



Februar 2018

Vertec-Software Release 6.2

Inhaltsverzeichnis

1 Einführung	5
1.1 Neuinstallation von Vertec	5
1.2 Update von Vertec	5
1.3 Das erste Aufstarten nach dem Konvert	5
1.4 Firebird Version 3.0.2.	5
2 Versionen	6
2.1 Highlights der Version 6.2	6
2.2 Eingestellte Features in Vertec 6.2	6
2.3 Vorankündigung Vertec 6.3	7
3 Vertec Phone App	9
3.1 Installation und Konfiguration	9
3.2 Angemeldet bleiben	10
3.3 Phone App Session Timeout	10
3.4 Mindestanforderungen	10
3.5 Login	11
3.6 Leistungen erfassen	13
3.7 Menü	14
3.8 Spesen erfassen	15
3.9 Aktivitäten erfassen	16
3.10 Aktivitäten pendent/offen	17
3.11 Einen Eintrag löschen	18
3.12 Adressen einsehen	18
4 Erweiterte Office-Berichte	21
4.1 Registrierung erweiterte Office-Berichte	22
4.2 Bericht-Definition	23
4.3 Aufbau des Python Codes	24
4.4 Feld-Typen	24
4.5 Das Context-Objekt	27
4.6 Before Report Logik	28
4.7 Aufbau der Word-Vorlagen	29
4.8 Expressions	29
4.9 Bands	29

4.10 Locale	29
4.11 Context-Expressions	30
4.12 Übersetzungen	31
4.13 Feldformatierungen	31
4.14 Bilder	32
4.15 Expressions in Kopf- und Fusszeilen	32
5 Config Sets	33
5.1 Der Config Set Dialog	33
5.2 Config Sets einspielen	33
5.3 Config Sets erstellen	35
5.4 Config Set XML	37
5.5 Verwendung der Eintrag Id	40
5.6 Behandlung von Spezialfällen	41
5.7 Config Sets entfernen/löschen	42
6 Customizing/Parametrisierbarkeit	43
6.1 Tags auf User-Einträgen	43
6.2 Key-Values auf User-Einträgen	44
6.3 Syntax-Highlighting in Code-Editoren	46
6.4 Control-Namen auf Rechnung-Page geändert	46
6.5 DatePicker in Listen	47
6.6 Unterstützung von Datumswerten in Custom Dialogen	47
6.7 Custom Dialoge: DatePicker mit Null-Wert	48
6.8 Neues Passwort verlangen	48
6.9 Neue OCL Operatoren	50
6.10 Neue Python Methoden	50
6.11 Python Methode vtcapp.sendfile()	52
6.12 Suchdialoge für Expressionordner	54
6.13 Ordner-Suche unmittelbar ausführen	57
6.14 Berechtigung für SQL-Abfragen	58
6.15 Berechtigungsprüfung bei persönlichen Ordnern geändert	60
7 Leistungen/CRM	61
7.1 Darstellung Saldi auf Übersicht Leistungen	61
7.2 Aktivitäten-Ordner auf Offerten/Angeboten	61

8 Rechnungsstellung	62
8.1 Unterstützung für Einzahlungsschein mit QR-Code	62
8.2 Unterstützung von IBAN Nummer auf Zahlungstyp	63
9 Reporting	64
9.1 Neuimplementierung Vertec-generierte Reports	64
9.2 Office Vorlagen auf neue Office Formate umgestellt	65
9.3 Vorlagen erstellen/bearbeiten in den Cloud Clients	66
9.4 Excel Export: Auswahlfelder werden nicht exportiert	67
9.5 Locale-Bands in Word-Reports akzeptieren neu auch OCL-Expressions	68
10 Schnittstellen	69
10.1 ProCall Addin	69
10.2 Abacus: Betrag darf nur 2 Nachkommastellen haben	71
10.3 Abacus: Land-Code aus Vertec Adresseintrag verwenden	71
10.4 SelectLine COM Schnittstellen Lizenzierung	71
11 Technik und Datenmodell	72
11.1 Einführung von freiem Referenzfeld	72
11.2 Festlegung minimaler Version für DB-Konvert	72
11.3 Kommandozeilen-Parameter	73
11.4 Parameter /noevents nur noch für Administratoren	73
11.5 Probleme mit Excel-Report VBA Makro behoben	73
11.6 Vertec.Updater self-autoupdate	74
11.7 CloudServer Services per Konfiguration aktivieren	74
11.8 Web App: Statische Files clientseitig cachen	74
11.9 "Angemeldet bleiben" im Batch-Modus	74
11.10 Brute-Force Prevention Mechanismus	75

1 Einführung

Dies sind die Release-Notes zur Vertec-Version 6.2. Bitte beachten Sie folgende Punkte:

1.1 Neuinstallation von Vertec

Für eine Neuinstallation verwenden Sie das zum Download verfügbare Installationsprogramm [VertecSetup-6.2.0.exe](#). Nähere Informationen zur Installation finden Sie in unserer Online Knowledge Base unter [www.vertec.com/de/kb/neuinstallation](#).

1.2 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/de/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
2.2 Eingestellte Features in Vertec 6.2	6
2.3 Vorankündigung Vertec 6.3	7
6.4 Control-Namen auf Rechnung-Page geändert	46
6.7 Custom Dialoge: DatePicker mit Null-Wert	48
6.14 Berechtigung für SQL-Abfragen	58
6.15 Berechtigungsprüfung bei persönlichen Ordnern geändert	60
8.2 Unterstützung von IBAN Nummer auf Zahlungstyp	63
9.1 Neuimplementierung Vertec-generierte Reports	64
9.2 Office Vorlagen auf neue Office Formate umgestellt	65
10.3 Abacus: Land-Code aus Vertec Adresseintrag verwenden	71

1.3 Das erste Aufstarten nach dem Konvert

Der erste Start von Vertec 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.

1.4 Firebird Version 3.0.2.

Vertec 6.2 liefert die neuste Firebird Version 3.0.2. mit. Das Update wird vom Vertec-Setup nach einer entsprechenden Abfrage automatisch vorgenommen.

Weitere Informationen Weitere Informationen zu Firebird finden Sie in der Online Knowledge Base unter [www.vertec.com/de/kb/firebird](#).

2 Versionen

2.1 Highlights der Version 6.2

Die Version 6.2 wartet wieder mit vielen Highlights auf. Die Spannendsten hier in einem kurzen Überblick:

Artikel	Seite
3 Vertec Phone App	9
4 Erweiterte Office-Berichte	21
5 Config Sets	33
6.1 Tags auf User-Einträgen	43
6.2 Key-Values auf User-Einträgen	44
8.1 Unterstützung für Einzahlungsschein mit QR-Code	62

2.2 Eingestellte Features in Vertec 6.2

Betrieb der Classic App wird eingestellt

Mit dem Release 6.2 wird der Betrieb der Classic App eingestellt. Die meisten Features, welche bisher nur in der Classic App laufen, wurden auf diesen Zeitpunkt in die neuen Apps übernommen. Folgende Features werden definitiv eingestellt:

- **Aktivitäten-Kalender:** Die Kalender-Ansicht für Aktivitäten gibt es nicht mehr. Die Kalenderwerte (von-bis Zeiten, Dauer, ganzer Tag) können auf den Aktivitäten nach wie vor eingetragen und die Aktivitäten so als Termine mit Outlook abgeglichen werden. Weitere Informationen dazu finden Sie in der Online Knowledge Base unter www.vertec.com/de/kb/kalender.
- **vtcForms-Dialoge:** vtcForms-Dialoge werden ebenfalls eingestellt. An ihrer Stelle stehen die neuen Python Dialoge zur Verfügung. Weitere Informationen finden Sie in der Online Knowledge Base unter www.vertec.com/de/kb/pythondialoge.

Betrieb der Classic Weberfassung wird eingestellt

Mit der Classic App verschwindet auch die Classic Weberfassung. Diese wird ab sofort nicht mehr unterstützt. Für die Erfassung via Internet-Browser steht die Vertec Web App zur Verfügung.

Bitte überprüfen Sie, ob Sie sämtliche Prozesse, die Sie bis jetzt mit der Classic Weberfassung ausführen, auch mit der Web App abwickeln können. Sollte dies nicht der Fall sein, wenden Sie sich bitte an Ihren Vertec-Betreuer.

Betrieb der iPhone App wird eingestellt

Mit der Classic App wird auf Version 6.2 auch die iPhone App eingestellt. Neu steht die Phone App zur Verfügung (siehe Kapitel 3).

XML Server kann nur noch über den Cloud Server Dienst betrieben werden

Der XML Server ist nun Teil des Cloud Server Dienstes. Der bisherige XML Server als Teil des Vertec Service Dienstes wird abgeschaltet. Die Authentisierung und das Session Handling wurde umgestellt und erweitert, sonstige Änderungen hat der XML Server nicht erfahren. Informationen dazu finden Sie in der Online Knowledge Base unter www.vertec.com/de/kb/xmlreferenz.

Systemeinstellungen und Funktionalität entfernt

Eingestellt werden folgende Systemeinstellungen, welche ab Vertec 6.0 nur noch in der Classic App angezeigt wurden.

Property	Sektion
Baumanzeige als Standard	Allgemein
Detailansicht mit Baumanzeige für Projekte und Mitarbeiter	Allgemein
Ordnerliste (weitere Info) anzeigen	Allgemein
Favoriten in Such-Edits anzeigen	Allgemein
Leistungstext in Wochenerfassung anzeigen	Classic Web
OpenID Authentifizierung unterstützen	Classic Web
Monatsfreigabe anzeigen	Classic Web
Nur externe Adressen verwenden	Adressen

Weitere Informationen zu den eingestellten Funktionalitäten finden Sie in unserer Online Knowledge Base unter www.vertec.com/de/kb/legacy.

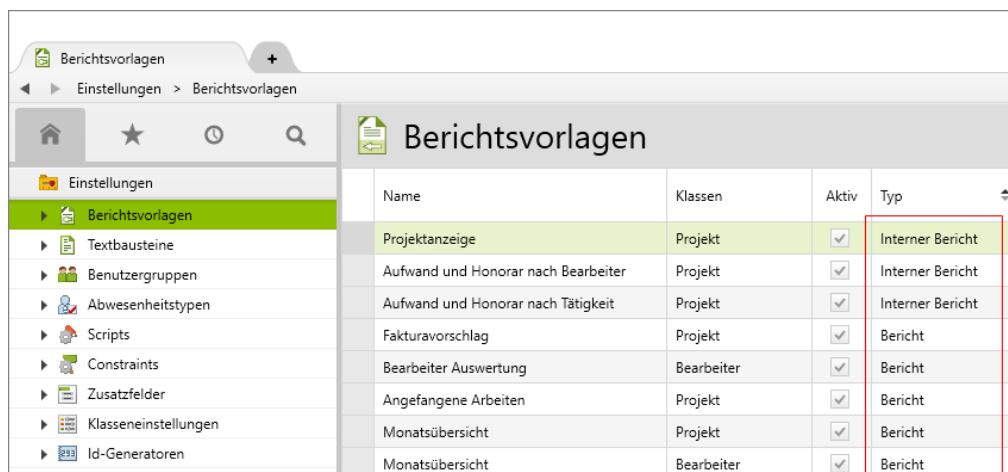
2.3 Vorankündigung Vertec 6.3

Mit der zukünftigen Vertec Version 6.3 werden die alten Reporting-Systeme **Interne Berichte** und **Bericht-Designer Berichte** (Rb-Reports) abgeschaltet.

Deshalb führt Vertec mit der vorliegenden Version 6.2 die erweiterten Office-Berichte ein (siehe Kapitel 4), welche die oben genannten Systeme ablösen.

Wichtig ist, dass Sie in der Zeit bis zur Vertec Version 6.3 Ihre Reports kontrollieren und diejenigen der veralteten Reporting-Systeme, welche Sie verwenden, auf das neue System überführen.

In den **Einstellungen > Berichtsvorlagen** geht es um alle Berichte, die vom Typ **Bericht** oder **Interner Bericht** sind:



Name	Klassen	Aktiv	Typ
Projektanzeige	Projekt	<input checked="" type="checkbox"/>	Interner Bericht
Aufwand und Honorar nach Bearbeiter	Projekt	<input checked="" type="checkbox"/>	Interner Bericht
Aufwand und Honorar nach Tätigkeit	Projekt	<input checked="" type="checkbox"/>	Interner Bericht
Fakturavorschlag	Projekt	<input checked="" type="checkbox"/>	Bericht
Bearbeiter Auswertung	Bearbeiter	<input checked="" type="checkbox"/>	Bericht
Angefangene Arbeiten	Projekt	<input checked="" type="checkbox"/>	Bericht
Monatsübersicht	Projekt	<input checked="" type="checkbox"/>	Bericht
Monatsübersicht	Bearbeiter	<input checked="" type="checkbox"/>	Bericht

Abbildung 1: Legacy Berichte in den Berichtsvorlagen

Bitte gehen Sie diese Liste durch und klären Sie ab, welche dieser Berichte Sie auch nach der Version 6.3 benötigen.



Vertec liefert einen Teil dieser Berichte standardmässig als erweiterte Berichte aus. Die entsprechende Liste finden Sie im Kapitel 4.

Diese von Vertec mitgelieferten erweiterten Office-Berichte werden in Vertec 6.2 inaktiv registriert und mit Version 6.3 automatisch aktiviert und die entsprechenden alten Berichte gelöscht. Wenn Sie diese Berichte ohne eigene Konfiguration "as is" verwenden, können Sie danach mit der gleichen Funktionalität weiterarbeiten. Ansonsten passen Sie bitte vorgängig auch diese Berichte gemäss Ihren Anforderungen an.

Bei Fragen wenden Sie sich bitte an Ihren Vertec-Betreuer.

3 Vertec Phone App

Linie: Standard, Expert | Modul: Leistung & CRM | Version 6.2

3.1 Installation und Konfiguration

Mit der Version 6.2 stellt Vertec die neue Phone App für iPhone und Android bereit. Diese steht zum Download zur Verfügung unter:

- App Store: <https://itunes.apple.com/app/id1198126173>
- Play Store: <https://play.google.com/store/apps/details?id=com.vertec.phoneapp>

Die entsprechenden Links sind auch auf dem Vertec Portal für Cloud Clients verfügbar (siehe www.vertec.com/de/kb/cloudportal).

Die Vertec Phone App ist wie die Web App und die Cloud App ein Cloud Client, jedoch nicht wie diese "full-featured", sondern eine "specialized" App.

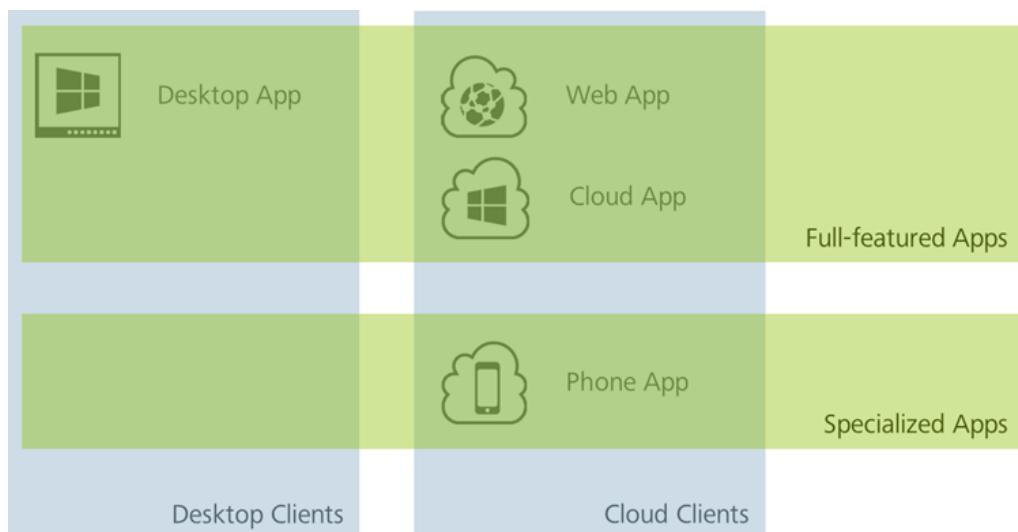


Abbildung 2: Die Phone App in der Vertec-Applikationslandschaft

Das bedeutet, dass die Phone App ebenfalls auf dem Cloud Server aufbaut, aber über eine eigene, spezialisierte Oberfläche sowie über einen reduzierten Funktionsumfang verfügt.

Der Zugriff erfolgt wie bei der Cloud App via den Vertec Webaccess-Service, welcher einen einfachen und per SSL verschlüsselten Zugriff aus dem Internet ermöglicht.

Sie können den Zugriff auf Vertec auch über den normalen Webserver betreiben, stellen also Ihren Server selbst ins Internet. In diesem Fall müssen Sie für die SSL-Konfiguration und Zertifikat sorgen und die Firewall-Einstellungen so vornehmen, dass Ihr Netzwerk nicht gefährdet ist.

Bitte beachten Sie, dass es grundsätzlich für alle möglich ist, sich via Internet an Ihrem Webserver anzumelden, sofern die URL und der Benutzernamen bekannt sind. Es ist deshalb äußerst wichtig, dass Sie für die Benutzer sichere Passwörter verwenden!

Weitere Informationen Detaillierte Informationen dazu finden Sie im Artikel [Vertec via Webaccess](#) unter www.vertec.com/de/kb/webaccess.

3.2 Angemeldet bleiben

Die Phone App unterstützt die Vertec **Angemeldet bleiben** Option. Beim Login wird automatisch ein Token erzeugt. Die Lebensdauer richtet sich nach den **Token Lifetime** Einstellungen im Vertec.ini File.

Angemeldet bleiben Weitere Informationen zur Angemeldet bleiben Option finden Sie im [Login-Artikel](http://www.vertec.com/de/kb/verteclogin#angemeldet-bleiben) unter www.vertec.com/de/kb/verteclogin#angemeldet-bleiben.

Ein Logout aus der Phone App entfernt den Token-Eintrag in der AuthToken-Tabelle.

Verliert die Phone App die Verbindung, wird bei einem Reconnect innerhalb des **Token Lifetime** das gleiche Token aktualisiert.

3.3 Phone App Session Timeout

Hat die Phone App keine Verbindung, wird eine entsprechende Meldung angezeigt.



Abbildung 3: Keine Netzwerkverbindung

Der Bearbeiter kann aber normal weiterarbeiten, solange die Session lebt. Die Dauer der Session wird (in Minuten) im Vertec.ini-File unter **Phone App Session Timeout** eingestellt:

```
[CloudServer]  
Phone App Session Timeout=75
```

Der eingetragene Wert muss grösser als 0 und kleiner gleich 2147483647 sein.

Ist nichts gesetzt, wird standardmässig der Wert von 60 [Minuten] angewandt.

Hinweis Achtung bei ausgeschaltetem Token Lifetime (siehe 3.2): Schalten Sie genügend Session-Zeit ein, sonst muss sich der Benutzer dauernd wieder neu anmelden.

3.4 Mindestanforderungen

Die Vertec Phone App unterstützt die Betriebssysteme Android ab Version 4.0.3 und iOS ab Version 8.



3.5 Login

Ein Klick auf das Vertec Icon startet die Applikation.



Abbildung 4: Das Phone App Icon

Die Sprache der Vertec Phone App richtet sich nach der System-Sprache des mobilen Geräts, mit welchem sie aufgerufen wird, sofern dieses eine der Vertec-Sprachen eingestellt hat (EN, DE, IT, FR). Ansonsten wird sie auf Englisch angezeigt.

Beim ersten Aufstarten der Phone App öffnen sich die Phone App Einstellungen. Hier geben Sie den Pfad zum Server an.

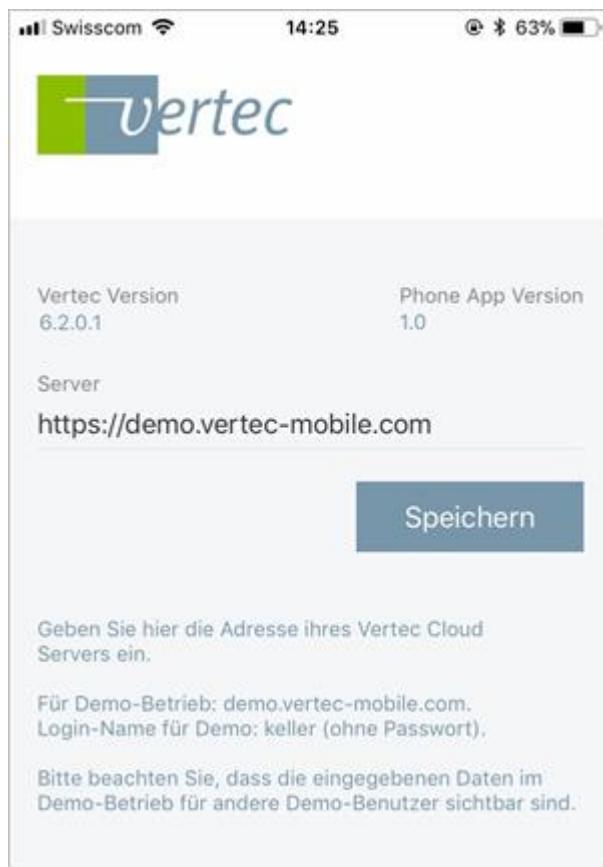


Abbildung 5: Server-Pfad Angabe in den Phone App Einstellungen

Danach erscheint der Login-Dialog:

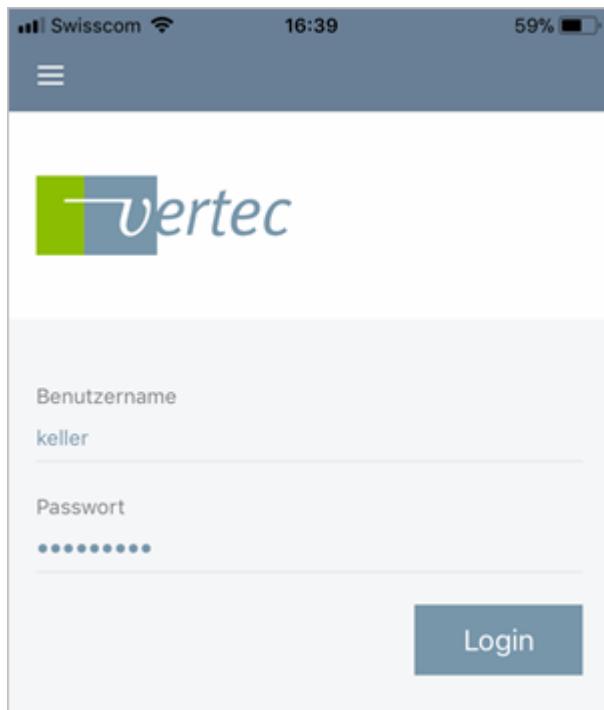


Abbildung 6: Der Phone App Login-Dialog

Die Phone App startet standardmässig mit der Leistungsliste des heutigen Tages.



Abbildung 7: Phone App Leistungsliste

Der Punkt neben dem Wochentag zeigt den heutigen Tag an. Der grüne Strich unterhalb zeigt den selektierten Tag. Durch Klick auf das Icon oben rechts kann man jederzeit wieder zum heutigen Tag in der Liste navigieren.

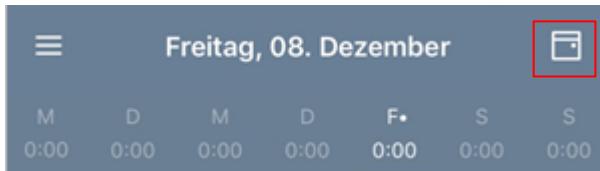


Abbildung 8: Zum heutigen Tag navigieren

3.6 Leistungen erfassen

Ein Klick auf den + Button wird eine neue Leistung erstellt. Zuerst erscheint die Projektliste für die Auswahl des Projektes.

Vorangezeigt werden alle Projekte, auf die der eingeloggte Bearbeiter innerhalb des letzten Monats Leistungen erfasst hat und auf die er erfassen darf.

Über die Lupe kann nach weiteren Projekten gesucht werden.

Sobald das Projekt ausgewählt ist, erscheint die Detailmaske der Leistung. Nun kann entweder der Timer gestartet oder der Aufwand eingegeben sowie die weiteren Angaben gemacht werden.

