkosten lizenziert ist.
Umsatz Leistungen TurnoverServices	Total nach Rabatt verrechnet (Valutadatum der Rechnung) <code>Rechnung.leistWertExtNachRabatt</code>	– Projekt – Projektbearbeiter
Umsatz Spesen TurnoverExpenses	Spesen verrechnet (Valutadatum der Rechnung) <code>Rechnung.spesenExt</code>	– Projekt – Projektbearbeiter
Vollzeitstellen Fte	Beschäftigungsgrad <code>->getBeschaefigungsgrad (Enddatum)</code>	– Projektbearbeiter

Zusätzlich werden folgende Kennzahlen standardmässig inaktiv mitgeliefert. Für die Verwendung ist die Lizenzierung des Moduls Budget & Teilprojekt, Linie Expert sowie eine genaue Budgetplanung Voraussetzung.

Bezeichnung / Int. Name	Bedeutung	Verfügbar auf
Auftragsbestand Honorar FeesOrderbacklog	Budget Honorare - kumuliertes Honorar extern. Nur erteilte und noch nicht abgeschlossene Phasen. <code>bdgvalue('planWertExt', Enddatum, -1) - leistSum.wertExtOffen +</code>	– Projekt – Projektbearbeiter – Projektphase Nur verfügbar, wenn Modul Budget & Teilprojekt lizenziert

Bezeichnung / Int. Name	Bedeutung	Verfügbar auf
	<code>leistSum.wertExtVerrechnet</code>	ist.
Auftragseingang Honorar <code>FeesOrderintake</code>	Budget Honorare der neu erteilten Phasen oder Änderungen dieses Budgetwerts während dieses Monats. <code>bdgvalue('planWertExt', Enddatum, -1) -</code> <code>bdgvalue('planWertExt', (Startdatum-1</code> <code>Tag), -1)</code>	– Projekt – Projektbearbeiter – Projektphase Nur verfügbar, wenn Modul Budget & Teilprojekt lizenziert ist.
Aufwand Budget Rest <code>MinutesBudgetRest</code>	Aufwand Budget Total – Aufwand Ist (Berechnung siehe bei den Kennzahlen dieses Namens)	– Projekt – Projektbearbeiter – Projektphase Nur verfügbar, wenn Modul Budget & Teilprojekt lizenziert ist.
Aufwand Budget Total <code>MinutesBudgetTotal</code>	Budget Aufwand von erteilten und noch nicht abgeschlossenen Phasen. <code>bdgvalue('planMinutenInt', Enddatum, -1)</code>	– Projekt – Projektbearbeiter – Projektphase Nur verfügbar, wenn Modul Budget & Teilprojekt lizenziert ist.
Aufwand Ist <code>MinutesCumulated</code>	Kumulierter Aufwand intern von erteilten und noch nicht abgeschlossenen Phasen. <code>leistSum.minutenIntOffen +</code> <code>leistSum.minutenIntVerrechnet</code>	– Projekt – Projektbearbeiter – Projektphase Nur verfügbar, wenn Modul Budget & Teilprojekt lizenziert ist.
Fertigstellungsgrad <code>PercentageOfCompletion</code>	Aufwand Ist / Aufwand Budget Total (Berechnung siehe bei den Kennzahlen dieses Namens)	– Projekt – Projektbearbeiter – Projektphase Nur verfügbar, wenn Modul Budget & Teilprojekt lizenziert ist.

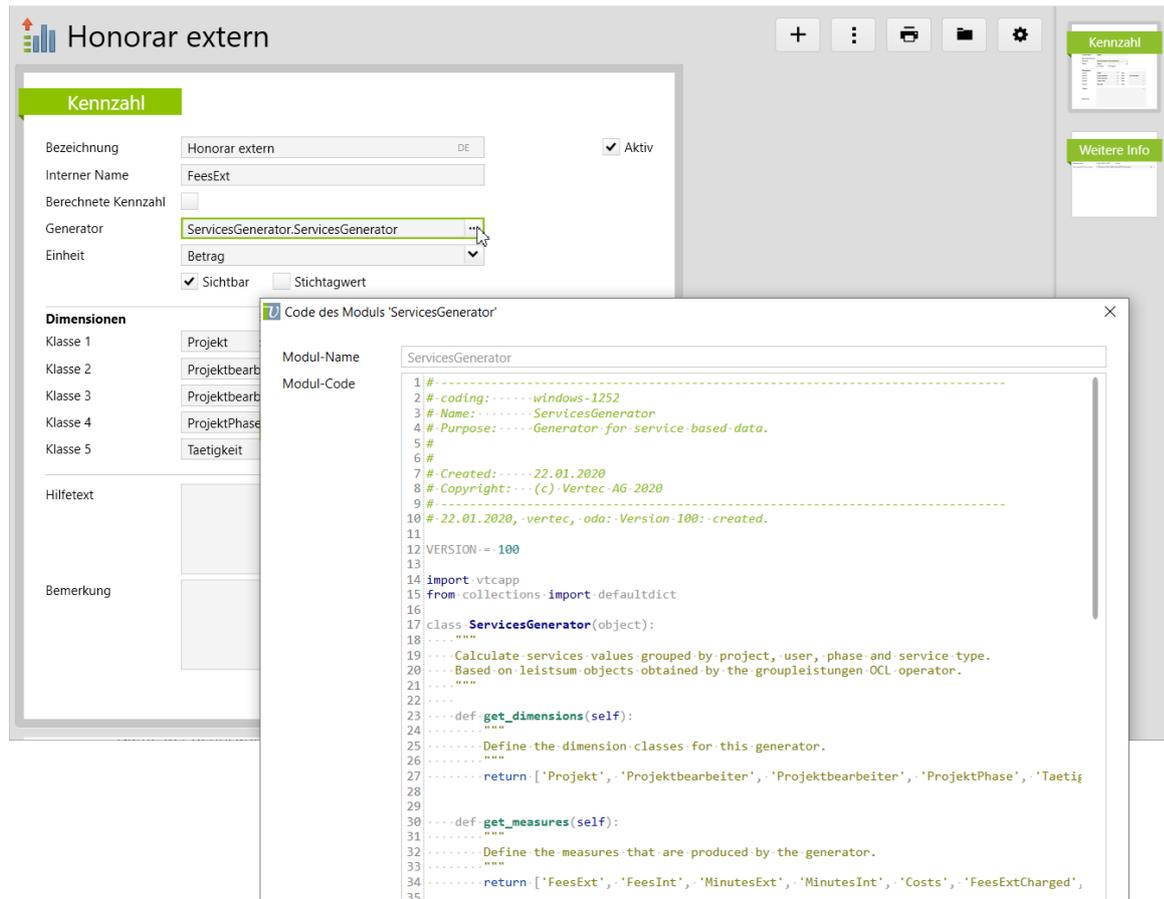
4.4 Generatoren

Für die Bereitstellung der BI Daten setzen wir einen Datenspeicher gemäss dem Datawarehouse Konzept ein, z.T. auch "OLAP Cube" genannt. Je nach Selektion der Dimensionen wird dann sinn gemäss ein Schnitt aus einem Würfel geschnitten.

Im Wesentlichen besteht das Datawarehouse aus einer grossen Datentabelle (Cube) mit vorberechneten Datenwerten. Dadurch wird ein effizienter, Businesslogik unabhängiger Zugriff auf die BI Daten und somit eine performante Anzeige auch mit grossen Datenmengen ermöglicht.

Die Berechnung erfolgt zeitlich unabhängig und mit voller Businesslogik Unterstützung via regelmässiger geplanter Aufgaben. So kann die Berechnung bei Bedarf in einzelnen Paketen (z.B. einzelne Monate) in eigenen Prozessen geschehen oder sogar parallelisiert werden. Siehe dazu Kapitel 4.6.

Die Berechnungslogik wird in "Generatoren" genannten Python-Scripts definiert. Diese sind auf den Kennzahlen (4.3) hinterlegt. Um den Code eines Generators einzusehen, klicken Sie auf den Button mit den drei Punkten beim Feld **Generator** auf der Kennzahl:



The screenshot shows the 'Kennzahl' configuration interface for 'Honorar extern'. The 'Generator' field is set to 'ServicesGenerator.ServicesGenerator'. A modal window titled 'Code des Moduls 'ServicesGenerator'' is open, displaying the following Python code:

```

1 #-----
2 # coding: ..... windows-1252
3 # Name: ..... ServicesGenerator
4 # Purpose: ..... Generator for service-based data.
5 #
6 #
7 # Created: ..... 22.01.2020
8 # Copyright: ..... (c) Vertec AG 2020
9 #-----
10 # 22.01.2020, vertec, oda: Version 100: created.
11
12 VERSION -- 100
13
14 import vtccpp
15 from collections import defaultdict
16
17 class ServicesGenerator(object):
18     """
19     Calculate services values grouped by project, user, phase and service type.
20     Based on leistungsum objects obtained by the groupleistungen OCL operator.
21     """
22
23     def get_dimensions(self):
24         """
25         Define the dimension classes for this generator.
26         """
27         return ['Projekt', 'Projektbearbeiter', 'Projektbearbeiter', 'ProjektPhase', 'Taetig
28
29
30     def get_measures(self):
31         """
32         Define the measures that are produced by the generator.
33         """
34         return ['FeesExt', 'FeesInt', 'MinutesExt', 'MinutesInt', 'Costs', 'FeesExtCharged',
35

```

Abbildung 11: Python-Code eines Generators

Der Generator muss für die Kennzahlen, auf denen er hinterlegt ist, eine Berechnung enthalten. Der interne Name der Kennzahl muss also mit den vom Generator zurückgelieferten Werten übereinstimmen.

4.5 Generatoren erweitern / erstellen

Generatoren können auch erweitert bzw. neue Generatoren geschrieben werden. Dafür erzeugen Sie im Ordner **Einstellungen > Scripts** einen neuen Scripteintrag und verfassen den Generator-Code als Python Script.

Die Bezeichnung des Scripteintrags ist der Modulname. Zusammen mit dem Klassen-Namen des Generators bildet er die Generator-ID. Diese können Sie bei den BI Kennzahlen, die durch den Generator berechnet werden, als Generator angeben.

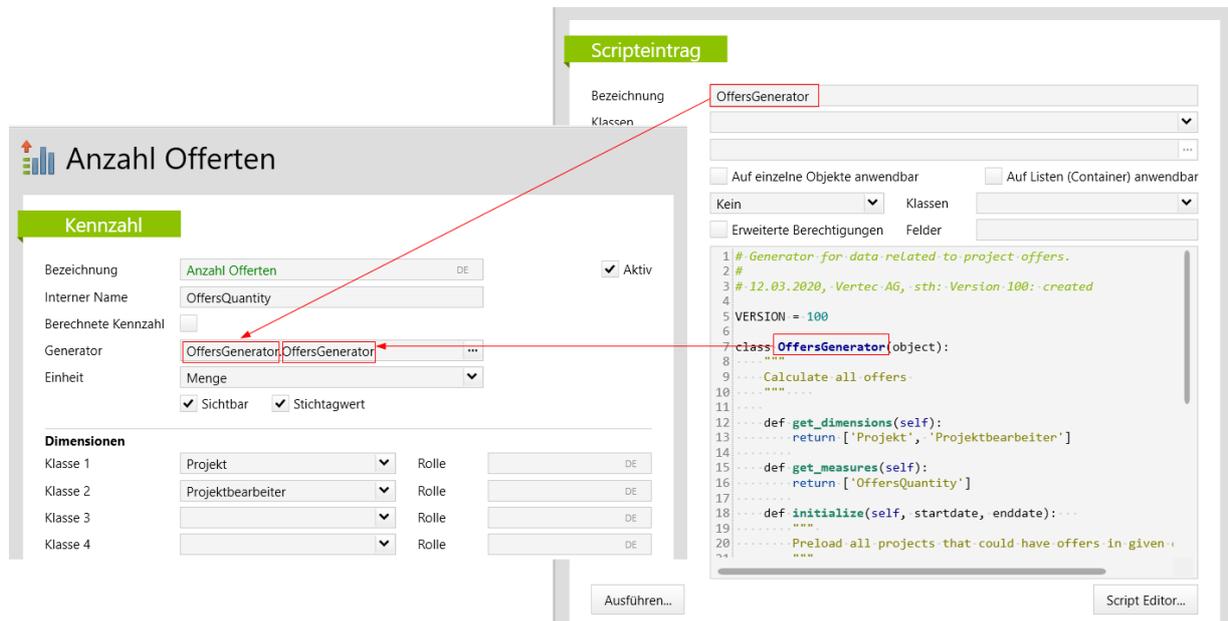


Abbildung 12 Die Bezeichnung des Scripteintrags (Modulname) plus der Generatorname im Script werden auf der Kennzahl angegeben
 Modulnamen müssen eindeutig sein. Sie "überschreiben" also nicht einen bestehenden Generator, indem Sie ein Modul gleichen Namens verfassen, sondern indem Sie einen neuen Generator erstellen und auf den entsprechenden Kennzahlen hinterlegen.

Die Deklaration des Generators ist wie folgt:

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

Auf das Generator-Objekt kann im Script jederzeit über `self` zugegriffen werden. Innerhalb des Generators stehen folgende Methoden zur Verfügung:

Methode

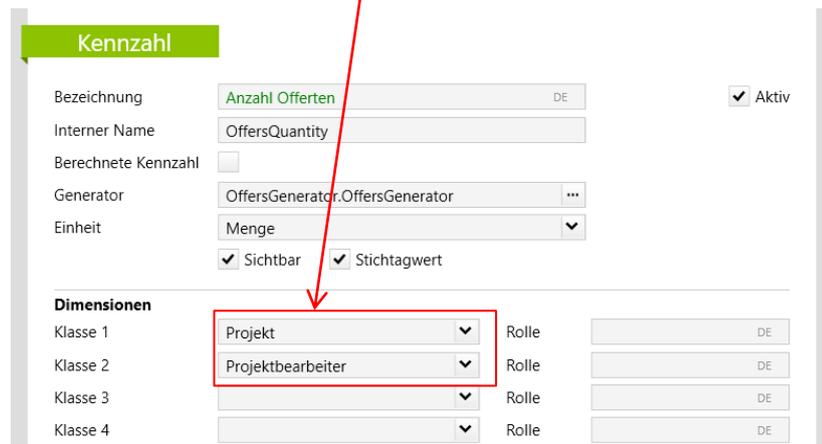
Beschreibung

get_dimensions(self)

Mit dieser Methode wird angegeben, für welche Dimensionen der Generator Werte berechnet:

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

In den Kennzahlen müssen genauso viele Dimensionen angegeben werden, wie hier geliefert werden, und zwar in der angegebenen Reihenfolge:



Möchte man also zum Beispiel Werte sowohl für Bearbeiter als Projektleiter als auch für Bearbeiter als Hauptbearbeiter anzeigen, muss die Dimension Projektbearbeiter zweimal angegeben werden:

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

Und auf der Kennzahl:



get_measures(self)

Mit dieser Methode wird angegeben, welche Kennzahlen damit berechnet werden. Die Rückgabewerte müssen den internen Namen auf den Kennzahlen entsprechen.

Methode	Beschreibung
	<pre>def get_measures(self) : return ['OffersQuantity']</pre> <p>Auf allen Kennzahlen, die hier zurückgeliefert werden, kann der Generator dann hinterlegt werden.</p>
initialize(self, startdate, enddate)	<p>Als erstes wird diese Methode durchlaufen. Die Variablen <code>startdate</code> und <code>enddate</code> enthalten das Datumsintervall der gesamten berechneten Periode (siehe auch Abschnitt 4.6 weiter unten).</p> <p>Hier kann man die Listen der verwendeten Objekte performanceoptimiert preloaden. So hat man alle relevanten Daten der gesamten Periode. In der Methode <code>generate()</code> werden die Daten dann pro Monat berechnet.</p>
generate(self, startdate, enddate)	<p>Die <code>generate</code>-Methode wird so viel Mal durchlaufen, wie es Monate in der berechneten Periode hat.</p> <p>Die Variablen <code>startdate</code> und <code>enddate</code> enthalten immer den ersten und den letzten Tag des Monats.</p> <p>Die im <code>initialize()</code> vorgeladenen Daten können hier nun pro Monat gefiltert und berechnet werden.</p> <p>Rückgabewert</p> <p>Als Rückgabewert wird ein Tuple aus Dimensionen und ein Tuple aus Measures (Kennzahlen) zurückgeliefert, in genau der oben deklarierten Reihenfolge.</p> <p>Der Rückgabewert wird nicht wie sonst in Python mit <code>return()</code> zurückgegeben, sondern mit <code>yield()</code>:</p> <pre>yield((project, project.projektleiter), (OffersQuantity,))</pre> <p>Die Verwendung von <code>yield()</code> hat zur Folge, dass es nicht nur einen Rückgabewert gibt (und alle Daten mühsam in einem Array zusammengefasst und dann zurückgegeben werden müssen), sondern jede Zeile zurückgegeben werden kann, bis der gesamte Code abgelaufen ist. So können zum Beispiel innerhalb einer Projekt-Schleife die berechneten Daten pro Projekt etc. laufend einfach "abgeliefert" werden (siehe Beispielcode weiter unten).</p> <p>Währungen</p> <p>Falls man in Vertec mit mehr als einer Währung arbeitet, muss man dies bei der Berechnung beachten. Dafür kann pro Rückgabezeile optional eine Währung mitgegeben werden (Beispiel aus dem mitgelieferten <code>InvoiceGenerator</code>):</p> <pre>yield ((project), turnover_tuple, project.waehrung)</pre> <p>Vertec kümmert sich dann selbst um die korrekte Umrechnung in die Leitwährung. Angezeigt werden die Werte immer in Leitwährung.</p>
finalize(self)	<p>Am Schluss wird noch die <code>finalize</code>-Methode durchlaufen. Darin können zum Beispiel nicht mehr benötigte Objekte aus dem Memory wieder entfernt werden.</p>

Beispielcode

```
# 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,))

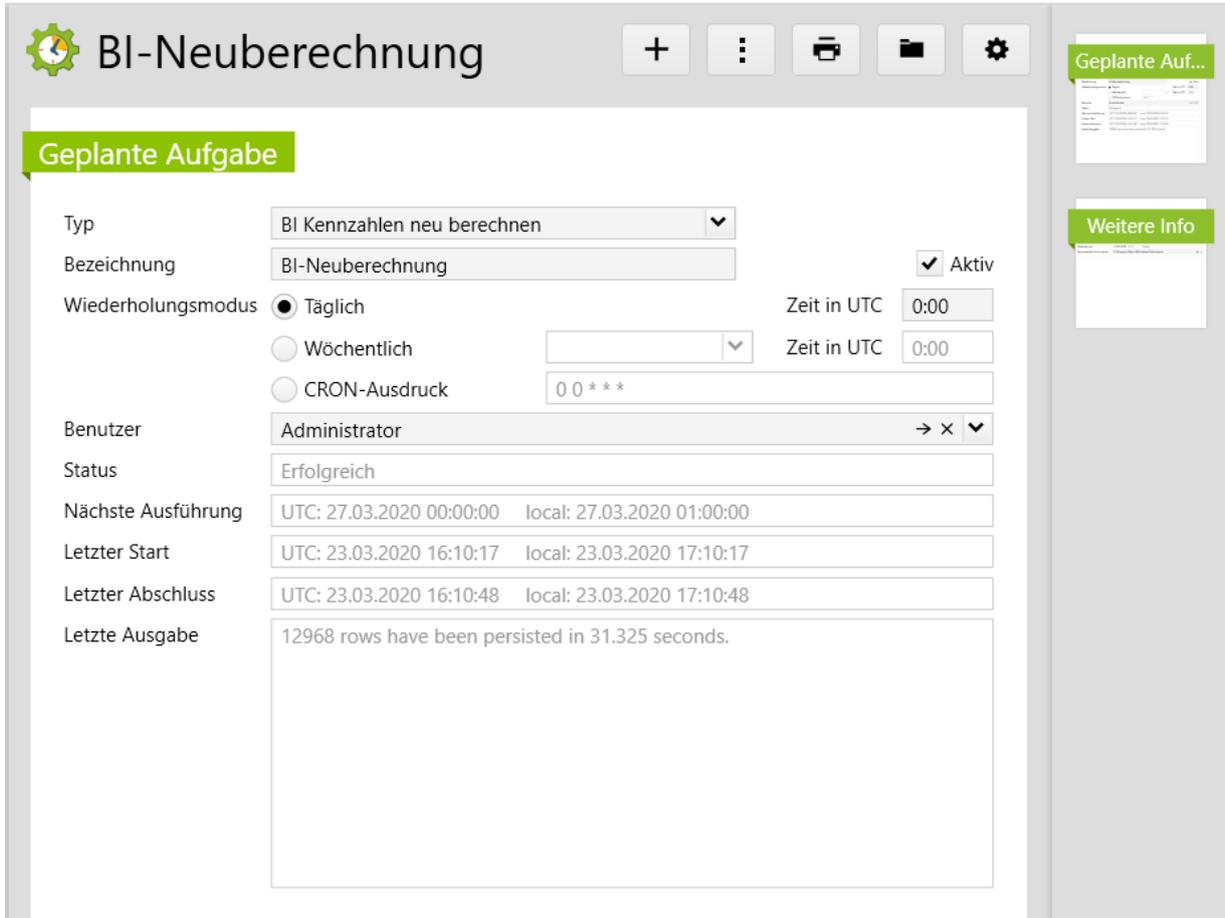
    def finalize(self):
        pass
```

