



September 2021

Vertec-Software Release 6.5

Inhaltsverzeichnis

1 Highlights der Version 6.5 und Einleitung	6
1.1 Rückwärtskompatibilität	6
1.2 Eingestellte Features in Vertec 6.5	7
2 Vertec Outlook App	8
2.1 Installation der Windows Edition	8
2.2 Installation der Web Edition	9
2.3 Eingehende E-Mails verwalten	11
2.4 E-Mails verfassen	15
2.5 Leistungen aufgrund von Kalendereinträgen erfassen	16
2.6 Die Aktivitätshistorie	16
2.7 E-Mails in Vertec verwalten	18
2.8 Technische Informationen	20
2.9 Security	21
3 Opportunitäten	22
3.1 Linktypen aktivieren	22
3.2 Opportunitäten erfassen	23
3.3 Opportunitätsstatus	25
3.4 Anlegen eines Projekts aus einer Opportunität	26
3.5 Aktivitäten	27
3.6 Berechtigungen	27
4 Fakturierung	29
4.1 Nur aktive Zahlungstypen in der ComboBox	29
4.2 Erfassen von neuen Kreditoren via QR-Code	29
4.3 Unterstützung von Rechnungsdatum in QR-Codes bei Debitoren	29
4.4 Rechnungsberichte mit fixierter Adresse von Rechnung	30
4.5 Schreibzugriff auf verr Attribute nicht mehr möglich	30
4.6 QR-Code Schweizerkreuz Dimensionen verbessert	31
4.7 QR-Zahlteil in Rechnungsreports Postfinance-konform	31
4.8 Zahlungstyp auf Rechnung festlegen	32
4.9 Persistente Attribute auf Rechnungen	32
5 Dokumente und Reports	33
5.1 Testdaten und Ausführen Button auf EOBs entfernt	33

5.2 Upload von Dokumenten auf einer Aktivität in der Web App	33
5.3 Speichern von PDF Belegen in der Phone App	33
5.4 Berichte ohne Filesystem	33
5.5 Rechnungsreports ohne Briefanrede	34
5.6 Umstellung Stunden/Minuten Darstellung in Excel EOB auf Stunden Dezimal	34
5.7 Beschrieb statt Code bei EOB Monatsübersicht mit Sollzeiten	34
5.8 Übergabe von Aktivität beim Speichern von Dokumenten in Addins	34
6 Controlling / BI	35
6.1 BI API	35
6.2 Speichern von BI Ansichten in Favoriten	38
6.3 Erweiterte BI URL	38
6.4 Berechtigungen für BI Kennzahlen	38
7 Customizing	40
7.1 Liste mit offenen und verrechneten Leistungen für Leistungserfassung	40
7.2 Unterstützung für fixierte Spalten in Listeneinstellungen	40
7.3 Icon Darstellung in Listen	40
7.4 Icon bei IX-Ordern	41
7.5 Icon Index für Subordner auf Ordern	42
7.6 Klassenspezifische Sortierung von Einträgen im Baum	42
7.7 Python Methode zum Erstellen einer E-Mail in Outlook	42
7.8 Bemerkungsfeld in Listeneinstellungen	43
7.9 Python Methode executereport liefert Bericht-Output als Byte-String	43
7.10 Anlegen von Systemeinstellungen mittels Config Set	43
7.11 vtcapp.executefile() unterstützt Commandline Parameter	44
7.12 Verbesserung der Anzeigebedingung für Container im BI Baum	44
7.13 Python Stub Files für builtin Modules	45
7.14 Checkbox für Stichwort-Ordner und einfache Ordner in Listen	45
7.15 LinkRollen in GetLinks sind via Eintrag Id identifizierbar	45
7.16 OCL Translate Operator	46
7.17 Zugriff auf Container und Parent-Objekt in OCL Expressions in Listen	46
7.18 Auswahldialoge aus Python aufrufbar	47
7.19 Python Executereport Methode mit Rückgabe von Aktivität	48
7.20 Python Funktion für Zahlungsabgleich	48
7.21 Script Editor öffnet sich nur mit Administratorrechten	48