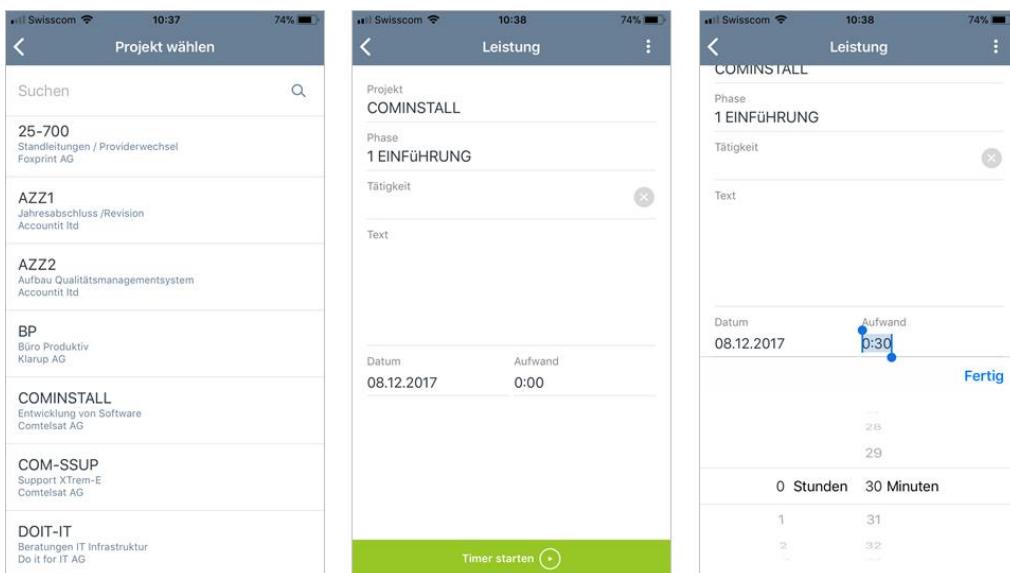


Abbildung 9: Eine Leistung erfassen

Eine Leistung mit gestartetem Timer erscheint in der Liste mit dem grünen Timer-Symbol:

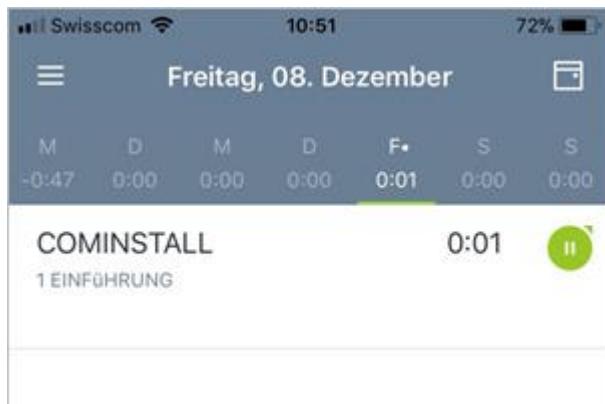


Abbildung 10: Leistung mit gestartetem Timer

Durch Klick auf das Symbol wird der Timer gestoppt.

3.7 Menü

Alle weiteren Möglichkeiten finden sich über das Phone App Menü, welches über das Menü-Symbol ausgeklappt wird:

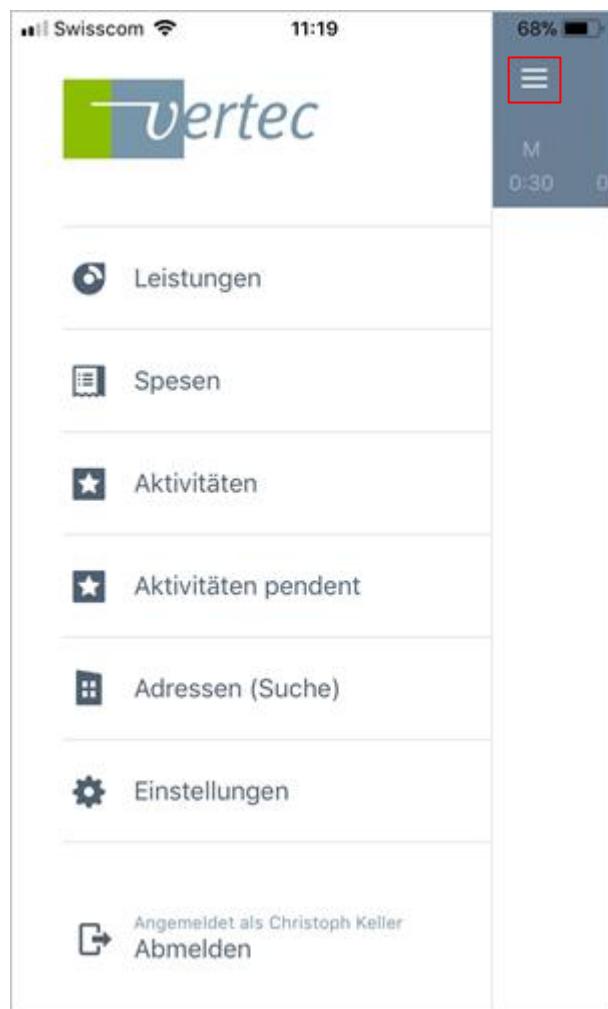


Abbildung 11: Das Phone App Menü

3.8 Spesen erfassen

Die Spesen erfassung erfolgt analog der Leistungserfassung. Für die Brutto-Erfassung sowie die Rückerstattung-Option stehen Buttons zur Verfügung:

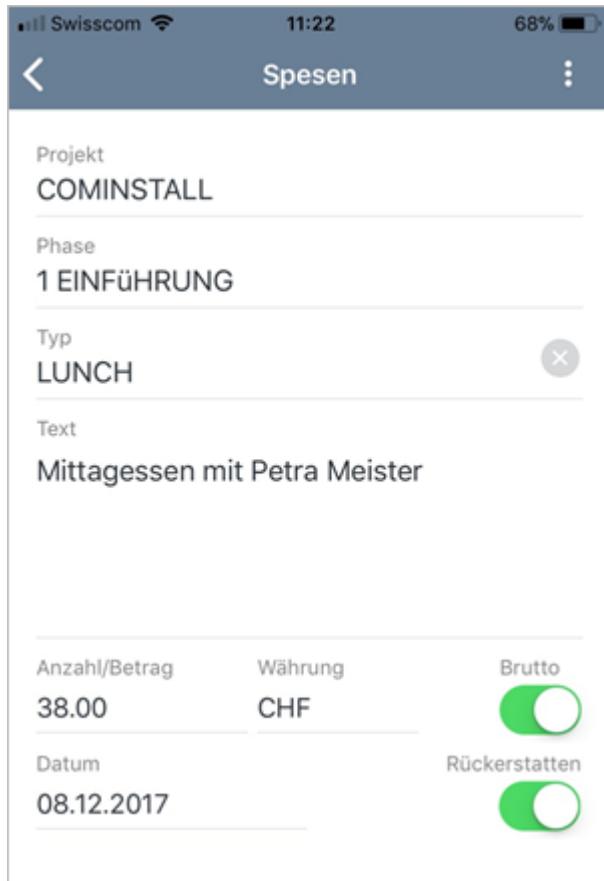


Abbildung 12: Spesen-Detail

Unter Währung kann die Erfassungswährung bei Bedarf umgestellt werden (Linie Expert):



Abbildung 13: Einstellen der Erfassungswährung

3.9 Aktivitäten erfassen

Über den Menüpunkt Aktivitäten werden die Aktivitäten erfasst.

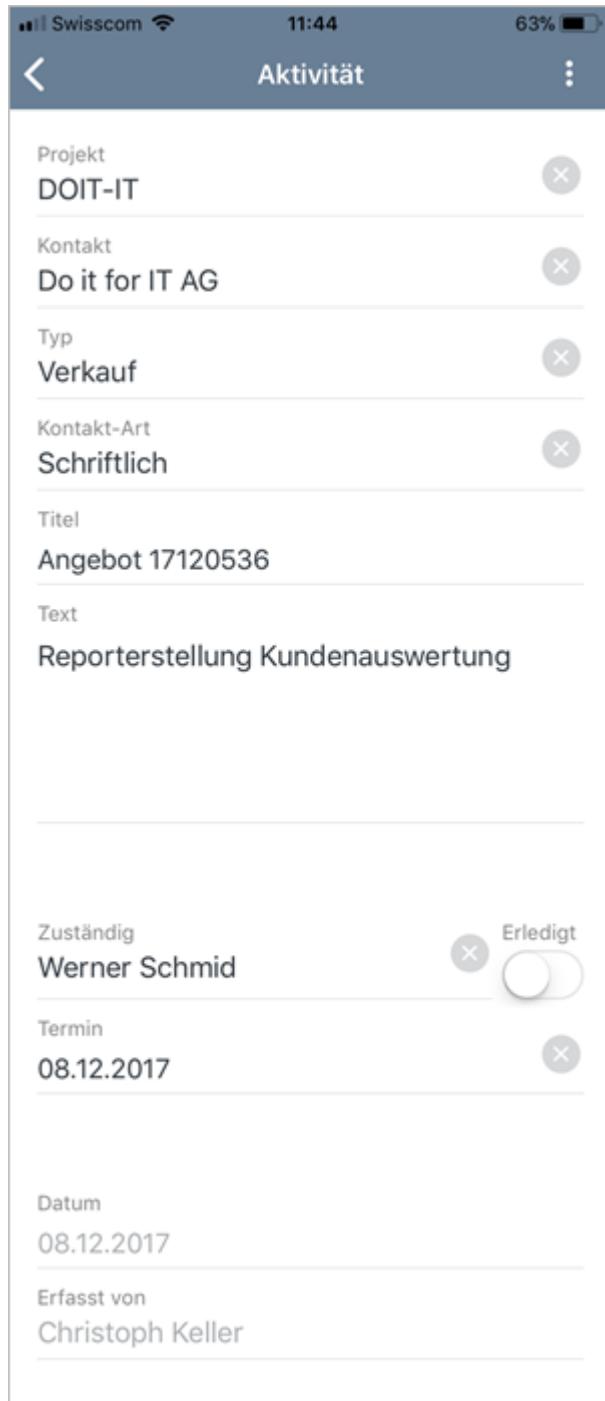
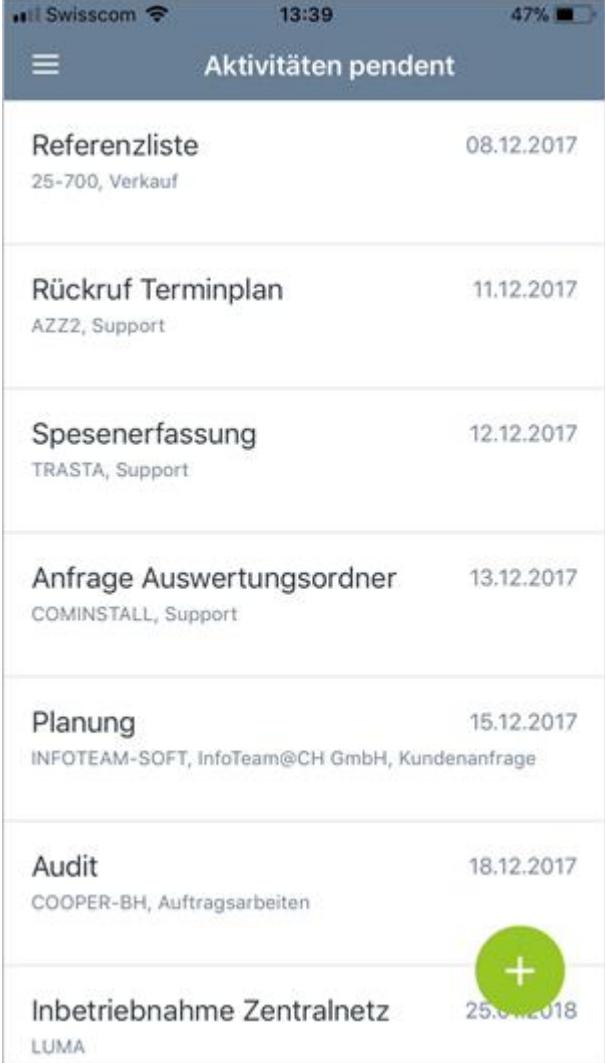


Abbildung 14: Eine Phone App Aktivität (gesamte Ansicht inkl. Scrolling)

Im oberen Teil werden die Details der Aktivität erfasst. Im unteren Teil kann ein Zuständiger sowie ein Datum erfasst werden. Die Aktivität erhält somit Pendenzcharakter und erscheint bei der zuständigen Person unter **Aktivitäten pendent/Aktivitäten offen** (Name je nach eingesetzter Sprache **Schweiz** oder **Deutschland**).

3.10 Aktivitäten pendent/offen

Die Liste Aktivitäten pendent/offen bringt alle noch nicht erledigten Aktivitäten des eingelogten Benutzers:



Aktivitäten pendent	
Referenzliste	08.12.2017
25-700, Verkauf	
Rückruf Terminplan	11.12.2017
AZZ2, Support	
Spesenerfassung	12.12.2017
TRASTA, Support	
Anfrage Auswertungsordner	13.12.2017
COMINSTALL, Support	
Planung	15.12.2017
INFOTEAM-SOFT, InfoTeam@CH GmbH, Kundenanfrage	
Audit	18.12.2017
COOPER-BH, Auftragsarbeiten	
Inbetriebnahme Zentralnetz	25.01.2018
LUMA	

Abbildung 15: Liste der offenen Aktivitäten

Nach Erledigung kann die Aktivität in der Detailansicht auf **Erledigt** gestellt werden. Sie verschwindet dann automatisch aus der Liste der offenen Aktivitäten.

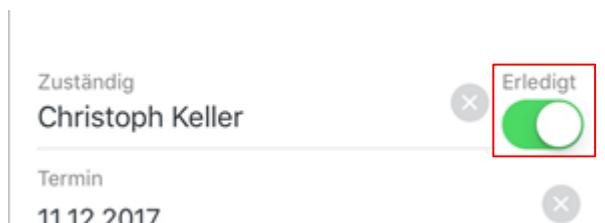


Abbildung 16: Eine Aktivität wird auf Erledigt gestellt

3.11 Einen Eintrag löschen

In der Detailansicht kann über den Button mit den drei Punkten das Kontextmenü aufgerufen werden.



Abbildung 17: Einen Phone App Eintrag löschen

Um den Eintrag zu löschen, klicken Sie auf **Eintrag löschen**. Der Eintrag wird ohne weitere Abfrage gelöscht.

3.12 Adressen einsehen

In der Phone App können keine neuen Adressen erfasst, aber die bestehenden eingesehen werden. Durch Klick auf den Menüpunkt **Adressen (Suche)** gelangen Sie in die Suchmaske. Geben Sie hier den gesuchten Begriff ein und klicken Sie auf die Lupe.

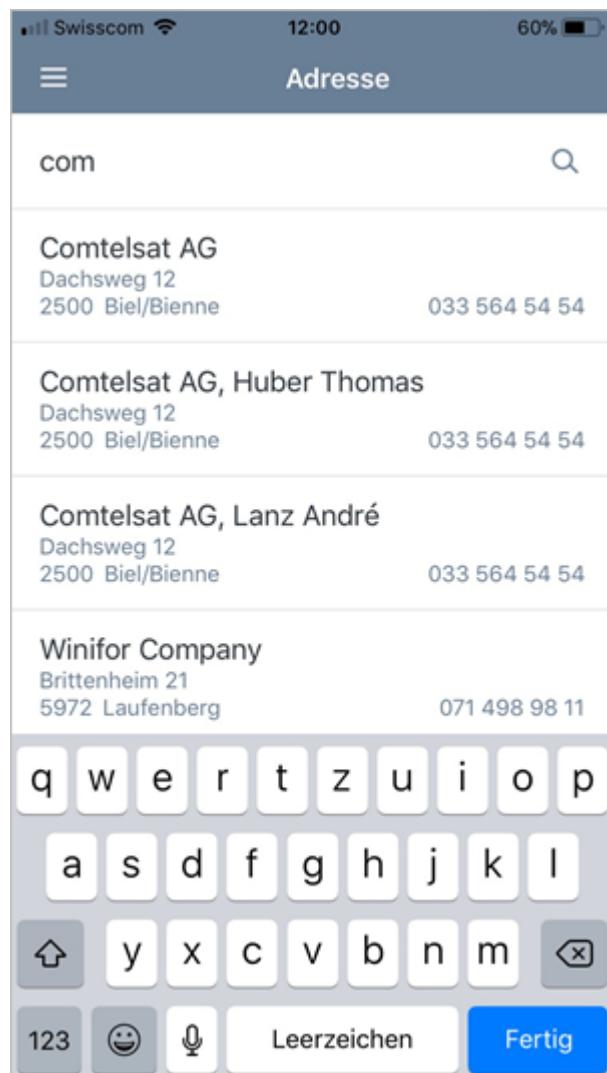


Abbildung 18: Suche nach Adressen in der Phone App

Ein Klick auf eine Adresse in der Liste öffnet die Detailansicht.

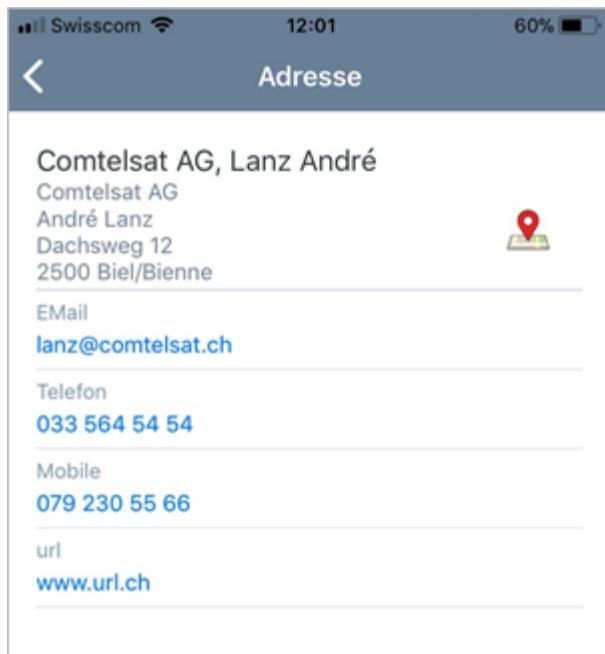


Abbildung 19: Eine einzelne Adresse in der Phone App

Von hier aus können Sie direkt telefonieren, eine E-Mail versenden, die Website aufrufen oder die Adresse in der Karten-App auf Ihrem Smartphone öffnen.

4 Erweiterte Office-Berichte

Linie: Standard, Expert | Modul: Leistung & CRM | Version 6.2

Office-Berichte, insbesondere Word-Berichte, gehören fest zu Vertec und haben sich sehr bewährt. Word-Berichte können seit Vertec 6.1 auch Vertec-generiert erstellt werden und stehen darum auch in der Web App zur Verfügung. Allerdings eignen sich Word-Berichte nicht für komplexe Auswertungen, weil kein Abfragedialog angezeigt werden kann und es schwierig ist, komplexe Logik im Bericht zu formulieren.

Ausserdem gibt es in Vertec noch 2 Bericht-Systeme, welche nicht cloud-fähig sind und auf die zukünftige Vertec Version 6.3 eingestellt werden (siehe 2.3).

Deshalb stellt Vertec einen neuen, Office-basierten und cloud-fähigen Reportgenerator zur Verfügung: Erweiterte Office-Berichte.

Erweiterte Office-Berichte bestehen aus der Kombination einer Word-Vorlage (ein .docx-File) für die Layout-Struktur, sowie einer Bericht-Definition in Form von Python Code für die Generierung des Inhalts.

Durch Vertec mitgelieferte erweiterte Office-Berichte

Mit Vertec 6.2 werden von Vertec folgende erweiterten Office-Berichte mitgeliefert. Diese sind im Standard Report-Verzeichnis abgelegt und in Vertec registriert, jedoch inaktiv.

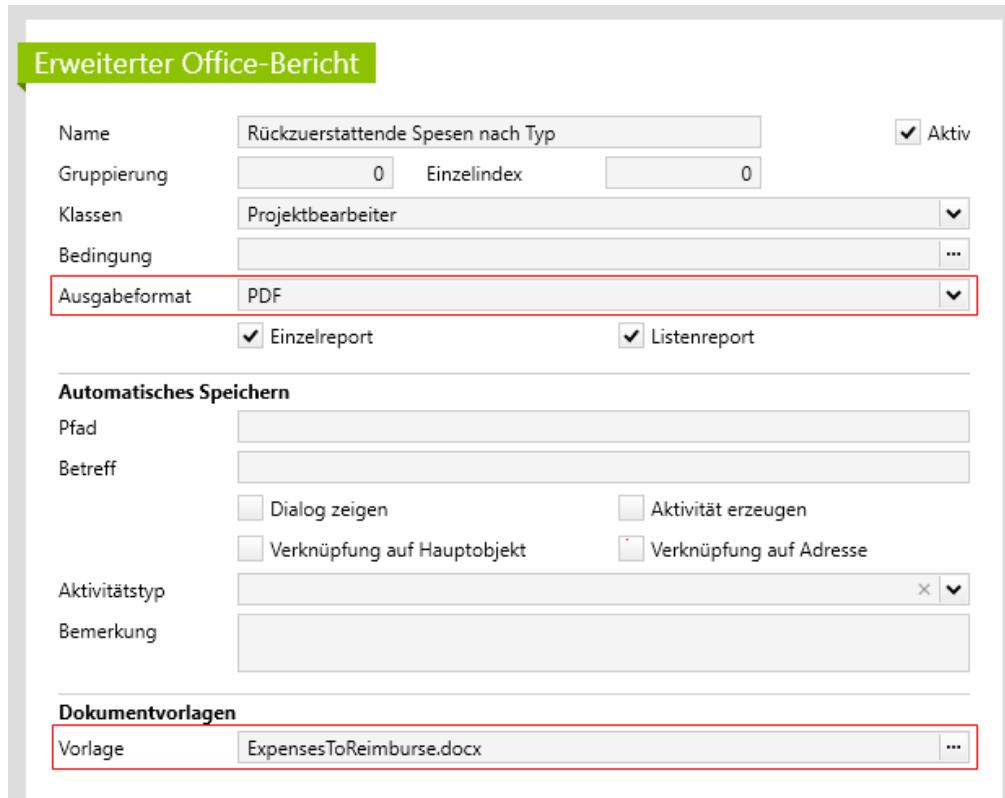
Mit Vertec 6.3 werden die aktuellen Berichtregistrierungen gelöscht und die erweiterten Office-Berichte aktiviert. Stellen Sie bis dahin sicher, dass alle entsprechenden Berichte, die Sie benötigen, als erweiterte Office-Berichte vorliegen. Bitte beachten Sie dazu das Kapitel 2.3.

Report	Registriert auf	Vorlagendokument
Fakturavorschlag	Projekt/Mandat	InvoiceProposal.docx
Angefangene Arbeiten	Projekt/Mandat	CommencedWork.docx
Ferienabrechnung	Bearbeiter	VacationReport.docx
Rekapitulation Buchungen	Projekt/Mandat	RecapitulationOfBookings.docx
Rückzuerstattende Spesen nach Typ	Bearbeiter	ExpensesToReimburse.docx
Überstundenkontrolle	Bearbeiter	OvertimeReport.docx

Als Ausgabeformat ist bei all diesen Berichten PDF voreingestellt. Falls Sie die Berichte stattdessen als Word-Dateien ausgeben möchten, können Sie dies in der entsprechenden Berichtregistrierung ändern.

4.1 Registrierung erweiterte Office-Berichte

Im Vergleich mit der herkömmlichen Office-Bericht Registrierung gibt es bei den erweiterten Office-Berichten folgende Unterschiede:



Erweiterter Office-Bericht

Name	Rückzuerstattende Spesen nach Typ	<input checked="" type="checkbox"/> Aktiv
Gruppierung	0 Einzelindex	0
Klassen	Projektbearbeiter	...
Bedingung	...	
Ausgabeformat	PDF	<input checked="" type="checkbox"/> Einzelreport <input checked="" type="checkbox"/> Listenreport
Automatisches Speichern		
Pfad		
Betreff		
	<input type="checkbox"/> Dialog zeigen	<input type="checkbox"/> Aktivität erzeugen
	<input type="checkbox"/> Verknüpfung auf Hauptobjekt	<input type="checkbox"/> Verknüpfung auf Adresse
Aktivitätstyp	x	
Bemerkung		
Dokumentvorlagen		
Vorlage	ExpensesToReimburse.docx	...

Abbildung 20: Registrierung eines erweiterten Office-Berichts

- Da die erweiterten Office-Berichte immer Vertec-generiert sind, entfällt die entsprechende Auswahl. Als Ausgabeformat kann aber nach wie vor gewählt werden zwischen Word und PDF.
- Unter **Vorlage** wird das entsprechende Vorlagendokument ausgewählt. Im Gegensatz zu den bisherigen Word-Reports (.dotx) ist dieses ein normales Word-Dokument mit der Dateiendung **.docx**.
- Es gibt nur noch eine Vorlage. Die Liste von Vorlagen pro Sprache, wie sie bei Office-Reports verwendet wird, entfällt. Die Übersetzungen erfolgen durch den normalen Vertec Übersetzungsmechanismus (siehe auch 4.12).

4.2 Bericht-Definition

Der Python Code zur eingefügten Vorlage wird bei der Bericht-Registrierung auf der Seite [Bericht-Definition](#) eingefügt.