Vertec liefert standardmässig folgende Generatoren mit:

Generator-Name	Verwendete Daten	Berechnete Kennzahlen
ServicesGenerator.ServicesGenerator	Leistungssummen (globaler groupLeistungen Operator) gruppiert nach Projekt, Phase, Bearbeiter und Tätigkeit.	<ul style="list-style-type: none"> – Honorar extern – Honorar intern – Aufwand extern – Aufwand intern – Kosten – Honorar extern verrechnet – Honorar extern offen – Honorar extern abgeschrieben – Deckungsbeitrag (DB)
ProjectGenerator.ProjectGenerator	Leistungssummen (globaler groupLeistungen Operator) gruppiert nach Projekt und Rechnung. Verwendet nur die produktiven Projekte.	<ul style="list-style-type: none"> – Bestand Vorschüsse – Angefangene Arbeiten
UsersGenerator.UsersGenerator	Projektbearbeiter, die vor dem Enddatum eingetreten und noch nicht ausgetreten sind bzw. deren Austrittsdatum nach dem Startdatum liegt.	<ul style="list-style-type: none"> – Vollzeitstellen – Sollzeit – Präsenzzeit – Überzeitsaldo – Feriensaldo – Feriensaldo abgegrenzt
InvoicesGenerator.InvoicesGenerator	Rechnungen, welche verrechnet sind und deren Valutadatum innerhalb des berechneten Datumsbereichs liegt.	<ul style="list-style-type: none"> – Umsatz Leistungen – Umsatz Spesen – Umsatz Auslagen
PhasesGenerator.PhasesGenerator	Projektphasen der obersten Ebene, welche ein Erteiltatum haben und deren Erteiltatum vor dem Enddatum liegt, und die nicht oder erst nach dem Startdatum abgeschlossen wurden: <ul style="list-style-type: none"> – BudgetChanges dieser Phasen – Leistungssummen dieser Phasen (groupLeistungenPWG) 	<ul style="list-style-type: none"> – Auftragsbestand Honorar – Auftragseingang Honorar – Aufwand Ist – Aufwand Budget Rest – Aufwand Budget Total

4.6 Daten berechnen

Die Daten werden regelmässig über eine (oder mehrere) geplante Aufgabe in der Nacht berechnet (siehe dazu auch Kapitel 8.1). Die geplanten Aufgaben finden Sie im Ordner **Einstellungen > Geplante Aufgaben**.



BI-Neuberechnung

Geplante Aufgabe

Typ: BI Kennzahlen neu berechnen

Bezeichnung: BI-Neuberechnung Aktiv

Wiederholungsmodus:

- Täglich Zeit in UTC: 0:00
- Wöchentlich Zeit in UTC: 0:00
- CRON-Ausdruck: 0 0 * * *

Benutzer: Administrator

Status: Erfolgreich

Nächste Ausführung: UTC: 27.03.2020 00:00:00 local: 27.03.2020 01:00:00

Letzter Start: UTC: 23.03.2020 16:10:17 local: 23.03.2020 17:10:17

Letzter Abschluss: UTC: 23.03.2020 16:10:48 local: 23.03.2020 17:10:48

Letzte Ausgabe: 12968 rows have been persisted in 31.325 seconds.

Abbildung 13: Eine geplante Aufgabe für die Neuberechnung der BI Werte

Bei diesem Vorgang werden alle bei den aktiven Kennzahlen hinterlegten Generatoren angestossen und die Werte für das Datumsintervall von Januar des vorletzten Jahres bis Ende des aktuellen Monats vorberechnet.

Auf der Aufgabe ist im Status sichtbar, ob die Berechnung erfolgreich war oder fehlgeschlagen ist. Im Ausgabe-fenster wird die Rückmeldung der Berechnung angezeigt.

Eine Vertec App muss dafür nicht laufen, nur ein laufender CloudServer ist Voraussetzung. Für die Batch-Berechnung wird auf der geplanten Aufgabe ein Benutzer hinterlegt, welcher für die Berechnung verwendet wird. Dieser Benutzer muss über Administratoren-Rechte verfügen, um die Berechnungen durchführen zu können.

Variante Cloud Abo

Im Cloud Abo kann die Zeit auf der Oberfläche nicht verändert werden. Die Berechnung erfolgt immer über Nacht. Welches der genaue Zeitpunkt ist, wird unter **Nächste Ausführung** angezeigt:

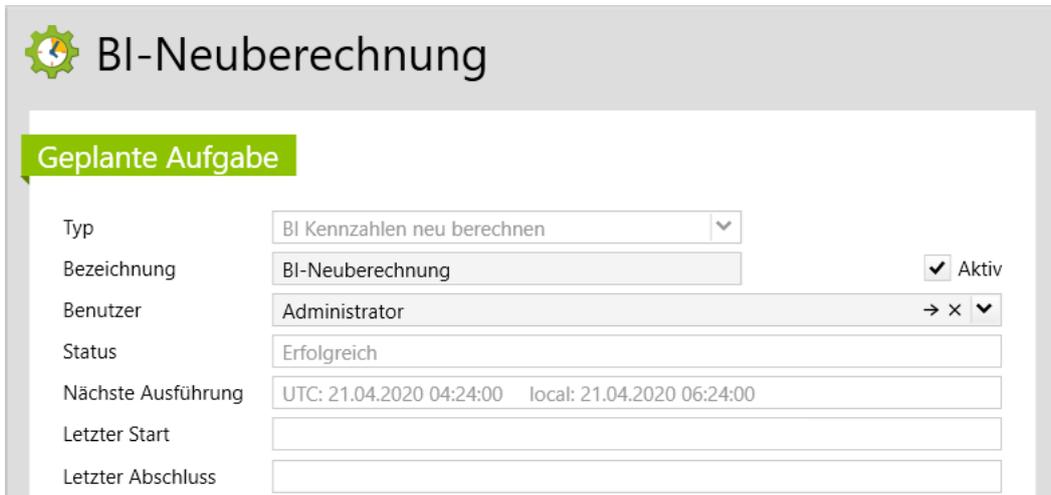
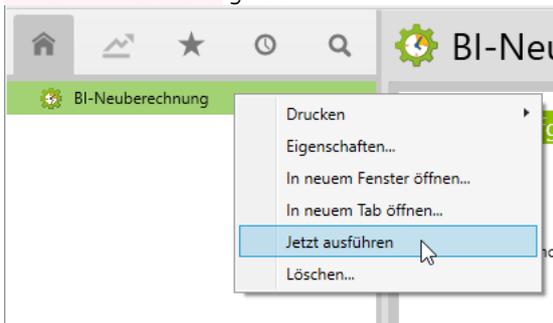


Abbildung 14: Die Steuerung der Ausführungszeit erfolgt zentral

Berechnung manuell anstossen

Eine BI-Berechnung kann auch manuell angestoßen werden. Hier gibt es zwei Varianten:

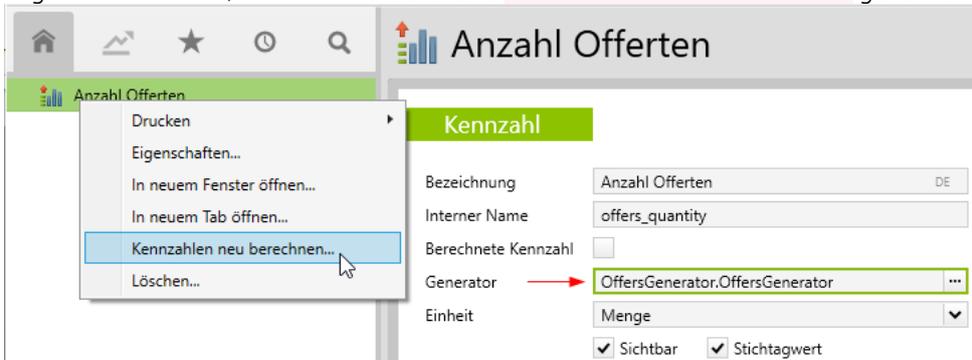
- **Geplante Aufgabe jetzt ausführen**: Dafür wird im Kontextmenü der geplanten Aufgabe der Menüpunkt **Jetzt ausführen** geklickt:



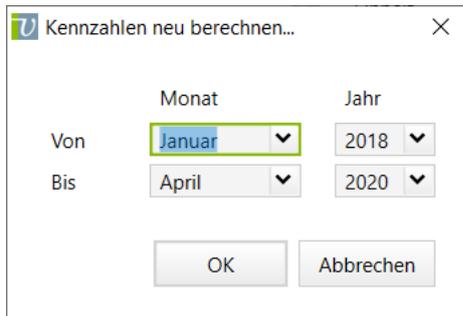
Dadurch wird die Berechnung aller Generatoren auf allen aktiven Kennzahlen angestoßen.

Achtung: Die Berechnung wird direkt in der laufenden Vertec App gestartet und nicht wie bei der zeitgesteuerten Ausführung in einem separaten Task Runner Prozess. Das sollte nicht im laufenden Betrieb gemacht werden, da die Berechnung sehr lange dauern und Vertec dadurch blockiert sein kann.

- **Anstoßen der Berechnung eines Generators**: Auf den einzelnen Kennzahlen kann die Berechnung manuell angestoßen werden, indem im Kontextmenü **Kennzahlen neu berechnen** gewählt wird:



Dabei wird der auf der Kennzahl hinterlegte **Generator** angestoßen und die entsprechenden Kennzahlen neu berechnet. Es kann jeweils das Datumsintervall noch gesetzt werden:



Am Ende kommt eine Meldung, welche Kennzahlen berechnet wurden.

Python Methoden zur Berechnung der BI Daten

Es gibt zwei neue Python Methoden für die Berechnung der BI Daten.

<code>vtcapp.standardprocessbi()</code>	<p>Löst die Berechnung aller Generatoren auf allen aktiven Kennzahlen aus. Benötigt Administrator-Rechte.</p> <p>Entspricht der Durchführung der standardmässig mitgelieferten geplanten Aufgabe, siehe oben.</p>
---	---

<code>vtcapp.processbi(von: String, bis: String, generator: String)</code>	<p>Löst die Berechnung des als Parameter mitgelieferten Generators aus. Benötigt Administrator-Rechte.</p> <p>Von- und Bis-Datum werden als String im Format "Jahr-Monat" angegeben.</p> <p>Der Generator muss gleich angegeben werden wie bei den Kennzahlen, also "<Modulname>.<Generatorname>".</p>
--	---

```
vtcapp.processbi("2018-01", "2020-02",
"OffersGenerator.OffersGenerator")
```

Entspricht der Berechnung der BI Daten auf einer Kennzahl, siehe oben.

4.7 BI Berechtigungen

Für die Abfrage von BI Daten wurde eine neue Berechtigung **BI Daten** eingeführt.

Der Benutzer muss dieses Recht auf der selektierten Klasse (siehe Kapitel 4.2 über die BI Anzeigen) haben, damit die Auswertung angezeigt werden kann.

Folgende Standard-Berechtigungen wurden eingeführt:

- Ein Projektleiter hat BI Daten Recht auf seinen eigenen Projekten
- Ein Projekt-Supervisor hat allgemeines BI Daten Recht
- Ein Teamleiter hat BI Daten Recht auf allen seinen Team-Mitgliedern

Zur Überprüfung gibt es einen neuen OCL-Operator `hasBiDataRight`, der zu jedem Eintrag Rückmeldung gibt, ob der eingeloggte Bearbeiter BI Daten Rechte darauf hat.

Was passiert bei fehlender Berechtigung

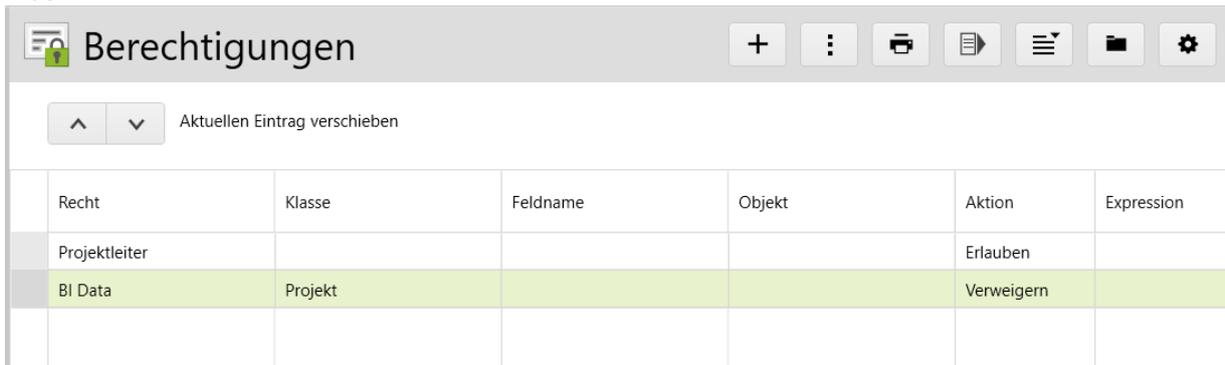
- Benutzer ohne BI Daten Recht auf Projekten sehen bei allen Auswertungen auf Projekten oder Projektlisten eine spezielle Grafik mit einem Hinweis ("Kein Zugriff auf BI Daten für Projekte").
- Mit Objektberechtigung kann die Berechtigung auf einem einzelnen Objekt gesteuert werden. Sobald der Benutzer auf einem Objekt kein BI-Leserecht hat, erhält er die Fehlermeldung "Der Zugriff auf ein Objekt vom Typ 'Projekt' wurde verweigert".

- Benutzer mit BI Daten Recht auf Projekten, aber nicht auf Bearbeitern, sehen bei Auswertungen auf Projekten nach Bearbeitern trotzdem die Auswertung. Für die Berechtigungsprüfung ist nur die Dimension massgebend, von der ausgegangen wird (Selektion in der BI Anzeige), nicht die Dimension, nach der Angezeigt wird (Gruppierung in der BI Anzeige).
- Bei Benutzern, welche über keinerlei BI Rechte verfügen, wird der Button für die BI Ansicht ausgeblendet. Dies erfolgt nur dann, wenn der Benutzer ganz sicher kein BI Rechte hat; bei komplizierten Rechte-Konstellationen kann es sein, dass der Button für die BI Ansicht angezeigt wird, obwohl der Benutzer tatsächlich keine BI Auswertung anschauen kann.

Einem Benutzer mit impliziter Berechtigung das BI Recht verweigern

Möchte man einem Benutzer mit impliziter Berechtigung (z.B. Projektleiter, siehe oben) das BI Recht trotzdem nicht gewähren, kann man dieses Recht wie folgt entziehen:

- Auf der entsprechenden Gruppe (also z.B. bei Projektleiter) eine weitere Zeile einfügen
- Das BI Daten Recht auswählen
- Wichtig: Es muss die Klasse angegeben werden, für die das BI Daten Recht verweigert werden soll. Ohne diese Angabe ist die Zeile wirkungslos.
- Möchte man sämtliche Berechtigung für BI Daten entziehen, muss als Klasse **UserEintrag** eingetragen werden.



Recht	Klasse	Feldname	Objekt	Aktion	Expression
Projektleiter				Erlauben	
BI Data	Projekt			Verweigern	

Abbildung 15: Den Projektleitern wird das BI Daten Recht auf Projekten verweigert

4.8 Rückwärtskompatibilität

Beim Business Intelligence Modul handelt es sich um ein neues Produkt, welches das bisherige Benchmarking zwar ablöst, aber sonst keine Rückwärtskompatibilität im herkömmlichen Sinne bietet.

Hier deshalb eine Auflistung der Unterschiede zum bisherigen Benchmarking.

Auswertungsperioden

Die Auswertungsperioden wie im Benchmarking gibt es nicht mehr. Diese Funktionalität ist neu zweigeteilt:

1. Die Werte vom vorletzten bis zum aktuellen Jahr werden jede Nacht vorberechnet und in der Datenbank gespeichert, siehe Kapitel 4.6.
2. In der BI Ansicht kann bei jeder Auswertung die Datumsperiode (innerhalb der vorberechneten Periode) ausgewählt werden, siehe Kapitel 4.1.
3. Alle Zahlen werden immer mindestens für einen ganzen Monat dargestellt.

Auswertungen

Alle im Benchmarking zur Verfügung stehenden Listen-Auswertungen sind im neuen BI Modul enthalten und werden standardmässig mitgeliefert.

Die Zusammenfassung auf den Detail-Ansichten gibt es nicht mehr.

Projektionen siehe am Schluss dieses Kapitels.

Ordnervergleiche

Das BI Modul unterstützt bei den Ordnervergleichen nur noch eine Hierarchiestufe. Dafür erscheinen sie automatisch auf allen BI-selektierbaren Objekten, siehe Abschnitt 4.2.

Grafik vs. Zahlen

Das BI Modul in der vorliegenden Version stellt die Auswertungen grafisch dar. Auf jedem Balken bzw. Linienpunkt können die konkreten Zahlen angezeigt werden, indem der Mauszeiger darauf positioniert wird.

Die gesamte Darstellung kann in Excel exportiert werden, dort sind dann alle Zahlen auf einen Blick ersichtlich.

Siehe zu diesem Thema Kapitel 4.1.

Berechtigungen

Das bisherige Recht "Benchmarking" steht nicht mehr zur Verfügung. Neu gibt es ein **BI Daten** Recht, welches in Abschnitt 4.7 detailliert erklärt ist. Diese Rechte werden bei einem Update nicht ausgetauscht, das heißt, Bearbeiter, die bisher über das Benchmarking-Recht verfügten, haben nicht automatisch das neue BI Daten Recht. Die Berechtigungen müssen deshalb nach dem Update bei Bedarf neu vergeben werden. Bitte beachten Sie dabei, dass gewisse BI-Rechte bereits implizit vergeben sind bei Projektleitern, Projekt-Supervisoren und bei Teamleitern. Weitere Informationen finden Sie im Kapitel 4.7.