7.22 Python Update	48
7.23 Aktivitätstypen sind Multilanguage-fähig und aktiv	48
7.24 Übersetzung von Konfigurationsdaten in der Businesslogik	49
7.25 Unterstützung von mehrfachen Klasseneinstellungen	49
7.26 Berechtigungen auf Abwesenheiten eingeschränkt	49
7.27 String-Darstellung von Projekteintrag-Typen enthält übersetzte Bezeichnung	50
7.28 String-Darstellung von Abwesenheit enthält übersetzte Bezeichnung	50
8 Schnittstellen	51
8.1 TAPI Extension funktioniert auch ohne CallerIdName	51
8.2 Übergabe von Aktivität in der SharePoint Online DMS Schnittstelle	51
8.3 Cloudfähige Abacus Web Debitoren Schnittstelle	52
8.4 Zeep auf Whitelist für Restrict Scripting	52
8.5 Abacus Web Kreditoren Schnittstelle ist cloudfähig	53
8.6 Unterstützung für ZUGFeRD 2.1	53
8.7 Bereinigung Adressenbehandlung bei cloudfähigen Schnittstellen	53
8.8 Einlesen von Zahlungsdaten für die cloudfähige DATEV Debitoren Schnittstelle	54
8.9 Topal Solutions Kreditoren Schnittstelle mit QR-Rechnung	54
8.10 Unterscheidung von ESR und QR Debitor Rechnungen	55
8.11 Unterstützung von mehr als 200 Sites in SharePoint Schnittstelle	55
8.12 Unterstützung von ESR Rechnungen in der Abacus Web Kreditoren Schnittstelle	55
8.13 SharePoint Online DMS Schnittstelle Unicode-fähig	55
8.14 Cloudfähige DATEV Kreditoren Schnittstelle	55
8.15 Setzen des EZ Verfahrens beim Buchen	55
8.16 Verbesserung Sage 200 Zahlungsabgleich	56
9 Performance	57
9.1 Performance-Verbesserung in Listen ohne aktiv Member	57
9.2 Performanceoptimierung beim Starten von Vertec	57
9.3 Shadow Attribute bei der Rechnung entfernt	57
9.4 Performanteres Laden der Budget Seite auf Projekt	57
10 Security	58
10.1 Restricted Session Prozess	58
11 Technik	59
11.1 Unicode Unterstützung	59
11.2 Neugestaltung des Login-Dialogs und Vertec Splash-Screens	61

11.3 Unterstützung von mehreren Cloud App Installationen	62
11.4 Zusätzliches Logging	62
11.5 Stoppen von hängenden TaskRunner Prozessen	62
11.6 Log Output an Loggly Dienst senden	62
11.7 Entfernung Fastviewer Client	63
11.8 Datenbank Konvert	63
11.9 Update von IBQuery	67
12 Verbesserungen Firebird Datenbank Unterstützung	68
12.1 Update auf Firebird 4.0	68
12.2 Update von Firebird FIB Komponenten	68
12.3 Firebird Dialect 3	68
12.4 Firebird Fehlermeldungen	70
13 Update von Vertec On-Premises Installationen	71
13.1 Neuinstallation von Vertec	71
13.2 Vor dem Update auf 6.5	71
13.3 Update von Vertec	71
13.4 Das erste Aufstarten nach dem Konvert	71

1 Highlights der Version 6.5 und Einleitung

Der Fokus des 6.5 Major Releases von Vertec liegt auf neuen CRM Funktionen: Der neuen Outlook App und einer neuen Klasse von Vertec Objekten, den Opportunitäten.

Die cloudfähige Outlook App ermöglicht eine komfortable und intuitive Ablage von Outlook-E-Mails als Vertec Aktivitäten – direkt aus Outlook heraus. Zudem sehen Sie in Outlook die vollständige in Vertec gepflegte Aktivitätshistorie. In Outlook können Sie auf die E-Mail-Adressen Ihrer Vertec-Kontakte zugreifen und sofort neue Kontakte in Vertec erfassen. Damit bietet die Vertec Outlook App eine integrierte Lösung für ein wertschöpfendes CRM.

Der Prozess vom ersten Kontakt mit einem (potentiellen) Kunden bis zum Projektabschluss lässt sich mit Opportunitäten leichtgewichtig darstellen. Hier erfassen Sie Verkaufswahrscheinlichkeiten und behalten den Überblick über den Status Ihrer laufenden Leads.