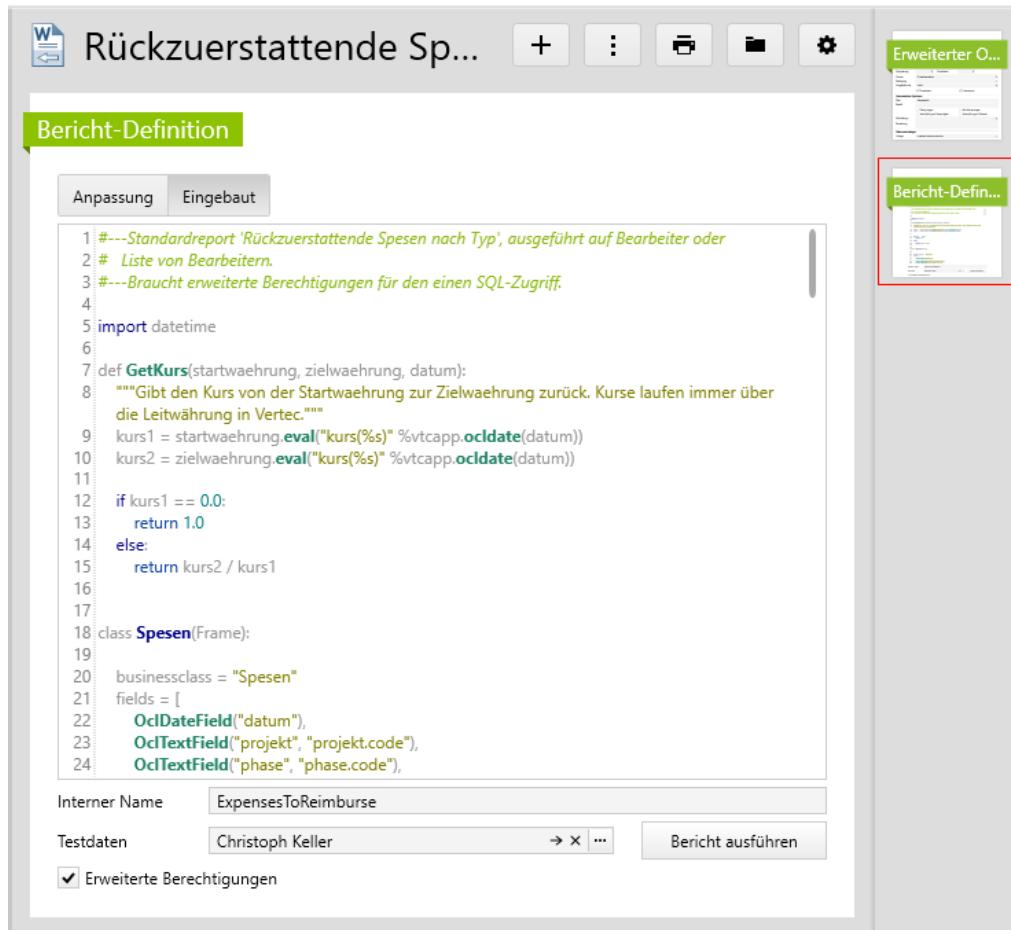


Abbildung 21: Bericht-Definition eines erweiterten Office-Berichts

Der interne Name bezeichnet den von Vertec mitgelieferten Code. Dieser heisst gleich wie der mitgelieferte Report und wird unter dem Reiter [Eingebaut](#) angezeigt.

Unter dem Reiter [Anpassung](#) kann der mitgelieferte Code überschrieben werden. Sobald unter Anpassung Code vorhanden ist, gilt nur noch dieser. Um etwas zu ändern, muss also der komplette Code aus Eingebaut kopiert und dann unter Anpassung geändert werden.

Vom Kunden selbst oder auf Auftrag erstellte erweiterte Office-Berichte enthalten keinen eingebauten Code und keinen Internen Namen, sondern der Code wird direkt im Feld Anpassung eingefügt.

Der Aufbau des Python Codes ist im Kapitel 4.3 genau beschrieben.

Um den Code zu testen, kann hier bei [Testdaten](#) ein Objekt aus Vertec ausgewählt werden. Ein Klick auf den Button [Bericht ausführen](#) führt dann den Report direkt für dieses Objekt aus.

Falls es im Python Code erweiterte Berechtigungen braucht, z.B. für einen SQL-Zugriff, muss das Häkchen [Erweiterte Berechtigungen](#) unterhalb der Bericht-Definition gesetzt werden.

4.3 Aufbau des Python Codes

Die Datenlogik eines erweiterten Office-Berichts basiert auf einer Hierarchie von sogenannten Frames. Ein Frame ist vergleichbar mit einer Tabelle, es hat Zeilen mit Datenfeldern drauf. Die Felder eines bestimmten Frames werden mit Hilfe von Feld-Definitionen festgelegt.

Der Python Code für einen erweiterten Office-Bericht besteht im Wesentlichen aus der Deklaration der Frames, startend mit dem Haupt-Frame (`main_frame`) des Reports.

Eine minimale Bericht-Definition muss eine Frame-Deklaration sowie die Zuweisung der `main_frame` Variable enthalten:

```
# Beispiel eines minimalen Mitarbeiter-Reports

class Bearbeiter(Frame):
    businessclass = "Projektbearbeiter"
    fields = [OclTextField("name"), OclTextField("kuerzel")]

main_frame = Bearbeiter
```

Hier wird ein Bearbeiter-Frame mit 2 Feldern, welche per OCL berechnet werden, definiert.

4.4 Feld-Typen

Es stehen folgende Typen von Feldern zur Verfügung. Die Feld-Typen werden jeweils mit Name und optionalen Berechnungs-Angaben definiert. Von jedem Typ (ausser `FrameField`) gibt es noch die Variante mit `Ocl` als Präfix (siehe `OCL Felder` weiter unten).

TextField	stellt einen String-Wert dar
CurrencyField	stellt eine Fixkomma Zahl dar, wird üblicherweise für Geld-Beträge verwendet. Formatierung gemäss Ländereinstellung.
IntegerField	stellt eine Ganzzahl dar
MinuteField	stellt einen Integer-Wert als Minuten dar. Formatierung gemäss Einstellung in Vertec.
BooleanField	stellt einen Wahr/Falsch Wert dar
DateField	stellt ein Datum dar
DateTimeField	stellt ein Datum mit Zeitanteil dar
ImageField	stellt ein Bild dar
FrameField	enthält als Wert ein weiteres Frame. Frame-Fields erlauben die Darstellung von hierarchischen Datenstrukturen.

Bei Frame-Feldern muss als zweiter Parameter immer der Frame-Typ angegeben werden, der dargestellt wird.

```

1 # Beispiel eines minimalen Mitarbeiter-Reports
2 class Bearbeiter(Frame):
3     businessclass = "Projektbearbeiter"
4     fields = [OclTextField("name"),
5               OclTextField("kuerzel"),
6               FrameField("leistungen", "Leistungen", "calculate_leistungen")]
7 ]
8
9     def calculate_leistungen(context):
10         leistungen = context.evalocl("offeneleistungen")
11         return Leistungen(context, leistungen)
12
13 main_frame = Bearbeiter
14
15 class Leistungen(Frame):
16     businessclass = "OffeneLeistung"
17     fields = [OclDateField("datum"),
18               OclTextField("projekt", "projekt.code"),
19               OclTextField("text"),
20               OclMinuteField("minutenext"),
21               OclCurrencyField("ansatzext"),
22               OclCurrencyField("wertext")]
23 ]

```

Abbildung 22: Angabe eines FrameFields

Der Frame-Typ kann als direkte Referenz oder als Name eines deklarierten Frames angegeben werden. Die Referenz via Name erlaubt die beliebige Anordnung der Frame-Deklarationen im Code. Bei Verwendung der direkten Referenzen muss die referenzierte Frame-Deklaration vorher im Code stehen.

Berechnete Felder

Die Felder eines Frames können automatisch berechnet werden. Die Berechnung erfolgt via OCL oder in Python Code.

OCL Felder

Ein Ocl-Field (OclTextField, OclCurrencyField, OclBooleanField etc.) wird unter Angabe eines Feld-Namens erstellt. Im einfachsten Fall ist der Feld-Name auch gleich die OCL Expression, die zum Berechnen verwendet wird:

```
OclTextField("code")
```

Optional kann die OCL Expression als separates, zweites Argument angegeben werden. Das ist insbesondere dann notwendig und sinnvoll, wenn es sich um eine längere Expression handelt.

```
OclTextField("name", "projektleiter.name")
```

OCL-Felder sind nur auf Businessobject-Frames zulässig, das heisst auf Frames, die mittels **businessclass=** einer bestimmten Klasse zugeordnet sind.

Manuelle bzw. in Python Code berechnete Felder

Die Feld-Typen ohne Ocl-Präfix müssen nur einen Namen erhalten. Ohne weitere Angaben stellen diese nicht automatisch berechnete Felder dar, die beim Berechnen des Frames im Code mit Werten gefüllt werden können.

Optional kann als zweites Argument eine Python Funktion angegeben werden, die zur Berechnung des Feldes aufgerufen wird. Die Funktion kann als direkte Funktions-Referenz oder als Name einer Funktion, welche auf dem Frame definiert ist (string) angegeben werden.

Berechnung von Feldern in Code

Bei der Berechnung eines Feldes mit Code, wird dem Feld eine Funktions-Referenz (direkt Referenz oder Name der Funktion) übergeben.

Die Berechnungs-Funktion muss ein Argument für den Berechnungs-Kontext enthalten und den berechneten Feld-Wert zurückgeben.

```
class Rechnung(Frame):
    fields = [CurrencyField("summe", "calc_summe")]

    def calc_summe(context):
        rechnung = context.currentobject
        sum = 0.0

        for leist in rechnung.leistungen:
            sum += leist.wertext

        return sum
```

Berechnung von Frame-Fields

Auch Felder vom Type `FrameField` können berechnet sein. Falls die Berechnung per OCL Expression erfolgt, muss das Ergebnis der Expression eine Liste von Businessobjekten sein, welche der Frame Deklaration entsprechen.

Falls die Berechnung per Python Methode erfolgt, muss die Methode eine Frame-Instanz des richtigen Typs zurückgeben.

In den meisten Fällen wird dabei die Frame-Instanz im Code aufgebaut. Dafür steht auf dem Frame Objekt eine `add_row` Methode zur Verfügung, welche ein zeilenweises Aufbauen des Frames erlaubt. Die auf dem Frame definierten Felder stehen als Eigenschaften des `Row`-Objektes zur Verfügung.

```
class DetailsFrame(Frame):
    businessclass="Projektphase"
    fields = [
        TextField("code"),
        CurrencyField("value1"),
        CurrencyField("value2")]

class InvoiceFrame(Frame):
    fields = [
        FrameField("details", DetailsFrame, calculate_details)]

    def calculate_details(context):
        phasen = context.evaluate('phasenaufrechnung')

        frame = DetailsFrame(context)

        for ph in phasen:
            row = frame.add_row(ph)
            row.value1 = 25.10
            row.value2 = 17.20
```

Berechnetes Main-Frame

Das Main-Frame des Berichts ist kein Field-Value, kann aber ebenfalls via Funktion oder OCL berechnet werden.

Das geschieht mittels `calculate_main_frame` in der Report-Definition. Folgende Varianten sind möglich:

- `calculate_main_frame` ist eine Funktion, welche ein Context-Objekt erhält und eine Frame-Instanz des richtigen Typs zurückgeben soll.

Signatur der Funktion ist `calculate_main_frame(context)`, Rückgabewert ist eine Frame Instanz.

Die Implementation sollte den Typ des erzeugten Frames anhand der Report-Definition überprüfen.

Beispiel:

Der Report ist auf einer Liste von Projekten registriert, das Main-Frame stellt aber eine Liste von Bearbeitern dar. Die Main-Frame Klasse heisst "Bearbeiter".

```
main_frame=Bearbeiter

def calculate_main_frame(context):
    bearbeiter_list = context.evalocl("bearbeiter")
    frame = Bearbeiter(context)

    for b in bearbeiter_list:
        row = frame.add_row(b)
        row.specialvalue = calculate_some_value()
    return frame
```

- `calculate_main_frame` ist ein string. Dieser wird als OCL Expression ausgewertet und muss eine Liste ergeben. Die Expression wird auf der ursprünglichen Liste des Reports ausgewertet und hat auch die OCL Variablen im Context zur Verfügung.

Beispiel, wie oben, aber nur mit Expression:

```
main_frame = Bearbeiter
calculate_main_frame = "bearbeiter"
```

4.5 Das Context-Objekt

Jeder Berechnung im Rahmen der Report-Generierung wird das Context-Objekt mitgegeben. Dieses ist damit in allen Berechnungsfunktionen verfügbar und hat folgende Eigenschaften:

Variablen

Die folgenden Variablen sind in jedem Report definiert:

currentobject Das jeweils aktuelle Objekt der Berechnung.

```
context.currentobject
```

currentdate Das heutige Datum, ohne Zeitteil.

```
context.currentdate
```

optarg	Das optionale Adressen-Argument bei Berichten. Um mit dem Objekt weiterzuarbeiten, kann wie folgt darauf zugegriffen werden:
--------	--

```

class Projekt(Frame):
    fields = [
        OclTextField("code"),
        OclTextField("beschrieb"),
        OclTextField("creationdatetime"),
        TextField('adresstext', 'calcadresse'),
    ]
    def calcadresse (context):
        return context.optarg.adresstext

main_frame = Projekt

```

rootlist	Die Liste, auf der der Report ausgeführt wurde. Normalerweise ist das die eintraege -Liste des Containers, auf dem der Report ausgeführt wurde, beziehungsweise eine Liste mit dem Objekt drin, auf dem der Report ausgeführt wurde.
----------	---

container	Der Container, auf dem der Report ausgeführt wurde.
-----------	---

var<frame>	Für jedes übergeordnete Frame ist eine Variable mit dessen aktuellem Objekt definiert.
------------	--

var<frame>List	Für jedes übergeordnete Frame ist eine Variable mit der Liste der Objekte des Frames definiert.
----------------	---

Weitere Variablen können selbst definiert und dem Context-Objekt zugewiesen werden.

```

if vtcapp.getpropertyvalue('Firma') <> None:
    context.firma = vtcapp.getpropertyvalue('Firma')

```

Diese können später auf dem Report direkt über eine Context Expression ausgegeben werden (siehe Kapitel 4.7).

Methode evalocl(expression)

Ausserdem verfügt das Context-Objekt über die Methode **evalocl(<expression>)**, welche eine OCL-Expression auf dem aktuellen Objekt (**currentobject**) auswertet.

```

phase = context.currentobject
context.leistungen.evalocl("self->select
    (phase.boldid=%s).wertext->sum" % (phase.objid))

```

4.6 Before Report Logik

Um vor der Ausführung des Reports Python Code ausführen zu können (um beispielsweise Dialoge anzuzeigen und Parameter auf dem Context-Objekt zu setzen), gibt es die Methode **before_report(context)**:

```

def before_report(context):
    """Frage den User nach dem Datum ab."""
    initialValues = {}
    initialValues["Stichdatum"] = vtcapp.currentdate()
    dlgDefinition = """
        <Dialog Title="{Translate 'Choose date'}" Width="400">

```

```
<Group Orientation="Vertical">
    <DatePicker Name="Stichdatum" Label="Stichdatum" />
</Group>
<Dialog.Buttons>
    <Button Text="OK" IsAccept="True" Command="{Binding OkCommand}" />
    <Button Text="Cancel" IsCancel="True" Command="{Binding CancelCommand}" />
</Dialog.Buttons>
</Dialog>
"""

ok, values = vtcapp.showcustomdialog(dlgDefinition, initialValues)
if not ok:
    return False

context.stichdatum = values["Stichdatum"]
```

4.7 Aufbau der Word-Vorlagen

Die erweiterten Office-Berichte verwenden im Gegensatz zu den bisherigen Word-Reports (.dotx) ganz normale Word-Dokumente (.docx) als Vorlagen.

4.8 Expressions

Da die Berechnungen neu im Code stattfinden und nicht mehr im Report, gibt es keine langen OCL-Expressions mehr direkt im Word-Bericht, sondern nur noch Referenzen auf Felder, die im Report-Code definiert sind.

Die Steuerung erfolgt nach wie vor über ein durch einen Kommentar markierten Text innerhalb eines Bandes. Die Referenzen befinden sich aber nicht wie bisher im Kommentar, sondern im Text, der markiert wird. Der Kommentar ist leer oder enthält das Schlüsselwort **translation**, um die Übersetzung für dieses Feld zu aktivieren (siehe unten).

4.9 Bands

Nach wie vor befindet sich alles, was gedruckt wird, in Bands. Es gibt also immer mindestens ein Haupt-Band, welches über alles geht. Sublisten innerhalb befinden sich ebenfalls wie bisher in einem Band.

Es gibt wie bisher Exp-, Cond-, und Locale-Bänder. Dabei gilt ebenfalls der Text, der markiert wird, als Steuerung (siehe oben). Die Referenz auf den Code wird mit **:name** angegeben. Als Band-Expression wird also zum Beispiel **bndSpesenExp:spesen** geschrieben, dieser Text markiert und mit einem leeren Kommentar versehen:

bndSpesenExp:spesen [sth1]

Weitere Informationen Weitere Informationen zu den Bändern in Word-Berichten finden Sie in der Online Knowledge Base unter <https://www.vertec.com/de/kb/wordreports>.

4.10 Locale

Zahlenwerte werden nach den lokalen Regionaleinstellungen formatiert. Mit Locale-Expressions können Zahlenwerte nach anderen Ländereinstellung formatiert werden.

Die Locale-Expressions werden analog zu Band- und Conditional-Expressions im Report Template formuliert. Als Präfix zur Kennzeichnung einer Locale-Expression wird **<Band-**

`Name>Locale` verwendet. Zum Festlegen von Regionaleinstellungen auf dem "bndLeistungen" Band wird also zum Beispiel `bndLeistungenLocale:locale` geschrieben, dieser Text markiert und mit einem leeren Kommentar versehen:

Datum	Projekt	Text	Aufwand	Ansatz	Honorar
<code>(datum)</code>	<code>(projekt)</code>	<code>(text)</code>	<code>bndLeistungenLocale:locale</code>	<code>(minutenext)</code>	<code>(ansatz)</code>

[sth4]

[sth5]

[sth6]

[sth7]

[sth8]

[sth9]

[sth10]

[sth11]

Abbildung 23: Locale-Band

Der Ländercode kann auch direkt angegeben werden, z.B.:

`bndLeistungenLocale:de-DE`

Alle Zahlenwerte innerhalb dieses Bands werden dann nach den angegebenen Ländereinstellungen formatiert.

Im Code wird dafür im entsprechenden Frame ein `OclTextField` mit dem angegebenen Namen (hier im Beispiel `locale`) eingefügt:

```

15 class Leistungen(Frame):
16     businessclass = "OffeneLeistung"
17     fields = [OclDateField("datum"),
18               OclTextField("projekt", "projekt.code"),
19               OclTextField("text"),
20               OclMinuteField("minutenext"),
21               OclCurrencyField("ansatz", "if minutenext > 0 then wertext/minutenext*60 else ansatztext endif"),
22               OclCurrencyField("wertext"),
23               OclTextField("locale", "de-DE")]
24

```

Abbildung 24: Feld für die Angabe der Locale-Expression

Resultat der angegebenen OCL-Expression muss ein Ländercode sein. Dieser kann entweder direkt angegeben werden wie hier oben im Beispiel ('`de-DE`'), oder z.B. aus einem Feld ausgelesen werden: `OclTextField("locale", "projekt.zusatzfeldasstring('locale')")`.

Weitere Informationen Weitere Informationen zu Locale-Expressions finden Sie in der Online Knowledge Base unter <https://www.vertec.com/de/kb/wordreports#locale>.

4.11 Context-Expressions

Mittels sogenannten Context-Expressions können Variablen direkt auf dem Report-Context ausgegeben werden, ohne dass dafür extra ein Feld auf einer Frame-Definition eingeführt werden muss.

Eine Context-Expression ist durch den Präfix `context:` gekennzeichnet und referenziert eine Context-Variable des Reports (siehe 4.5). Beispiel:

```
context:stichdatum
```

Der Datentyp des Feldes richtet sich nach dem Wert der Context-Variable. Falls es sich bei der Context-Variable um eine Objekt-Referenz oder eine Liste handelt, gibt die Context-Expression als Wert einen String (String-Representation) zurück.

Anwendungsfall: Auswertungs-Reports zum Beispiel beinhalten oft ein Band, welches iteriert wird; der Rest des Reports besteht jedoch vorwiegend aus statischen Inhalten wie Titel etc. Reports dieser Art können so implementiert werden, dass das iterierte Band als Main-Band (und damit als main_frame) definiert ist. Möchte man nun beispielsweise im Titel trotzdem einen Report-Parameter angeben, z.B. das Stichtdatum der Auswertung, muss dafür kein neues Rahmenframe mit berechnetem FrameField eingeführt werden, sondern es kann direkt die Context-Expression angegeben werden.

4.12 Übersetzungen

Es gibt keine verschiedenen Word-Vorlagen mehr für die verschiedenen Sprachen. Stattdessen wird auf das Vertec Übersetzungssystem zurückgegriffen.

Um ein Feld zu übersetzen, wird als Kommentar das Schlüsselwort `translation` eingetragen. Der Übersetzungsmechanismus schaut dann in der Übersetzungsdatei nach dem Begriff und setzt, falls vorhanden, die entsprechende Übersetzung ein.

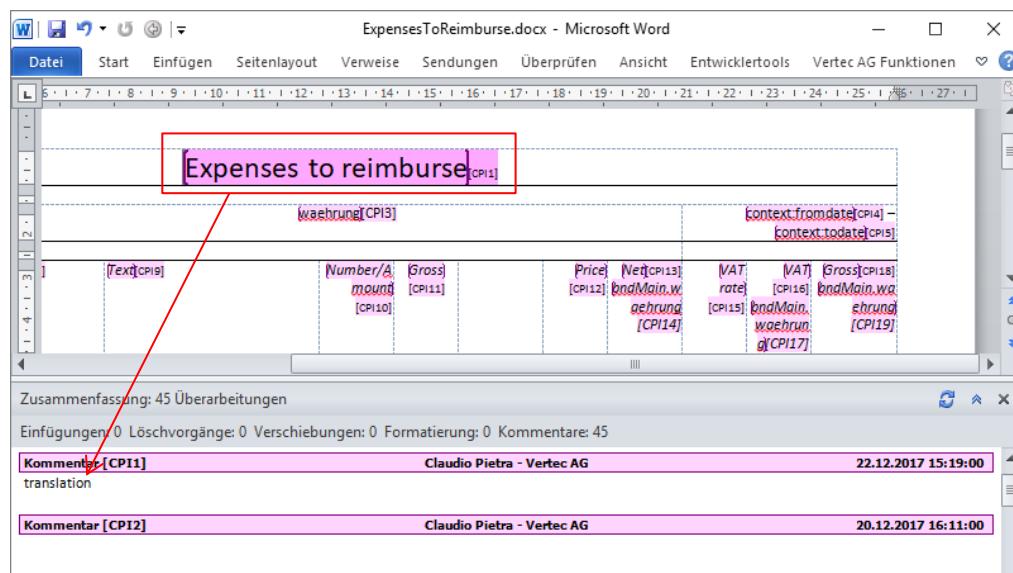


Abbildung 25: Der Begriff wird übersetzt

Das setzt voraus, dass Begriffe, die ausschliesslich im Report verwendet werden (zum Beispiel der Report-Titel) in die Übersetzungsdatei eingefügt werden. Wie das geht, finden Sie in der Online Knowledge Base unter <https://www.vertec.com/de/kb/sprachen#userdict>.

4.13 Feldformatierungen

Auch in den erweiterten Office-Berichten können Datum- und Zahlenwerte nach den Visual Basic Formatierungsregeln formatiert werden.

Die Formatierung wird dabei in den Kommentar des Feldes eingefügt, welches formatiert werden soll:

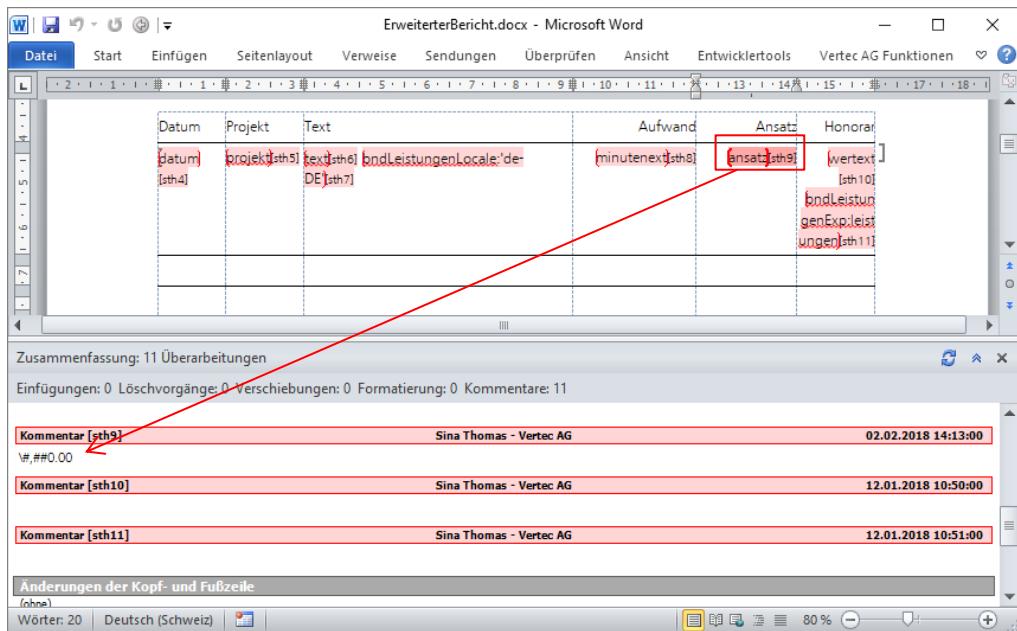


Abbildung 26: Die Formatierung wird im Kommentar eingefügt

Damit die Formatierung greift, muss es sich um einen Zahlenwert (Currency, Integer) handeln. Für die Formatierung werden die lokalen Regionaleinstellungen verwendet. Um andere Regionaleinstellungen zu verwenden, kann eine Locale-Expression auf dem Band eingefügt werden (siehe dazu Abschnitt 4.10).

Weitere Informationen Weitere Informationen zur Formatierung von Nummern und Datumsformaten finden Sie in der Online Knowledge Base unter <https://www.vertec.com/de/kb/vbformatierung>.

4.14 Bilder

Bilder aus der Datenbank (ImageData) werden im Python Code als `ImageField` bzw. `OclImageField` deklariert (siehe 4.3). Im Report-Dokument können sie als normale Datenfelder (kommentierter Text) eingefügt werden.