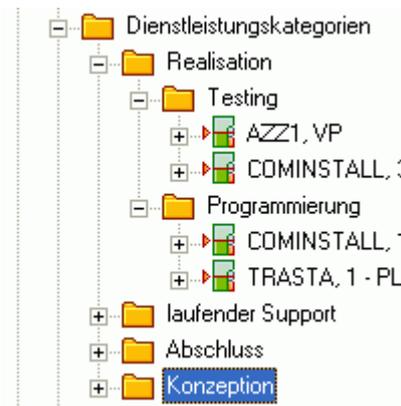
Projektionen

Bestehende Projektionen aus dem Benchmarking werden nicht übernommen, sondern müssen mit dem BI Modul neu erstellt werden (Gruppierung nach Ordner in der BI Anzeige, siehe 4.2).

Hier nachfolgend ein Beispiel einer Projektion im Benchmarking und wie diese korrekt im BI Modul erstellt wird.

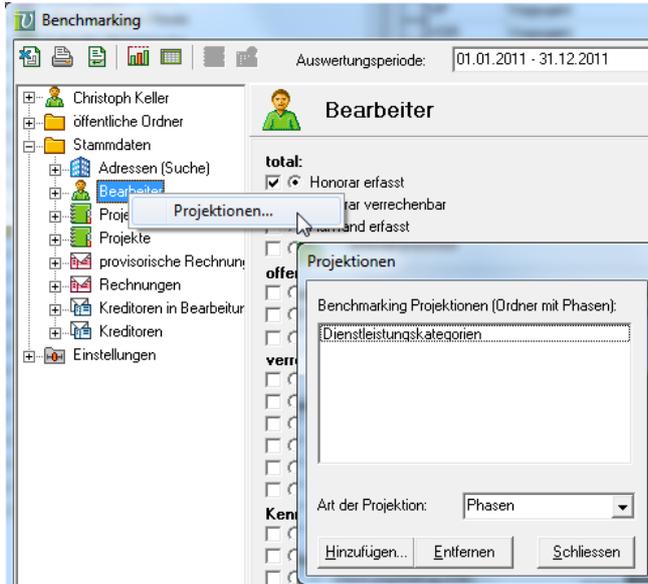
Projektion im Benchmarking

Im Beispiel ist eine Ordnerstruktur **Dienstleistungskategorien** zu sehen.



Der Überordner **Dienstleistungskategorien** gibt den Namen der Projektion vor, die direkten Unterordner definieren die Unterteilung der Werte. Die Ordnerhierarchie kann mehrere Ebenen umfassen, auf der untersten Ebene sind dann die eigentlichen Objekte zugeordnet. Im Benchmarking mussten das Projekte, Projektbearbeiter oder Projektphasen sein, neu können Objekte von allen Klassen, für welche Kennzahlen zur Verfügung stehen, zugeordnet sein (siehe 4.3).

Damit der Ordner **Dienstleistungskategorien** im Benchmarking als Projektion angezeigt wird, muss dieser einem beliebigen Ordner zugeordnet werden, zum Beispiel dem Ordner **Bearbeiter**. Die Zuordnung geschieht im Benchmarkingfenster durch Rechtsklick auf den Ordner:



Wird nun im Benchmarking-Fenster der Bearbeiter Ordner angewählt, so ist eine neue Auswertung namens Dienstleistungskategorien sichtbar, in welcher alle Leistungszahlen der Bearbeiter nach Dienstleistungskategorien gruppiert dargestellt werden.

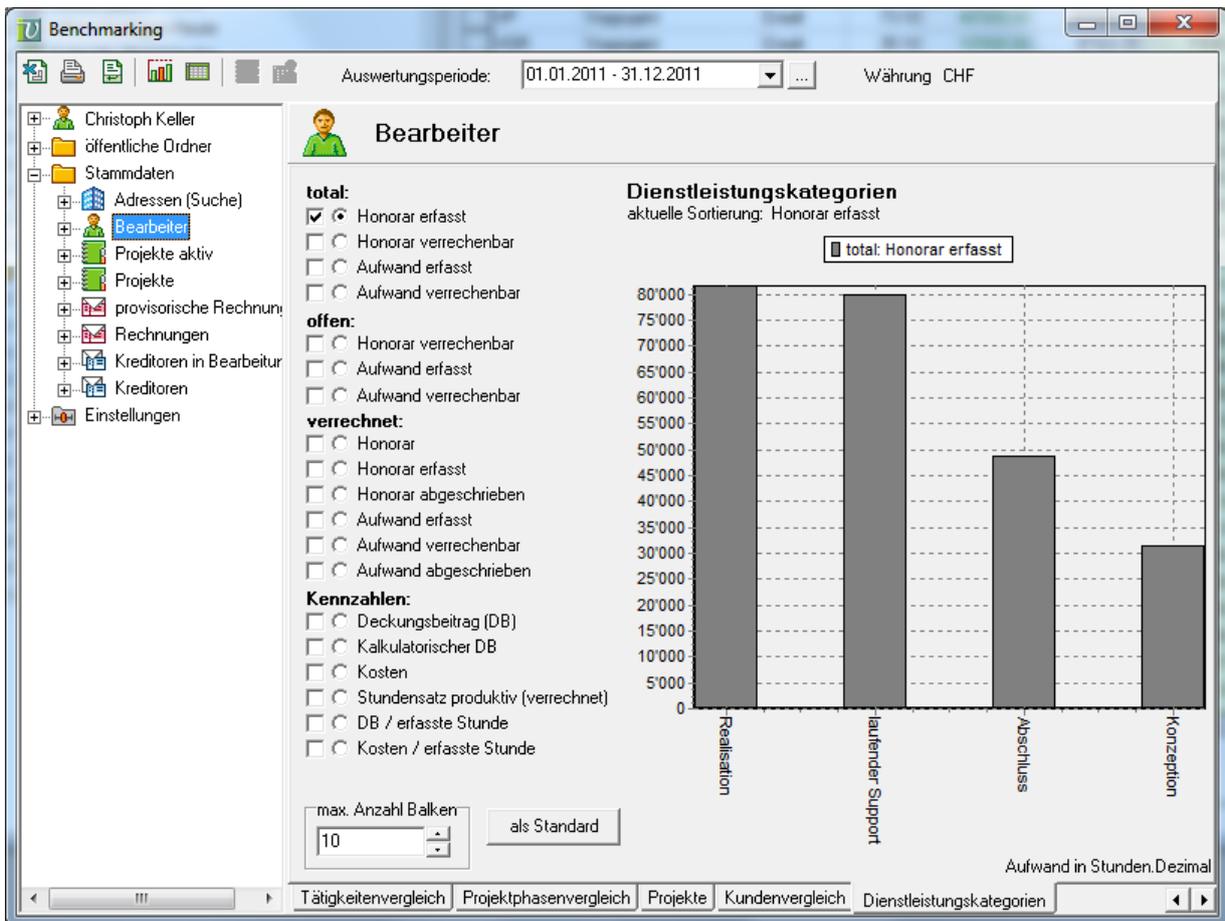
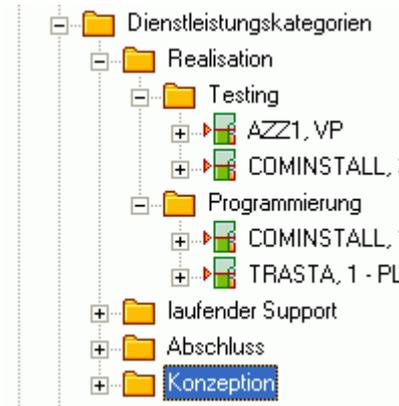


Abbildung 16: Benchmarking-Projektion

"Projektion" im BI Modul

Diese Projektion wie oben legen wir nun im neuen BI Modul an.

Wir verwenden dazu dieselbe Ordnerstruktur **Dienstleistungskategorien**:



Neu wird jedoch nur noch eine Hierarchieebene unterstützt. Zuerst müssen wir also die Ordnerstruktur entsprechend anpassen:



Abbildung 17: Ordnerstruktur mit einer Hierarchieebene

Weitere Informationen Für das Verschieben der Ordner wurde das Script **Ordner verschieben** auf der Online Knowledge Base verwendet:
www.vertec.com/kb/changeparentordner

Nun definieren wir eine neue BI Anzeige (siehe 4.2).

Da wir die Projektion auf Bearbeitern anzeigen wollen, wählen wir als Selektion **Bearbeiter** aus.

Bei der Gruppierung wählen wir die Klasse aus, die wir in der Ordnerstruktur zugeordnet haben. In unserem Fall sind das Projektphasen.

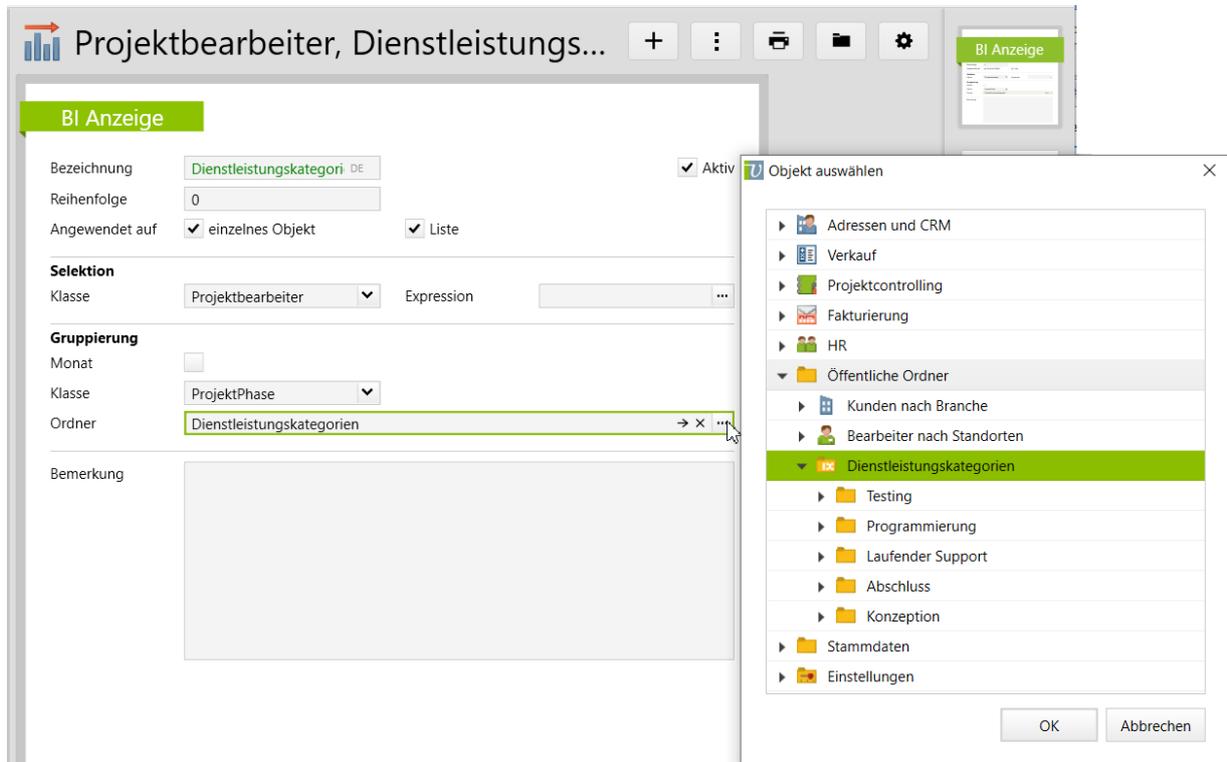


Abbildung 18: Definition der BI Anzeige für die Ordnerprojektion

Wird nun in der BI Ansicht ein Bearbeiter oder eine Liste von Bearbeitern angewählt, erscheint die Auswertung namens **Dienstleistungskategorien**, in welcher alle Leistungszahlen der Bearbeiter nach Dienstleistungskategorien gruppiert dargestellt werden:

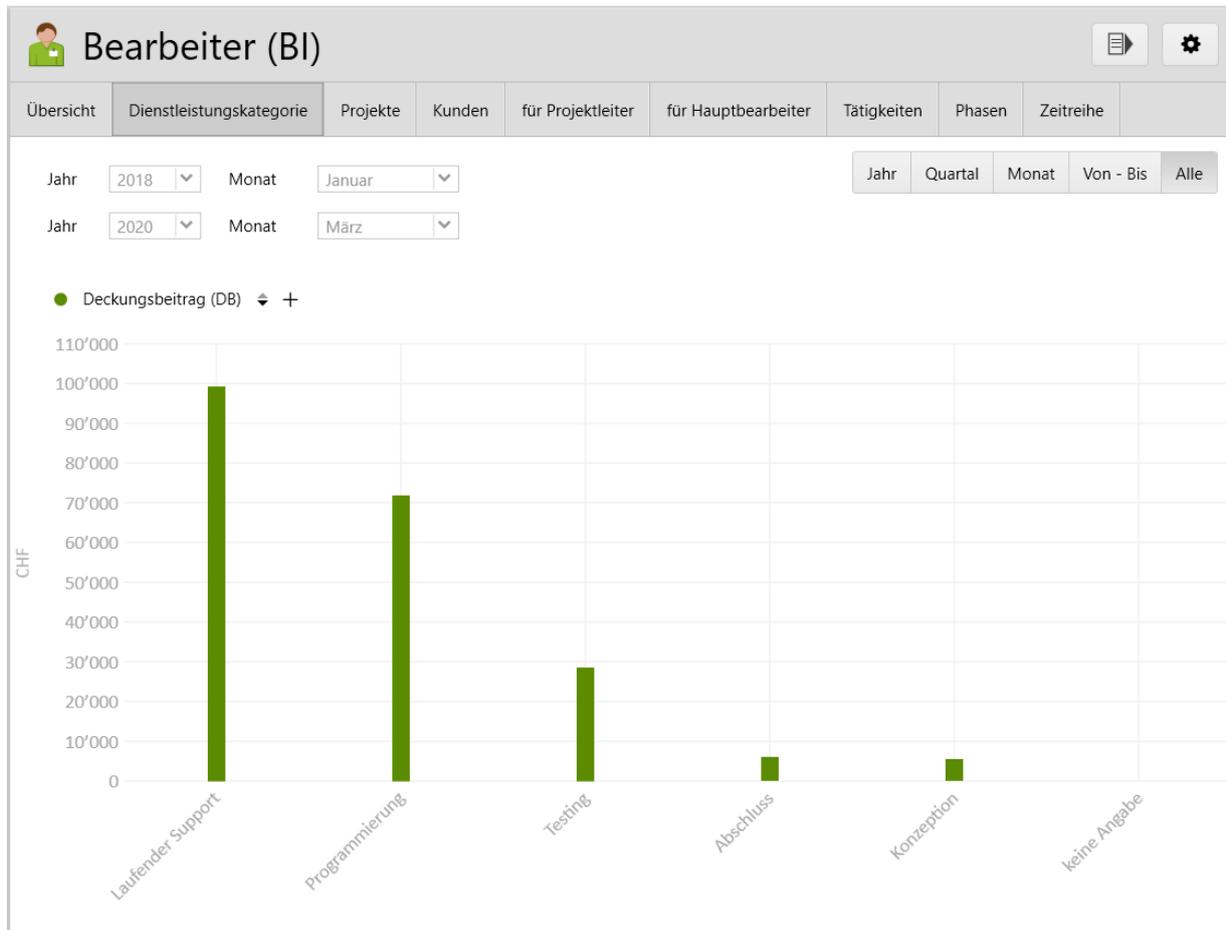


Abbildung 19: Bearbeiterwerte gruppiert nach Dienstleistungskategorien¹

Wie Sie sehen, ist das neue BI Modul hier mächtiger als das Benchmarking:

- Jede berechnete Klasse kann projiziert werden, nicht mehr nur Projektphasen.
- Die "Projektion" steht automatisch auf allen Objekten der selektierten Klasse zur Verfügung, also in unserem Beispiel nicht nur auf dem Ordner "Bearbeiter", sondern auf allen Bearbeiter-Listen und einzelnen Bearbeitern.
- Alle definierten Kennzahlen (siehe 4.3) stehen automatisch auch auf der neuen Auswertung zur Verfügung.

4.9 Inbetriebnahme

Wenn Sie Vertec 6.4 das erste Mal aufstarten, sind die BI Daten noch nicht berechnet (siehe Kapitel 4.6). Das bedeutet, Sie sehen in der BI Ansicht keine Zahlen.

Die Daten werden automatisch in der Nacht berechnet. Sie können also einfach auf den nächsten Tag warten und die Auswertungen dann einsehen.

Eine andere Variante ist, die Berechnung manuell einmal anzustossen. Wie das geht, ist im Abschnitt **Berechnung manuell anstossen** im Kapitel 4.6 beschrieben. Seien Sie sich dabei bitte bewusst, dass die Berechnung bei grossen Datenmengen viele Ressourcen brauchen wird und Vertec unter Umständen während dieser Zeit für den normalen Betrieb blockiert ist.

¹ Hinweis: Der Abbildung 19 liegen andere Daten zugrunde als der Abbildung 16, die Zahlen können nicht verglichen werden.

5 Leistungs- und Spesenerfassung

5.1 PDF Dokumente als Spesenbelege

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Alle | Version: 6.4

Neu können auch PDF Belege als Spesenbilder geladen werden. Der Vorgang ist gleich wie beim Laden der Bilder: Klicken Sie mit der rechten Maustaste ins Feld auf der Seite **Belegbild** und wählen Sie **Bild laden...**

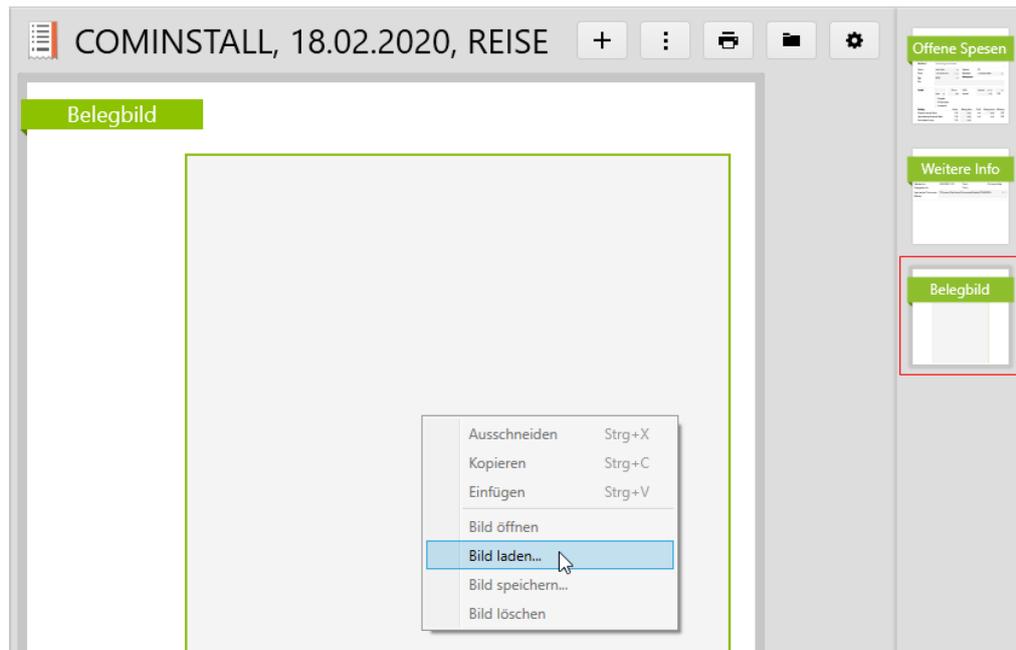


Abbildung 20: Einen Spesenbeleg laden

Neu kann neben Grafikdateien auch ein PDF Dokument ausgewählt werden:

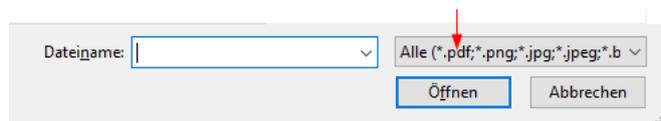


Abbildung 21: Neu können auch PDFs ausgewählt werden

Im Feld wird eine Vorschau der ersten Seite angezeigt. Durch erneuten Rechtsklick auf das Bild kann das Dokument auch geöffnet werden.