Auch im Zusammenhang mit CRM muss eine grössere Umstellung auf 6.5 gesehen werden, nämlich die Umstellung der Zeichensätze, die in Vertec verwaltet werden können. Aus historischen Gründen waren die in Vertec verwaltbaren Zeichen auf den westeuropäischen Zeichensatz (ANSI) eingeschränkt. Mit 6.5 öffnet sich Vertec auch für ost-europäische und internationale Sprachen (Unicode). Damit können z.B. auch russische Adressen korrekt in kyrillisch in Vertec verwaltet werden.

Daneben ist vor allem eine neue Web-Schnittstelle ins Vertec BI erwähnenswert: Durch diese können durch das Vertec BI berechnete Kennzahlen auch in Drittsystemen (z.B. Power BI, aber auch Microsoft Excel) weiterverarbeitet und ausgewertet werden.

Die Version 6.5 bietet wieder viele **Highlights**. Die Spannendsten hier in einem kurzen Überblick:

Artikel	Seite
2 Vertec Outlook App	8
3 Opportunitäten	22
6.1 BI API	35
7.3 Icon Darstellung in Listen	40
11.1 Unicode Unterstützung	59

1.1 Rückwärtskompatibilität

Bei den Features, bei welchen die Rückwärtskompatibilität eine Rolle spielt, wird das im entsprechenden Artikel im Abschnitt **Rückwärtskompatibilität** genau beschrieben:

Artikel	Seite
6.4 Berechtigungen für BI Kennzahlen	38
11.1 Unicode Unterstützung	59
12.3 Firebird Dialect 3	68

1.2 Eingestellte Features in Vertec 6.5

Systemeinstellung für Konfliktcheck entfernt

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.21

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Das **Plug-in Konfliktcheck** löst das bisherige Vorgehen der Konfliktsuche ab. Bei Neuinstallationen wird die Systemeinstellung **Angaben zu Mandatsverflechtungen anzeigen (Konfliktsuche)** nicht mehr angezeigt. Falls die alte Konfliktsuche verwendet wurde, kann diese in den Systemeinstellungen auf **False** gesetzt werden, um die dazugehörigen Menüpunkte und Standardfelder auszublenden.

Weitere Informationen Alle Informationen zum neuen Plug-in finden Sie im Artikel Plug-in: Konfliktcheck in der Online Knowledge Base unter www.vertec.com/kb/plugin-konfliktcheck/.

2 Vertec Outlook App

Linie: Standard, Expert | Module: Leistung & CRM | Apps: Outlook App | Version: 6.5.0.1

Die Vertec Outlook App stellt CRM Informationen in Outlook dar und bietet Unterstützung bei alltäglichen Aufgaben.

Sie ist ein **Vertec Cloud Client** und läuft somit ohne lokale Installation einer weiteren Vertec App direkt über den Vertec Cloud Server. Es handelt sich um eine **Specialized App** und bietet dementsprechend gewisse Funktionen, jedoch nicht den vollen Funktionsumfang einer full-featured App wie z.B. die Web App oder Cloud App.

Die Outlook App ermöglicht eine enge Integration von E-Mails und Kalendereinträgen aus Outlook in Vertec. Die Sprache richtet sich nach dem eingeloggten Benutzer. Zuvor wird die Sprache des Client-Rechners verwendet.

Die Outlook App deckt folgende Themen ab:

- Speichern von E-Mails in einer Aktivitätshistorie sowie direktes Anlegen der Aktivität in Vertec
- Anzeige von Informationen aus Vertec zu E-Mails in Outlook
- Direkte Navigation zu den relevanten Objekten in Vertec
- Auswahl von Kontakten aus Vertec beim Verfassen einer E-Mail
- Darstellung der Aktivitätshistorie und E-Mail Konversation
- Kalendereinträge zu Leistungen machen

Die Vertec Outlook App wird in **zwei Editionen** bereitgestellt: die **Windows Edition**, in der die Outlook App im lokalen Windows Client von Outlook läuft und die **Web Edition**, in der die Outlook App im Office 365 installiert wird. Im Folgenden werden beide Installationsmöglichkeiten beschrieben.