Die Angabe eines Pfades für eine Bilddatei (`ImagePath`) ist für erweiterte Office Berichte noch nicht implementiert.

4.15 Expressions in Kopf- und Fusszeilen

Context-Variablen (siehe 4.11) können auch in Kopf- und Fusszeilen verwendet werden.

Da in Kopf- und Fusszeilen Kommentare nicht verfügbar sind, werden die Variablen hier in doppelt geschweifte Klammern geschrieben, z.B.:

```
{ {context:projekt} }
```

5 Config Sets

Linie: Standard, Expert | Modul: Leistung & CRM | Version 6.2

Vertec Installationen können sehr weitgehend auf die Bedürfnisse von Kunden angepasst werden. Der Grossteil der möglichen Konfiguration erfolgt in der Datenbank des Kunden.

Neu können solche Konfigurationen thematisch zusammengefasst in einem Datenbank-unabhängigen Format als File gespeichert und in einer anderen Installation angewendet werden. Eine solche Konfigurations-Definition heisst **Config Set**.

5.1 Der Config Set Dialog

Die Steuerung der Config Sets erfolgt über den Config Set Dialog. Dieser öffnet sich über das Menü **Einstellungen > Config Sets**.

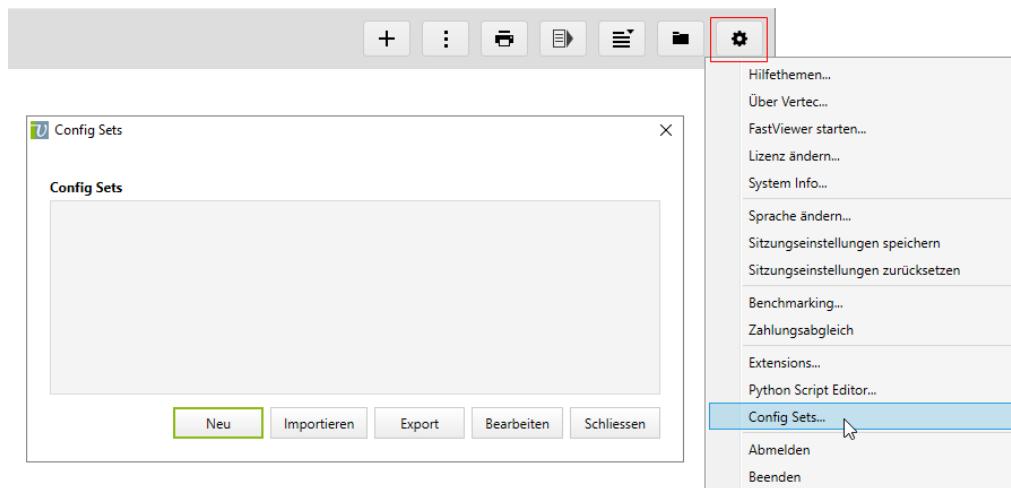


Abbildung 27: Der Config Set Dialog

5.2 Config Sets einspielen

Ein Config Set wird über eine **Import** Aktion im Config Set Dialog eingespielt. Durch Klick auf den **Import** Button erscheint ein Explorer, in welchem das Config Set XML-File ausgewählt werden kann.

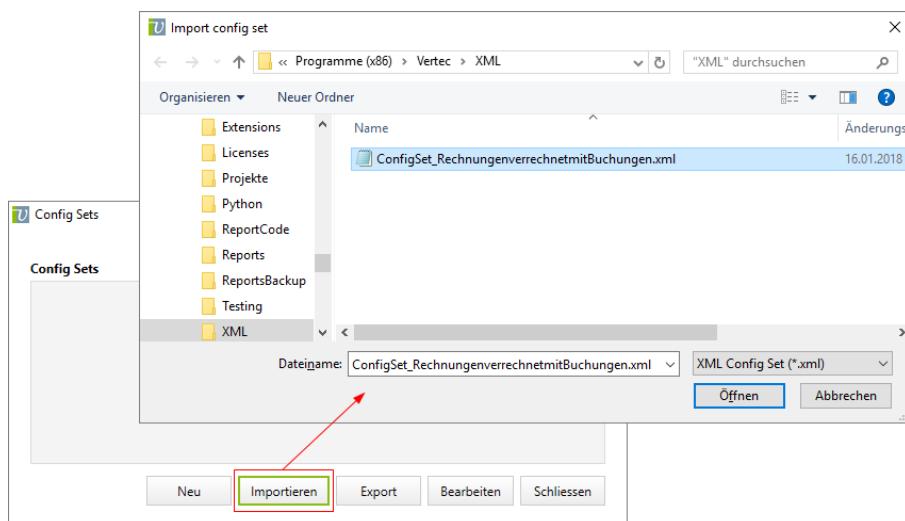


Abbildung 28: Ein Config Set wird importiert

Plausibilitätsprüfung

Beim Importieren wird das Config Set zuerst auf Plausibilität geprüft. Folgende Bedingungen müssen erfüllt sein, damit das Set angewendet wird:

- Im Config Set explizit formulierte Vorbedingungen sind erfüllt
- Alle aufgeführten externen Referenzen sind vorhanden
- Alle internen Aliases sind eindeutig
- Members mit Referenzen verweisen auf definierte Aliases
- Im Config Set enthaltene Zusatzklassen sind im System noch nicht vorhanden
- Im Config Set enthaltene Zusatzfelder sind im System noch nicht vorhanden

Falls Fehler auftreten, wird eine Fehlermeldung ausgegeben und das Config Set nicht angewendet.

Sind keine Fehler vorhanden, wird das Config Set direkt angewendet und die entsprechenden Objekte in Vertec erzeugt.

Nach dem Anwenden eines Config Sets erscheint dieses in der Liste im Config Set Dialog.

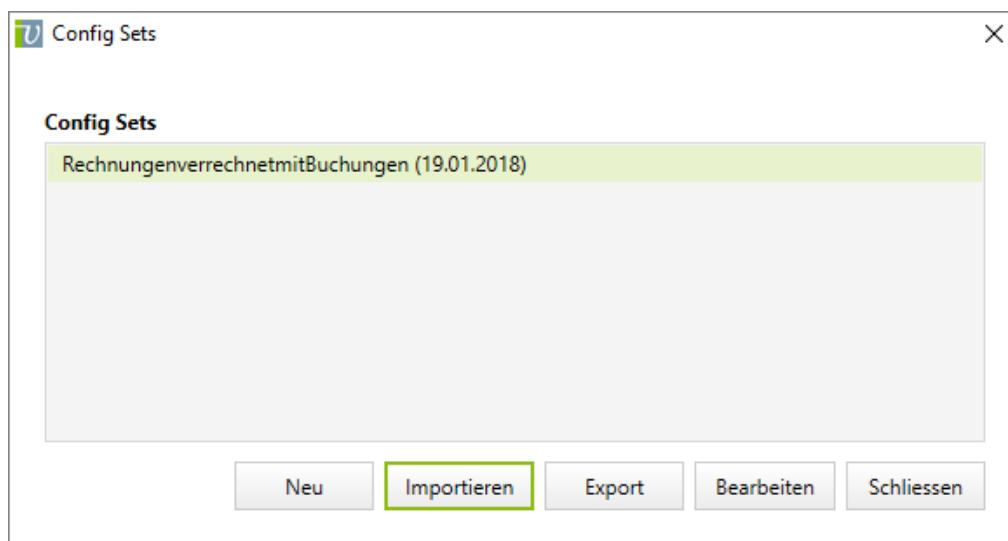


Abbildung 29: Das importierte Config Set erscheint in der Liste

Config Set via Python anwenden

In Python gibt es eine neue Methode `vtcapp.importconfigset()`, welche ein Config Set als String akzeptiert (XML-Text). Beispiel:

```
vtcapp.importconfigset("""<?xml version="1.0" encoding="utf-8"?><configset name="test" author="test"> <requirements /> <references /> <objects /> <settings> <system-setting name="GlobalWordTemplatePath">C:/TestConfigSet/</system-setting> </settings></configset>""")
```

Das auf diese Art importierte Config Set wird ebenfalls in der Liste im Config Set Dialog eingetragen.

5.3 Config Sets erstellen

Für das Erstellen von Config Sets steht in Vertec der [Config Set Builder](#) zur Verfügung. Dieser ermöglicht das interaktive Zusammenstellen der Konfiguration und exportiert daraus das Config Set in Form eines XML Datenfiles.

Um ein neues Config Set zu erstellen, klicken Sie im Config Set Dialog auf den Button [Neu](#). Es öffnet sich der Config Set Builder:

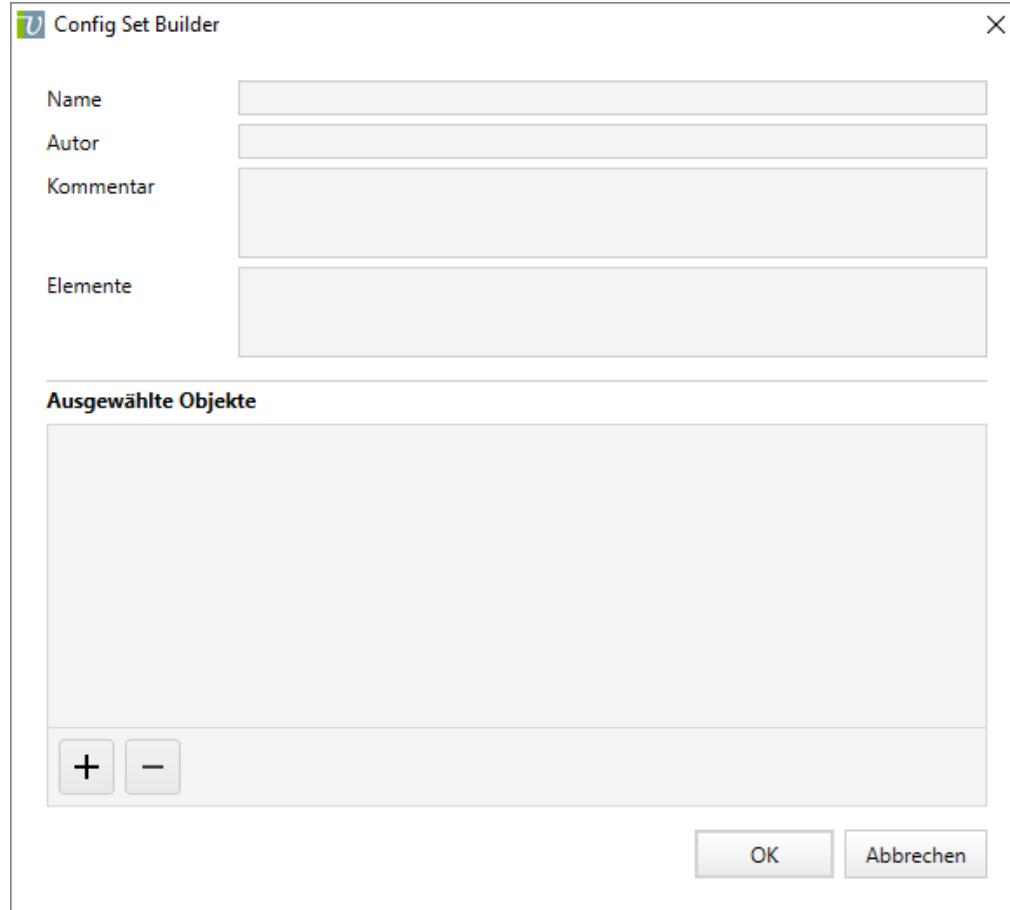


Abbildung 30: Der Config Set Builder

Er hat folgende Felder:

Name Name des Config Sets.

Da die Config Sets untereinander referenziert werden können, ist darauf zu achten, dass es sich um eindeutige Namen handelt.

Autor Autor des Config Sets.

Kommentar Freier Kommentartext für eine genauere Beschreibung des Config Sets.

Elemente In diesem Feld können Requirements, References und Settings eingegeben werden. Dieser Text muss in gültigem XML verfasst sein (siehe 5.4).

Folgende Elemente werden unterstützt:

- ocl-requirement
- configset-requirement
- configset-reference
- ocl-reference
- system-setting

Die einzelnen Elemente können direkt im Feld angegeben werden, ohne übergeordnete Node. Das heisst, dass statt

```
<references>
  <configset-reference ref-name="ConfigSet" />
</references>
```

nur

```
<configset-reference ref-name="ConfigSet" .../>
```

eingegeben werden muss.

Vertec-Objekte hinzufügen/entfernen

Unter **Ausgewählte Objekte** zeigt der Config Set Builder eine Liste von Vertec-Objekten, die im Config Set enthalten sind.

Um ein bestimmtes Objekt in die Liste aufzunehmen, wird dieses im aktiven Vertec-Fenster selektiert und anschliessend im Config Set Builder auf den **+** Buttons geklickt oder direkt per Drag&Drop ins Feld gezogen.

Um ein Objekt aus der Liste zu entfernen wird es markiert und der **-** Button geklickt.

Bei hierarchischen Objekten, welche Ihre Unterobjekte "besitzen", werden die Unterobjekte automatisch mitgenommen. Im Falle einer Ordnerstruktur muss z.B. nur der Überordner eingefügt werden.

Config Sets exportieren

Durch Klick auf den Button **Exportieren** wird vom Config Set Builder ein XML File erstellt, im lokalen Temp-Verzeichnis gespeichert und direkt zur Bearbeitung geöffnet.

Dieses Config Set kann dann in der Zielinstallation importiert werden (siehe Kapitel 5.2).

Falls beim Exportieren eine Fehlermeldung der Art:

```
Der Datei ist kein Programm zum Ausführen der Aktion zugeordnet...
```

erscheint, dann müssen Sie in Ihrem Windows dem Dateityp **.xml** eine Standard-App zuordnen, z.B. den Editor.

5.4 Config Set XML

Die Config Set Definition besteht aus den folgenden Teilen:

Attribute

author	Autor des Config Sets. Der Wert aus dem Feld Autor im Config Set Builder (5.3) wird automatisch eingetragen.
name	Name des Config Sets. Der Wert aus dem Feld Name im Config Set Builder (5.3) wird automatisch eingetragen.
ref-name	Name eines referenzierten Config Sets.
class	Klassenname des Objekts.
entryid	Eintrag Id des Objekts in Vertec (siehe 5.5)
alias	Alias des Elements innerhalb des Config Sets
ref-alias	Alias eines referenzierten Config Sets
expression	Expression-Attribut von OCL-References

Elemente

```
<requirements>
  <ocl-requirement />
  <configset-requirement />
</requirements>

<comment />

<references>
  <entryid-reference />
  <configset-reference />
  <ocl-reference />
</references>

<settings>
  <system-setting />
</settings>

<objects>
  <object>
    <reference-object />
    <member />
  </object>
</objects>
```

Requirements

Mit Requirements können Vorbedingungen definiert werden, welche erfüllt sein müssen, damit das Config Set eingelesen werden kann. Dies kann ein anderes Config Set sein oder eine beliebige OCL-Expression.

Die Requirements werden im Feld **Elemente** im Config Set Builder eingetragen (siehe 5.3).

Es gibt folgende Arten von Requirements:

- **configset-requirement**: Benennt ein zugrundeliegendes Config Set, welches bereits installiert sein muss.

```
<configset-requirement>Filialordner_Luzern</configset-requirement>
```

- **ocl-requirement**: Beliebige OCL Expression, welche True ergeben muss. Damit können Vorbedingungen abgefragt werden, welche für den Import des Config Sets erfüllt sein müssen, z.B. die Sicherstellung, dass eine bestimmte Zusatzklasse noch nicht belegt ist (siehe dazu 5.6):

```
<ocl-requirement>ClassSettings.allInstances->select(klasse='ZusatzKlasse0')->size = 0</ocl-requirement>
```

Referenzen

Enthält eine Auflistung von Referenzen, welche innerhalb des Config Sets verwendet werden. Jede Referenz erhält einen innerhalb des Config Sets eindeutigen Namen.

Es gibt folgende Arten von Referenzen:

- **entryid-reference**: Referenziert ein Objekt aufgrund seiner Klasse und seiner Eintrag Id. Auf diese Art werden im Zielsystem bereits vorhandene Objekte identifiziert (siehe Kapitel 5.5).

```
<entryid-reference class="OrdnerOeffentlich" alias="Alias_Parentordner" entryid="FolderPublicFolders" />
```

entryid-references werden beim Erstellen vom Config Set via Config Set Builder automatisch eingetragen.

- **configset-reference**: Referenziert ein benanntes Objekt aus einem anderen Config Set. Dieses muss bereits vorhanden sein, sollte also in den Requirements eingetragen werden.

```
<configset-reference ref-name="ConfigSet" ref-alias="Alias" alias="my_alias" />
```

configset-references werden im Feld **Elemente** im Config Set Builder eingetragen (siehe 5.3).

- **ocl-reference**: Referenziert ein Objekt durch Angabe einer OCL Expression. Die OCL Expression wird global ausgewertet. So kann ein im Zielsystem vorhandenes Objekt identifiziert und im Config Set damit gearbeitet werden.

```
<ocl-reference expression="TimSession.allInstances->first.login" alias="aktBearb" />
```

ocl-references werden im Feld **Elemente** im Config Set Builder eingetragen (siehe 5.3).

Settings

Es können auch Einstellungen gesetzt werden. Diese werden im Feld **Elemente** im Config Set Builder (siehe 5.3) eingetragen.

- **system-setting**: Damit kann eine Systemeinstellung gesetzt werden. Als Name wird der **PropertyName** angegeben. Diesen finden Sie bei der Beschreibung der jeweiligen Systemeinstellung in der Online Knowledge Base unter www.vertec.com/kb.

```
<system-setting
    name="GlobalWordTemplatePath">C:/Testpfad</system-setting>
```

Objekte

Das Config Set enthält im Wesentlichen eine Liste von Objekten, welche durch das Config Set angelegt werden sollen. Die Objekte enthalten Werte von Eigenschaften und Links.

```
<objects>
    <object class="Ordner" alias="Ordner_ProjektenachFilialen0">
        <member name="Bezeichnung">Projekte nach Filialen</member>
        ...
    </object>
</objects>
```

Bei Links wird entweder ein im selben Config Set erstelltes Objekt oder eine am Anfang definierte Referenz mit dem Alias angegeben (siehe nächsten Abschnitt).

```
<member name="Parentordner">
    <reference-object
        alias="Ordner_ProjektenachFilialen0_Parentordner" />
</member>
```

Objekt Aliases

Im Config Set enthaltene Objekte können optional mit einem Alias versehen und damit explizit referenziert werden, innerhalb des Config Sets oder von einem anderen Config Set aus. Innerhalb eines Config Sets muss ein Alias eindeutig sein.

Wird ein Objekt im Config Set Builder hinzugefügt, erhält dieses einen automatischen Alias. Wenn Sie diesen ändern möchten, klicken Sie mit der rechten Maustaste auf das Objekt in der Liste > umbenennen, und vergeben Sie den gewünschten Alias.

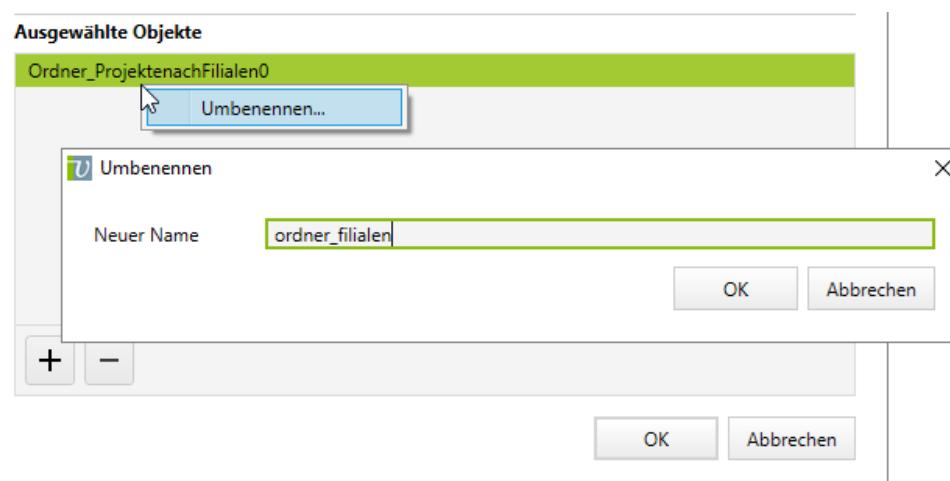


Abbildung 31: Ein Objekt umbenennen

Bestehende Objekte

In bestimmten Situationen wird für ein Objekt-Element in einem Config Set kein neues Objekt angelegt, sondern die Eigenschaften eines bestehenden Objekts geschrieben.

- Falls das Objekt eine Eintrag Id hat, wird zuerst in der entsprechenden Klasse nach dem bestehenden Objekt gesucht und dieses aktualisiert. Siehe dazu auch das Kapitel 5.5 über die Eintrag Id.
- Klasseneinstellungen (Class-Settings) werden speziell behandelt: Aufgrund des Klassennamens wird nach dem bestehendem Objekt gesucht. Diese Spezialbehandlung gilt nicht für Class-Settings von Zusatzklassen – dort wird ein Fehler gemeldet, falls Class-Settings für eine bestehende Zusatzklasse verändert werden sollen (siehe 5.6)
- Das Setzen von Eigenschaften auf ihren Defaultwert (z.B. String auf Leerstring) bei bestehenden Objekten wird nicht unterstützt.

5.5 Verwendung der Eintrag Id

Die Eintrag Id von Vertec Objekten ist zentral bei der Handhabung von Config Sets. Über die Eintrag Id wird identifiziert, ob das Objekt in der Zieldatenbank bereits besteht oder nicht.

Die Eintrag Id ist im **Eigenschaften** Dialog bei den entsprechenden Objekten (rechte Maustaste > Eigenschaften) zu finden:

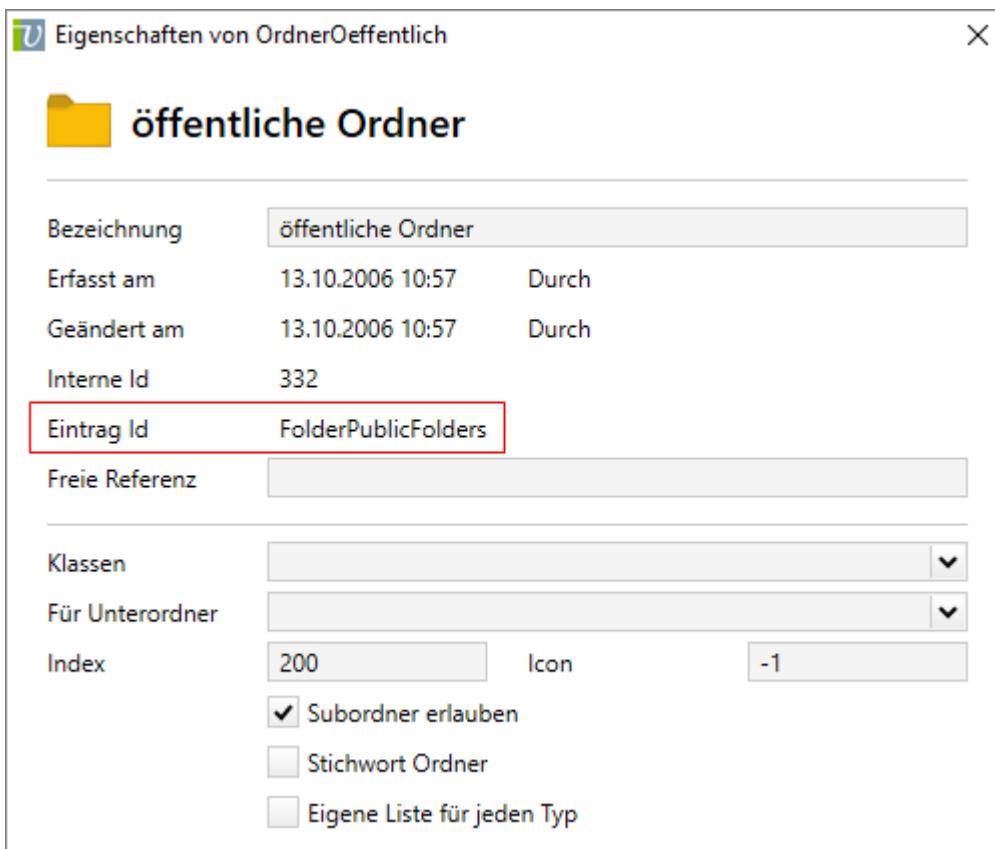


Abbildung 32: Eintrag Id eines öffentlichen Ordners

Objekte, die ohne Eintrag Id Referenz (`entryid="xx"`) übergeben werden, werden im Zielsystem neu erzeugt.

Bei der Erstellung eines Config Sets (siehe 5.3) wird beim Einfügen eines Objekts darauf geachtet, ob der Überordner (Parentordner) eine Eintrag Id hat. In diesem Fall wird davon aus-

gegangen, dass der Überordner im Zielsystem bereits besteht und das Objekt darin erzeugt werden kann.

Wird ein Objekt, welches mit einer Eintrag Id referenziert wird, im Zielsystem nicht gefunden, wirft der Import einen Fehler (siehe 5.2).

Hat der Überordner eines Objekts noch keine Eintrag Id, wird davon ausgegangen, dass der gesamte Ordner im Zielsystem nicht vorhanden ist. In Konsequenz wird der Überordner sowie auch all seine Unterordner – siehe Hierarchie unter 5.4 – exportiert bzw. im Zielsystem neu erzeugt.