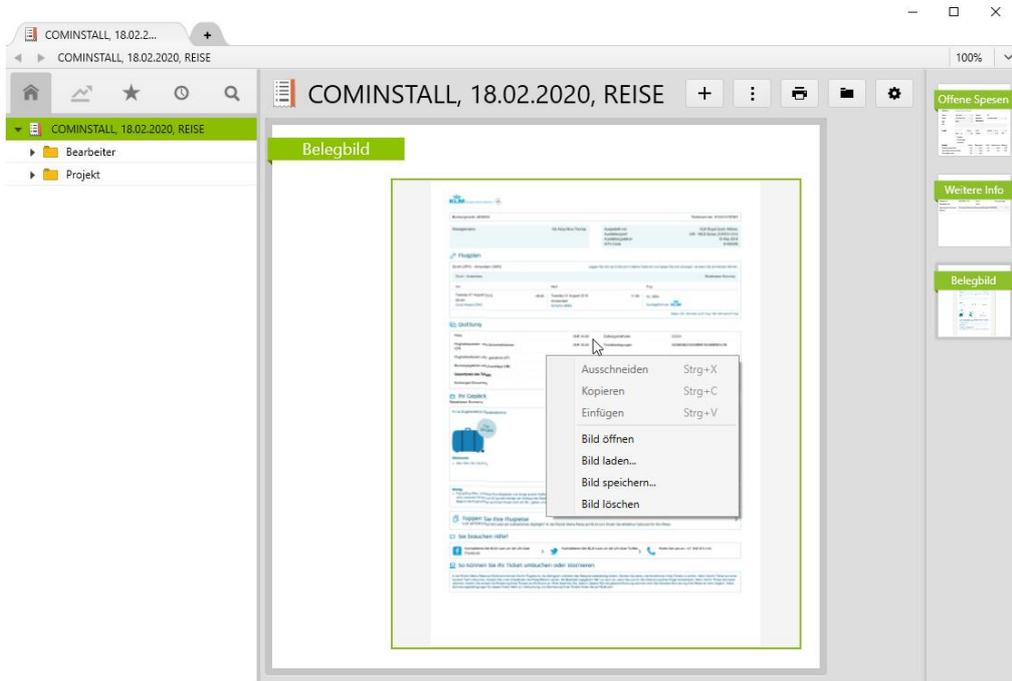


Abbildung 22: Ein PDF ist als Spesenbild hinterlegt

Diese Funktionalität ist wie folgt verfügbar:

- PDF Belege up- und downloaden in Desktop-, Cloud- und Web App
- Vorschau der ersten Seite anzeigen in Desktop-, Cloud-, Web- und Phone App
- Verwendung der Vorschau als Belegbild in erweiterten Word-Berichten
- Das Erfassen von PDF Belegen in der Phone App folgt in einer Version nach 6.4.

5.2 Direktes Setzen von Projekt in Suchdialog (edtProjekt)

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.3.0.4

Der Projekt-Suchdialog in der Leistungserfassung (konfigurierbar als Steuerelement **edtProjekt** in der Spalteneinstellung) wurde so erweitert, dass bei Eingabe eines eindeutigen Projekt-Codes das Projekt direkt gewählt wird, ohne Erscheinen des Such-Dialogs.

Sobald man den Code eingegeben hat und TAB klickt, wird das Projekt eingetragen.

	Timer	Projekt	Phase	Datum	Mitarbeiter	Tätigkeit	Text
	▶	AZZ1	BERATUNG	06.01.2020	Christoph Keller	BM	Frau Meier
*		azz1 → × ...					

Abbildung 23: Der eindeutige Code wird bei TAB direkt übernommen

Dies kann besonders bei Kunden mit vielen Projekten zu einer Performance-Verbesserung führen.

5.3 Berechnung und Darstellung von Ferienwerten

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.3.0.17

An der Darstellung und der Berechnung von Ferienwerten wurden folgende Anpassungen vorgenommen:

- Lag das Eintrittsdatum eines Mitarbeiters in der aktuellen Ferienperiode (normalerweise aktuelles Jahr), dann wurde ein Ferienvortrag nicht korrekt berücksichtigt (Vortrag immer 0). Vorträge werden neu unabhängig vom Eintrittsdatum berücksichtigt.
- Im Modus, wo ein Vortrag innerhalb einer Ferienperiode erfasst wird (nicht auf den 1. Tag der Ferienperiode), ersetzt dieser die Berechnung des Ferienguthabens, weil er einfach den Saldo zu einem bestimmten Zeitpunkt definiert. Die Darstellung zeigt dann unter Ferienguthaben nicht mehr 0.00 an, sondern nichts (es ändert nur die Darstellung, nicht die Berechnung).
- Beim Ferienguthaben wird neu als Enddatum das Austrittsdatum angezeigt.
- Die Darstellung und Zahlen der Ferientabellen auf **Vorgaben** und **Übersicht Leistungen** verhalten sich nun gleich.

Weitere Informationen Weitere Informationen zu diesem Thema finden Sie in der Online Knowledge Base unter www.vertec.com/kb/faq_sollzeit/#ferienberechnung.

6 Berichte

6.1 Rechnungen nach ZUGFeRD 2.0 Standard (XRechnung)

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.3.0.12

Mit Vertec können ab Version 6.3.0.12 Rechnungen nach ZUGFeRD 2.0 Standard erstellt werden. Dafür werden im Code des erweiterten Office-Berichts die XML-Metadaten generiert und dem Reporting-Mechanismus übergeben, welcher die Metadaten ins PDF integriert.

Dafür gibt es in den Erweiterten Office-Berichten eine Methode

```
metadata_zugferd(context)
```

Diese gibt ein Tuple zurück bestehend aus:

- **profilename** : Name des Zugferd Profiles. Hier muss es sich entweder um den Namen eines "eingebauten" Profils handeln oder es muss als 3. Argument eine Schema-Definition zurückgegeben werden. Ansonsten meldet der Bericht beim Ausführen einen Fehler.
- **data** : ZUGFeRD XML Daten. Das XML Dokument welches die ZUGFeRD Daten zur Rechnung darstellt.
- **schema**: optional XMP Metadatenstruktur. Für nicht eingebaute Profile kann hier optional ein XML Datenstring übergeben werden, welcher das Schema des zu verwendenden Profils definiert.

Als eingebaute Profile werden die aktuellen ZUGFeRD 2.0 Profile unterstützt:

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

Die mitgelieferte Standardbericht **Rechnung mit Leistungsliste** enthält einen entsprechenden Beispielcode für das Profil **EN 16931**. Kunden, die ihre Rechnungen nach ZUGFeRD 2.0 Standard oder als X-Rechnungen verwenden wollen, müssen diesen Code entsprechend anpassen.

Weitere Informationen Weitere Informationen zum Thema Vertec und ZUGFeRD finden Sie in der Online Knowledge Base unter www.vertec.com/kb/zugferd.

6.2 Rechnung mit Bearbeitersummen als erweiterter Office-Bericht

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.3.0.17

Der Standard-Bericht **Rechnung mit Bearbeitersummen** wird neu als erweiterter Office-Bericht mitgeliefert.

Bei neuen Vertec Installationen ist nur noch diese Variante des Berichts vorhanden und aktiv registriert.

Bei bestehenden Installationen wird der erweiterte Office-Bericht mitgeliefert und inaktiv registriert. Falls Sie ihn verwenden möchten, können Sie ihn in den Berichtsvorlagen aktivieren.

6.3 Rechnung mit Phasensummen als erweiterter Office-Bericht

Linie: Standard, Expert | Module: Budget & Teilprojekt | Apps: Full featured | Version: 6.3.0.17

Der Standard-Bericht **Rechnung mit Phasensummen** wird neu als erweiterter Office-Bericht mitgeliefert.

Bei neuen Vertec Installationen ist nur noch diese Variante des Berichts vorhanden und aktiv registriert.

Bei bestehenden Installationen wird der erweiterte Office-Bericht mitgeliefert und inaktiv registriert. Falls Sie ihn verwenden möchten, können Sie ihn in den Berichtsvorlagen aktivieren.

6.4 Erweiterte Rechnungen mit QR-Code an Spezifikation 2.1 angepasst

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.3.0.17

Bei allen mitgelieferten erweiterten Office-Berichte für Rechnungen wurde die Seite mit dem QR-Code an die aktuelle Spezifikation 2.1 angepasst. Dies betrifft die Berichte

- Rechnung mit Leistungsliste
- Rechnung mit Phasensummen (siehe 6.3)
- Rechnung mit Bearbeitersummen (siehe 6.2)

Rückwärtskompatibilität

Wichtiger Hinweis Bei kundenspezifischen Vorlagen muss die QR-Seite manuell angepasst werden. Eine Anleitung, wie Sie genau vorgehen müssen, finden Sie in unserer Online Knowledge Base unter www.vertec.com/kb/rechnungsvorlagen-fuer-qr-code-bereit-machen.

6.5 Vorschuss Bericht neu mit Stichdatum

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.3.0.17

Der erweiterte Office-Bericht **Vorschüsse** wurde umgebaut, mit folgenden Änderungen:

- Einführung eines Stichdatums. Voreingestellt ist der Letzte des letzten Monats. Damit sind ein paar sinngemäße Änderungen verbunden:
 - Es werden nur noch Vorschüsse angezeigt, die einen Restbetrag haben (bzw. hatten zum Stichdatum), sonst wird die Nuller-Liste immer länger.
 - Es werden nur Abzüge auf Rechnungen angezeigt, die zum Stichdatum bereits verrechnet waren (man hat sonst keine Chance, herauszufinden, wann der Vorschuss auf die Rechnung kam). Dadurch entfällt auch die Spalte "Verrechnet".
- Das Stichdatum wird auf dem Bericht als Untertitel ausgewiesen.
- Die Listen sind gruppiert nach Währung.

6.6 Report Ferienabrechnung zeigt Vortrag immer an

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.3.0.13

Beim Erweiterten Office-Bericht **Ferienabrechnung** wurde bisher nur ein Vortrag angezeigt, wenn es einen manuell eingetragenen Vortrag gab.

Neu wird der Vortrag vom Anfang der ausgewerteten Periode immer angezeigt:

Ferienabrechnung

Paul Thomson | Eintritt 01.01.2006

Ferienbezug 01.01.2018 - 31.12.2019

Jahr	Total	Januar	Februar	März	April	Mai
2018	119:45	0:00	0:00	0:00	0:00	28:00
2019	0:00	0:00	0:00	0:00	0:00	0:00
Total	119:45					

Ferienvortrag 01.01.2018	75:08
Ferienguthaben 01.01.2018 - 31.12.2019	110:05
Ferienbezug 01.01.2018 - 31.12.2019	119:45
Feriensaldo per 31.12.2019	65:28

- Der Vortrag berechnet sich mit `->getFerienvortrag(startDatum)`.
- Das Ferienguthaben wird berechnet: `Saldo + Ferienbezug - Vortrag`.

6.7 Rekapitulation Buchungen nach Währung gruppiert

Linie: Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.3.0.17

Beim erweiterten Office-Bericht **Rekapitulation Buchungen** sind die Buchungen neu nach Währung gruppiert und das Total mit Vermerk der Währung ausgegeben.

6.8 Performance Verbesserung bei Fakturavorschlag

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.3.0.17

Der erweiterte Office-Bericht Fakturavorschlag wurde aus Performancegründen komplett umgebaut, mit folgenden Änderungen:

- Es gibt nur noch ein Bis-Datum, kein Von-Datum mehr. Wird kein Bis-Datum eingegeben, erscheint eine Meldung, und die Ausführung wird abgebrochen.
- Pauschalphasen ohne Planwert werden nicht mehr berücksichtigt.
- Die jeweilige Projektwährung wird beim Projektitel angedruckt.

6.9 Performance Verbesserung bei Budget Übersicht

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.3.0.17

Der erweiterte Office-Bericht **Budget Übersicht** wurde aus Performancegründen komplett umgebaut und läuft nun bei grossen Datenmengen performanter.

Zusätzlich ist er neu gruppiert nach Währung.

6.10 Expressions in Kopf- und Fusszeilen von Vertec-generierten Word-Berichten

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.3.0.4

Nun wird auch in Vertec-generierten Word-Berichten die Möglichkeit unterstützt, Expressions in Kopf- und Fusszeilen einzufügen.

Die Expressions werden wie bei den erweiterten Office-Berichten in doppelte geschweifte Klammern gesetzt und so beim Ausführen als OCL-Expressions interpretiert und mit den entsprechenden Daten befüllt.

Beispiel

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

fügt den Namen des in Vertec eingeloggten Benutzers in die Kopf- bzw. Fusszeile ein.

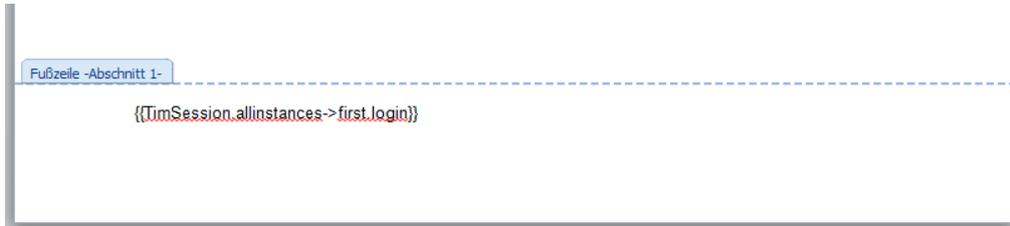


Abbildung 24: Der Name des eingeloggten Bearbeiters wird in eine Fusszeile eingefügt

6.11 Summierung über mehrere Stufen in erweiterten Excel-Berichten

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.3.0.7

Die Summierung von Zellen in Excel Berichten via Band-Summen summiert alle Zellen einer Spalte innerhalb eines Bands.

In vielen Fällen möchten man einzelne Zellen eines Bands summieren, welches durch mehrere Hierarchiestufen von der Summe entfernt ist.

Dies wurde nun ermöglicht. Neu wird bei Summierung von nur einer Zelle eine Summe über alle Kopien dieser Zellen gemacht.

Beispiel

Bei der Summierung von nur einer Zelle in einem Band wird beim Ausführen eine Summe über alle Kopien dieser Zelle gemacht:

	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	

The image shows an Excel spreadsheet with a formula bar at the top containing `=SUMME(D10)`. Red arrows point from the formula bar to cell E3 (containing '0') and cell D10 (containing 'total').

Abbildung 25: Die Summierung bezieht sich auf die Total-Zelle

E3		fx		=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	<i>Datum</i>	<i>Text</i>		<i>WertExt</i>	
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	

Abbildung 26: Nach dem Ausführen werden die beiden Zellen summiert

7 Ordner und Listeneinstellungen

7.1 Mehrstufige Sortierung in Listen

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.4

Neu können Vertec Listen nach mehr als einer Spalte sortiert werden.

Die Einstellung erfolgt in den Listeneinstellungen:

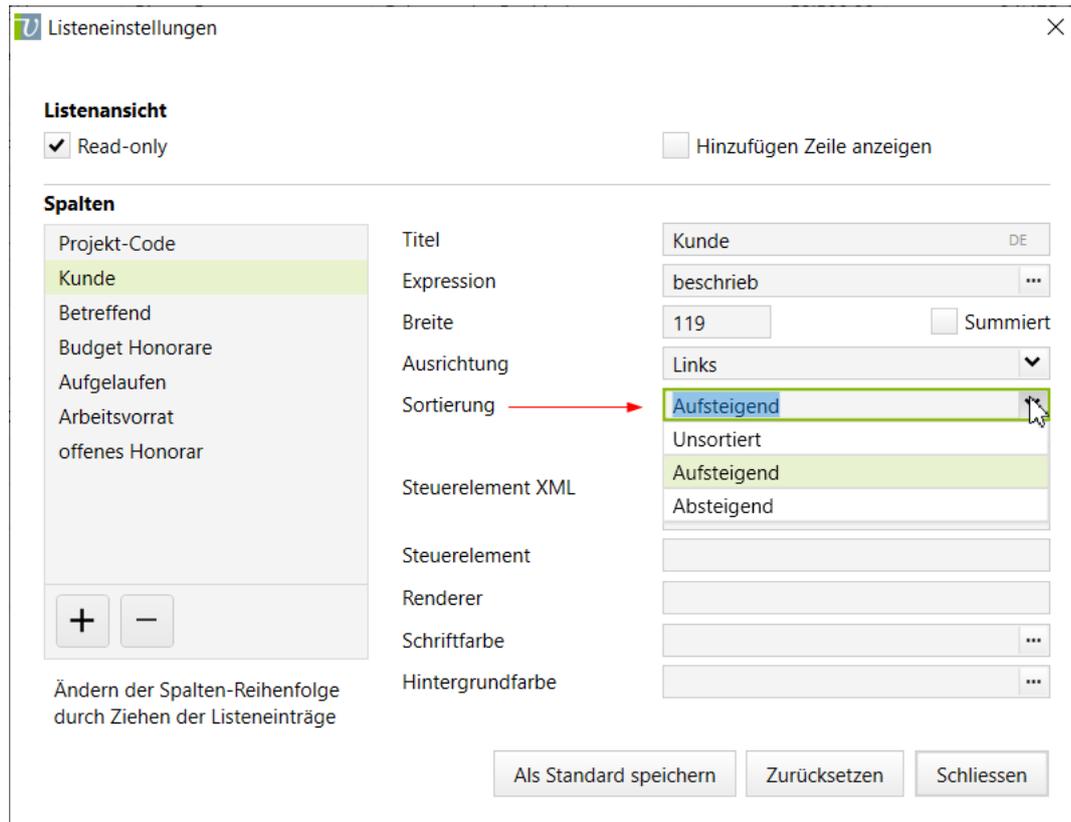


Abbildung 27: In den Listeneinstellungen wird die Sortierung der einzelnen Spalten eingestellt

- Die Sortierung erfolgt in der Reihenfolge der Spaltendefinitionen, also von links nach rechts.
- Alle an der Sortierung beteiligten Spalten werden auf der Oberfläche im Titel als sortiert dargestellt.

Eigene Projekte						
Projekt-Code	Kunde	Betreffend	Budget Honor.	Aufgelaufen	Arbeitsvorrat	
ACC-EINF	Accountit Ltd	Vertec Einführung	0.00	3'000.00	-3'000.00	
COOPER-BH	Bistro Cooper	Führung der Buchhaltungyz	52'500.00	24'475.40	28'024.60	
COMINSTALL	Comtelsat AG	Entwicklung von Software	133'000.00	79'785.55	53'214.45	
COM-SSUP	Comtelsat AG	Support XTrem-E	8'000.00	6'635.85	1'364.15	
DOIT-EINF	Do it for IT AG	Einführung Vertec	2'000.00	0.00	2'000.00	
ELIWAG-STEUER	Eliwag AG	Steuerberatung	0.00	0.00	0.00	
EUROPROJEKT	Europrojekt	Projekt in EURO	0.00	0.00	0.00	

Abbildung 28: Die Liste ist aufsteigend nach Kunde und innerhalb des Kunden absteigend nach Budget sortiert

Wird in der Liste auf einen Sortierpfeil geklickt, wird die Liste nur noch nach der gewählten Spalte sortiert. Die so veränderte Suche bleibt solange aktiv, bis die Liste wieder neu angezeigt wird. Dann ist wieder die Sortierung gemäss Listeneinstellung aktiv.