2.1 Installation der Windows Edition

Vorgehen

1. Öffnen Sie Ihr **Vertec App Portal**. Hier finden Sie neu die **Outlook App**.
2. Schliessen Sie Ihr Outlook, falls Sie dieses geöffnet haben.
3. Klicken Sie auf den Button **Installer Download** und öffnen Sie die heruntergeladene Datei.

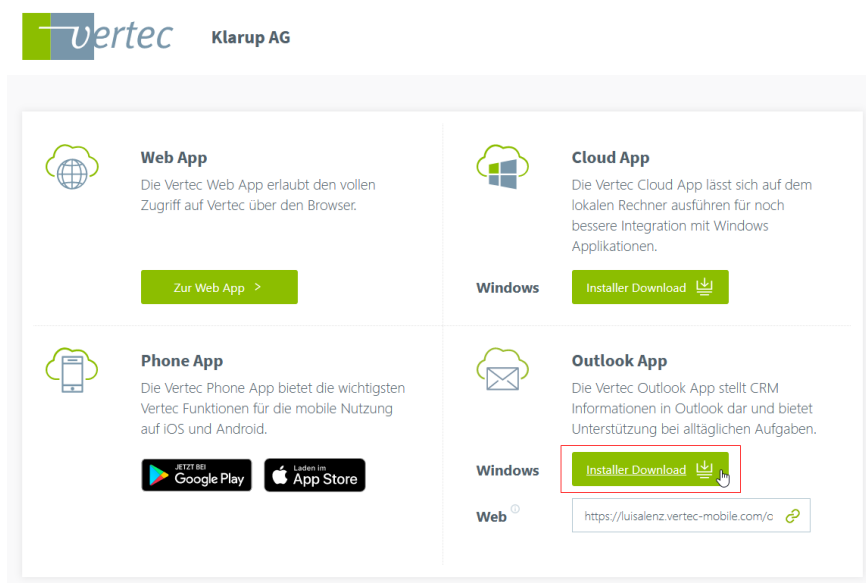


Abbildung 1: Installer Download im Vertec App Portal

Der Outlook App Installer wird anschliessend automatisch gestartet. Die Installation der Outlook App ist mit folgender Meldung abgeschlossen:

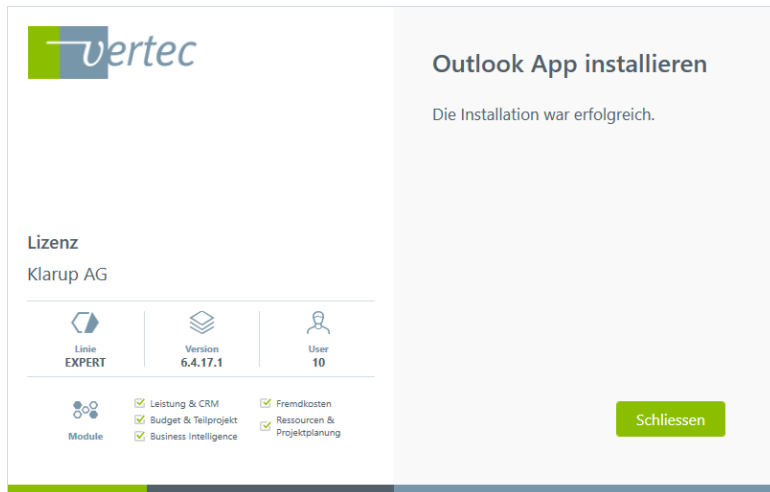


Abbildung 2: Erfolgreiche Installation

4. Wählen Sie **Schliessen**.
5. Öffnen Sie nun Ihren Outlook Client. Die Vertec Outlook App ist als Sidebar eingeblendet.
6. Loggen Sie sich mit Ihrem Vertec Login ein.

Die Anzeige der Outlook App kann in der Menüleiste via **Vertec** > **Anzeigen** gesteuert werden.