Wichtig ist deshalb, dass alle von Vertec mitgelieferten Strukturdaten (Ordner, Linktypen) in Vertec eine Eintrag Id haben. Dies wird von Vertec mit Version 6.2 sichergestellt, siehe dazu Kapitel 11.1.

Hier nachfolgend zwei Szenarien, die verdeutlichen, in welcher Art auf die Eintrag Id beachtet werden muss.

Szenario Einspielen von neuen Konfigurationen

Bei diesem Szenario geht es darum, dass der Kunde ein Config Set erhält, welches er einspielt, um eine bestimmte Konfiguration zu erhalten.

Hier wird davon ausgegangen, dass die von Vertec standardmäßig mitgelieferten Strukturdaten wie Ordner (Ordner Stammdaten, Öffentliche Ordner, Linktypen etc.) vorhanden sind und referenziert werden können.

Ist eine solche Struktur in Vertec beim Import eines Config Sets nicht vorhanden, wird eine Fehlermeldung angezeigt.

Im diesem Szenario muss sich der Kunde also nicht um die Eintrag Id kümmern.

Szenario Testumgebung – Livesystem

Ein anderes Szenario ist es, wenn auf einem Testsystem Konfigurationen ausgearbeitet und später auf das Livesystem übertragen werden sollen. In diesem Fall ist die Eintrag Id zentral.

Hier geht es darum, ob die neuen Objekte im Zielsystem in eine bestehende Struktur eingefügt oder ob auch diese Struktur neu erzeugt werden soll.

Soll beispielsweise ein neuer Root-Ordner im Zielsystem erzeugt werden, darf dieser im Quellsystem keine Eintrag Id haben. Will man aber später auf dem Zielsystem auf diesen Root-Ordner zugreifen, um beispielsweise neue Unterordner zu erzeugen, dann muss dieser eine Eintrag Id haben. In diesem Fall muss die Eintrag Id sowohl im Quell- als auch im Zielsystem manuell gesetzt werden.

Dies kann via Python über `argobject.eintragid="XX"` auf dem entsprechenden Objekt geschehen.

5.6 Behandlung von Spezialfällen

Zusatzklassen

Soll im Rahmen eines Config Sets eine bestimmte Zusatzklasse konfiguriert werden, muss sichergestellt werden, dass diese im Zielsystem noch nicht verwendet wird. Das Config Set kann dies beim Import nicht prüfen, sondern dies muss bereits beim Erstellen der Konfiguration berücksichtigt werden (insbesondere wenn sonstige Konfigurationen, z.B. Scripts, welche ebenfalls mit dem Config Set mitgeliefert werden, auf diese Zusatzklasse verweisen).

Um zu verhindern, dass 'aus Versehen' eine Zusatzklasse importiert wird, welche bereits besteht, kann ein entsprechendes OCL-Requirement formuliert werden (siehe 5.4).

Zusatzfelder

Falls ein Config-Set Zusatzfelder anlegt, muss sichergestellt sein, dass die Zieldatenbank keine Zusatzfelder gleichen Namens auf der gleichen Klasse definiert hat. Dies wird beim Import nicht geprüft, sondern die Zusatzfelder gegebenenfalls doppelt angelegt.

Auch hier muss dies, wie bei den Zusatzklassen (siehe oben), bereits bei der Erstellung des Config Sets berücksichtigt werden.

Um zu verhindern, dass 'aus Versehen' ein Zusatzfeld importiert wird, welches bereits besteht, kann ein entsprechendes OCL-Requirement formuliert werden (siehe 5.4).

5.7 Config Sets entfernen/löschen

Config Sets können entweder entfernt oder gelöscht werden. Klicken Sie mit der rechten Maustaste im Config Set Dialog (siehe 5.1) auf das entsprechende Config Set:

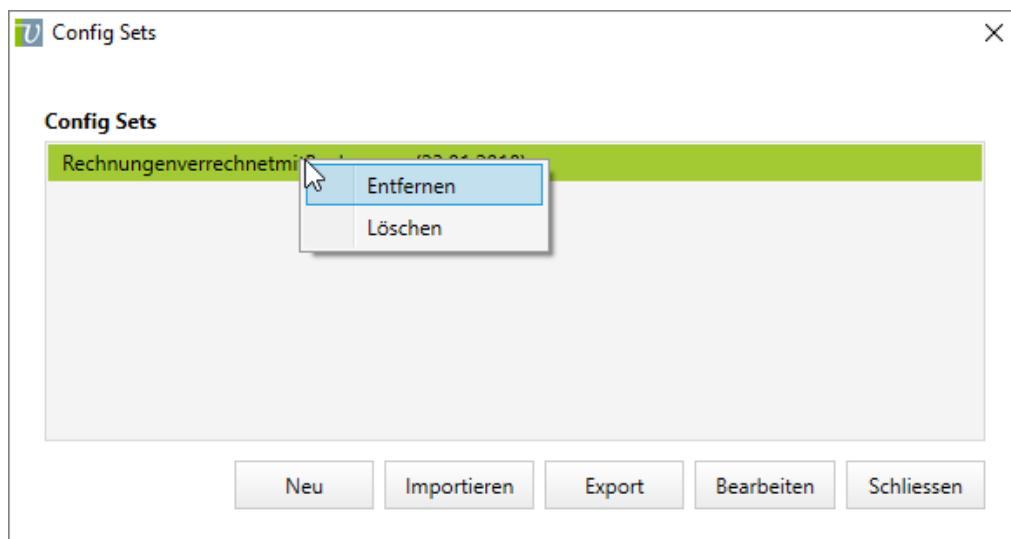


Abbildung 33: Config Set entfernen oder löschen

- **Entfernen:** Das Config Set wird aus der Liste entfernt, die Objekte verbleiben jedoch im System. Das Entfernen von Config Sets dient vor allem dazu, diese vor einem allfälligen Löschvorgang zu schützen.
Das Config Set kann nicht erneut importiert werden, da die Objekte sonst doppelt vorhanden sind. Es gibt keine Möglichkeit, ein Config Set in die Liste aufzunehmen, ohne damit die Objekte neu zu erzeugen.
- **Löschen:** Das Config Set wird gelöscht. **Achtung! Es werden auch alle dazugehörigen Objekte in Vertec gelöscht.** Dieser Vorgang dient dazu, alle durch das Config Set erzeugten Objekte wieder aus Vertec zu löschen. Wurde mit diesen bereits weitergearbeitet, gehen damit auch alle Änderungen verloren. Um ein Config Set vor dem Löschen zu schützen, kann es aus der Liste entfernt werden ([Entfernen](#)).

6 Customizing/Parametrisierbarkeit

6.1 Tags auf User-Einträgen

Linie: Expert | Module: Leistung & CRM | Version 6.1.0.10

In Vertec werden Objekte (z.B. Projekte, Leistungen) vielfach über Boolean Zusatzfelder kategorisiert. Die daraus resultierenden OCL-Abfragen der Art

```
leistungen->select(zusatzfeldbool('istmarkiert'))
```

sind nicht performant, da für jedes Element in der Liste die Zusatzfelder geladen werden müssen.

Um diese Anforderung performant abilden zu können, wurde die Möglichkeit geschaffen, User-Einträge mit Tags zu versehen, nach denen gefiltert werden kann.

Dafür gibt es ein **Tags** Feld auf jedem UserEintrag (UserEinträge sind alle Objekte in Vertec, welche in Ordnern abgelegt werden können, wie Adressen, Projekte, Bearbeiter, Leistungen, etc.), welches (via Python) mit beliebigen Stichworten (Tags) befüllt werden kann.

Setzen eines Tags

Gesetzt werden die Tags mit der Python Methode `addtag(tagname)`, z.B.

```
argobject.addtag('mytag')
```

Abstände in den Tagnamen sind erlaubt. Ist ein Tag bereits vorhanden, wird er nicht erneut hinzugefügt.

Die maximale Länge aller auf einem User-Eintrag gesetzten Tags ist gesamthaft 1000 Zeichen. Wird diese Länge überschritten, erscheint eine Fehlermeldung.

Abfragen von Tags

Abfragen werden können die Tags via Python oder OCL. Die Abfrage lautet in beiden Fällen:

```
hastag(tagname) : boolean
```

Python

OCL

```
argobject.hastag('mytag')
```

```
hasTag('mytag')
```

Der Rückgabewert ist `true`, wenn der Tag gesetzt ist, und `false`, wenn es ihn nicht gibt.

So können Listen mit Abfragen der Art

```
leistungen->select(hasTag('istmarkiert'))
```

schnell und performant gefiltert werden.

Löschen von Tags

Um Tags zu löschen, steht die Python Methode `removetag(tagname)` zur Verfügung, z.B.

```
argobject.removetag('mytag')
```

Wird ein Tag gelöscht, den es nicht gibt, dann geschieht nichts, es gibt keinen Fehler.

6.2 Key-Values auf User-Einträgen

Linie: Expert | Module: Leistung & CRM | Version 6.1.0.10

Um lange Listen aufgrund verschiedener Kriterien performant filtern zu können, wurde die Möglichkeit geschaffen, kundenspezifische Werte auf User-Einträgen ablegen zu können, welche via OCL abgefragt werden können.

Dafür gibt es ein neues Feld **Keys** auf UserEintrag (UserEinträge sind alle Objekte in Vertec, welche in Ordern abgelegt werden können, wie Adressen, Projekte, Bearbeiter, Leistungen, etc.), in welches (via Python) die Key-Values Wertepaare eingetragen werden. Es werden folgende Datentypen unterstützt:

- String
- Boolean
- Date, DateTime
- Integer
- Currency

Setzen eines Key-Values

Die Key-Values werden mit der Python Methode `setkeyvalue(key, value)` gesetzt, z.B.

```
argobject.setkeyvalue('datum', argobject.eval("date"))
```

Als **value** werden die oben genannten Datentypen akzeptiert. Die Umwandlung in den entsprechenden Key-Value-Typ ist wie folgt:

Datentyp	Eingabe	Beispiel
String	String	<code>argobject.setkeyvalue("Mykey", "Mein Text hier")</code>
Boolean	"True", "False"	<code>argobject.setkeyvalue("Erledigt", "True")</code>
Date	ISO Datestring ("2017-02-03")	<code>argobject.setkeyvalue('datum', vtcapp.currentdate())</code>
DateTime	ISO Datetime ("2017-02-03T08:30")	<code>argobject.setkeyvalue('datum', vtcapp.currentdatetime())</code>
Integer	Ganzzahl	<code>argobject.setkeyvalue("Anzahl", 5)</code>
Currency	Fliesskommazahl	<code>argobject.setkeyvalue("Betrag", 10573.65)</code> <code>setkeyvalue()</code> schreibt numerische Typen mit so wenig Dezimalstellen wie möglich und maximal 4 Dezimalstellen. Es wird auf die 4. Dezimalstelle gerundet.
None	Leerstring	<code>argobject.setkeyvalue("Mykey", None)</code> oder <code>argobject.setkeyvalue("Mykey", "")</code>

Die maximale Länge aller auf einem User-Eintrag gesetzten Key-Values Wertepaare ist gesamthaft 1000 Zeichen. Wird diese Länge überschritten, erscheint eine Fehlermeldung.

Abfragen eines Key-Values

Zum Abfragen eines Key-Values stehen sowohl Python-Methoden als auch OCL-Operatoren zur Verfügung. Abgefragt wird jeweils typspezifisch.

Python

Methode	Resultat
getkeystring(key)	Liefert genau was im Key-Value drin steht
getkeybool(key)	liefert False für Leerstring, "False" (case insensitive) oder numerische 0-Werte, sonst true
getkeycurr(key)	liefert einen Float oder 0 als float
getkeyint(key)	liefert einen Integer oder 0 als integer
getkeydate(key)	liefert ein Datum mit Zeitteil oder None

`getkeyint(key)` und `getkeycurr(key)` geben nummerische Werte des jeweils anderen Typs (integer vs. float) zurück, wenn kein Datenverlust stattfindet.

OCL

Operator	Resultat
keystring(key)	alles -> string, sonst ""
keybool(key)	"", "False", "0", "0.0" -> False, sonst True
keydate(key)	iso-date (mit oder ohne time) -> datetime, sonst nil
keyint(key)	integer string -> integer, sonst 0
keycurr(key)	currency string -> currency, sonst 0.00

Leere oder ungültige Werte führen zu einem Defaultwert des entsprechenden Typs. Defaultwerte sind:

- String: ""
- Boolean: ""
- Date: None (bzw. nil)
- Integer: 0
- Currency: 0.00

Löschen eines Key-Values

Es gibt keine Methode zum Löschen von Key-Values. Das Entfernen eines solchen Wertepaars geschieht über das Setzen des entsprechenden Keys mit None bzw. Leerstring:

```
setkeyvalue('mykey', None)
```

oder

```
setkeyvalue('mykey', Leerstring)
```

Hinweis zu Currency Key-Values

In den Vertec Versionen 6.1.0.10 bis 6.1.0.12 wurden Currency Key-Values fälschlicherweise regionalspezifisch gespeichert. Während dieser Zeit gespeicherte Currency-Werte erscheinen beim Auslesen als String, bzw. führen beim Auslesen als Float zu einem Fehler. In diesen Fällen müssen die Werte neu gesetzt werden.

6.3 Syntax-Highlighting in Code-Editoren

Linie: Standard, Expert | Module: Leistung & CRM | Version 6.2

Die Code-Editoren für OCL, XML, Python und VBScript verfügen in der Desktop App und der Cloud App neu über ein Syntax-Highlighting. Dadurch wird der Code lesbarer und übersichtlicher.

```
76     return None
77
78 def main():
79     leistung = argobject
80
81     # Wenn eine Phase zugewiesen ist
82     if leistung.phase:
83         # 1. Priorität: Tätigkeit auf Phase
84         if leistung.typ and leistung.eval("phase.taetigkeitphaselink->select
85             (name=self.typ.code).ansatzExt->first"):
86             showPriority(1)
87             return
88
89         # 2.1 Priorität: Bearbeiter auf Phase -> Explizit gesetzter Ansatz
90         if leistung.eval("phase.bearbeiterPhasen->select
91             (bearbeiter=self.bearbeiter).ansatzext->first"):
92             showPriority(2)
93             return
```

Abbildung 34: Python Script mit Syntax-Highlighting

Ausser bei OCL gibt es außerdem eine Zeilenummerierung, und die Klammern zeigen Anfang und Ende an, wenn sie angeklickt werden:

```
82 	" leistung,phase,  
83 	# 1. Priorität: Tätigkeit auf Phase  
84 	if leistung.typ and leistung.eval("phase.taetigkeitphaselink->select  
85 	(name=self.typ.code).ansatzExt->first"):  
86 		showPriority(1)  
87 	return
```

Abbildung 35 Klammern zeigen Anfang und Ende an

6.4 Control-Namen auf Rechnung-Page geändert

Linie: Expert | Module: Leistung & CRM | Version 6.2

Im XML der Rechnung gibt es zwei Controls, welche einen nicht korrekten Namen hatten und deshalb umbenannt wurden:

- TotalAfterCosts: In der Tabelle `TotalAfterCostsRow` gibt es einen TextBlock namens `TotalAfterCosts`. Dieser stellt das Total nach Rabatt dar und heisst neu daher `TotalAfterDiscount`.
 - FixedPrice: Das Feld `FixedPrice` heisst neu `FlatExpenses`.

Rückwärtskompatibilität

Die Umsetzung ist rückwärtskompatibel. Auch bei Kunden, welche die Elemente im Customizing bereits überschrieben haben, funktioniert die Anzeige nach wie vor. Auf der Oberfläche ist also von der Änderung nichts zu spüren.

In diesem Fall ist aber "von Auge" keine Zuordnung des überschriebenen Elementes zum entsprechenden Element im eingebauten XML mehr möglich.

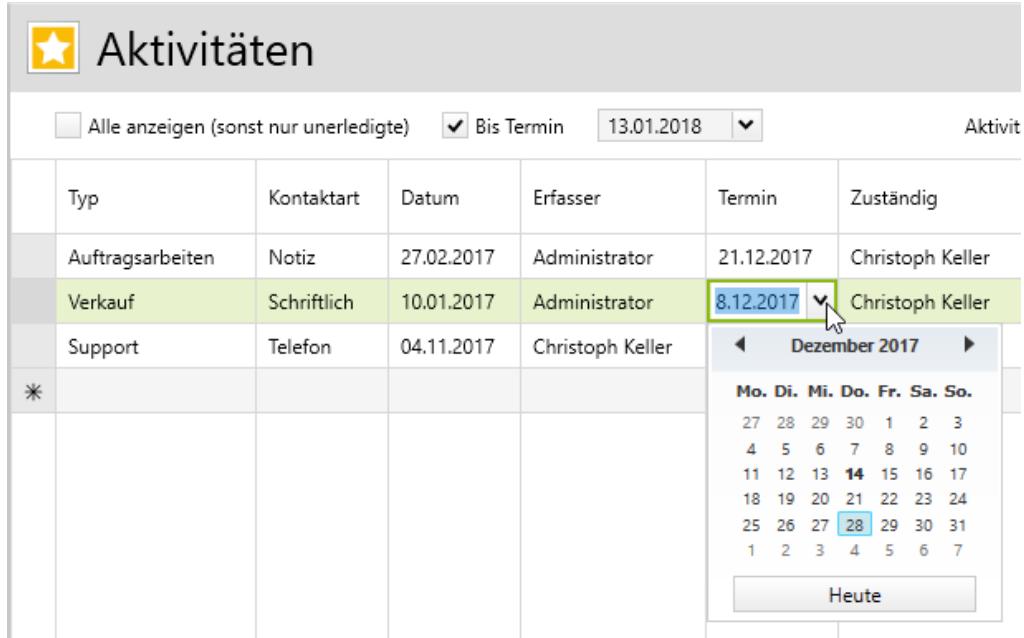
6.5 DatePicker in Listen

Linie: Expert | Module: Leistung & CRM | Version 6.2

Neu kann der DatePicker auch in der Liste verwendet werden. Mittels Customizing des Spalten-XML kann der Benutzer einen DatePicker einsetzen.

Beispiel des `Steuerelement XML` beim Termin auf einer Aktivitätenliste:

```
<DatePicker ValueExpression="termin" />
```



	Typ	Kontaktart	Datum	Erfasser	Termin	Zuständig
	Auftragsarbeiten	Notiz	27.02.2017	Administrator	21.12.2017	Christoph Keller
	Verkauf	Schriftlich	10.01.2017	Administrator	8.12.2017	Christoph Keller
	Support	Telefon	04.11.2017	Christoph Keller		
*						

Abbildung 36: Aktivitätenliste mit Datepicker

6.6 Unterstützung von Datumswerten in Custom Dialogen

Linie: Expert | Module: Leistung & CRM | Version 6.1.0.14

Die Angabe eines Datumswertes (`datetime`) als Initialisierungswert für einen Datepicker in Custom Dialogen führte bisher zu einem Serialisierungsfehler. Datumswerte konnten nur über einen Workaround verwendet werden.

Dieses Problem wurde behoben. Ab sofort können Datumswerte ganz normal übergeben werden:

```
from datetime import date
initValues["DatumVon"] = date(2017, 1, 1)
```

Rückwärtskompatibilität

In Vertec Versionen bis 6.1.0.14 wurden Datumswerte anders behandelt. Scripts dieser Versionen, welche Datumswerte verwenden, müssen entsprechend angepasst werden:

– Datumswerte konnten wie folgt übergeben werden:

```
von = vtcapp.firstdayofmonth(vtcapp.incmonth(date.today(), -1))
bis = vtcapp.lastdayofmonth(von)
```

```
initValues = {}  
initValues["DatumVon"] = von.isoformat() + "T00:00:00"  
initValues["DatumBis"] = bis.isoformat() + "T00:00:00"
```

Neu muss die Übergabe so erfolgen:

```
initValues["DatumVon"] = date.today() #oder:  
initValues["DatumVon"] = date(2017, 1, 1)
```

- Der Rückgabewert konnte wie folgt in ein **date** umgewandelt werden:

```
#Bsp. 01.11.16  
von = datetime.strptime(values['DatumVon'], '%Y-%m-%dT%H:%M:%S')  
#Bsp. 31.01.17  
bis = datetime.strptime(values['DatumBis'], '%Y-%m-%dT%H:%M:%S')
```

Ab 6.1.0.14 kommt es damit zur Fehlermeldung:

```
<type 'exceptions.TypeError'>: must be string, not datetime.datetime
```

Lösung: Der Rückgabewert ist bereits vom Typ **date** und kann direkt zugewiesen werden:

```
von = values['DatumVon']  
bis = values['DatumBis']
```

6.7 Custom Dialoge: DatePicker mit Null-Wert

Linie: Expert | Module: Leistung & CRM | Version 6.2.

DatePicker in Custom Dialogen ergaben bisher einen speziellen Datumswert (1.1.0001), wenn sie leer waren. Neu wird **None** zurückgeliefert.

Rückwärtskompatibilität

Scripts, die den speziellen Datumswert sinngemäss wie folgt abgefragt haben:

```
if values['Datum'] == "0001-01-01 00:00:00":
```

müssen angepasst werden und neu auf **None** abfragen:

```
if values['Datum']:
```

6.8 Neues Passwort verlangen

Linie: Standard, Expert | Module: Leistung & CRM | Version 6.1.0.11

Neu hat der Administrator die Möglichkeit, vom Benutzer einen Passwortwechsel zu erzwingen beim nächsten Login. Diese Option ist nur vorhanden, wenn die Authentisierung nicht über LDAP erfolgt.

Dafür gibt es eine neue Checkbox auf dem Bearbeiter, welche nur dem Administrator angezeigt wird:

Bearbeiter

Name	Ralf Maurer	<input checked="" type="checkbox"/> Aktiv
Kürzel	RM	Login Name maurer
Stufe	Senior Consultant	Passwort Passwort ändern...
		<input checked="" type="checkbox"/> Neues Passwort verlangen
Teamleiter	Christoph Keller	Stellvertreter Judith Feller
Erfassungswährung	CHF	
Eintritt per	01.03.2016	Austritt per
Adresse	Maurer Ralf	

Abbildung 37: Neues Passwort verlangen wird gesetzt

Sobald sich der entsprechende Bearbeiter das nächste Mal an Vertec anmeldet, erscheint der Passwort ändern Dialog, und der Bearbeiter muss ein neues Passwort setzen, welches den eingestellten Password-Policies entspricht. Dieses kann nicht dasselbe sein wie das aktuelle, sonst erscheint eine entsprechende Meldung:

Passwort ändern

Aktuelles Passwort	••••••••••••
Passwort	••••••••••••
Passwort-Bestätigung	••••••••••••
Bisheriges und neues Passwort können nicht identisch sein.	
OK	Abbrechen

Abbildung 38: Es muss ein neues Passwort gesetzt werden

Password Policy Weitere Informationen zu den Passwort-Einstellungen finden Sie in der Online Knowledge Base unter www.vertec.com/de/kb/verteclogin.

6.9 Neue OCL Operatoren

Linie: Expert | Module: Leistung & CRM | Version 6.2

Folgende OCL Operatoren wurden neu eingeführt:

Funktion	Auf	Beschreibung
datetostrgerman	Datum	<p>Gibt das Datum in deutschem Datumsformat als String zurück.</p> <p>Es braucht an gewissen Orten manchmal Datumswerte im deutschen Format, unabhängig von aktuellen Regionaleinstellungen (siehe z.B. von/bis-Argumente bei groupLeistungen Operatoren unter https://www.vertec.com/de/kb/leistungssummen#syntax).</p> <pre>argobject.eval('creationdatetime.datetostrgerman')</pre>
getPresenceEntries(datum)	Bearbeiter	<p>Gibt eine Liste der Präsenzzeit-Einträge eines Tages für den Bearbeiter aus.</p> <pre>bearb.getPresenceEntries(date)</pre>

6.10 Neue Python Methoden

Linie: Expert | Module: Leistung & CRM

Folgende Python Methoden wurden neu eingeführt:

Auf dem Modul "vtcapp"

Funktion	Version	Beschreibung
datetostr(date): string	6.1.0.14	<p>Wandelt ein Datum in einen String um und gibt diesen gemäss Ländereinstellungen zurück. Zum Beispiel in Australian English:</p> <pre>>>> datum = vtcapp.strptime('13/06/17') >>> print datum 2017-06-13 >>> string = vtcapp.datetostr(datum) >>> print string 13/06/2017</pre>
datetostrgerman(date): string	6.2.	<p>Es braucht an gewissen Orten in Scripts manchmal Datumswerte im deutschen Format, unabhängig von aktuellen Regionaleinstellungen (siehe z.B. von/bis-Argumente bei groupLeistungen Operatoren unter https://www.vertec.com/de/kb/leistungssummen#syntax).</p> <p>Die datetostrgerman(date) Methode wandelt ein Datum in einen String um und gibt diesen in deutschem Datumsformat zurück.</p> <pre>>>> vtcapp.datetostrgerman(argobject.eval('creationdatetime')) 13.10.2016</pre>
currentdate(): date	6.1.0.10	<p>Gibt das aktuelle Datum zurück.</p> <pre>>>> vtcapp.currentdate() 2017-11-24</pre>

Funktion	Version	Beschreibung
currentdatetime(): datetime	6.1.0.10	Gibt das aktuelle Datum und die aktuelle Uhrzeit zurück. >>> vtcapp.currentdatetime() 2017-11-24 15:01:12.880000
executeserverwordreport(rootObj, optarg, templatePath, [outputFormat, saveAs, showDialog, showApp])	6.1.0.11	Diese Methode dient dazu, Vertec-generierte Word-Reports ohne Bericht-Objekt auszuführen. vtcapp.executeserverwordreport(argobject, None, r"C:\Program Files (x86)\Vertec\Reports\Deutsch\Brief.dotx", "", "C:\Dokumente\Vertec\Test", True, True)
createlist(classname, [list]): list	6.1.0.14	Erstellt eine Vertec-Liste, welche Typinformationen enthält und die Konstruktion einer Vertec-Liste direkt aufgrund einer anderen Liste (Vertec- oder Python Liste) mylist = vtcapp.createlist('Projekt') mylist.append(argobject) oder liste = vtcapp.createlist("Projekt", vtcapp.evalocl("projekt ->select(code.sqllike('A%'))")) liste2 = liste.evalocl("self->select(aktiv)")
importconfigset(xmltext: string)	6.2	Importiert ein Config Set in Vertec und wendet es an (siehe Kapitel 5.2). Als Config Set wird gültiger XML-Text als String übergeben.