Rückwärtskompatibilität

Bestehende Listeneinstellungen ändern sich nicht.

Eine Veränderung zu vorher ergibt sich, wenn die Sortierung von Listenspalten in den Listeneinstellungen geändert wird. Bisher hat das Setzen einer Sortierung auf einer Spalte die Sortierung auf allen anderen Spalten zurückgesetzt. Das ist neu nicht mehr so. Möchte man nach einer Spalte nicht mehr sortieren, muss sie explizit auf **Unsortiert** gesetzt werden.

7.2 Auswahl-Suchfeld für Expression-Ordner Abfragen

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.4

Den bei SQL-Abfragen seit 6.3 verfügbare Feld-Typ **Auswahl** steht nun auch auf Expression-Ordnern zur Verfügung.

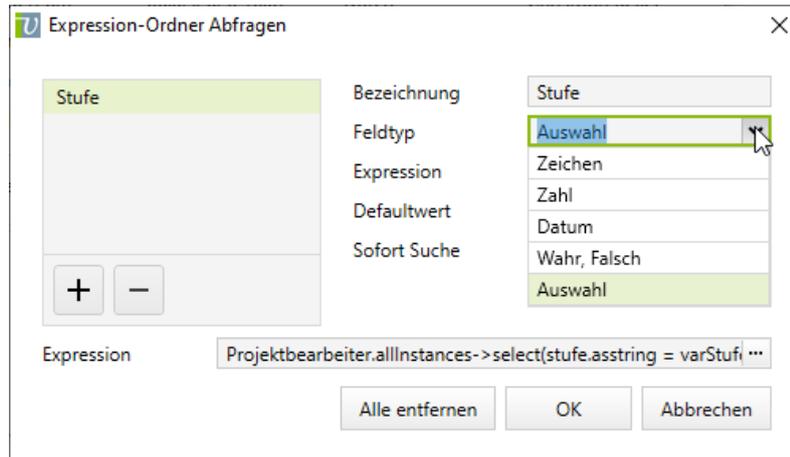


Abbildung 29: Feldtyp Auswahl

Sobald dieser Typ gesetzt ist, wird gleich darunter ein Feld für die Expression eingeblendet.

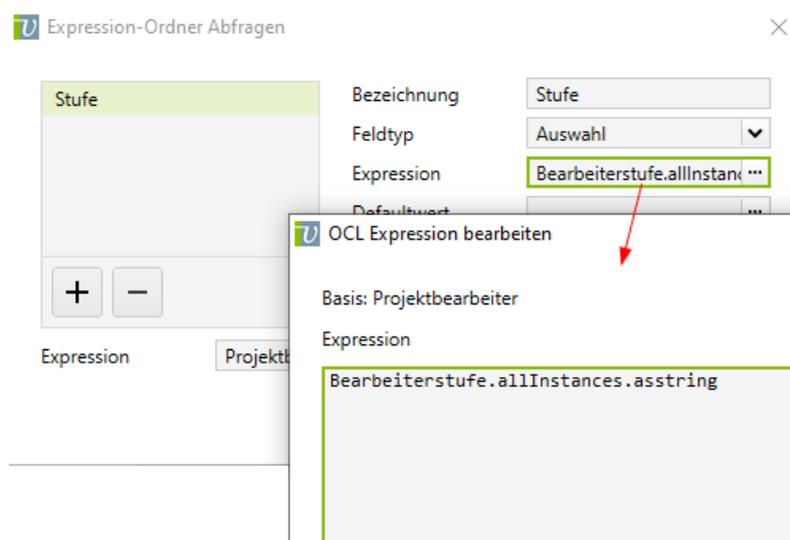


Abbildung 30: Die Expression für die Auswahl

Die Expression muss als Resultat eine Liste von Strings zurückgeben.

Auf der Oberfläche wird dann ein Auswahlfeld angezeigt, aus welchem der gewünschte Wert ausgewählt werden kann.

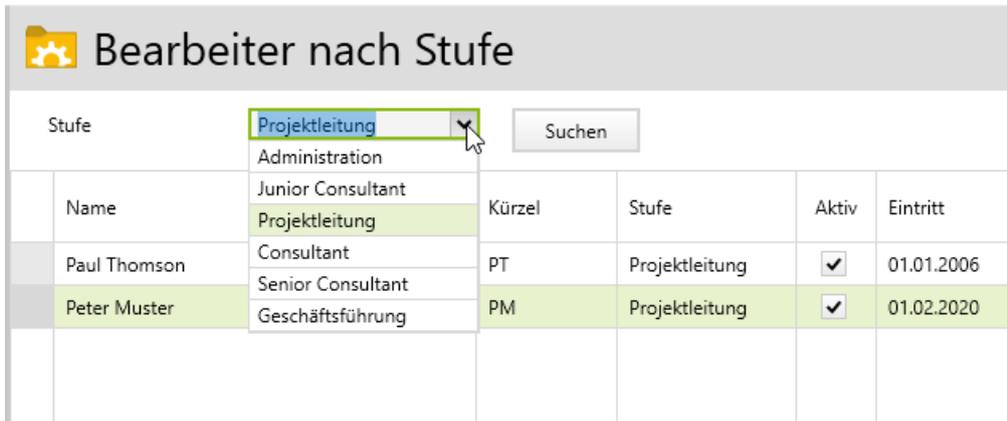


Abbildung 31: Der gewünschte Wert kann aus dem Auswahlfeld ausgewählt werden

Falls ein Defaultwert definiert ist, wird standardmässig dieser angezeigt.

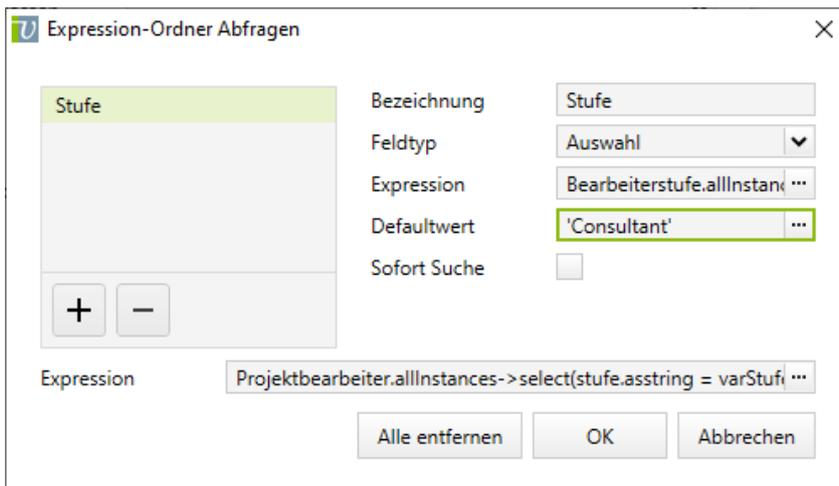


Abbildung 32: Ein Standardwert wird definiert

Falls der angegebene Defaultwert nicht in der Auswahlliste vorhanden ist oder es keinen Defaultwert hat, wird der erste Eintrag der Auswahlliste genommen.

7.3 SQL Ordner setzt "Suchdialog anzeigen" wieder standardmässig

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.3.0.4

Seit Vertec 6.2 wurde die Option **Suchdialog anzeigen** nicht mehr standardmässig gesetzt, wenn man einem SQL Ordner das erste Suchfeld hinzugefügt hat. Dies wurde nun wieder geändert.

Bei SQL Ordnern wird die Option **Suchdialog anzeigen** standardmässig gesetzt, sobald ein Abfragefeld hinzugefügt wird. Will man keinen Suchdialog anzeigen, muss man die Option manuell deaktivieren.

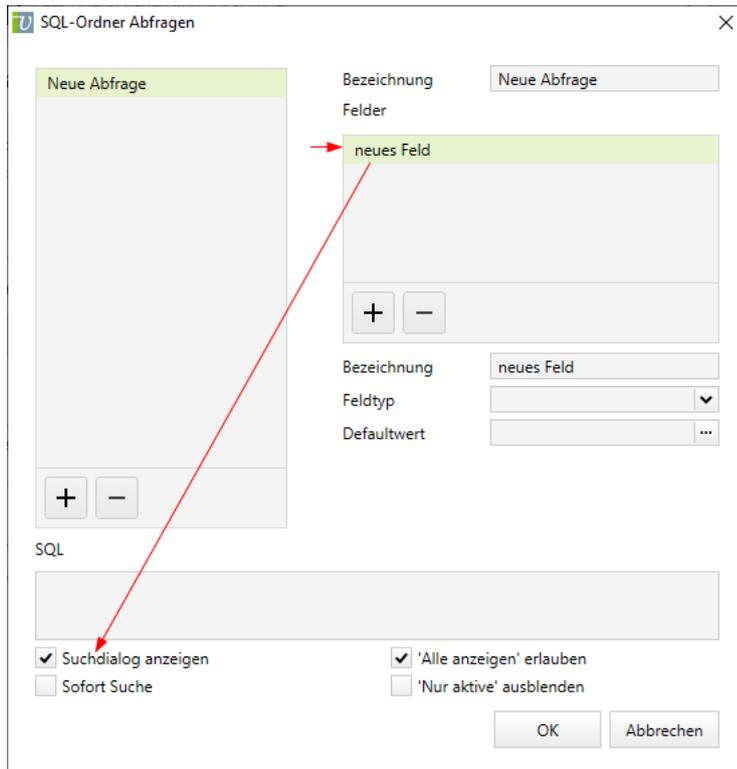


Abbildung 33: Sobald ein Suchfeld hinzugefügt wird, aktiviert sich die Option "Suchdialog anzeigen" automatisch

8 Einstellungen

8.1 Geplante Aufgaben

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.4

Die Neuberechnung der Daten für das Business Intelligence Modul erfolgt, wie in Kapitel 4.6 beschrieben, über eine geplante Aufgabe in Vertec.

Diese geplanten Aufgaben können auch für andere Aufgaben verwendet werden.

Im Ordner **Einstellungen > Geplante Aufgaben** können neue geplante Aufgaben erzeugt werden.



Geplante Aufgabe

Typ: Script ausführen

Bezeichnung: Projektstatusbericht erstellen Aktiv

Wiederholungsmodus:

 Täglich

 Wöchentlich

 CRON-Ausdruck

Zeit in UTC: 23:00

Freitag

Zeit in UTC: 23:00

CRON-Ausdruck: 0 23 * * 5

Benutzer: Administrator

Funktionsreferenz: PythonModule.projektstatusbericht

Status: Keiner

Nächste Ausführung: UTC: 20.03.2020 23:00:00 local: 21.03.2020 00:00:00

Letzter Start:

Letzter Abschluss:

Letzte Ausgabe:

Abbildung 34: Eine geplante Aufgabe

Typ	Neben dem Typ BI Kennzahlen neu berechnen (siehe Kapitel 4.6) gibt es auch den Typ Script ausführen . Damit kann ein in Vertec registriertes Script zu einem definierten Zeitpunkt ausgeführt werden.
Bezeichnung	Bezeichnung des Scripts
Aktiv	Es werden nur aktive geplante Aufgaben durchgeführt
Wiederholungsmodus	Hier kann zwischen Täglich und Wöchentlich gewählt werden. Die Zeit wird jeweils in der koordinierten Weltzeit UTC angegeben. Alternativ dazu kann ein CRON-Ausdruck angegeben werden. Informationen

über den Aufbau von CRON-Ausdrücken finden Sie im Internet, zum Beispiel auf Wikipedia: <https://de.wikipedia.org/wiki/Cron>.

Im Hintergrund führt jeder dieser Angaben zu einem CRON-Ausdruck, mit welchem der Zeitpunkt der nächsten Ausführung ermittelt wird.

Im Feld **Nächste Ausführung** weiter unten können Sie die Ihre Eingaben kontrollieren, dort wird jeweils der Zeitpunkt der nächsten Ausführung sowohl in UTC als auch in lokaler Uhrzeit angezeigt.

Benutzer	Die geplante Aufgabe wird unter diesem Benutzer durchgeführt. Es muss hier also ein Benutzer hinterlegt werden, welcher über genügend Rechte verfügt, um dieses Script auszuführen.
Funktionsreferenz	Als Funktionsreferenz wird der Name des Scripteintrags, gefolgt von der innerhalb dieses Scripts auszuführenden Funktion, durch Punkt abgetrennt, angegeben. Es muss sich um eine Funktion ohne Argumente handeln. <Scripteintrag>.<Funktionsname> Einmal eingegeben, kann der Code über den Button mit den drei Punkten in einem Popup angezeigt werden. Über einen Button In Script Editor öffnen im Popup öffnet sich der Scripteintrag direkt im Script Editor und kann dort bearbeitet werden.
Status	Hier wird angegeben, ob die Durchführung erfolgreich war oder fehlgeschlagen ist. Ist die Durchführung fehlgeschlagen, wird im Feld Letzte Ausgabe weiter unten die Fehlermeldung angezeigt.
Nächste Ausführung	Hier wird jeweils der Zeitpunkt der nächsten Ausführung sowohl in UTC als auch in lokaler Uhrzeit angezeigt. Die Ermittlung der nächsten Ausführung erfolgt aufgrund des CRON-Ausdrucks (siehe Feld Wiederholungsmodus weiter oben) und der aktuellen Uhrzeit. Läuft jedoch der Cloud-Server zu diesem Zeitpunkt nicht, kann die Ausführung nicht durchgeführt werden. Sobald der Cloud-Server wieder läuft, ermittelt dieser die nächste Ausführung. Vergangene "verpasste" Ausführungen werden beim Start des Cloud-Servers nicht nachgeholt, um zu verhindern, dass plötzlich viele Tasks gleichzeitig ausgeführt werden.
Letzter Start	Startzeitpunkt der letzten Ausführung.
Letzter Abschluss	Endzeitpunkt der letzten Ausführung.
Letzte Ausgabe	Output der letzten Ausführung. Ist diese fehlgeschlagen, wird hier die Fehlermeldung angezeigt.

Direkt Ausführen Im Kontextmenü der geplanten Aufgabe gibt es einen Menüpunkt **Jetzt ausführen**. Hier ist es wichtig zu wissen, dass dieser Vorgang nicht die geplante Aufgabe auf den jetzigen Zeitpunkt setzt, sondern lediglich das Script ausführt – genau gleich, wie wenn das Script bzw. die aufgerufene Methode im Script Editor ausgeführt würde, und zwar unter dem eingeloggten Benutzer und nicht unter dem Benutzer, der in der geplanten Aufgabe hinterlegt ist.

Parameter im Vertec.ini-File

Betrieb: On Premises

Es gibt im Vertec.ini-File im Sektor [CloudServer] folgende neue Parameter:

Task Scheduler	<p>Steuert, ob das Task Scheduling aktiv ist oder nicht. Falls nicht, werden keine geplanten Aufgaben ausgeführt.</p> <p>Ist der Parameter nicht angegeben, gilt der Standard =True. Möchte man also mit einer CloudServer-Section keine geplanten Aufgaben durchführen, kann das ausgeschaltet werden mit</p> <pre>Task Scheduler = False</pre> <p>Beim Betrieb von mehreren Cloud Servern darf nur einer davon die Aufgabe übernehmen, Tasks laufen zu lassen. Bei allen anderen muss dieser Parameter auf False gestellt werden.</p>
Task Scheduler Polling Minutes	<p>Steuert, alle wie viel Minuten auf ausstehende geplante Aufgaben überprüft wird.</p> <p>Natürliche Zahl, Standard ist 1.</p>
Task Scheduler Concurrent Tasks	<p>Steuert, wie viele geplante Aufgaben maximal parallel ausgeführt werden können.</p> <p>Natürliche Zahl, Standard ist 1.</p>
BI Task Scheduler Range Start BI Task Scheduler Range End	<p>Damit kann das Zeitfenster, in welchem die BI-Daten über Nacht berechnet werden, definiert werden. Innerhalb dieses Zeitfensters wird vom CloudServer eine Zeit gewürfelt, in welcher der Prozess angestossen wird.</p> <p>Diese Parameter sind vor allem für das Vertec Cloud Abo von Bedeutung (siehe Abschnitt 4.6), können natürlich aber auch sonst verwendet werden.</p> <pre>BI Task Scheduler Range Start=23:00 BI Task Scheduler Range End=05:00</pre> <p>So wird gewürfelt: Ausgehend von der Startzeit werden die Anzahl Minuten bis zur Endzeit ermittelt. Dann wird eine Zufallszahl von 0 bis zur Anzahl Minuten gewürfelt und dieser Wert der Startzeit hinzugezählt. Das ergibt den CRON-Ausdruck für die Ermittlung der nächsten Ausführung (siehe dazu die Beschreibung des Feldes Nächste Ausführung weiter oben).</p>

Die Applikation, die geplante Aufgaben ausführt, heisst **Vertec.TaskRunner.exe** und befindet sich standardmässig im Vertec Installationsverzeichnis.

8.2 2 Faktor Authentisierung

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Cloud Clients | Version: 6.3.0.12

Ab Version 6.3.0.12 unterstützt Vertec für das Login in Cloud Clients (Cloud App, Web App, Phone App) einen 2. Faktor via Authenticator App, z.B. Google Authenticator.

Bei der Authenticator App muss es sich um eine "Soft-Token" App handeln, welche ein zeitbasiertes One-time-password generiert. Bei diesem Prinzip ist keine Kommunikation zwischen der Authenticator App und Vertec notwendig, es muss nur einmal ein gemeinsames Secret ausgetauscht werden. Aufgrund dieses Secrets können die Authenticator App und Vertec beide unabhängig dieselben Codes generieren und vergleichen.

Voraussetzung ist, dass die Zeit auf dem Mobilgerät und auf dem Vertec Server übereinstimmt.

Einrichten in Vertec

Unter Systemeinstellungen > Authentisierung gibt es eine Einstellung **Zwei Faktor Authentisierung erforderlich**:



Abbildung 35 Aktivierung der Zwei Faktor Authentisierung in den Vertec Systemeinstellungen

Ist diese Einstellung aktiviert, geschieht folgendes:

- Meldet sich der Bearbeiter zum ersten Mal nach der Aktivierung der Systemeinstellung mit Username und Passwort an Vertec an, erhält er einen Dialog zur Einrichtung der Zwei Faktor Authentisierung (2FA). Diese Seite kann er nur abrechnen oder geordnet verlassen. In beiden Fällen wird der Benutzer direkt wieder ausgeloggt und muss sich anschliessend neu anmelden.
- Bei Abbruch wird das Secret nicht gespeichert. Beim nächsten Anmeldeversuch erscheint wieder der Einrichten-Dialog.
- Bei Speichern wird das Secret gespeichert. Beim nächsten Anmeldeversuch muss er den entsprechenden Authenticator Pin angeben.
- Andere Aktionen in Vertec sind nicht möglich, bis 2FA für den Benutzer eingerichtet wurde und sich dieser neu anmeldet.