Ausserdem lässt sich in den Outlook App Einstellungen festlegen, ob sich Vertec in der Web App oder in der Cloud/Desktop App öffnen soll:

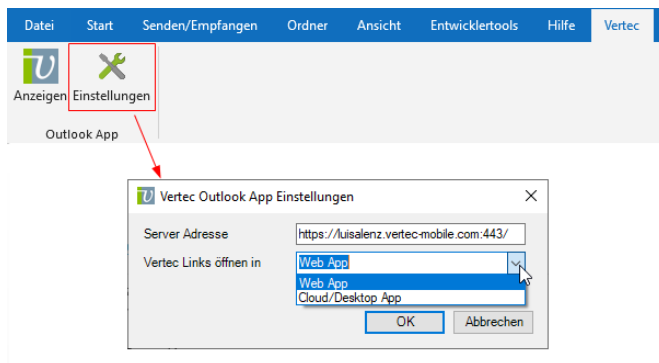


Abbildung 3: Outlook App Anzeigen und Einstellungen

Sobald die Einstellungen geändert werden, wird die Outlook App neu gestartet.

2.2 Installation der Web Edition

Vorgehen

1. Öffnen Sie Ihr Office 365, beziehungsweise Ihr Outlook im Webbrowser.
2. Öffnen Sie Ihr **Vertec App Portal**. Hier finden Sie neu die **Outlook App**.
3. Kopieren Sie den **Link** unter Outlook App manuell oder mit folgendem Icon:

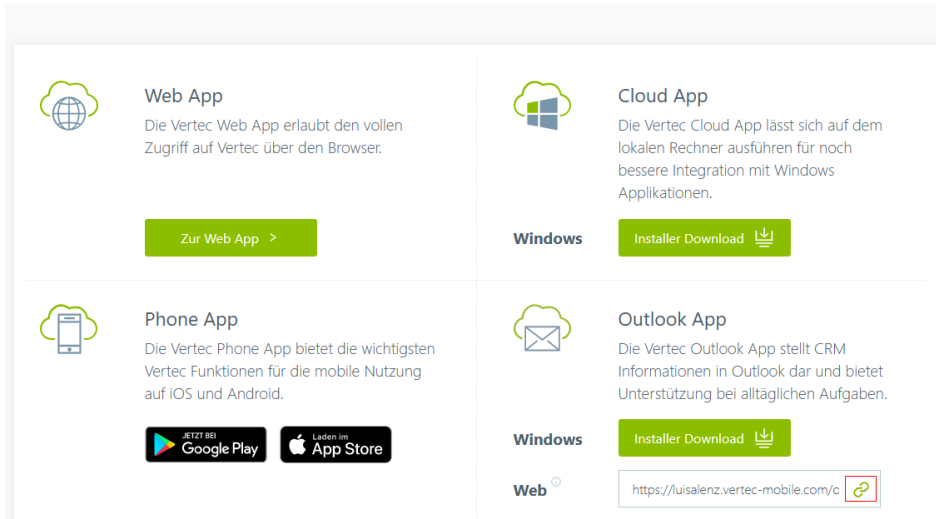


Abbildung 4: Das neue Vertec App Portal

- Öffnen Sie ein beliebiges E-Mail aus Ihrem Outlook.
- Klicken Sie auf **weitere Aktionen** und wählen Sie anschliessend **Add-Ins erhalten**:

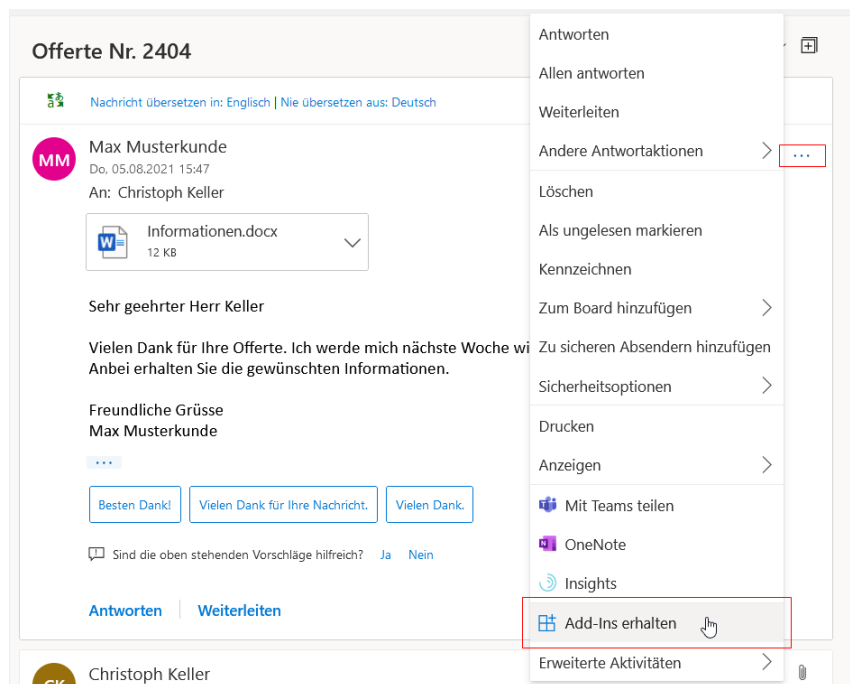


Abbildung 5: Add-Ins erhalten

- Klicken Sie unter **Meine Add-Ins** auf **+ Benutzerdefiniertes Add-In hinzufügen** und wählen Sie **Aus URL hinzufügen**.

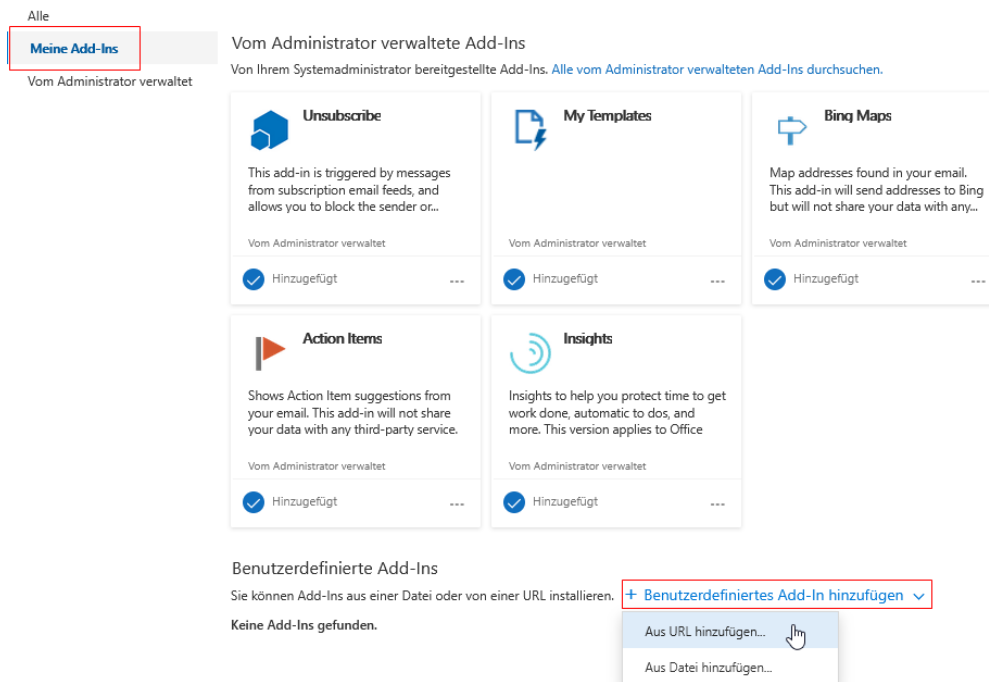


Abbildung 6: Add-In aus URL hinzufügen

7. Anschließend kann der zuvor aus dem Vertec App Portal kopierte Link eingefügt und mit **OK** bestätigt werden.
8. Wählen Sie **Installieren** im nächsten Schritt.

Damit haben Sie das Add-In erfolgreich hinzugefügt.

Third-Party Cookies erlauben

In der Web Edition läuft die Vertec Outlook App im Rahmen der Web Edition von Outlook. Sie ist damit eine Drittanwendung in der outlook.com Applikation. Die Outlook App nutzt eigene Cookies für die Authentisierung und das Session Management – diese Cookies sind aus Sicht des eingesetzten Browsers «Third-Party Cookies», weil sie nicht direkt an die Hauptseite geschickt werden, sondern an eine Drittseite.

Bei vielen Browsern gibt es Möglichkeiten, die Verwendung solcher Third-Party Cookies einzuschränken, weil die häufig zum «Tracking» (für Werbeanzeigen) verwendet werden. Blockiert ein Browser aber die entsprechenden Cookies der Outlook App, so kann diese nicht funktionieren – Third-Party Cookies müssen darum für die outlook.com Seite erlaubt sein.