Auf einzelnen Objekten

Funktion	Version	Beschreibung
evalocl(ocl:string)	6.1.0.10	Evaluiert eine OCL-Expression auf dem Objekt. Macht dasselbe wie eval, wurde eingeführt, damit es gleich heißt wie die globale Methode. projektcode = argobject.evalocl("code")
Aktivitaet.setpfad(pfad:string)	6.1.0.14	Setzt den Dokumentenpfad auf der Aktivität. argobject.setpfad('C:\Dokumente\text.txt') Benötigt Schreibrechte auf der Aktivität.
Waehrung.getkursto(waehrung: object, date: date): currency	6.1.0.14	Gibt den Umrechnungskurs zum entsprechenden Datum zurück. chf.getkursto(eur, vtcapp.currentdate())
Leistung.updatesatz()	6.1.0.14	Stundensatz wird neu berechnet (aus Tarifsystem). argobject.updatesatz() Benötigt Berechtigung Projekt-Administrator oder Super .

Auf Vertec-Listen

Funktion	Version	Beschreibung
extend()	6.2	<p>Mit der extend() Methode erhalten Vertec-Listen in Python die Möglichkeit, eine Python-Liste (z.B. aus einer OCL-Evaluation) anzuhängen. Bisher gab es nur die append() Methode mit der Möglichkeit, einzelne Objekte anzuhängen.</p> <p>Dabei wird geprüft, ob es sich bei der Vertec-Liste um eine derived Liste (z.B. Container.eintraege) handelt. Der Zugriff wird in diesem Fall verweigert.</p> <pre>cominstall = vtcapp.evalocl("projekt->select(code='COMINSTALL')->first") comssup = vtcapp.evalocl("projekt->select(code='COM-SSUP')->first") cominstall.offeneleistungen.extend(comssup.offeneleistungen)</pre>
idstring()	6.2	<p>Eine häufige Anforderung in Python Scripts ist, aufgrund einer Liste von Vertec Objekten eine SQL-Abfrage mit <code>bold_id in (...)</code> zu formulieren.</p> <p>Dazu müssen die IDs einer Liste von Objekten in einen comma-delimited String umgewandelt werden.</p> <ul style="list-style-type: none"> – <code>List.idstring()</code> gibt die IDs der Vertec-Objekte als comma-delimited string zurück. – Falls die Liste nicht persistente Objekte enthält (Objekt-ID ist not storables) gibt die Methode eine Fehlermeldung zurück. <pre>vtcapp.evalocl('Projekt->allinstances').idstring()</pre>

An folgenden Funktionen wurden Änderungen vorgenommen:

Funktion	Version	Änderung
vtcapp.evaltovariable()	6.1.0.14	"self" darf als Variablenname nicht mehr verwendet werden, da dies ein Schlüsselwort ist. Es erscheint eine entsprechende Fehlermeldung.
vtcapp.executereport(rootObj, optarg, berichtObj, [saveAs, showDialog, doPrint, showApp])	6.1.0.14	Mit dem Parameter <code>showDialog</code> wird neu auch die Anzeige von Meldungen wie "existierendes File überschreiben" und "neuen Pfad erstellen" über diese Einstellung gesteuert. Wenn der Parameter nicht gesetzt wird, wird die entsprechende Einstellung aus dem Report Objekt (<code>berichtObj</code>) übernommen.

Detaillierte Informationen zu den verschiedenen Funktionen finden Sie im Artikel [Python Interfaces](#) in der Online Knowledge Base unter www.vertec.com/de/kb/pythoninterfaces.

6.11 Python Methode vtcapp.sendFile()

Linie: Expert | Module: Leistung & CRM | Version 6.2

Mit der neuen Python Methode ist es möglich, ein File oder einen String (als File) an einen Client zu senden. Die Syntax lautet:

```
vtcapp.sendFile(file, filename, [showsavedialog], [openfile])
```

Parameter

file: string	Hier kann ein Pfad zu einem File angegeben werden oder ein String. Daher funktioniert das auch in Cloud Clients ohne Filesystemzugriff.
filename: string	Name des Files, falls es gespeichert wird.
showsavedialog : boolean	<p>Optional. Das übertragene File soll auf dem Client je nach Anforderung und Möglichkeiten des Clients gespeichert oder gleich ausgeführt (Shell-Execute) werden.</p> <ul style="list-style-type: none"> – Web App: Nur Download – Desktop und Cloud App: Dialog und Ausführung möglich. <p>In der Desktop- und Cloud-App bewirkt ein gesetztes <code>showsavedialog</code> Argument, dass ein Speichern-Dialog angezeigt wird. Standardmäßig wird dabei der Desktop als Speicherort angezeigt.</p> <p>Falls <code>showsavedialog=false</code> ist, wird das File im Temp-Folder des Clients gespeichert, ein bestehendes File wird dabei überschrieben. Das ist sinnvoll im Zusammenhang mit <code>openfile=true</code>, dann wird das File in Temp gespeichert und gleich geöffnet.</p> <p>Standard ist <code>false</code>.</p>
openfile: boolean	<p>Optional. Wenn <code>true</code>, wird das File nach dem Speichern geöffnet.</p> <p>Standard ist <code>true</code>.</p>

Mit der Web App als Client erscheint das File immer als Download im Webbrowser. Die Argumente `showsavedialog` und `openfile` haben keine Bedeutung.

Hinweis Die Kombination `showsavedialog=false` und `openfile=false` ist nicht sinnvoll, da so das File im Temp-Verzeichnis gespeichert wird und sonst nichts passiert.

Die Methode funktioniert auch im Restrict Scripting Modus.

Beispiele

Beispiel eines einfachen Projektexports:

```
projekte = argobject.evalocl("einträge.list")
projektstr = ""

for projekt in projekte:
    projektstr = projektstr + projekt.code + "\r\n"
    vtcapp.sendfile(projektstr, 'projekte.txt', True, True)
```

Beispiel wie ein lokal vorliegendes Bild geschickt werden kann:

```
filename = r"C:\Arbeitsverzeichnis\python_editor.png"

# opening for [r]eading as [b]inary
with open(filename, 'rb') as afile:
    vtcapp.sendfile(afile, 'aFilenameHere.jpg', True, True)
```

Files sollten explizit geschlossen werden nach Gebrauch. Ansonsten werden sie erst vom Garbage Collector geschlossen oder im Fehlerfall sogar erst bei der nächsten Exception. Es wird empfohlen, `open` mit `with` zu verwenden wie im obigen Beispiel.

6.12 Suchdialoge für Expressionordner

Linie: Standard, Expert | Module: Leistung & CRM | Version 6.2

Mit der vorliegenden Version wurde das Handling der Expression-Ordner dem der SQL-Ordner angeglichen. So können nun auch bei Expression-Ordnern Abfragen definiert und die eingegebenen Parameter in der Ordner-Expression sowie in den Spalten-Expressions verwendet werden.

Es kann wie bisher einfach eine Expression eingegeben werden, ohne dass eine Abfrage definiert werden muss. In diesem Fall verhält sich der Expression-Ordner wie gewohnt, und die bereits bestehenden Expression-Ordner verhalten sich gleich wie bis anhin.

Möchte man dieser Expression jedoch gewisse Suchparameter übergeben können, welche vom User abgefragt werden, können dafür nun Abfragen definiert werden.

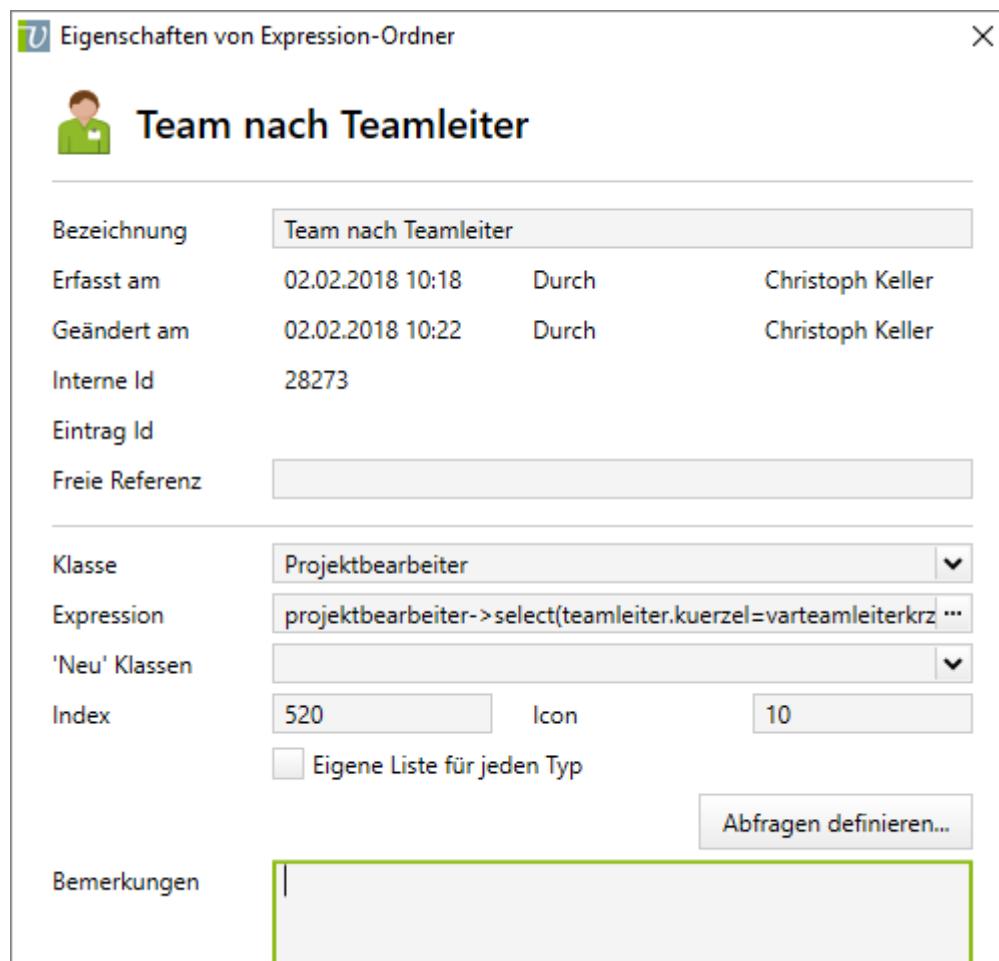


Abbildung 39: Eigenschaften-Dialog eines Expression-Ordners

Über den Button **Abfragen definieren...** öffnet sich der Dialog für die Erstellung der Abfragen.

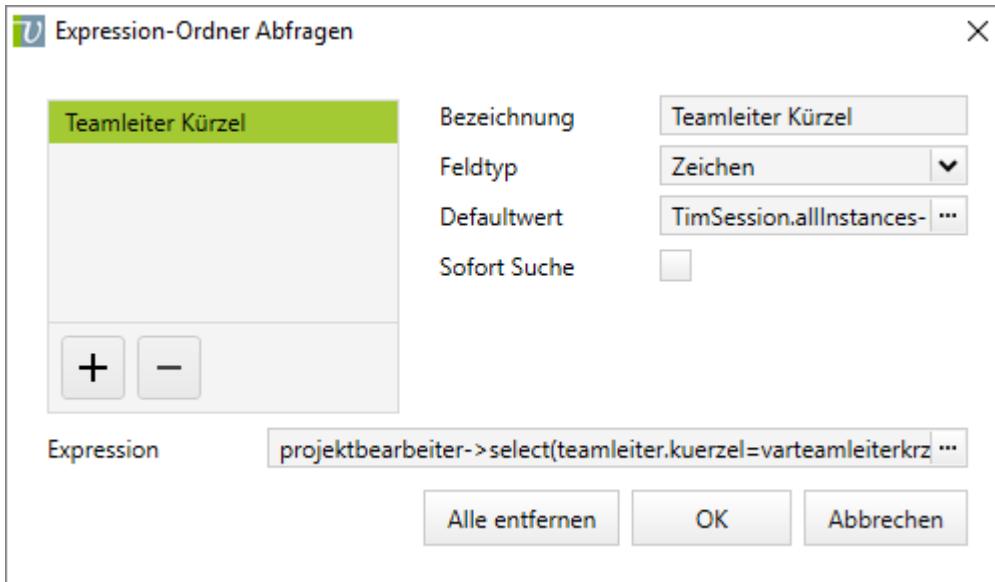


Abbildung 40: Der Abfrage-Dialog

Die Feld-Bezeichnungen werden wie bei den SQL-Ordnern als Variablennamen übernommen, mit vorangestelltem `var`. Im obigen Beispiel gibt es also aus diesem Feld eine Variable `varteamleiterkrzel`, welche in der OCL Expression sowie in den Spaltenexpressions der Listeneinstellungen verwendet werden kann:

```
projektbearbeiter->select (teamleiter.kuerzel=varteamleiterkrzel)
```

Der OCL Expression Editor kennt diese Variablennamen zeigt die Expression als gültig an.

Es können wie beim SQL-Ordner auch mehrere Suchfelder definiert werden. Diese werden mit dem `+` im Abfrage-Dialog erstellt.

Einzelne Abfragefelder können mit dem `-` Button wieder entfernt werden.

Sobald man im Abfrage-Dialog auf `OK` klickt, wird die Abfrage angelegt und auf der Oberfläche das entsprechende Feld angezeigt:

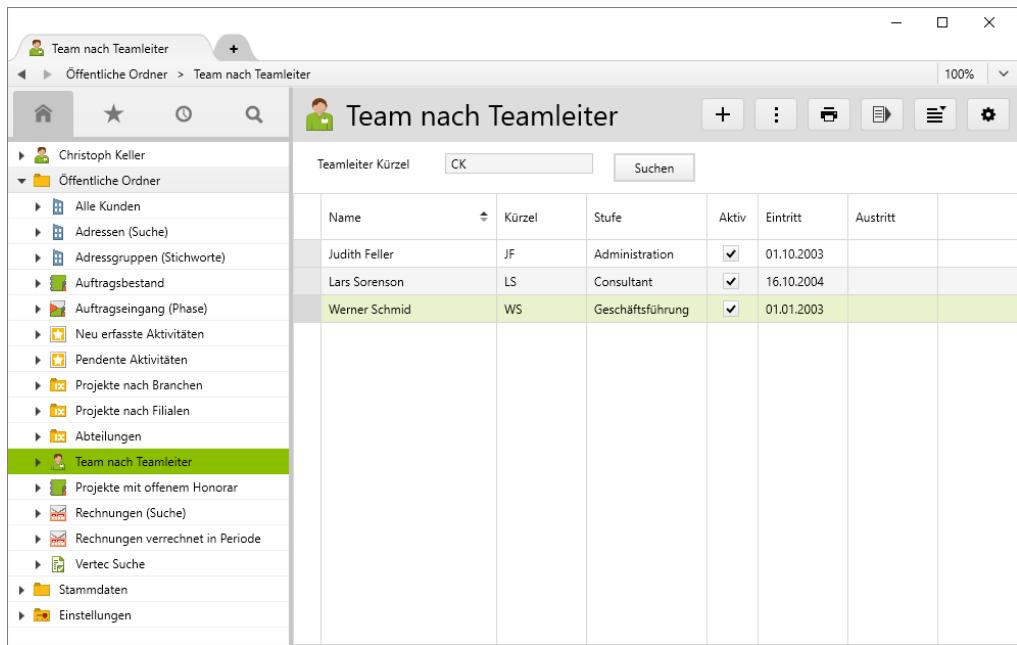


Abbildung 41: Expression-Ordner mit Abfrage

Sofort Suche Die Ordner-Suche kann auch unmittelbar ausgeführt werden, indem im Abfrage-Dialog die **Sofort Suche** aktiviert wird. Siehe dazu die separate Beschreibung im Kapitel 6.13.

Die gesamte Abfrage wird durch Klick auf den Button **Alle entfernen** im Abfrage-Dialog gelöscht, und der Expression-Ordner verhält sich wieder wie ein "gewöhnlicher" Expression-Ordner.

Bei Expression-Ordnern mit Abfrage wird auf dem Eigenschaften-Dialog der Klassennamen angezeigt:

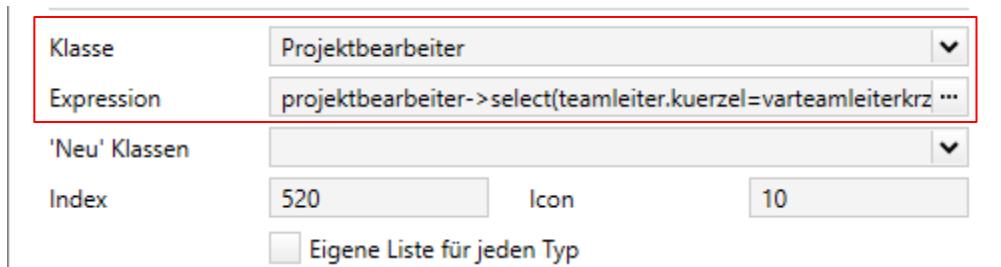


Abbildung 42: Expression-Ordner mit Abfragen zeigen den Klassennamen an

Bei Expression-Ordnern ohne Abfrage ist dieses Feld ausgeblendet, da es nicht benötigt wird, weil sich die Klasse aus der Expression von selbst ergibt.

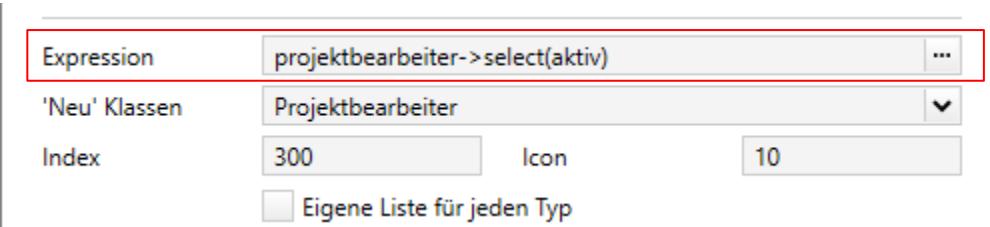


Abbildung 43: Expression-Ordner wie bisher

6.13 Ordner-Suche unmittelbar ausführen

Linie: Standard & Expert | Module: Leistung & CRM | Version 6.2

SQL-Ordner und Expression-Ordner mit Abfragen (siehe 6.12) können neu mit Standard-Suchparametern direkt ausgeführt werden, ohne dass noch auf **Suchen** geklickt werden muss.

Dies kommt dann zur Anwendung, wenn sinnvolle Standardwerte in den Suchparametern bereits hinterlegt sind (z.B. der angemeldete Benutzer), welche üblicherweise nur noch ausgeführt werden.

Dafür gibt es eine neue Checkbox **Sofort Suche**:

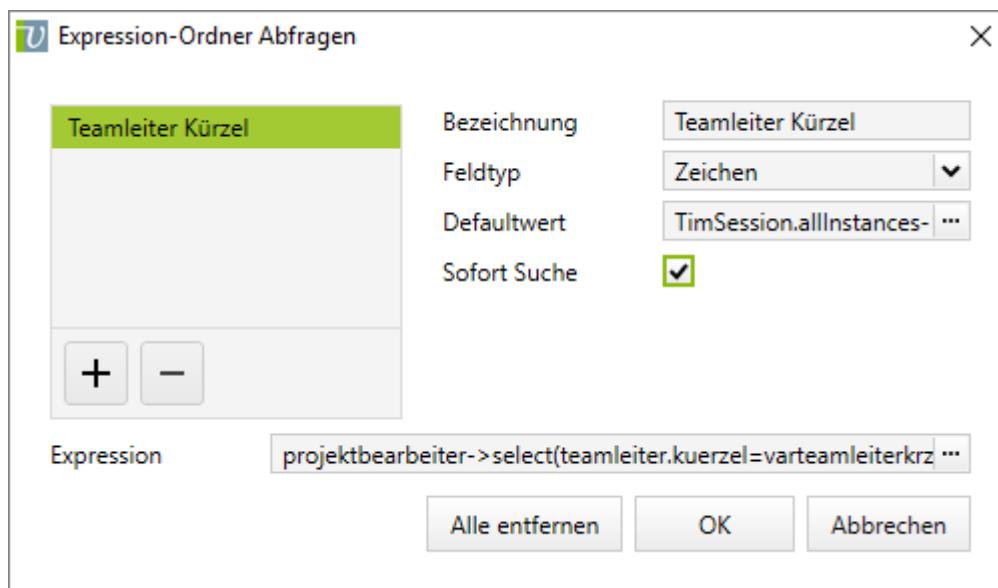


Abbildung 44: Expression-Ordner Abfrage mit Sofort Suche

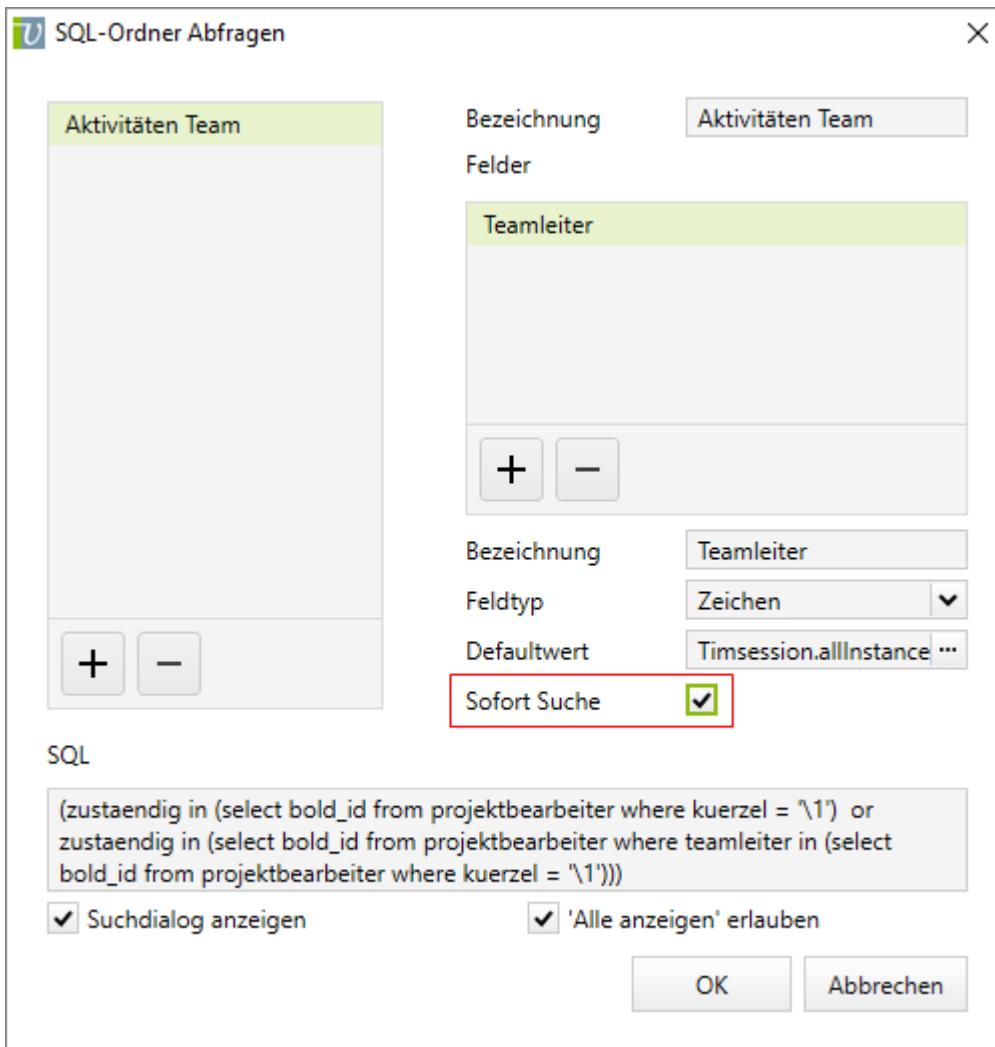


Abbildung 45: SQL-Ordner Abfrage mit Sofort Suche

Damit wird die Suche direkt beim Klicken auf den Ordner ausgelöst und die Resultate in der Liste angezeigt.

Weitere Suchen können wie gewohnt durch Ändern der Suchparameter und mittels Suchbutton erfolgen.

6.14 Berechtigung für SQL-Abfragen

Linie: Standard & Expert | Module: Leistung & CRM | Version 6.2

Für SQL-Abfragen werden neu entweder Administratorenrechte oder ein spezielles SQL-Recht namens **SQL Query** benötigt, welches mit dieser Version eingeführt wird. Ist keine entsprechende Berechtigung vorhanden, erscheint beim SQL-Aufruf eine Fehlermeldung.

SQL-Abfragen gibt es in folgenden Bereichen:

Scripts (Python und VBScript)

In Scripts geht es um die Verwendung der **GetWithSQL** und **GetMemberWithSQL** Script-Methoden. Werden diese Methoden in Scripts aufgerufen, muss der ausführende Benutzer über Administratorenrechte oder über das **SQL Query** Recht verfügen.