Der Einrichten-Dialog sieht wie folgt aus:

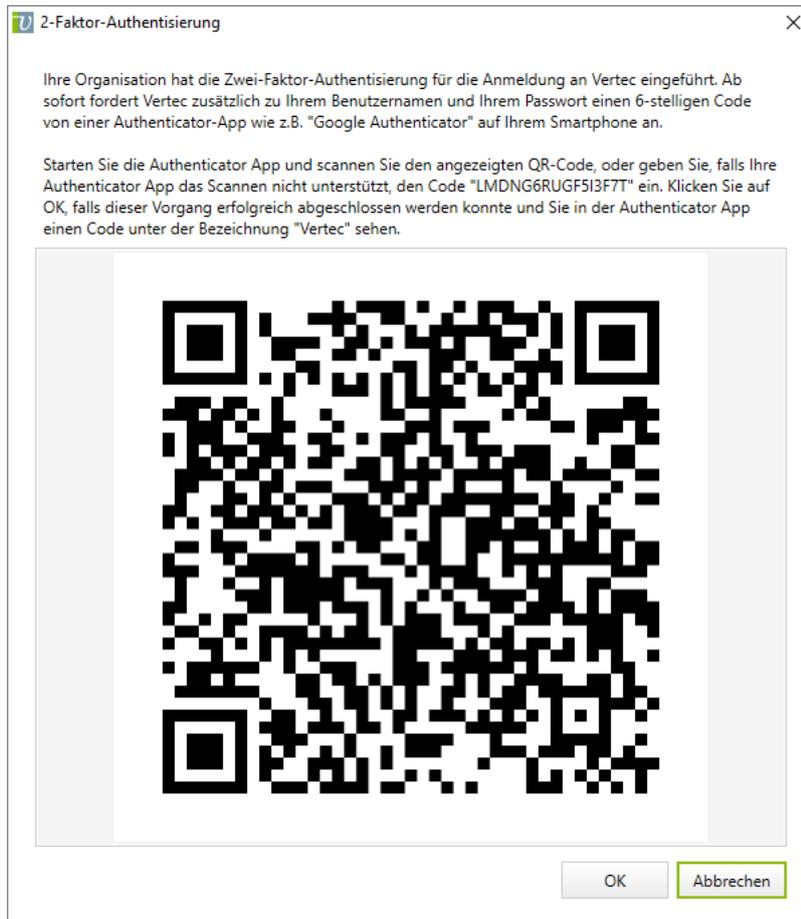


Abbildung 36: Dialog zum Einrichten der Zwei Faktor Authentisierung

Starten Sie nun Ihre Authenticator App und scannen Sie den angezeigten Code bzw. geben Sie den angegebenen Code darin ein. Sobald in Ihrer App ein Code mit der Bezeichnung "Vertec" erscheint, können Sie auf **OK** klicken. Ab diesem Zeitpunkt sind die beiden Systeme gekoppelt. Ab dann geben Sie beim Login in Vertec den von der Authenticator App generierten Code als zweiten Faktor ein.

Wichtiger Hinweis Die Zwei Faktor Authentisierung greift auch in der Phone App, die Einrichtung der Authentisierung wie hier beschrieben muss jedoch in der Cloud App oder der Web App erfolgen. Nur dort erscheint der Einrichten Dialog.

Jeder Benutzer **muss sich also zuerst an einer Cloud App oder Web App anmelden** und die Zwei Faktor Authentisierung einrichten, bevor er die Phone App mit der Zwei Faktor Authentisierung verwenden kann.

Ein Secret zurücksetzen

Bei Notfällen wie z.B. Handy- oder Datenverlust kann der Administrator das Secret eines Bearbeiters zurücksetzen. Dafür kann sich der Administrator anmelden und auf dem entsprechenden Bearbeiter über Kontextmenü oder Menü Aktionen > **2. Faktor für Authentisierung bearbeiten** den Dialog aufrufen und auf den Button **Token löschen** klicken.

8.3 Native Sprache in MLString Attributen

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.4

An verschiedenen Orten in Vertec kommen sogenannte Multi-Language-Strings (MLStrings) zum Einsatz:

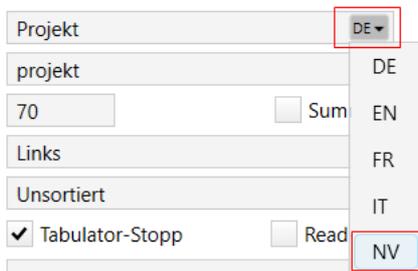
- Titel in den Listeneinstellungen
- Texte bei Tätigkeiten, Spesen- und Auslagetypen
- Bezeichnung von Zahlungskonditionen
- Bezeichnung von Zusatzfeldern
- Bezeichnung von Abwesenheitstypen
- Im neuen Business Intelligence Modul (siehe Kapitel 4)

Dabei kann pro Sprache ein Begriff eingegeben werden, welcher auf der Oberfläche oder bei fremdsprachigen Kunden z.B. in Ausdrucken angezeigt werden.

Das war bisher ein Problem, wenn die gleiche Sprache dynamisch in den Jargon übersetzt werden sollte (Projekt – Mandat); hier musste bisher fix der jeweilige Begriff eingegeben werden.

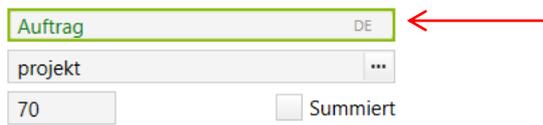
Neu gibt es nun zusätzlich eine "native" Sprache, in welcher der zu übersetzende Begriff eingegeben werden kann. Dieser wird vom Übersetzungssystem automatisch in die jeweils massgebliche Sprache im jeweiligen Jargon übersetzt.

Zum Beispiel bei den Listeneinstellungen:



Dabei gilt:

- Hat der MLString bei der Darstellung auf der Oberfläche keinen Text in der aktuellen Sprache hinterlegt, dann wird der native Text verwendet und via Übersetzungssystem übersetzt. In diesem Fall wird der Begriff in schwarzer Schrift angezeigt.
- Hat der MLString bei der Darstellung auf der Oberfläche einen Text in der aktuellen Sprache hinterlegt, dann wird dieser Text verwendet und der Begriff in grüner Schrift angezeigt:



Durch Löschen des Begriffs in der aktuellen Sprache wird wieder der übersetzte Begriff (in schwarzer Schrift) angezeigt.

- Wird der native Begriff vom Übersetzungssystem nicht gefunden, wird er unübersetzt in allen Sprachen angezeigt, in denen es keine Überschreibung hat.
- Ist weder ein nativer Begriff noch eine Überschreibung der aktuellen Sprache vorhanden, ist der Text leer.
- Überschriebene Begriffe in der aktuellen Sprache haben immer Vorrang gegenüber den übersetzten Begriffen.

Rückwärtskompatibilität

Beim ersten Aufstarten nach dem Konvert (siehe 1.4) prüft Vertec die bestehenden MLString-Attribute. Hat ein Attribut nur einen EN oder einen DE Text definiert, wird dieser Begriff als native Sprache, also als NV, eingetragen:

- Falls ein EN Text definiert ist, wird dieser als NV Text gesetzt
- Ist kein EN Text, aber ein DE Text definiert, wird dieser als NV Text gesetzt
- Ist weder ein EN noch ein DE Text definiert, wird kein NV Text gesetzt

Da die bereits gesetzten Texte nicht verändert und bei der Darstellung Vorrang haben gegenüber den nativen Begriffen, ändert sich bei bestehenden Installationen dadurch nichts. Die Anpassung wäre erst spürbar, wenn später auf eine andere Oberflächensprache umgestellt würde, welche vorher noch keinen Wert gesetzt hatte.

8.4 Veränderung bei den Shortcuts der Navigationsansichten

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.4

Aufgrund des neuen BI Moduls (siehe Kapitel 4) ergibt sich eine Veränderung in den Shortcuts für die Navigationsansichten.

Die Standardansicht ist neu über das Tastaturkürzel **F8** (statt wie bisher F9) zu erreichen.

Die neue BI Ansicht öffnet sich über das Tastaturkürzel **F9**.

Neu



F8	Standardansicht
F9	BI Ansicht
F10	Favoriten
F11	Verlauf
F6 / F12	Globale Suche

Bisher



F9	Standardansicht
F10	Favoriten
F11	Verlauf
F6 / F12	Globale Suche

8.5 Sprache auswählen Dialog per Systemeinstellung ausblenden

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.3.0.12

In den **Systemeinstellungen Allgemein** gibt es eine neue Einstellung namens **Sprachauswahl anzeigen**: Sie bewirkt, dass der Menüpunkt **Sprache ändern...** im Menü **Einstellungen** angezeigt und die Vertec Programmsprache vom Benutzer angepasst werden kann.

PropertyName: **AllowLanguageSelection**. BooleanProperty. Standardeinstellung: Ja.

9 Phone App Erweiterungen

9.1 Sprache gemäss Einstellungen des Vertec-Users

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Phone | Version: 6.3.0.12

Die Phone App richtet sich nicht mehr nach der Sprache in den Geräte-Einstellungen, sondern nach der für den Vertec-User eingestellten Sprache.

9.2 Umbau Wochenansicht

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Phone | Version: 6.3.0.15

Die Menüauswahl für Leistungen, Spesen, Aktivitäten und neu Abwesenheiten wird nun in einer Zeile am unteren Bildschirm-Rand dargestellt:



Abbildung 37: Neue Menüführung in der Phone App

Das Hauptmenü der Phone App umfasst nur noch Optionen für:

- Wochenansicht
- Aktivitäten pendent/offen
- Adressen
- Einstellungen

9.3 Abwesenheiten erfassen

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Phone | Version: 6.3.0.15

Abwesenheiten können nun direkt in der Phone App erfasst werden.

Ein Klick auf den grünen + Button erzeugt eine neue Abwesenheit.

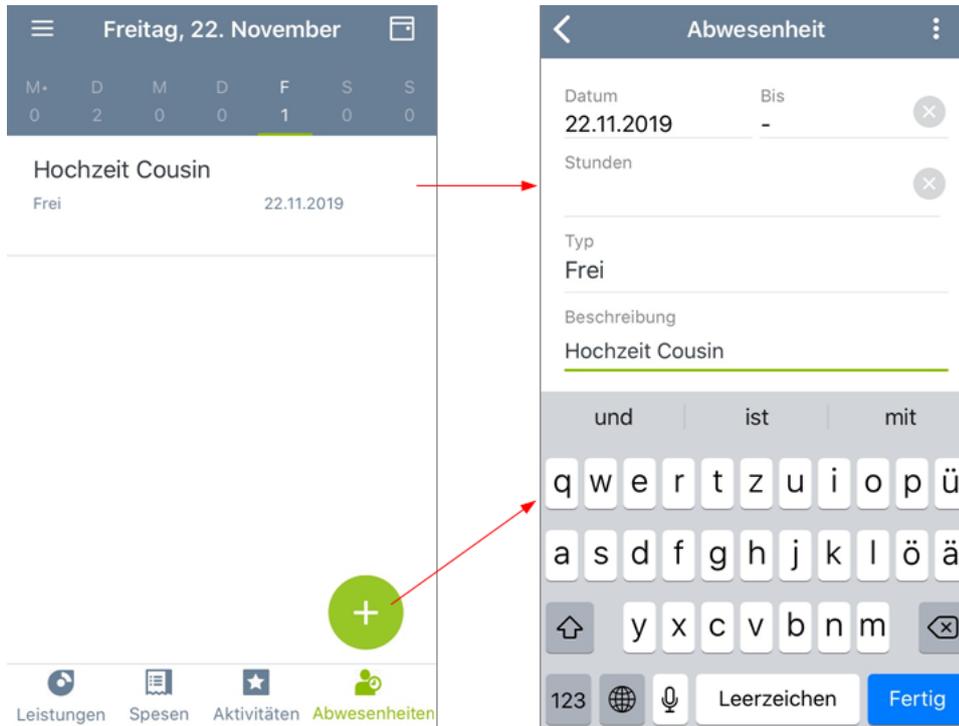


Abbildung 38: Erfassung einer Abwesenheit in der Phone App

Ein Klick auf den Typ bringt die Liste der in Vertec hinterlegten Abwesenheitstypen zur Auswahl.

Für den Tag erfasste Abwesenheiten werden in den Tagesdetails (siehe 9.4) angezeigt.

9.4 Tagesdetails

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Phone | Version: 6.3.0.15

Für jeden Tag können folgende Tagesdetails ein- und ausgeklappt werden:

M*	D	M	D	F	S	S
7:00	0:00	0:00	0:00	–	0:00	0:00
Tagesdetails ▲ ←						
Sollzeit					8:00	
Abwesenheitsgrund: Arztbesuch						
Summe Leistungen					7:00	
Päsenzzeit					7:30 	
Differenz					-0:30	

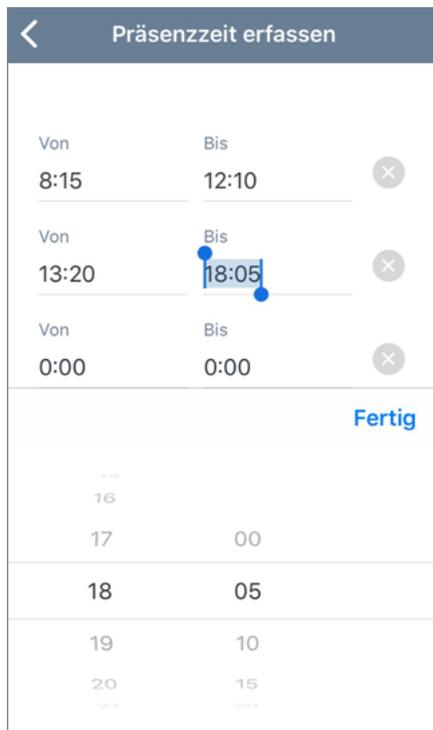
- Sollzeit. Diese wird nicht angezeigt, wenn in Vertec die Systemeinstellung Projekt/Mandat > **Sollzeit Übersicht für Leistungserfassung anzeigen** auf **Nein** gestellt ist oder der Benutzer keinen Überzeitsaldo hat.
- Abwesenheiten, falls für diesen Tag vorhanden (siehe 9.3).
- Summe der Leistungen des Tages

- Präsenzzeit (Summe). Die Präsenzzeiten können auch direkt hier erfasst werden, indem auf das Stift-Symbol geklickt wird (siehe 9.5). Beides wird nur angezeigt, wenn die Systemeinstellung Projekt/Mandat > **Präsenzzeitenliste für Leistungserfassung anzeigen** auf **Ja** gestellt ist.
- Differenz (zwischen Leistungen und Präsenzzeit)

9.5 Präsenzzeiten erfassen

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Phone | Version: 6.3.0.15

Über einen Klick auf das Stift-Symbol hinter den Präsenzzeiten in den Tagesdetails öffnet sich die Präsenzzeiterfassung:



Von	Bis	
8:15	12:10	✕
13:20	18:05	✕
0:00	0:00	✕

Fertig

16	
17	00
18	05
19	10
20	15

Abbildung 39: Präsenzzeiten des Tages erfassen

10 Schnittstellen und Addins

10.1 Cloudfähige DATEV Debitoren Schnittstelle

Linie: Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.3.0.6

Die neue DATEV Debitoren Schnittstelle erlaubt es, Debitorstammdaten und Debitorbuchungen von Vertec nach DATEV zu übertragen.

Die Schnittstelle ist cloudfähig und implementiert das listenbasierte Buchen. Da bei jedem Buchungsvorgang direkt eine Datei erzeugt wird, können damit Buchungen direkt auf ganzen Listen durchgeführt werden statt für alle Objekte einzeln.

Detaillierte Informationen zur DATEV Debitoren Schnittstelle finden Sie in der Online Knowledge Base unter www.vertec.com/kb/datev-debitoren-schnittstelle.

10.2 Run my Accounts Debitoren Schnittstelle

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.3.0.9

Bei Run my Accounts (www.runmyaccounts.ch) handelt es sich um einen Online-Buchhaltungs-Service. Mit der listenbasierten und cloudfähigen Vertec Run my Accounts Schnittstelle können Debitoren gebucht und storniert werden.

Die Schnittstelle ist cloudfähig und implementiert das listenbasierte Buchen. Da bei jedem Buchungsvorgang direkt eine Datei erzeugt wird, können damit Buchungen direkt auf ganzen Listen durchgeführt werden statt für alle Objekte einzeln.

Detaillierte Informationen Eine detaillierte Beschreibung über Installation und Betrieb der Run my Accounts Debitoren Schnittstelle finden Sie in der Online Knowledge Base unter www.vertec.com/kb/rma-debitoren-schnittstelle.

10.3 BMD Kreditoren Schnittstelle

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.4

Bei der BMD Kreditoren Schnittstelle handelt es sich um eine Dateischnittstelle, welche CSV Dateien erzeugt. Damit können Kreditor Rechnungen in BMD gebucht und storniert werden.

Die Schnittstelle ist cloudfähig und implementiert das listenbasierte Buchen. Da bei jedem Buchungsvorgang direkt eine Datei erzeugt wird, können damit Buchungen direkt auf ganzen Listen durchgeführt werden.

Detaillierte Informationen Eine detaillierte Beschreibung über Installation und Betrieb der BMD Kreditoren Schnittstelle finden Sie in der Online Knowledge Base unter www.vertec.com/kb/bmd-kreditoren-schnittstelle.

10.4 Mehrere Dokumentbibliotheken via Sharepoint-Schnittstelle anbinden

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.3.0.18

Mit der Vertec Sharepoint Schnittstelle können nun bei Bedarf mehrere verschiedene Dokumentbibliotheken (Libraries) angesprochen werden. Die verschiedenen Libraries werden dabei via ihren Dateipfad in Vertec unterschieden. Das bedeutet, dass die Library URL Einstellung in Vertec nicht mehr unbedingt die komplette URL zu

einer Document Library darstellt. Erst zusammen mit dem Dateipfad muss eine gültige Sharepoint URL für eine bestimmte Document Library entstehen.

Möchte man mehrere Dokument-Libraries nutzen und Dokumente pro Klasse speichern, muss man folgendes beachten:

- Unter Systemeinstellungen Allgemein > Dokumentenpfad muss der Root-DMS-Pfad hinterlegt werden, z.B. `Sharepoint:\MultiSite2\Shared Documents`. Dies ist der Standard-Pfad und wird für alle Objekte verwendet, die nicht einen anderen Pfad hinterlegt haben (siehe nächster Punkt). Der Standard-Pfad muss als eigenständiger Pfad gültig sein.
- Im Ordner Einstellungen > Klasseneinstellungen muss für alle Klassen, die vom Standard-Verzeichnis **abweichen**, ein absoluter Sharepoint-Pfad eingetragen werden, bei Adressen zum Beispiel `Sharepoint:\MultiSite1\Shared Documents\Kontakte\%if oclistypeof(Kontakt) then oclastype(Kontakt).firma.name else name endif%`.

10.5 Aufnahme des ProCall-Add-ins in den Vertec Setup

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Desktop, Cloud | Version: 6.3.0.13

Das ProCall Addin wurde in den Vertec Setup integriert und kann nun wie die anderen Addins im Setup-Dialog ausgewählt werden:

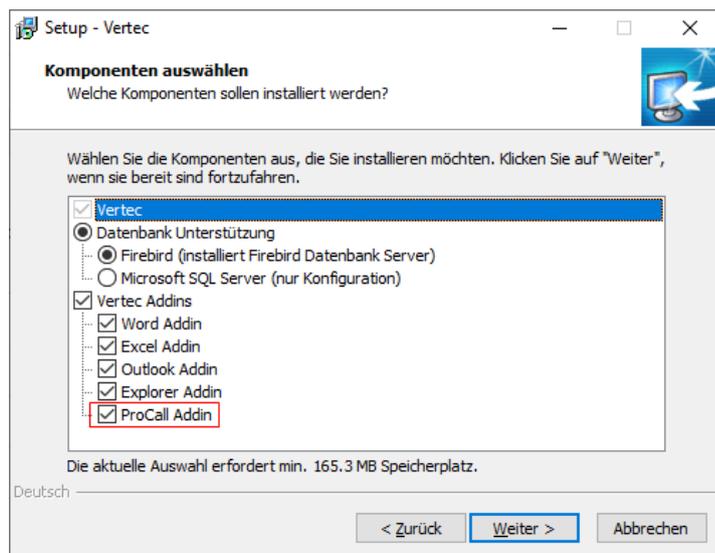


Abbildung 40: Auswahl des ProCall Addins beim Vertec Setup

Es wird dann vom Setup automatisch installiert.