Hinweis Die Vertec Phone App verwendet eine ähnliche Architektur wie die Outlook App. Einige Umstellungen im Zusammenhang mit der Web Edition der Outlook App und Third-Party Cookies führen auch zu Umstellung für die Phone App. Alte Phone App Versionen sind daher **nicht** mit der neuen Serverversion kompatibel. Neue Phone App Versionen sind jedoch rückwärtskompatibel mit alten Serverversionen.

2.3 Eingehende E-Mails verwalten

Nachdem die Outlook App erfolgreich installiert wurde, kann sie als Sidebar aufgerufen werden.

Wählen Sie hierfür ein beliebiges E-Mail aus Ihrem Outlook an. Über **Weitere Aktionen** kann die Outlook App aufgerufen werden:

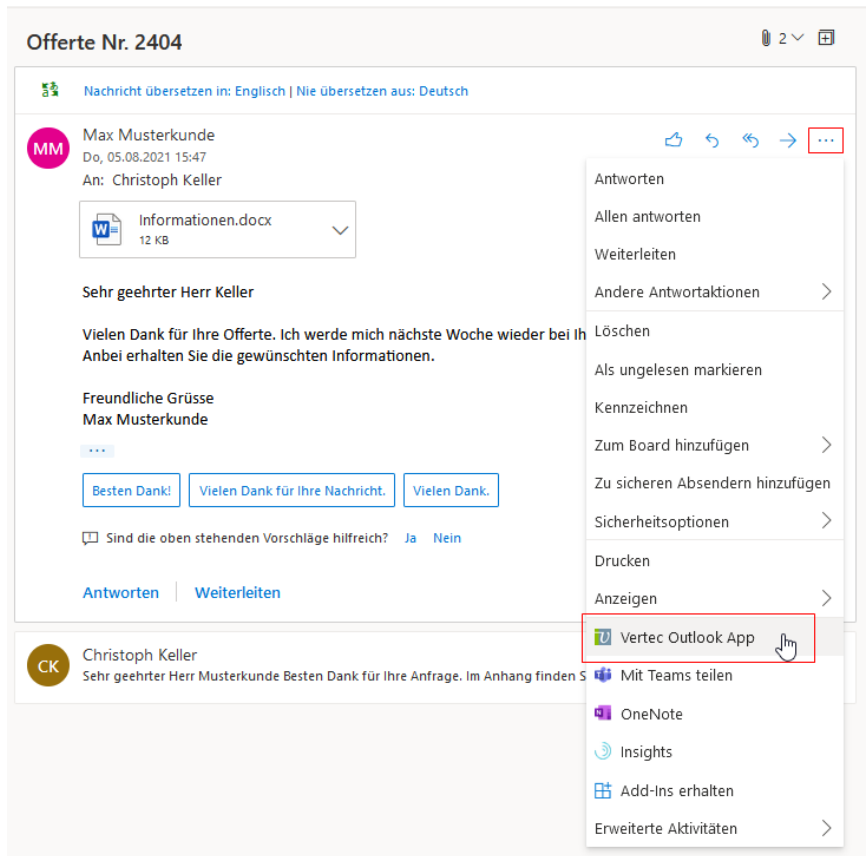


Abbildung 7: Outlook App aufrufen

Loggen Sie sich mit Ihrem Vertec Login ein.

Nach Belieben kann die Outlook App mit folgendem Button angepinnt, bzw. fixiert werden:

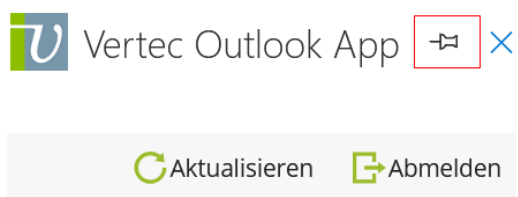


Abbildung 8: Outlook App anpinnen

Wenn Sie nun eine eingehende Mail anwählen, gibt es zwei Szenarien: der Kontakt, von dem die E-Mail gesendet wurde, ist bereits in Vertec angelegt oder noch unbekannt. Im Folgenden werden die beiden Möglichkeiten beschrieben:

Einen neuen Kontakt anlegen

Die von Ihnen angewählte E-Mail kommt von einem Kontakt, welcher noch nicht in Vertec erfasst ist. Sie haben nun die Möglichkeit, über den Button + **Neue Adresse** die Person, bzw. den Kontakt, zu speichern, bzw. anzulegen.