Bei Scripts gibt es zusätzlich die Variante, erweiterte Berechtigungen einzuschalten. Diese vergeben dann für den Bereich zwischen `beginSystemContext` und `EndSystemContext` dem ausführenden Benutzer Administratorenrechte.

Erweiterte Berechtigungen Weitere Informationen zum Thema Erweiterte Berechtigungen in Scripts finden Sie in der Online Knowledge Base unter www.vertec.com/de/kb/erweiterte-berechtigungen.

ActiveX/COM

Bei Aufrufen via COM (Word, Excel) geht um die Methoden `GetWithSQL` und `MemberWithSql`.

Der ausführende Benutzer muss über Administratorenrechte oder über das `SQL Query` Recht verfügen.

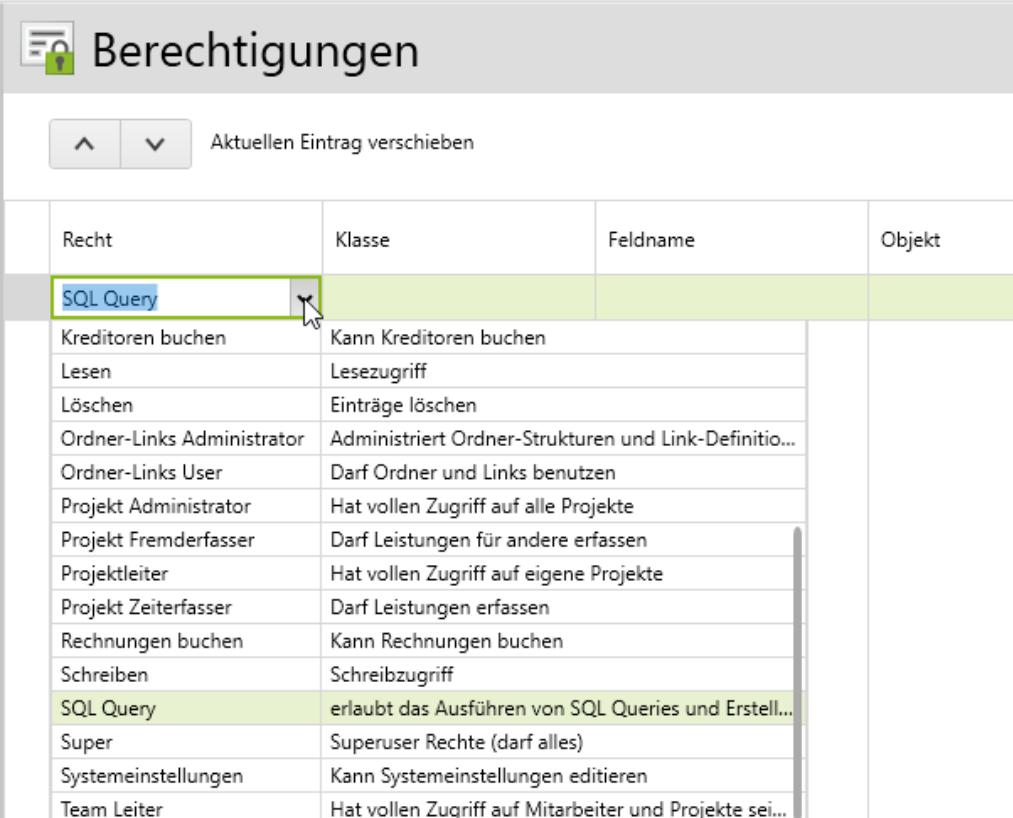
XML

Bei XML heisst die entsprechende Methode `sqlwhere`.

Der ausführende Benutzer muss über Administratorenrechte oder über das `SQL Query` Recht verfügen.

SQL Query Recht

Es gibt eine neue Berechtigung namens `SQL Query`:



Recht	Klasse	Feldname	Objekt
SQL Query			
Kreditoren buchen	Kann Kreditoren buchen		
Lesen	Lesezugriff		
Löschen	Einträge löschen		
Ordner-Links Administrator	Administriert Ordner-Strukturen und Link-Definitio...		
Ordner-Links User	Darf Ordner und Links benutzen		
Projekt Administrator	Hat vollen Zugriff auf alle Projekte		
Projekt Fremderfasser	Darf Leistungen für andere erfassen		
Projektleiter	Hat vollen Zugriff auf eigene Projekte		
Projekt Zeiterfasser	Darf Leistungen erfassen		
Rechnungen buchen	Kann Rechnungen buchen		
Schreiben	Schreibzugriff		
SQL Query	erlaubt das Ausführen von SQL Queries und Erstell...		
Super	Superuser Rechte (darf alles)		
Systemeinstellungen	Kann Systemeinstellungen editieren		
Team Leiter	Hat vollen Zugriff auf Mitarbeiter und Projekte sei...		

Abbildung 46: SQL Query Recht

Wir empfehlen, die **SQL Query** Berechtigung nur sehr restriktiv zu vergeben. Mit dieser Berechtigung wird den entsprechenden Benutzern weitreichender SQL-Zugriff auf die Daten ermöglicht.

OCL hasright Der OCL **hasright**-Rechtname heisst **sqlquery**. Siehe dazu den Abschnitt über die User-Rechte im Artikel OCL unter www.vertec.com/de/kb/ocl#user-rechte.

Rückwärtskompatibilität

Auch bei bestehenden Datenbanken werden neu entweder Administratorenrechte oder ein spezielles SQL-Recht benötigt, um SQL-Abfragen abzusetzen. Es kann also sein, dass Sie nach dem Update in Scripts, Excel-Berichten und bei XML-Aufrufen Fehlermeldungen erhalten.

Bei Scripts empfehlen wir, diese so anzupassen, dass der Aufruf innerhalb des **beginSystemContext/endSystemContext** erfolgt, und dem Script erweiterte Berechtigungen zuzuweisen.

Bei Aufrufen via COM und XML muss den entsprechenden Benutzern das **SQL Query** Recht erteilt werden. Wir empfehlen, dafür eine eigene Benutzergruppe zu erstellen.

6.15 Berechtigungsprüfung bei persönlichen Ordnern geändert

Linie: Standard | Module: Leistung & CRM | Version: 6.2

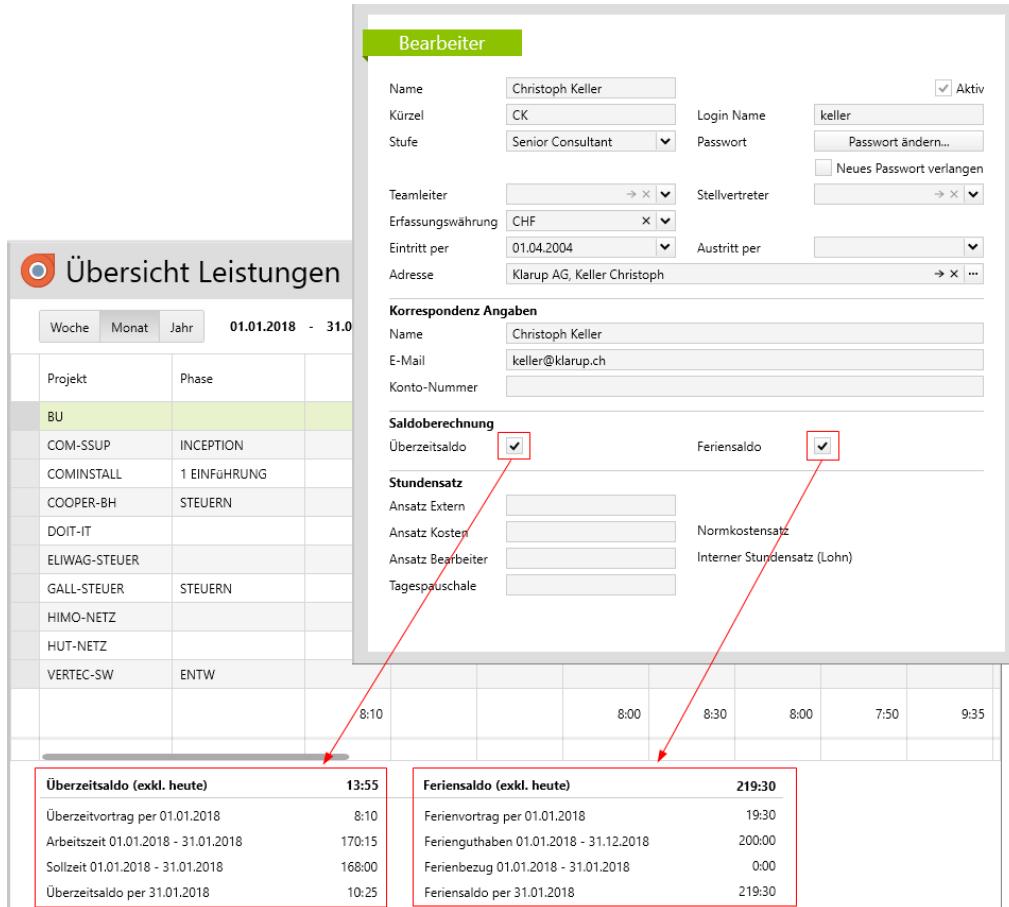
Die Berechtigungsprüfung bei persönlichen Ordnern wurde angepasst. Falls Sie mit persönlichen Ordnern arbeiten (das erkennen Sie daran, dass Sie einen Root-Ordner **Persönliche Ordner** haben) kann es sein, dass Sie eine Änderung feststellen – zum Beispiel gewisse Auswertungen nicht mehr aufrufen können – und die Berechtigungen entsprechend anpassen müssen.

7 Leistungen/CRM

7.1 Darstellung Saldi auf Übersicht Leistungen

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.2

Die Darstellung der Überzeit- und Feriensi Saldi auf **Übersicht Leistungen** richten sich neu nach den entsprechenden Optionen für die Saldoberechnung auf dem Bearbeiter.



Überzeitsaldo (exkl. heute)		13:55
Überzeitvortrag per 01.01.2018	8:10	
Arbeitszeit 01.01.2018 - 31.01.2018	170:15	
Sollzeit 01.01.2018 - 31.01.2018	168:00	
Überzeitsaldo per 31.01.2018	10:25	

Feriensaldo (exkl. heute)		219:30
Ferienvortrag per 01.01.2018	19:30	
Ferienguthaben 01.01.2018 - 31.12.2018	200:00	
Ferienbezug 01.01.2018 - 31.01.2018	0:00	
Feriensaldo per 31.01.2018	219:30	

Abbildung 47: Steuerung der Anzeige durch die Saldoberechnungsoptionen

Ist für einen Bearbeiter eine oder beide der Optionen deaktiviert, werden die entsprechenden Zahlen auf Übersicht Leistungen automatisch ausgeblendet.

7.2 Aktivitäten-Ordner auf Offerten/Angeboten

Linie: Expert | Module: Budget & Teilprojekt | Version 6.2

Es gibt einen neuen Linktyp für Aktivitäten auf Offerten/Angeboten. Dieser heisst **Offerte – Aktivitäten** bzw. **Angebot – Aktivitäten** und ist standardmäßig verfügbar.

8 Rechnungsstellung

8.1 Unterstützung für Einzahlungsschein mit QR-Code

Linie: Expert | Module: Leistung & CRM | Version: 6.2

Es gibt in der Schweiz einen neuen Standard für Einzahlungsscheine. Die bisherigen OCR-B Referenzzeile wird dabei durch einen QR-Code mit allen relevanten Rechnungsangaben ersetzt.

QR-Rechnungen Nähere Informationen dazu finden Sie unter

- <https://www.iso-20022.ch/lexikon/einheitsbeleg/>
- <https://www.postfinance.ch/de/biz/zv.html>

Dafür gibt es neu auf der Rechnung ein derived Member `qrCode`, welches den QR-Code als Binärgrafik (PNG) berechnet. Dieses Member kann dann als Image im Word-Report eingesetzt werden.

- Inhalt des QR-Codes ist ein ANSI Latin-1 encodierter Text.
- Die einzelnen Datenfelder sind durch CR/LF getrennt.
- Die Bedeutung der Felder ist durch ihre Position (Reihenfolge) gegeben.

Mitgelieferte Rechnungsreports

Die von Vertec mitgelieferten Standard-Rechnungsvorlagen mit Einzahlungsscheinen sind wie folgt angepasst:

- ESR und QR-Code werden je auf einer einzelnen Seite ausgegeben.
- Der QR-Code wird dann ausgegeben, wenn eine IBAN-Nummer auf dem Zahlungstyp angegeben ist (siehe 8.2).
- Für die Ausgabe des ESR ist die Voraussetzung wie bisher das Vorhandensein einer Teilnehmer-Nummer.

Gibt es aus irgendeinem Grund einen Fehler bei der Berechnung des QR-Codes, wird statt des QR-Codes eine Fehlermeldung angedruckt. Das entsprechende Attribut heisst `qrCodeError`.

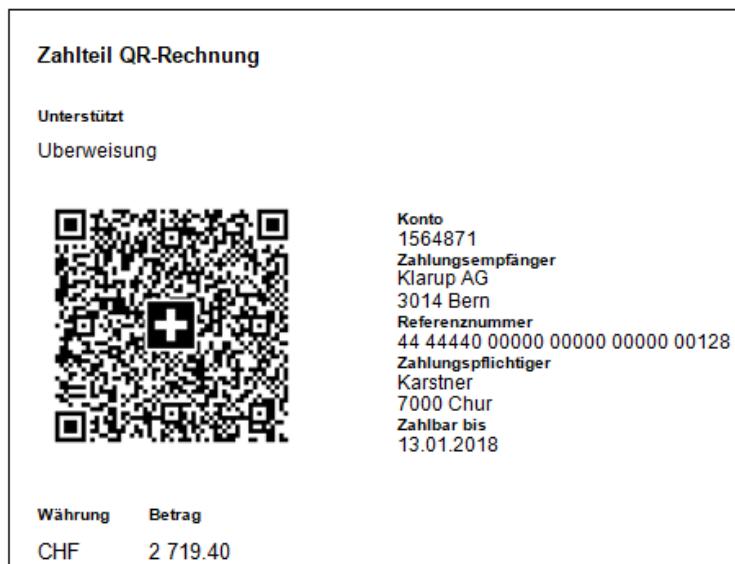


Abbildung 48: Zahlteil QR-Rechnung

8.2 Unterstützung von IBAN Nummer auf Zahlungstyp

Linie: Expert | Module: Leistung & CRM | Version: 6.2

Für die Unterstützung von Einzahlungsscheinen mit QR-Code (siehe 8.1) wurde auf dem Zahlungstyp ein neues Feld **IBAN** eingeführt.

Zusätzlich dazu gibt es neu in den **Systemeinstellungen Rechnung** eine Einstellung **Standard-Zahlungs-Typ**.

Die Projekttypen sowie die Projekte erben diese Einstellung, können aber auf beiden Ebenen noch übersteuert werden.

Rückwärtskompatibilität

Bei einem Update wird ein neuer Standard-Zahlungstyp erzeugt und die bisherigen Systemeinstellungen für ESR Angaben darin übernommen. Dieser Zahlungstyp wird in den Systemeinstellungen Rechnung als Standard Zahlungs-Typ eingesetzt.

Die folgenden Einstellungen werden übernommen und die entsprechenden Systemeinstellungen entfernt:

- BESR-Bank
- BESR-Kunden-Identifikations-Nummer
- VESR-Belegart
- VESR-Referenznummerlänge
- VESR-Teilnehmer-Nummer
- VESR oder BESR verwenden

Das bedeutet, dass die ESR Angaben in Zukunft vom Zahlungstyp und nicht mehr aus den Systemeinstellungen genommen werden. Im Betrieb merken Sie davon aber nichts.

Bestehende Zahlungstypen werden nicht angetastet und weiterhin überall dort verwendet, wo sie gesetzt sind (Projekttyp, Projekt).

Möchten Sie neu einen bereits bestehenden Zahlungstyp als Standard verwenden, wählen Sie diesen in den Systemeinstellungen Rechnung als Standard aus.

Weitere Informationen Die Systemeinstellung **BESR-Bank** war bisher eine Referenz auf einen Vertec-Adresse. Neu ist es ein Textfeld auf dem Standard Zahlungstyp. Das spielt insofern keine Rolle, als dass auf den Report schon immer nur der Text verwendet wurde. Der Adresseintrag der Bank selbst hat allerdings nun keine Bedeutung mehr für den ESR-Vorgang.

9 Reporting

9.1 Neuimplementierung Vertec-generierte Reports

Linie: Standard, Expert | Modul: Leistung & CRM | Version 6.2

Im Zuge der Einführung von erweiterten Office-Berichten (siehe Kapitel 4) wurde das Reporting-System für Vertec-generierte Word-Reports komplett neu implementiert.

Dies sollte für den Benutzer nicht spürbar sein, es ergeben sich aber einige wenige Unterschiede:

Locale-Bands

Bei der Verwendung eines Locale-Bands wurden bisher alle Zahlen- und Datumswerte innerhalb des Bands, welche nach den Visual Basic (VBA) Formatierungsregeln formatiert waren (z.B. `\#, ##0.00`), nach den in der Locale-Expression angegebenen Ländereinstellung formatiert.

Dieses System wurde in den Vertec-generierten Word-Reports so erweitert, dass ALLE Zahlen- und Datumswerte innerhalb des Bands automatisch nach der angegebenen Locale-Expression formatiert werden. Es ist somit nicht mehr nötig, Zahlenwerte, die in anderen Ländereinstellungen formatiert werden sollen, extra mit VBA-Formatierungen zu versehen (bereits bestehende VBA-Formatierungen stören nicht).

Unterschiede Office- und Vertec-generiert

Die Unterschiede in der Darstellung von Office-generierten und Vertec-generierten Word-Reports haben üblicherweise mit der Behandlung von leeren Abschnitten (Paragraphs) zu tun.

Je nachdem, wie die Textmarke (Bookmark) eines Bands definiert ist, wird der Umfang des Bands in Office- und Vertec-generierten Berichten geringfügig anders interpretiert.

Es gibt dabei 2 generelle Arten von Unterschieden:

Bands über Dokumenthierarchiestufen hinweg

Im Gegensatz zum VBA-basierten Report-Generator der Office-generierten Berichte unterstützt das server-basierte Reporting System für die Vertec-generierten Reports nur Band-Definitionen, deren Anfang und Ende sich auf gleicher Ebene befindet. Das bedeutet folgendes:

Band-Start	Band-Ende	Vertec-generierte Reports
Text (normaler Paragraph)	Text (normaler Paragraph)	OK
Text (normaler Paragraph)	Text (innerhalb Table)	nicht OK
Text (normaler Paragraph)	Ende von Tabellenzeile	nicht OK
Beginn Tabellenzeile	Ende von Tabellenzeile	OK
Beginn Tabellenzeile	Text in einer Tabellenzelle, nicht am Ende	nicht OK

Damit diese Bedingungen möglichst gut erfüllt sind, werden Band-Start und Band-Ende möglichst grosszügig interpretiert.

Wenn sich z.B. ein Band-Start am Ende eines Absatzes, welcher Text enthält befindet, aber nur die Absatzmarke umfasst, ist das eigentlich nicht gültig. Damit der Bericht trotzdem funktioniert, wird der Band-Start nach vorne an den Beginn des nächsten Elements (Absatz oder Table) geschoben, falls das Band-Ende am Ende einer Tabellenzeile oder eines Absatz ist.

Text im Absatz

Tabellen	☒	☒
☒	☒	☒

Diese automatische Anpassung führt natürlich dazu, dass ein Absatzwechsel weniger dargestellt wird, ist also im Output sichtbar.

Solche Bands sollten nach Möglichkeit nicht definiert werden, sondern möglichst immer ganze Abschnitte oder Start und Ende innerhalb des Textes als Band definiert werden.

Befindet sich der Start des Bands sogar mitten im Absatztext, wie in der folgenden Abbildung dargestellt, dann erscheint eine Fehlermeldung betreffend ungültiger Band-Definition.

Text im Absatz

Tabellen	☒	☒
☒	☒	☒

Ununterscheidbare Varianten

Es gibt Fälle von Band-Definitionen, welche im server-basierten Code nicht unterschieden werden können, obwohl sie in Word visuell unterscheidbar sind.

Dies ist zum Beispiel dann der Fall, wenn eine Textmarke am Schluss eines leeren Absatzes endet, welcher auf eine Tabelle folgt:

Tabellen	☒	☒
☒	☒	☒

In diesem Fall befindet sich das Bookmark-Ende innerhalb des leeren Absatzes. Endet das Band aber am Ende der Tabelle, enthält den leeren Absatz also nicht, dann befindet sich das Bookmark-Ende ebenfalls im darauffolgenden Abschnitt.

Bookmark-Enden in einem sonst leeren Absatz werden deshalb immer als Ende der vorangehenden Tabelle interpretiert.

Soll ein leerer Absatz nach einer Tabelle explizit zum Band gehören, kann dies erreicht werden, indem im Absatz irgendein Text (z.B. ein Leerzeichen) eingefügt wird. Dann wird der Absatz als zum Band zugehörig interpretiert.

Rückwärtskompatibilität

Haben Sie bereits Vertec-generierte Reports im Einsatz oder möchten Ihre Office-generierten Reports neu von Vertec generieren lassen, überprüfen Sie diese auf die genannten Punkte und passen Sie sie bei Bedarf entsprechend an.

9.2 Office Vorlagen auf neue Office Formate umgestellt

Linie: Standard, Expert | Modul: Leistung & CRM | Version 6.2

Neu werden von Vertec keine Office-Report Vorlagen im Format vor Office 2007 (xlt, dot) mehr mitgeliefert:

- Die Excel Berichtsvorlagen im **.xlt**-Format werden nicht mehr mitgeliefert
- Sämtliche Word-Vorlagen wurden von **.dot** auf **.dotx** umgestellt

- Das Makro-File [VertecReport.dot](#) wird durch [VertecReport.dotm](#) ersetzt. Ist [VertecReport.dotm](#) nicht vorhanden, können keine Reports mehr ausgeführt werden.

Rückwärtskompatibilität

Die [.dot](#)- und [.xlt](#)-Files von bestehenden Installationen werden belassen.

Sind im gleichen Verzeichnis dieselben Files in älterer und neuerer Version vorhanden, haben die neueren Files in der Logik Vorrang, wenn der Report in Vertec ohne Endung registriert ist (Standard).

Das bedeutet: Sind Ihre Vertec Reports mit Endung registriert, werden die veralteten Versionen verwendet. Falls Sie die neuen Vertec Vorlagen verwenden möchten, entfernen Sie die Endungen aus den Registrierungen.

9.3 Vorlagen erstellen/bearbeiten in den Cloud Clients

Linie: Standard, Expert | Modul: Leistung & CRM | Version 6.1.0.7

Zur Bearbeitung von Dokumentvorlagen via Cloud Clients steht neu ein Upload/Download Mechanismus zur Verfügung. Dieser funktioniert wie folgt:

Klicken Sie mit der rechten Maustaste neben die Vorlage, die Sie bearbeiten möchten, und wählen Sie [Vorlage bearbeiten...](#):

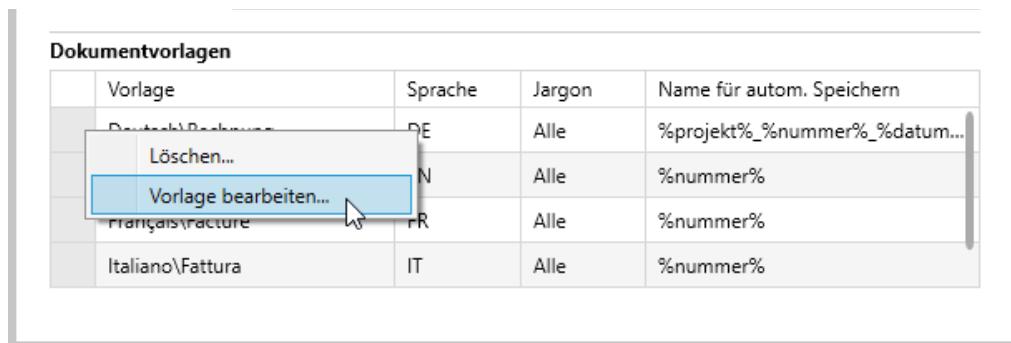


Abbildung 49: Vorlage bearbeiten wählen

oder klicken Sie doppelt in die entsprechende Zeile. Es erscheint der Upload/Download Dialog:

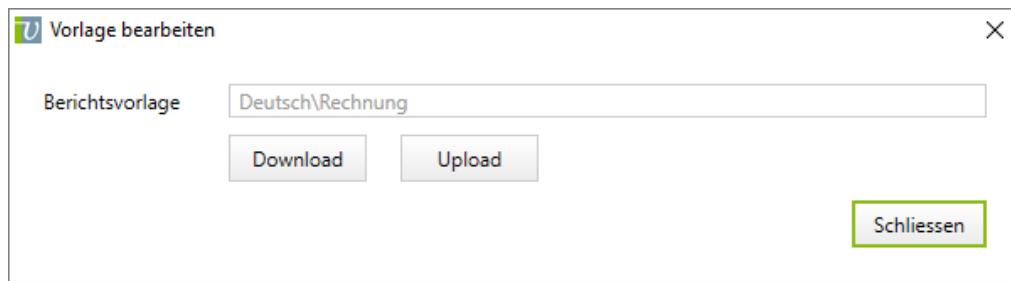


Abbildung 50: Der Dialog für Upload/Download der Berichtsvorlagen

Einen bestehenden Bericht anpassen

- **Download:** Um einen bestehenden Bericht zu bearbeiten, klicken Sie auf [Download](#). Der Bericht wird heruntergeladen und Sie werden gefragt, wo Sie ihn lokal speichern möchten (in der Web App wird der Report ins lokale Temp-Verzeichnis abgelegt und direkt zur Bearbeitung geöffnet).