Auch im Installer der Cloud App ist das ProCall Addin integriert:

- Cloud App via Portalpage installieren
- "Erweiterte Installationen" zulassen
- Das ProCall Addin wird in die Cloud App Installation kopiert

Wichtiger Hinweis Vor der Installation des ProCall Addins muss ProCall beendet werden. Weitere Angaben dazu finden Sie unter www.vertec.com/kb/procall-addin.

11 Scripting

11.1 OCL Call Operatoren für Custom Businesslogik

Linie: Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.3.0.2

Mit den OCL Call Operatoren können bestimmte Python Funktionen aus OCL aufgerufen werden. Das ermöglicht es, Ergänzungen zur Businesslogik in Python zu schreiben und diese an verschiedenen Orten anstelle der fest eingebauten Businesslogik verwenden zu können, z.B. in Scripts und für die Darstellung in Listen oder Berichten.

Da OCL statisch und streng typisiert ist, braucht es für jede benötigte Kombination aus Rückgabewert und Argumenten einen separaten OCL Operator. Folgende stehen zur Verfügung:

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

Die OCL Operatoren werden auf BusinessclassesRoot angewendet und haben alle als erstes Argument den Namen der implementierenden Python-Funktion. Dieser Name wird in der Form `<Modul>.<Funktionsnamen>` angegeben.

Beispiel

Es wird ein Script-Modul namens `customlogic` angelegt, darin eine Funktion mit der Signatur

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

Der Aufruf in OCL würde dann wie folgt aussehen:

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

das Ergebnis der OCL Expression ist ein Currency Wert.

Der Name des verwendeten Call-Operators enthält den Typ des Rückgabewertes und der Argumente (Rückgabewert Currency "Curr", Argument "Date").

11.2 Template Engine in Python

Linie: Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.3.0.9

Es gibt neu eine allgemein verwendbare Template Engine in Python. Sie basiert auf der Jinja2 Template Engine (<http://jinja.pocoo.org>).

Für die einfache Verwendung in Vertec führen wir auf dem vtcapp Modul eine Methode `rendertemplate(templateString, data)` ein.

Diese Methode ist auch im restricted scripting Modus (Cloud-Abo) verfügbar und liefert einen Unicode String zurück.

- Das `templateString` Argument kann ein String oder Unicode-String sein und kann Jinja2 spezifische Markups enthalten.
- Zur Übergabe der Daten stehen als weitere Argumente der Funktion folgende Möglichkeiten zur Verfügung:
 - Python Dictionary mit String-Werten als Keys. Definiert damit die im Template verfügbaren Variablen.
 - Beliebige Anzahl von Keyword Argumenten, welche die im Template verfügbaren Variablen definieren.
 - Kein weiteres Argument, das Template wird dann ohne Daten verarbeitet.

Die wichtigsten Strukturen sind die folgenden:

- Ein Block ist mit `{% ... %}` eingefasst. Er enthält eine Control-Anweisung oder eine Variablen-Zuweisung.
 - `{% if proj.code == "ABC" %} ... {% endif %}`. Ein if-Statement in einem Block wird verwendet, um einen Bereich des Templates konditional auszugeben. Ein if muss durch ein endif abgeschlossen werden. Optional ist auch ein `{% else %}` dazwischen erlaubt.
 - `{% for proj in projects %} ... {% endfor %}`. Ein for-Statement erlaubt die Wiederholung (Iteration) eines Template Bereichs. Muss durch endfor abgeschlossen werden.
- Eine Expression wird mit `{{ ... }}` abgetrennt. Eine Expression wird im Context des Templates ausgewertet und als String in die Ausgabe des Templates gemischt.

Beispiel Code für Ausgabe einer Projektliste

```
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
```

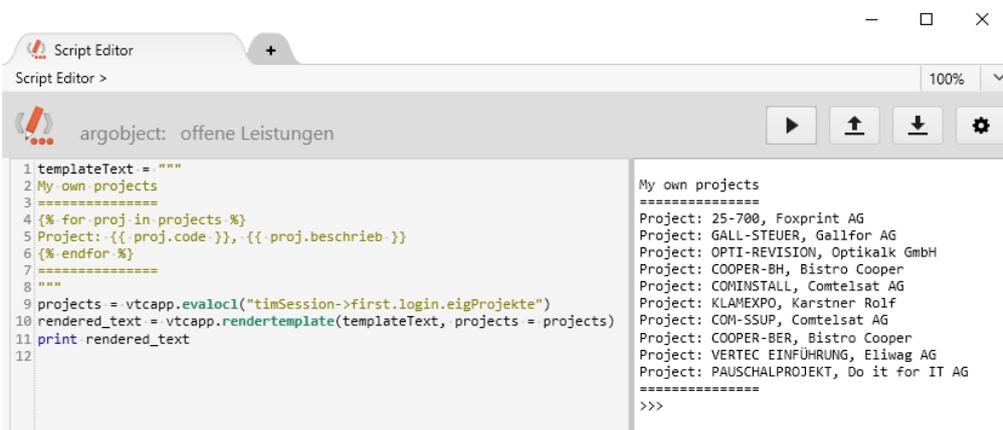


Abbildung 41: Das Beispiel-Script im Script Editor ausgeführt

Weitere Informationen Weitere Informationen dazu finden Sie in der Online Knowledge Base unter www.vertec.com/kb/pythoninterfaces#rendertemplate.

11.3 Eingebautes Python-Modul ziputils

Linie: Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.3.0.2

Ab Vertec 6.3.0.2 gibt es ein Vertec Modul namens ziputils, welches ermöglicht, in Python ZIP-Files zu generieren. Es ist auch im eingeschränkten (Restrict Scripting) Modus verfügbar und enthält die Methode:

createzip(contentlist: tuples): bytestream

Erzeugt aus einer Liste von tuples (filename,content) ein zipfile und gibt es als bytestream zurück. Dieser kann einfach mit einem Filename.zip abgespeichert/heruntergeladen werden.

Beispiel

```
import ziputils

zip = ziputils.createzip(['file.txt', 'Hallo das ist ein Text'])
vtcapp.sendfile(zip, 'ziptest.zip', True)
```

11.4 Python Methode für File Upload

Linie: Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.3.0.4 / 6.3.0.12

Mit der neuen Python Methode für den File Upload kann der Benutzer in der Client Applikation ein lokales File auswählen. Dieses wird dann an den Server übertragen und steht als Rückgabewert der Python Methode in Form eines Binär-Streams zur weiteren Verarbeitung zur Verfügung.

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

- **title**: Titel des Dialogs
- **path**: Pfad des im Dialog zu öffnenden Verzeichnisses. Falls es den Pfad nicht gibt, wird der Desktop angezeigt.
- **filter**: Filterausdruck, um die Auswahl von Dateien im Dialog einzuzugrenzen. Er besteht aus Filterpaaren, wobei jedes Filterpaar zwingend Name und Ausdruck benötigt.

```
"Office Documents|*.docx;*.xlsx|Text|*.txt|Xml|*.xml"
```

Sowohl zwischen den Paaren als auch innerhalb eines Pairs zwischen Name und Ausdruck müssen Pipes | als Trennzeichen verwendet werden.

Im Falle mehrerer Dateierendungen müssen diese durch Semikola getrennt werden.

- **abspath**: Absoluter Pfad der hochzuladenden Datei. Dieser Parameter ist optional. Wird er angegeben, so wird kein Dialog angezeigt.

Die Methode gibt ein Tupel bestehend aus absolutem Dateinamen und Dateiinhalt zurück.

Die Maximalgröße der hochzuladenden Datei beträgt 50 MB.

Es existiert eine Blacklist von Verzeichnissen, aus der keine Dateien angefordert werden dürfen, z.B. die Windows-Umgebungsvariable SYSTEMROOT (typischerweise C:\Windows). In diesem Fall erscheint eine Fehlermeldung.

Beispiele

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

Öffnet einen Dialog mit dem Titel "Hallo Welt" im Verzeichnis C:\MyDirectory, welcher den Upload von Office-, TXT- oder XML-Dateien erlaubt .

Es wird auch eine einfache Syntax unterstützt:

```
vtcapp.requestfilefromclient("Hallo Welt", r"C:\MyDirectory", "*.txt")
```

Öffnet einen Dialog mit dem Titel "Hallo Welt" im Verzeichnis C:\MyDirectory, welcher den Upload von TXT-Dateien erlaubt.

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

Lädt die Datei ohne das Öffnen eines Dialogs hoch.

Limitierungen in der Web App

Der Filedialog kann weder einen benutzerdefinierten Dialogtitel anzeigen, noch lässt er sich auf ein Startverzeichnis setzen. Somit werden die ersten zwei Parameter `title` und `path` ignoriert.

Wird ein clientseitiger Pfad angegeben (Parameter `abspath`), erscheint in der Web App eine Fehlermeldung.

11.5 getwithsql Python Methode mit Objektliste und Id-Feld

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Full featured | Version 6.3.0.8

SQL Abfragen, welche auf Listen von Objekten basieren wurden bisher mit einer „...in (id1, id2, id3,...)“ Konstruktion in `getwithsql` implementiert. Diese Art der Abfrage führt mit Firebird Datenbanken zu Fehlermeldungen, wenn mehr als 1500 Ids geprüft werden. Um dies zu umgehen und SQL Abfragen mit Referenzen auf eine Objektliste zu vereinfachen, wurde die `getwithsql` Methode um 2 optionale Argumente für eine Objektliste und ein Id-Feld erweitert.

Aufruf vorher

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

Aufruf nachher

```
getwithsql(klasse: string, where: string, order: string, [idfilterfeld: string, objectlist: list])
```

- **idfilterfeld**: Name des Members, nach welchem aufgrund der Liste gefiltert werden soll.
 - **objectlist**: Liste der Objekte, nach welchen im unter `idfilterfeld` angegebenen Feld gesucht werden soll.
- Die neuen Parameter sind optional..

Weitere Informationen und Beispiele zu diesem Thema finden Sie in der Online Knowledge Base unter www.vertec.com/kb/pythoninterfaces.

11.6 Member basierte SQL String Suche

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Full featured | Version 6.3.0.8

Es gibt neu eine Methode, um per SQL ein einzelnes Member auslesen zu können. Bei einer solchen Suche können wir den Suchparameter im Code genügend absichern und auch die Zugriffsberechtigung auf der Klasse vorgängig prüfen. Die Verwendung dieser Methode ist deshalb auch ohne explizites SQL-Recht erlaubt. Die Syntax ist wie folgt:

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

Die Expression muss ein Vergleichs-String (ohne Delimiter) sein. Er kann %-Platzhalter enthalten, etwaige String-Delimiter innerhalb des Strings werden escaped.

Falls die Expression Wildcard-Zeichen enthält, erfolgt der Vergleich mit dem SQL "LIKE" Operator, ansonsten mit dem "=" Operator.

Der Vergleich ist case-insensitive, das heisst, es muss nicht auf Gross- / Kleinschreibung geachtet werden.

Beim Aufruf der Funktion wird die Berechtigung des aktuellen Bearbeiters auf dem für die Selektion verwendeten Member geprüft. Der Benutzer muss klassenweites Lese-Recht auf dem Member haben, damit die Suche funktioniert, sonst erscheint eine Fehlermeldung.

Beispiel:

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

Findet alle Projekte mit Projekt-Code "ABC", "aBc" oder "abc".

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

Findet alle Projekte, deren Code mit "AB", "ab", etc. beginnt.

Umstellung ProCall Addin

Das ProCall Addin wurde auf die neue Methode umgestellt und der Standard Benutzergruppe das Leserecht auf das Feld `normalizedziel` der Klasse KommMittel erteilt.

Ab der Vertec Version 6.3.0.8 wird also für den Betrieb des ProCall Addins kein SQL-Recht mehr benötigt.

11.7 Angabe von Speicherpfad bei SendFile Mechanismus

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Full featured | Version 6.3.0.8

Die Python Methode `sendfile()` ermöglicht es, vom Server ein File auf dem Client zu speichern. Diese Funktionalität gibt es seit Vertec 6.2.

Bisher wurde nur die Angabe eines Filenamens unterstützt, und mit der Desktop- und Cloud App wurde das File immer im Temp-Verzeichnis des Benutzers gespeichert.

Neu kann ein kompletter Speicherpfad angegeben werden. Falls möglich, wird das File unter diesem Pfad auf dem Client gespeichert. Dabei gelten folgende Regeln:

- Falls der Zielpfad nicht existiert, wird er ohne Nachfrage erstellt.
- Falls das Speichern (oder Erstellen des Pfades) aufgrund mangelnder Zugriffsrechte nicht möglich ist, gibt es eine Fehlermeldung.
- Das Speichern mit Pfad funktioniert mit der Desktop- und der Cloud App identisch. In der Web App wird der angegebene Pfad ignoriert und nur der Filename für den Download berücksichtigt.

Die Methode hat dafür neu auch einen Rückgabewert und liefert `True` oder `False` zurück:

- True, falls die Datei gespeichert wurde,
- False, falls die Datei nicht gespeichert wurde,
- False, falls das Speichern aufgrund von Benutzerangaben abgebrochen wird (auf Abbrechen klicken)
- Die Web App liefert immer True zurück, da dort nicht überprüft werden kann, ob das File gespeichert wurde. Bei der Verwendung der Web App empfehlen wir, im Browser den automatischen Download einzuschalten.

Weitere Informationen und Beispiele zu diesem Thema finden Sie in der Online Knowledge Base unter www.vertec.com/kb/pythoninterfaces.

11.8 executereport() auch für erweiterte Office-Berichte

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.3.0.4

Die Python-Methode `executereport()` wurde so erweitert, dass sie auch mit erweiterten Office Berichten funktioniert. Die Methode erwartet als Argument eine Berichts-Definition (registrierter Bericht in Vertec).

Weitere Informationen Weitere Informationen zu dieser Python-Methode finden Sie in der Online Knowledge Base unter www.vertec.com/kb/pythoninterfaces.

11.9 Neue Python Module im Restricted Scripting Modus

Linie: Expert | Module: Leistung & CRM | Apps: Full featured | Version 6.3.0.2/6.3.0.16

Die folgenden Python Module sind neu für die Verwendung im Restricted Scripting Modus verfügbar:

Ab Version 6.3.0.2

- ziputils (siehe Abschnitt 11.3)

Ab Version 6.3.0.16

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

sowie:

- Python Package `email` mit allen ihren Sub-Modulen
- Builtin function `help`

Weitere Informationen Weitere Informationen zum Thema **Restrict Scripting** finden Sie in unserer Online Knowledge Base unter www.vertec.com/kb/restrict-scripting.

11.10 Geänderte Python Script-Module automatisch neu laden

Linie: Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.4

Beim Importieren von Scripteinträgen mit Python-Code als Python Module wurden Änderungen im Code erst berücksichtigt, wenn Vertec neu gestartet wurde.

Neu ist es so, dass bei der Code-Änderung eines Scripteintrags ein allfällig geladenes Modul aufgrund seiner Bezeichnung aus dem Zwischenspeicher gelöscht wird. Dadurch wird es beim nächsten Import des Moduls durch anderen Python-Code wieder geladen und somit der aktuellste Code verwendet.

Bei den Extensions hat sich nichts geändert. Das heisst, dass beim Ändern des Extension-Codes in einem Script-eintrag die Extension immer noch reloaded werden muss. Allerdings ist dann das Modul bereits aktuell.

11.11 Unterstützung von erweiterten Berechtigungen in importieren Script-Modulen

Linie: Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.3.0.14

Einem Script können erweiterte Rechte erlaubt werden. Die bisherige Behandlung dieser erweiterten Berechtigungen basierte darauf, dass zu einem bestimmten Zeitpunkt ein Script-Text in einem bestimmten Context ausgeführt wird. Beim Ausführen von Code aus importierten Modulen via OCL Call Operatoren (siehe 11.1) oder via Scripts war das nicht möglich, da der Code dann in einem anderen Context ausgeführt wird.

Dieser Mechanismus wurde nun so erweitert, dass die Markierung von Scripteinträgen mit erweiterten Berechtigungen auch funktioniert, wenn der Scripteintrag als Modul in Python importiert oder via OCL Call Operator aufgerufen wird.

Weitere Informationen Weitere Informationen zu erweiterten Berechtigungen in Scripts finden Sie in der Online Knowledge Base unter www.vertec.com/kb/erweiterte-berechtigungen.

11.12 Python ODBC Library pyodbc mitgeliefert

Linie: Expert | Module: Leistung & CRM | Apps: Desktop | Version: 6.4

Ab Vertec 6.4 liefern wir die **pyodbc** Package im Python Folder im Vertec Installationsverzeichnis mit. Sie heisst pyodbc.pyd und erlaubt Datenbankzugriffe mittels ODBC, zum Beispiel für Migrationen.

Durch `import pyodbc` in Vertec Python Scripts steht die ganze Funktionalität zur Verfügung (nur im nicht eingeschränkten Modus, also ohne Restrict Scripting).

Weitere Informationen Eine detaillierte Dokumentation der Python ODBC Library **pyodbc** erhalten Sie beim Hersteller unter <https://github.com/mkleehammer/pyodbc>.

12 Performance

12.1 Optimierung von Objekt-Referenzen in OCL

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.4.

OCL wertet Expressions streng von links nach rechts aus und verwendet das Resultat der ersten Teilexpression dann als Quelle für die nächste Teilexpression. Einfache Expression wie

```
projekt (bzw. Projekt.allInstances)
```

bestehen nur aus einer Teilexpression und machen im Normalfall auch keine Performance Probleme.

Erweitert man nun diese Expression um ein persistentes Member, z.B. "aktiv":

```
projekt->select(aktiv)
```

so dauert das genau gleich lang, denn die Projekte sind mit allen Members schon geladen.

Selektiert man nun aber auf eine Subliste wie zum Beispiel

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

(alle Projekte, die mindestens eine Phase haben), so wird es plötzlich viel langsamer. Das ist so, weil die Berechnung wie oben beschrieben von links nach rechts abgearbeitet wird:

1. Alle Projekte reinladen
2. Dann für jedes Projekt die Expression `phasen->size>0` auswerten. Das bedeutet, dass für jedes einzelne Projekt in der Liste die Phasen dieses Projektes einzeln reingeladen werden und dann entschieden wird, ob es in die Liste kommt oder nicht.
3. Dann beim nächsten Projekt das gleiche, bis zum Ende der Liste.

Hier greift nun die neue Optimierung. Da in unserem Beispiel klar ist, dass alle Phasen der Projekte in der Liste benötigt werden, wird nun bei der Durchführung einer solchen Expression die gesamte Phasenliste auf einmal ins Memory geladen. Es erfolgt also nur ein einmaliger Datenbankzugriff. Die Abarbeitung ist danach viel schneller, weil alle benötigten Phasen bereits geladen sind.

Diese Optimierung findet automatisch statt, das heisst, ab sofort werden Expressions dieser optimiert durchgeführt, und zwar immer dann, wenn solche OCL Expressions mit Sublisten direkt ausgeführt werden, z.B. als Expression in Python oder in einem Expression-Ordner.

Optimierung von Objekt-Referenzen in Vertec-Listen

Die beschriebene Optimierung wurde auch bei den Spaltenexpressions in Vertec-Listen umgesetzt. Häufig wird da ja auf Sublisten oder Linkobjekte zugegriffen, z.B. `projektleiter` auf Projekten oder eben Phasen.

Wann greift die Optimierung in Listen:

- Bei allen Spalten, wo sowieso alle Daten benötigt werden, also immer dann, wenn danach summiert bzw. sortiert wird.
- Für alle Objekte, die beim Aufruf einer Liste standardmässig geladen werden. Das sind die dargestellten plus eine Seite mehr.
- Bei einer Total-Anzahl von Objekten bis ca. 300, weil dann die Wahrscheinlichkeit gross ist, dass er User scrollen wird. Hier tritt beim Erstaufbau der Liste unter Umständen eine Verlangsamung ein, und bei mehr als 300 Einträgen wird die Liste dann merklich langsamer, ohne dass der User vielleicht alle Einträge sehen will.

Wann greift sie nicht:

- Wenn die Expression zu komplex ist.
- Bei Sublisten von Sublisten wie zum Beispiel Subphasen.
- wenn keine der oben genannten Konditionen erfüllt ist.

12.2 Beschleunigung Berechtigungsprüfung auf Projekt

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.3.0.4

Die Reihenfolge der Berechtigungsprüfung auf Projekten wurde umgestellt. Das führt zu einer Performance-Verbesserung beim Öffnen der Projekt-Combobox in der Leistungserfassung bei sehr vielen Projekten.

12.3 Performance-Optimierung bei der Auswahl von Tätigkeiten

Linie: Standard, Expert | Module: Budget & Teilprojekte | Apps: Full featured | Version: 6.3.0.6

Bei der Auswahl von Tätigkeiten in der Leistungserfassung muss vom System geprüft werden, ob die Tätigkeit für die gewählte Phase aufgrund von Zuordnungen zur Erfassung zugelassen ist oder nicht.

Diese Überprüfung wurde so optimiert, dass unterschieden wird, ob bereits eine Projektphase ausgewählt wurde oder nicht. Ist bereits eine Phase ausgewählt, erfolgt die Prüfung nur noch auf dieser einzelnen Phase, nicht mehr auf allen, was zu einer Performance-Verbesserung führt.

Weitere Informationen Weitere Informationen zum Thema Phasenzuordnungen finden Sie in der Online Knowledge Base unter www.vertec.com/kb/projektphasen.

12.4 Outlook Addin: cc: Adressen nicht mehr speichern

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Desktop, Cloud | Version: 6.3.0.6

Performance-Optimierung beim **Outlook Addin**: Kunden-E-Mail-Adressen, welche nur im CC einer E-Mail erscheinen, werden nicht mehr als Empfänger-Adresse in der Aktivität gespeichert.

13 Customizing

13.1 Auch Nicht-Stichwort-Ordner in Seiten einblendbar

Linie: Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.3.0.12

Einzelne Stichwörter können per Customizing mit dem Element `KeywordGroup` in anderen Seiten platziert werden. Dabei wird eine Gruppe mit der Bezeichnung des Stichwort-Ordners und einer Auswahl je nach Art mit Checkboxes oder einer Combobox dargestellt.

Ab Vertec 6.3.0.12 können auch Ordner, die nicht als Stichwort-Ordner markiert sind, auf diese Art in den Seiten platziert werden.

Weitere Informationen Weitere Informationen zum Platzieren von Stichwörtern in Seiten finden Sie in der Online Knowledge Base unter www.vertec.com/kb/seiten#stichwort.

13.2 Anpassung von Code Editoren

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.3.0.3

In den Code Editoren wird nun ein monospaced Font (im Gegensatz zu einem proportionalem Font) verwendet. Schriftart ist `Consolas`, wie bereits in der Python-Konsole (F3) eingesetzt.

Betrifft folgende Editoren:

- Code bei den registrierten Scripts (ScriptEintrag)
- Code im Script-Editor
- Berichtdefinition bei erweiterten Office Berichten
- XML-Codeboxen in Listeneinstellungen und Klasseneinstellungen
- OCL Expression Editor

Ausserdem erscheint ein horizontaler Scrollbalken in der Python-Konsole (F3) sowie dem Code-Fenster im Script Editor.

13.3 Web App: Script Editor mit Syntax Highlighting

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Web | Version 6.3.0.8

Das mit Vertec 6.2 eingeführte Syntax Highlighting in den Code Editoren wurde nun auch in der Web App umgesetzt.

13.4 Tastaturkürzel für den Script Editor

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Full featured | Version 6.3.0.8

Im Script Editor wurden folgende neue Tastaturkürzel eingeführt:

- `Ctrl+E` (Script ausführen)
- `Ctrl+Z` (Undo)

14 Config Sets

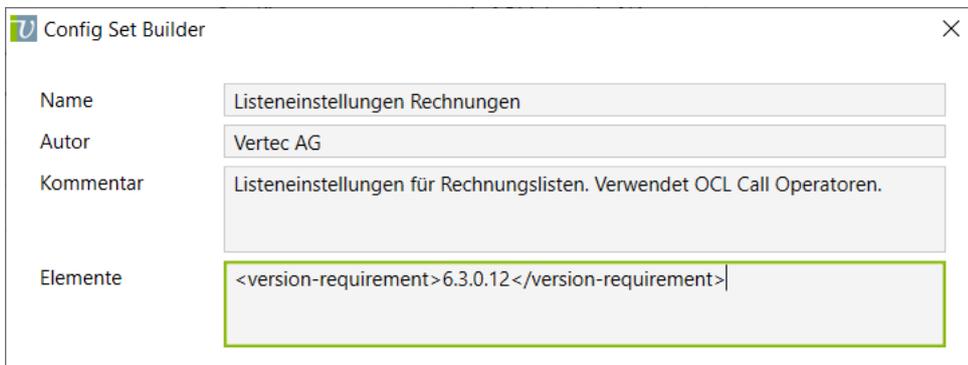
14.1 Minimale Applikations-Version in Config Sets

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Desktop, Cloud | Version: 6.4

Neu kann in einem Config Set optional die minimale Vertec Version angegeben werden, die benötigt wird, um das Config Set zu verwenden. Die Umsetzung erfolgt, analog zu den bestehenden Requirements, mit einem manuellen XML-Element. Die Syntax ist wie folgt:

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

Eingegeben wird das im Feld **Elemente** im Config Set Builder:



Name	Listeneinstellungen Rechnungen
Autor	Vertec AG
Kommentar	Listeneinstellungen für Rechnungslisten. Verwendet OCL Call Operatoren.
Elemente	<version-requirement>6.3.0.12</version-requirement>

Abbildung 42: Dieses Config Set benötigt eine Vertec Version von mindestens 6.3.0.12

- Wird ein Config Set mit einem Version-Requirement importiert und die Vertec Version ist kleiner als die angegebene Version, wird ein Fehler gemeldet und das Config Set kann nicht importiert werden.
- Wird ein Config Set mit Version-Requirement in einer älteren Version von Vertec ohne Implementierung des Version-Requirement Features importiert, wird ebenfalls ein Fehler gemeldet, und das Config Set kann nicht importiert werden. Für Config Sets mit dieser Angabe wird also ein Vertec ab Version 6.4 benötigt.

14.2 Unterstützung von Zusatzfeldern in Config Sets

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Desktop, Cloud | Version: 6.4

Wird ein Objekt, welches über Zusatzfelder verfügt, zu einem Config Set hinzugefügt, dann werden neu automatisch auch die Feldwerte und die entsprechenden Zusatzfeld-Definitionen hinzugefügt. Diese müssen nicht wie andere Objekte über eine Eintrag Id verfügen, sondern werden via Klasse und Name eindeutig identifiziert.

Falls sich die Zusatzfelder auf einer Zusatzklasse befinden, wird die Referenz ebenfalls direkt eingetragen. Hier muss sichergestellt sein, dass im Zielsystem nicht bereits eine andere Zusatzklasse mit derselben Nummer befindet, denn die Klasse wird auf den Zusatzfeldern einfach entsprechend eingetragen und wären somit falsch verknüpft.

Möchte man die Klasseneinstellungen der Zusatzklasse ebenfalls übertragen, müssen diese dem Config Set separat hinzugefügt werden.

Rückwärtskompatibilität

Ab Version 6.4 erstellte Config Sets, welche Objekte mit Zusatzfeldern beinhalten, können in ältere Vertec Versionen nicht mehr importiert werden.

Wir empfehlen, in diesen Fällen eine Minimalversion anzugeben (siehe Kapitel 14.1).

14.3 Berücksichtigung von MLStrings in Config Sets

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Desktop, Cloud | Version: 6.4

Neu wird bei MLStrings (siehe Kapitel 8.3) nicht mehr nur der Text in der aktuellen Sprache, sondern alle angegebenen Sprachen inklusive der mit dieser Version eingeführten nativen Sprache übertragen.

Rückwärtskompatibilität

- Ist nur ein Begriff in der aktuellen Sprache angegeben, wird das Attribut als normaler String übertragen. In diesem Fall funktioniert der Import weiterhin mit allen Versionen.
- In jedem anderen Fall (also auch wenn nur ein nativer Begriff angegeben ist) werden die angegebenen Sprachen als `mltext`-Attribute übergeben. Dieses Element besteht erst ab Version 6.4, was bedeutet:
 - Früher erstellte Config Sets können auch mit neueren Vertec Versionen importiert werden
 - Ab Vertec 6.4 erstellte Config Sets können mit älteren Vertec Versionen nicht importiert werden, wenn mehr als ein Begriff in der aktuellen Sprache in den MLString-Attributen angegeben ist.

15 Technik / Diverses

15.1 Beenden der Session nach Verbindungsabbruch verzögern

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Cloud Clients | Version: 6.3.0.18

Es gibt in Vertec Logik, welche bei Verbindungsabbruch eines Cloud Clients die serverseitige Session beendet, auch wenn diese sich nicht kontrolliert beendet hat. Beispiel: Schliessen des Browsers, in welchem die Web App läuft.

Dies wird gemacht, um hängende, nicht mehr verwendete Prozesse auf dem Server zu vermeiden.

Das Beenden der Session wurde nun um 5 Minuten verzögert, um gewissen Prozessen die Gelegenheit zu geben, ihre Arbeiten vollständig durchführen zu können (z.B. die Beendigung eines laufenden Eventscripts).

Da dies im Hintergrund abläuft, ist für den Benutzer keine Veränderung zu vorher festzustellen.

15.2 Instanzname in Login- und Über Vertec-Dialog anzeigen

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Desktop | Version: 6.4

Eine Datenbank-Sektion kann im Vertec.ini-File einen Wert namens `InstanceName` enthalten. Auf dem Taskbar-Icon der Vertec Desktop App wird dies dann entsprechend angezeigt.

Neu wird dieser Instanzname auch im Login- und im Über Vertec-Dialog angezeigt:

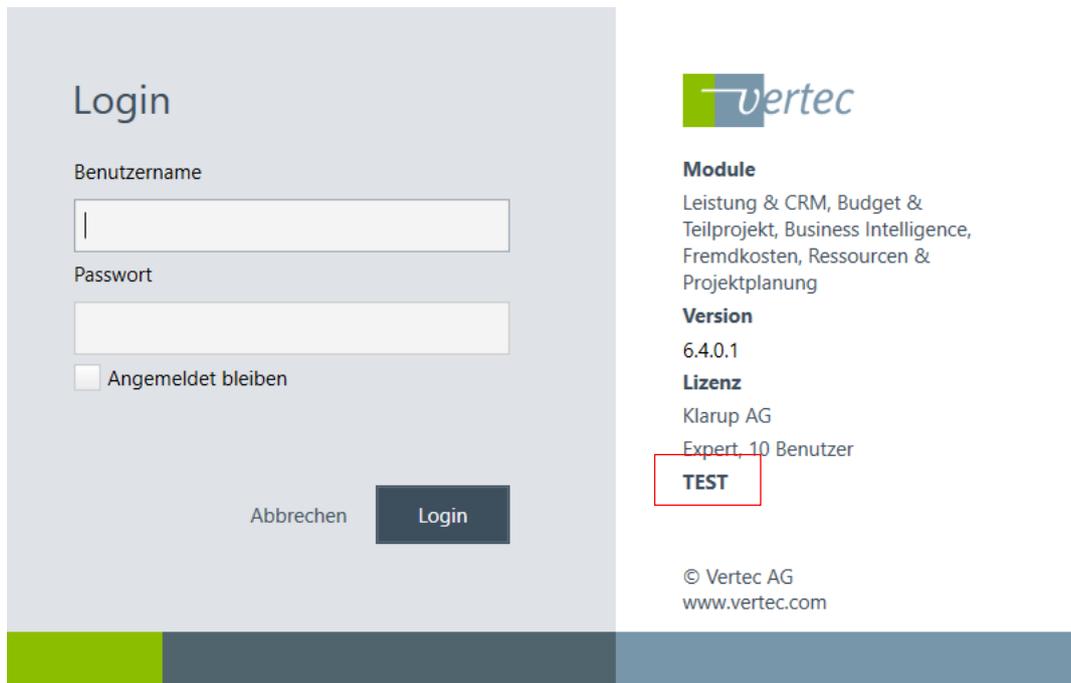


Abbildung 43: Der Instanzname wird im Login-Dialog angezeigt

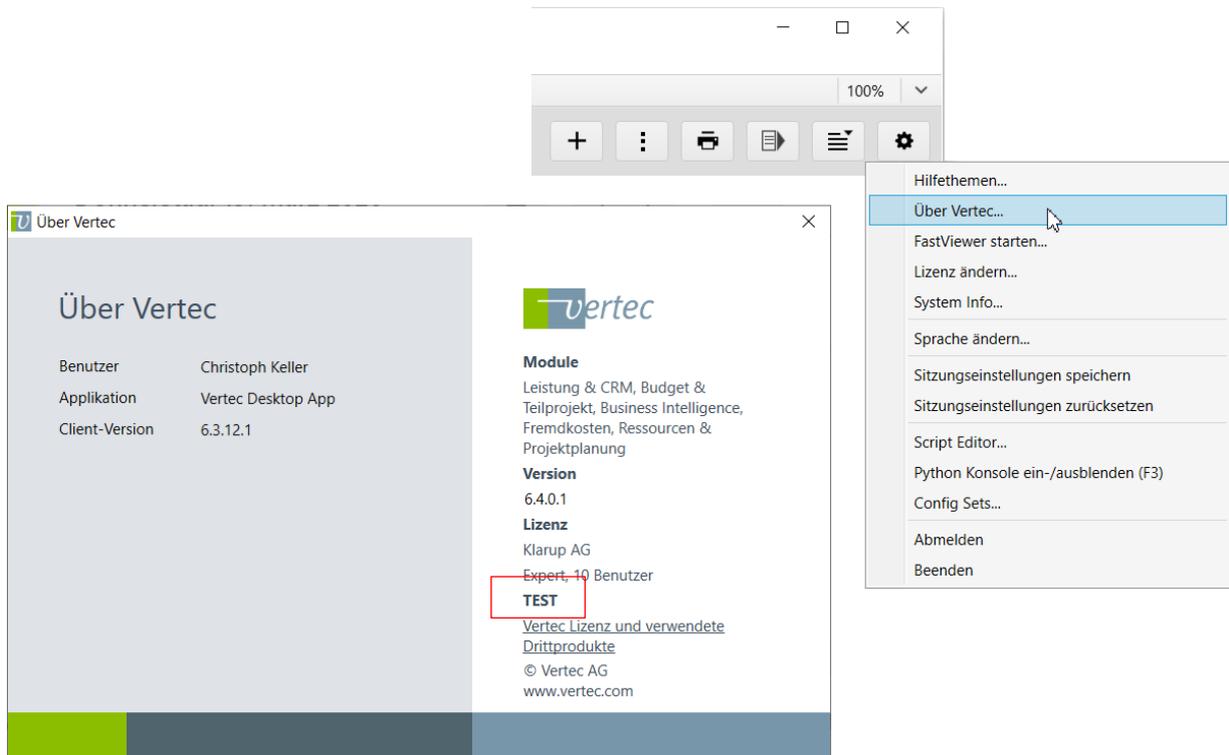


Abbildung 44: Im Über Vertec Dialog wird der Instanzname angezeigt

Weitere Informationen Weitere Informationen dazu finden Sie in der Online Knowledge Base unter www.vertec.com/kb/ini#testinstallationen.

15.3 Eintrag Id für Admin User vergeben

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.3.0.14

Der standardmässig mitgelieferte Administrator erhält eine Eintrag Id. Dies ist wichtig, damit er in Config Sets in Zukunft referenziert werden kann:

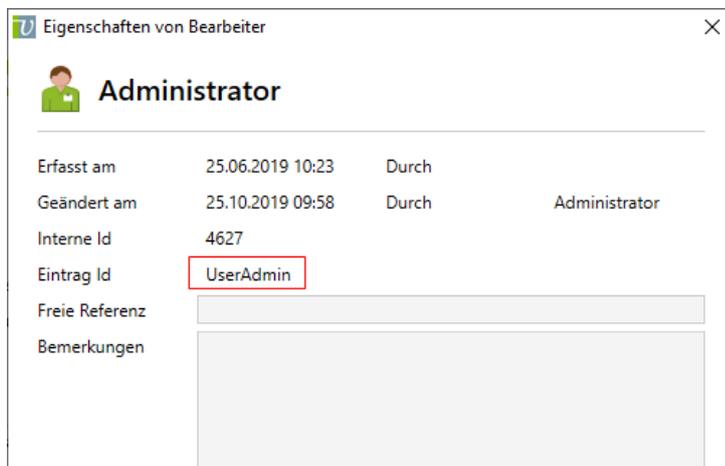


Abbildung 45: Eintrag Id beim Administrator

Der Wert wird auch in bestehenden Datenbanken beim Administrator eingetragen, falls dort noch kein Wert vorhanden ist.

15.4 CompanyName Angaben auf verrechneten Rechnungen

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Full featured | Version: 6.3.0.17

Das Feld **CompanyName** auf der Rechnung sowie weitere verwandte Felder wurden mit Vertec 6.3 eingeführt. Beim Verrechnen einer Rechnung werden die aktuellen Werte für diese Felder in persistenten Feldern mit `verr` Präfix gespeichert. Bei verrechneten Rechnungen wird zur Darstellung jeweils auf diese fixierten Werte zugegriffen.

Bei Rechnungen, welche vor dem Update auf Version 6.3 verrechnet wurden, sind diese fixierten Company-Angaben (alle Felder, die mit `verrCompany` beginnen) deshalb nicht gesetzt. Das führt z.B. dazu, dass alte Rechnungen, wenn sie nochmal ausgedruckt werden, keine Adresse anzeigen. Um dieses Problem zu umgehen, führen wir bei der Berechnung der derived `CompanyXXX` Member folgende Fallunterscheidung ein:

- Falls alle `verrCompanyXXX` Members leer sind, wird der aktuell gültige, nicht fixierte Wert Firma (via Zahlungstyp bzw. Systemeinstellungen) verwendet.
- Falls mindestens ein `verrCompanyXXX` Member nicht leer ist, verwenden wir bei allen den `verr` Wert (wie bisher).

15.5 Verbindungen zu AbstractAdresseintrag aufgehoben

Linie: Standard, Expert | Module: Leistung & CRM, Fremdkosten | Apps: Full featured | Version: 6.4

Verschiedene Associations im Modell zeigten direkt auf **AbstractAdresseintrag**, die Basisklasse von **Adresseintrag**. Das führte dazu, dass bei einem Zugriff zum Beispiel auf den Kunden eines Projekt noch gecastet werden musste:

```
argobject.eval("kunde.name")
```

 funktionierte nicht, man musste noch einen Cast auf den Adresseintrag machen:

```
argobject.eval("kunde.oclastype(Adresseintrag).name").
```

Das Modell wurde nun so angepasst, dass die Associations direkt auf die Klasse Adresseintrag gehen. Damit entfallen diese Casts. Folgende Verbindungen wurden geändert:

- Projekt ⇔ Gegenparteien
- Projekt ⇔ Kunde
- Auslage ⇔ Lieferant
- Auslage ⇔ xRechnungsadresse
- Auslage ⇔ xLieferadresse
- Kreditor ⇔ Lieferant

Rückwärtskompatibilität

Da die Expressions mit Cast auch nachher noch funktionieren und die Klasse AbstractAdresseintrag weiterhin im Modell als Basisklasse von Adresseintrag besteht, ändert sich auch bei bestehenden Installation nichts.