- **Upload:** Um den Bericht nach dem Bearbeiten auf dem Server zu speichern, klicken Sie auf **Upload**. Im Explorer wählen Sie den vorher bearbeiteten Report aus. Dieser wird automatisch im unter Vorlagen-Pfad für Word-Berichte angegebenen Pfad gespeichert.

Einen neuen Bericht registrieren

Um einen neuen Bericht zu registrieren, füllen Sie den gewünschten Datei-Namen des Reports in das Feld bei Vorlage, und klicken dann auf **Vorlage bearbeiten** > **Upload**.

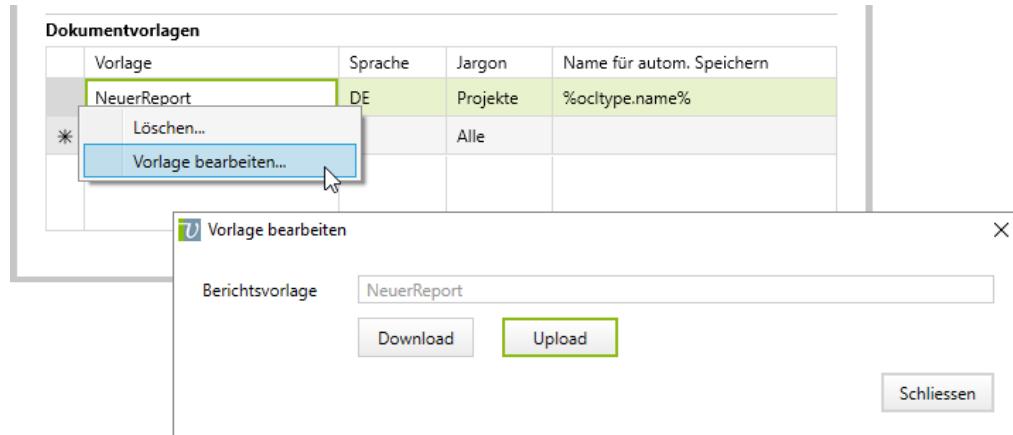


Abbildung 51: Einen neuen Report erstellen

Sie können den gewünschten Report auswählen, und er wird unter dem angegebenen Namen im unter Vorlagen-Pfad für Word-Berichte angegebenen Pfad abgelegt.

Weitere Informationen Die Grösse für Uploads ist auf 1 MB beschränkt. Die Beschränkung wird Client- und Serverseitig überprüft.

Download der Berichtsvorlagen beim Ausführen (Cloud App)

Ab Version 6.1.0.7 sind Client-Side Office Reports möglich mit der Cloud App, auch wenn kein Zugriff auf das Filesystem auf dem Server besteht. Dafür wird beim Ausführen eines Office Reports die entsprechende Vorlage ins lokale temp-Verzeichnis heruntergeladen und der Report dann mit dieser Vorlage gestartet. Excel-Reports sowie Nicht-Vertec-generierte Word-Reports verhalten sich neu so.

- Die Vorlagen müssen sich unterhalb des Vorlagen-Pfads befinden. Dateien ausserhalb des Report Pfades können damit nicht heruntergeladen werden.
- Existiert auf dem Client im temp-Verzeichnis eine Vorlage bereits, wird dieses überschrieben. Falls das nicht möglich ist, wird dem Namen eine Nummerierung angefügt.

9.4 Excel Export: Auswahlfelder werden nicht exportiert

Linie: Standard, Expert | Modul: Leistung & CRM | Version 6.1.0.14

Auswahlfelder in Listen wurden in Excel nicht korrekt dargestellt. Statt des Wertes erschienen Klassennamen wie

`Vertec.ServerControls.AdditionalFields.AdditionalFieldIntValue.`

Dieses Problem wurde behoben.

9.5 Locale-Bands in Word-Reports akzeptieren neu auch OCL-Expressions

Linie: Standard, Expert | Modul: Leistung & CRM | Version 6.2.

Mit sogenannten Locale-Bands kann in Word-Reports angegeben werden, welche Regional-einstellungen innerhalb des Bands für Zahlen- und Datumsformatierungen verwendet werden sollen.

Bisher musste hier ein Ländercode angegeben werden. Neu kann auch eine OCL-Expression angegeben werden, welche als Ergebnis einen Ländercode als String zurückgibt. Konkret funktionieren neu folgende Varianten:

- Angabe von Locale-Name als Locale Expression: `de-CH`
 - Angabe von Ocl-Expression, die als Ergebnis einen Locale-Namen liefert: `'de-CH'`
-

Weitere Informationen Weitere Informationen zu den Locale-Bands finden Sie in der Online Knowledge Base unter <https://www.vertec.com/de/kb/wordreports#locale>.

10 Schnittstellen

10.1 ProCall Addin

Linie: Standard, Expert | Module: Leistung & CRM | Version 6.1.0.11

Vertec bietet mit dem ProCall Addin eine Telefonie-Integration in Vertec via ProCall von Estos (<https://www.estos.de/>). Das Addin läuft mit ProCall Versionen ab 5.1.104.41061.

Das Telefonie-Journal

Bei ein- und ausgehenden Anrufen zeigt ProCall ein Telefoniefenster (Popup) an. Ist das Vertec ProCall Addin installiert, erscheint dort ein Button **Adresse in Vertec öffnen**. So können Sie zum Beispiel bei eingehenden Anrufen die entsprechende Adresse mit der Kundenhistorie direkt in Vertec öffnen und haben alle nötigen Daten per Knopfdruck griffbereit.



Abbildung 52: Das ProCall Telefoniefenster

Auch im Journal sowie bei den Kontakten ist diese Funktion über den rechten Mausklick verfügbar:

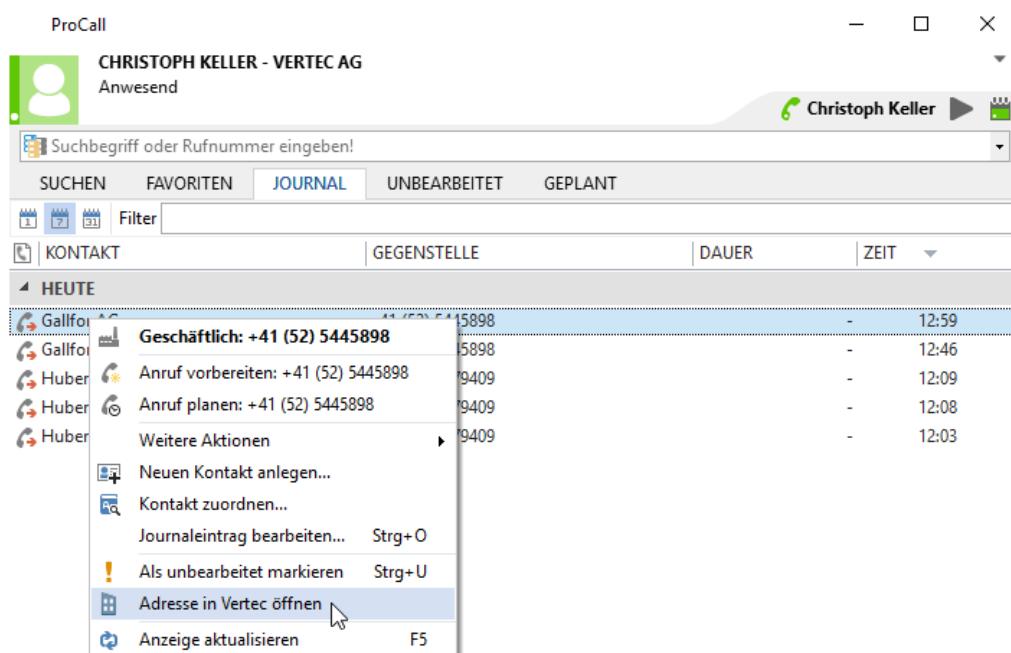


Abbildung 53: Das ProCall Telefoniejournal

Der Zugriff auf die Vertec Adressen erfolgt via COM.

Es können auch mehrere Adressen selektiert und in Vertec geöffnet werden. Es wird für jede selektierte Adresse ein neuer Tab in Vertec geöffnet.

Ausgehende Anrufe

Ist das ProCall Addin installiert, kann aus Vertec direkt via ProCall gewählt werden. Die Nummern werden via URI an den ProCall Client übermittelt.

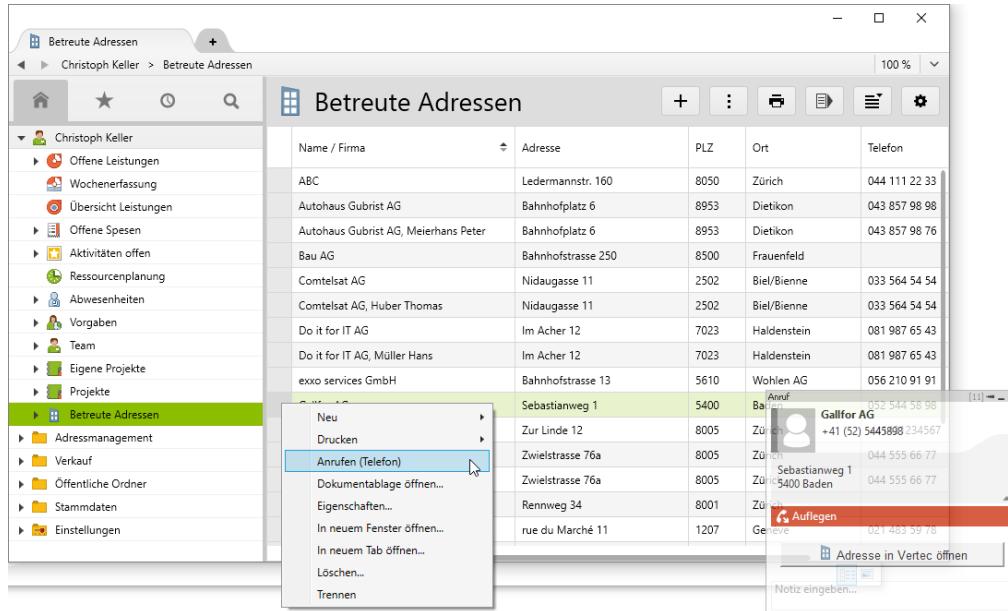


Abbildung 54: Direkt aus Vertec wählen

Eingehende Anrufe

Bei eingehenden Anrufen durchsucht ProCall die KommMittel nach der zu auflösenden Nummer und übernimmt Kontaktdaten wie Name, Ort, Adresse. Falls mehrere Adressen in Vertec die gleiche Nummer besitzen wird diejenige genommen, die zuerst erfasst wurde (niedrigste ID).

Installation

Um das Vertec ProCall Addin installieren zu können, muss ProCall beendet werden.

Das Addin `VertecProCall.dll` befindet sich im Unterordner `Addins` in Ihrem Vertec-Installationsverzeichnis.

Um das Addin zu installieren, führen Sie mit lokalen Administratorenrechten die Datei `InstallProCallAddin.bat` aus, welche sich im Unterverzeichnis `Addins` der Vertec Installation befindet.

Deinstallation

Um das Addin zu deinstallieren, führen Sie mit lokalen Administratorenrechten die Datei `UninstallProCallAddin.bat` aus, welche sich im Unterverzeichnis `Addins` der Vertec Installation befindet.

Dies ist nur notwendig, wenn das Addin nicht mehr verwendet werden soll (keine Buttons/Menüs mehr in den entsprechenden Applikationen). Für das Update (Ersetzen des DLL Files) ist keine Aufhebung der Registrierung oder Neuregistrierung notwendig. Es muss lediglich sichergestellt sein, dass ProCall nicht läuft, dann kann die DLL überschrieben werden.

10.2 Abacus: Betrag darf nur 2 Nachkommastellen haben

Linie: Standard, Expert | Module: Leistung & CRM | Version 6.1.0.12

Alle Abacus Schnittstellen dürfen Beträge nur mit 2 Nachkommastellen schreiben. Dies wurde mit der aktuellen Version umgesetzt.

10.3 Abacus: Land-Code aus Vertec Adresseintrag verwenden

Linie: Standard, Expert | Module: Leistung & CRM | Version 6.1.0.12

Die Abacus Fibu Extensions haben eine eingebaute Mapping Tabelle zur Bestimmung des bei Buchungen zu verwendenden Country-Codes.

Diese Tabelle erlaubt es, dass Klartext Ländernamen in Vertec Adresseinträgen auf die von Abacus verwendeten Länderkürzel umgesetzt werden.

Diese Mapping Tabelle ist im Extension Code fix enthalten und enthält nur Einträge für Schweiz, Deutschland, Italien und Frankreich. Ist ein Land nicht in dieser Tabelle, wird beim Buchen ein leerer Ländercode verwendet und Abacus meldet Fehler.

Neu wird das Land auf der Adresse (`standardland` auf Adresseintrag) als Countrycode für die Schnittstelle verwendet, falls das Land in der Mapping Tabelle nicht vorhanden ist.

Rückwärtskompatibilität

Bisher wurden alle Werte, welche nicht in der Mapping Tabelle vorkommen, als "CH" erzeugt. Falls Werte wie "Kanarische Inseln" etc. eingetragen sind, gibt das neu einen Fehler. Es muss sichergestellt werden, dass entweder ein gültiger Countrycode im Feld steht oder nichts (in diesem Fall wird "CH" verwendet).

10.4 SelectLine COM Schnittstellen Lizenzierung

Linie: Standard, Expert | Module: Leistung & CRM | Version 6.2

Die COM-Schnittstelle der SelectLine Software, zu der wir Extensions-Schnittstellen anbieten, ist für SelectLine Auftrag ab Version 15.1 und für Rechnungswesen und Lohn ab Version 16.1 lizenpflichtig.

Ab SelectLine Version 16.1 besteht die Möglichkeit, die COM-Schnittstelle direkt durch das Drittprogramm des Schnittstellenanbieters freischalten zu lassen, ohne dass der Kunde eine Lizenz kaufen muss. Dies bietet Vertec nun an.

Da jede neue SelectLine Hauptversion einen neuen COM-Freischaltcode benötigt, muss dieser vom Kunden in Vertec eingetragen werden. Der Kunde erhält diesen Freischaltcode direkt von SelectLine.

Unter Systemeinstellungen > Buchhaltung erscheint neu ein Feld `Selectline COM Freischaltcode`, sobald eine SelectLine Extension installiert wurde. Geben Sie hier den Freischaltcode ein, welchen Sie von Ihrem SelectLine-Betreuer erhalten.

11 Technik und Datenmodell

11.1 Einführung von freiem Referenzfeld

Linie: Standard, Expert | Modul: Leistung & CRM | Version 6.2.

Neu gibt es auf jedem Vertec Objekt ein Feld namens **Freie Referenz** (`freeReference`). Dieses kann frei verwendet werden.

Der Grund dafür ist, dass das Feld Eintrag Id nicht mehr frei verwendet werden kann, da dies neu für die Config Sets (siehe Kapitel 5) verwendet wird.

Das Feld Eintrag Id wurde teilweise für das Speichern von Referenzen in externe Systeme beim Import von Daten (Daten-Migrationen) verwendet. Dafür kann neu das **Freie Referenz** Feld verwendet werden.

Auf der Oberfläche gibt es dafür das Feld neu im Eigenschaften-Dialog (rechte Maustaste > Eigenschaften):

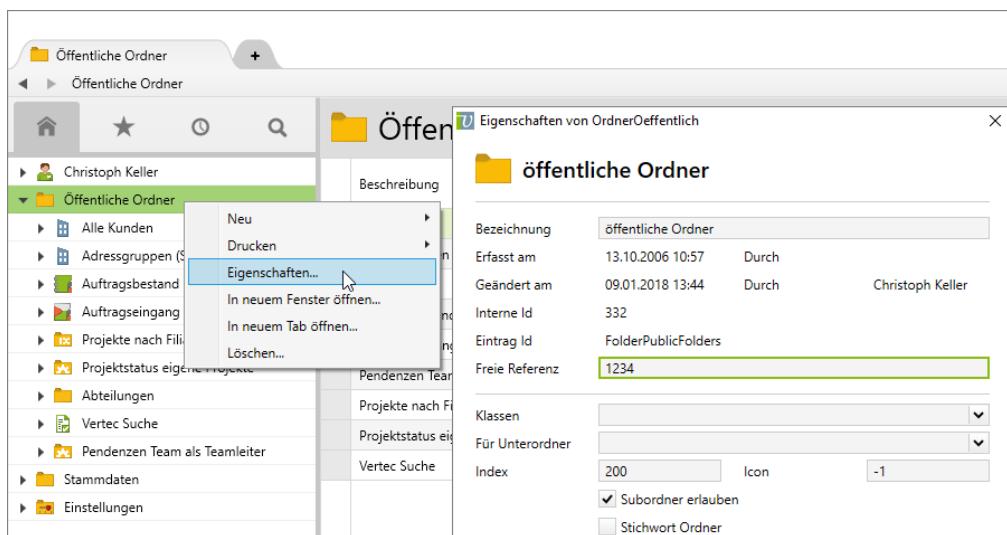


Abbildung 55: Feld für die Freie Referenz im Eigenschaften-Dialog

Der Wert kann auch über die Liste oder im Code gesetzt werden. Das entsprechende Attribut in OCL heisst `freereference`.

Update auf Version 6.2

Beim ersten Aufstarten von Vertec mit Version 6.2 (siehe 1.3) werden alle Strukturdaten, welche ursprünglich von Vertec erzeugt wurden, mit einer Eintrag Id versehen.

11.2 Festlegung minimaler Version für DB-Konvert

Linie: Standard, Expert | Modul: Leistung & CRM | Version 6.2.

Beim Update von älteren Vertec Versionen auf die neueste Version kann ein Schritt über eine Zwischenversion nötig sein. Eine entsprechende Liste finden Sie im Abschnitt **Update von älteren Vertec Versionen** im Artikel **Update von Vertec** in der Online Knowledge Base unter <https://www.vertec.com/de/kb/updateinstallation#olderversions>.

Neu wird das von Vertec automatisch erkannt. Wird ein Konvert von einer zu frühen Version versucht, welche eigentlich einen Zwischenschritt erfordern würde, erscheint eine Fehlermeldung der Art:

Eine Konvertierung der Datenbankversion 5.7.0 auf die Programmversion 6.2.0.1 ist nicht direkt möglich. Verwenden Sie eine Vertec Version vor 6.2.0 zur Durchführung des ersten Konvertier-Schrittes.

11.3 Kommandozeilen-Parameter

Linie: Standard, Expert | Modul: Leistung & CRM | Version: 6.2

Diverse Parameter wurden neu eingeführt oder angepasst.

Parameter	Version	Beschreibung
/script	6.1.0.11	<p>Zusätzlich zu den registrierten Scripts werden auch Python Scripts, welche auf dem Filesystem liegen (mit Endung <code>.py</code>) unterstützt. In diesem Fall muss als Parameter der Dateipfad übergeben werden.</p> <p>Für diesen Vorgang braucht es Vertec Administratorrechte.</p>
/setpassword	6.1.0.11	Erlaubt das Setzen des Passworts eines Users in Vertec Desktop.
/batch_nologin	6.1.0.12	<p>Startet Vertec im Batch-Modus ohne den Login Dialog. Dabei werden nur die Vorgänge VOR dem Login durchgeführt und Vertec dann wieder beendet. Es kann damit also nicht das Login umgangen werden.</p> <p>Um Vertec im Batch-Modus auszuführen, ohne ein Login einzugeben, wurde die Möglichkeit des "Angemeldet bleiben" auf den Batch-Modus erweitert (siehe 11.9).</p>

Weitere Informationen Detaillierte Informationen zu den verschiedenen Parametern finden Sie im Artikel [Kommandozeilen-Parameter](#) in der Online Knowledge Base unter www.vertec.com/de/kb/parameter.

11.4 Parameter `/noevents` nur noch für Administratoren

Linie: Standard, Expert | Modul: Leistung & CRM | Version 6.2.

Das Starten von Vertec mit dem `/noevents` Parameter ist nur für Debugging und Daten-Migrationen gedacht und kann im produktiven Betrieb zu inkonsistenten Daten führen. Deshalb ist dieser Parameter ab sofort nur noch bei Benutzern mit Administratorrechten wirksam.

Um zu überprüfen, ob der eingeloggte Benutzer Administrator ist, kann `/noevents` erst nach dem Login ausgewertet werden. Das bedeutet, dass Eventscripts vom Typ [Login](#) nach wie vor ausgeführt werden.

11.5 Probleme mit Excel-Report VBA Makro behoben

Linie: Standard, Expert | Modul: Leistung & CRM | Version 6.1.0.12

Folgende Probleme mit Excel Reports wurden behoben:

- COM Fehler aus Excel bei Abfragen von grösseren Datenmengen, auch wenn `ProcessMessages` aufgerufen wird.
- Vertec/Desktop ist nach Beenden des Reports blockiert.

Veränderungen

- Vertec ist während der Ausführung des Excel Reports nicht mehr blockiert. Das sollte keine Probleme verursachen, sofern der Excel Report nur das übergebene Objekt verwendet und nicht selbständig auf `vertec.argobject` oder `vertec.currentobject` zugreift.
- Die `vertec.ProcessMessages` Aufrufe sind nicht mehr notwendig, schaden aber auch nichts.
- Stürzt Excel während der Ausführung des Reports ab oder wird abgeschossen, kann folgende Fehlermeldung erscheinen:

Die angegebene Umwandlung ist ungültig.

11.6 Vertec.Updater self-autoupdate

Linie: Standard, Expert | Modul: Leistung & CRM | Version 6.1.0.14

Die Cloud App lädt den **Vertec.Updater** bei einem Update automatisch herunter. Das musste bisher manuell gemacht werden.

Der **Vertec.Updater** ist das Hilfsprogramm für das automatische Update der Cloud App, welches über die Einstellung `AutoUpdate` in der Konfigurationsdatei **Vertec.ini** gesteuert wird.

11.7 CloudServer Services per Konfiguration aktivieren

Linie: Standard, Expert | Modul: Leistung & CRM | Version 6.2

Die verschiedenen Services einer CloudServer-Instanz können im **Vertec.ini**-File separat gestartet werden. So kann beispielsweise auf einer CloudServer Instanz, welche ins Internet gestellt wird, ausschliesslich die Phone-App zur Verfügung gestellt werden, um die Angriffsrisiken zu minimieren.

In der Section **[CloudServer]** gibt es folgende Einstellungen:

```
[CloudServer]
XML Server=true
UISync Server=true
Phone App Server=true
```

Defaultwert ist für alle true. Das bedeutet: Um einen der Services auszuschalten, muss er auf false gesetzt werden.

11.8 Web App: Statische Files clientseitig cachen

Linie: Standard, Expert | Modul: Leistung & CRM | Version 6.1.0.12

Neu werden die statischen Files (Bilder, Stylesheets etc.) auf dem Client gecached. Das bedeutet, dass nicht mehr bei jedem Laden übertragen werden, sondern nur beim ersten Zugriff sowie bei einem Update des CloudServers.

Dies hat einen positiven Einfluss auf die Performance der Web App.

11.9 "Angemeldet bleiben" im Batch-Modus

Linie: Standard, Expert | Modul: Leistung & CRM | Version 6.2

Vertec.Desktop kann im Batch-Modus (ohne Oberfläche) aufgerufen werden. Bisher wurde dabei User und Login via Parameter angegeben. Das ist etwas unschön, denn die Credentials tauchen so in einem etwaigen scheduled Script auf.

Ab Version 6.2 erfolgt die Authentisierung auch im Batch-Modus über das "Angemeldet bleiben". Die Änderung ist dabei, dass die Desktop App ein neues AuthToken für das nächste Login generiert und dieses so nicht abläuft.

Das Vorgehen ist wie folgt: Der entsprechende User loggt sich auf dem Server einmalig in Vertec ein und setzt die Option **Angemeldet bleiben**. Wird Vertec danach im Batch-Modus gestartet, wird der User automatisch angemeldet.

Weitere Informationen Weitere Informationen zum Automatisieren von Vorgängen in Vertec finden Sie in der Online Knowledge Base unter www.vertec.com/kb/task.

11.10 Brute-Force Prevention Mechanismus

Linie: Standard, Expert | Modul: Leistung & CRM | Version 6.2

Um zahlreiche erfolglose Login-Versuche, wie sie bei Brute-Force Authentisierungs-Attacken verwendet werden, zu erschweren, wurde neu ein Brute-Force Prevention Mechanismus eingebaut.

Der Authentisierungs-Server merkt sich fehlgeschlagene Versuche pro Login-Namen und zählt sie. Hat die Anzahl fehlgeschlagener Versuche für einen bestimmten Login-Namen eine Schwelle erreicht, werden für eine bestimmte Zeitspanne keine Authentisierungs-Anfragen für diesen Login-Namen mehr verarbeitet.

Die Anzahl möglicher Versuche sowie die Zeitspanne können im Vertec.ini-File konfiguriert werden:

Login Max Failures=10 (Standard)
Login Ban Minutes=10 (Standard)

Ist eine der Einstellungen auf 0, so ist der Brute Force Prevention Mechanismus deaktiviert.

Falls vor Erreichen der Schwelle ein erfolgreiches Login stattfindet, wird der Zähler für diesen Login-Namen zurückgesetzt.