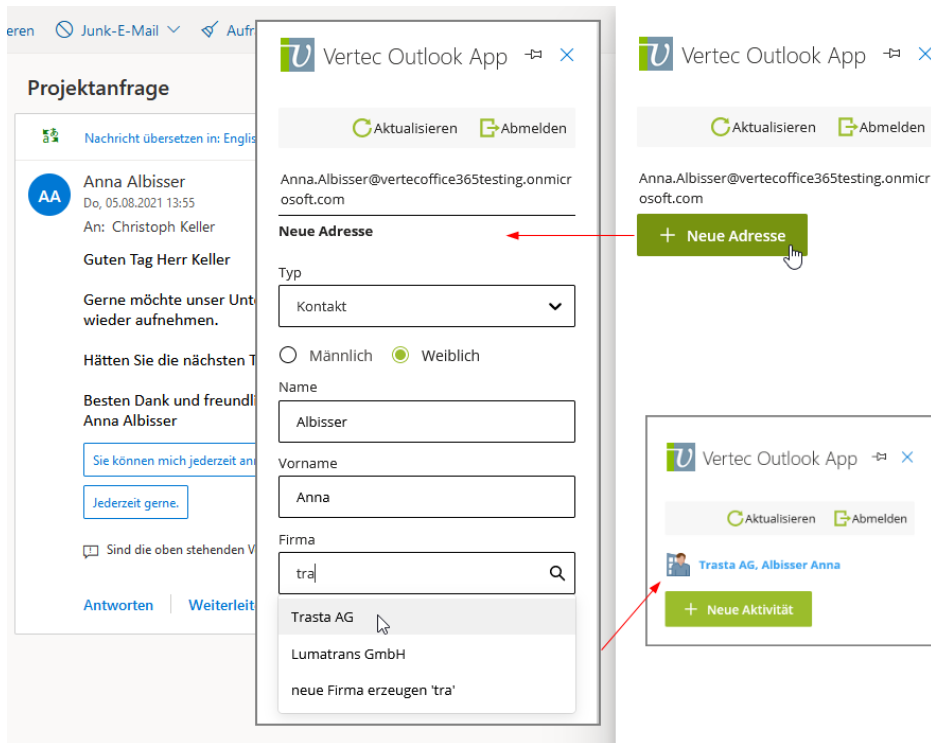


Abbildung 9: Neuer Kontakt hinzufügen

Unter **Typ** kann zwischen **Kontakt** oder **Person** gewählt werden. Beim Auswählen von **Kontakt** erscheint zusätzlich das Feld **Firma**, ansonsten nur die Felder **Name** und **Vorname** sowie die Auswahl des Geschlechts.

Wenn kein Adresseintrag zu der ausgewählten E-Mail-Adresse gefunden wurde, kann ein neuer Eintrag erstellt werden. Der Name kann manuell eingegeben und mit **Enter** bestätigt werden. Daraufhin erscheint im Feld **Firma** die Meldung «Es wird eine neue Firma mit dem Namen 'XYZ' erstellt.» und die Firma wird automatisch in Vertec angelegt.

Mit dem Button **Sichern** wird der Kontakt direkt in Ihrem Vertec angelegt. In der Outlook App wird er durch das Icon und einen Link mit Firma und Name dargestellt, welcher beim Anklicken direkt den Kontakt im Vertec öffnet. Zudem wechselt der Button **+ Neue Adresse** zu **+ Neue Aktivität**.

Eine neue Aktivität anlegen

Der Kontakt, von welchem die E-Mail stammt, ist bereits bekannt und in Ihrem Vertec gespeichert. In diesem Fall erscheint in der Outlook App zuoberst der Kontakt mit der dazugehörigen Firma und anschliessend die bisherige **Aktivitätshistorie**. Darin sind auch die Aktivitäten anderer Mitarbeitenden mit derselben Kontaktperson aufgelistet, wie in diesem Fall das Telefonat zwischen Herrn Musterkunde und Lars Sorenson.

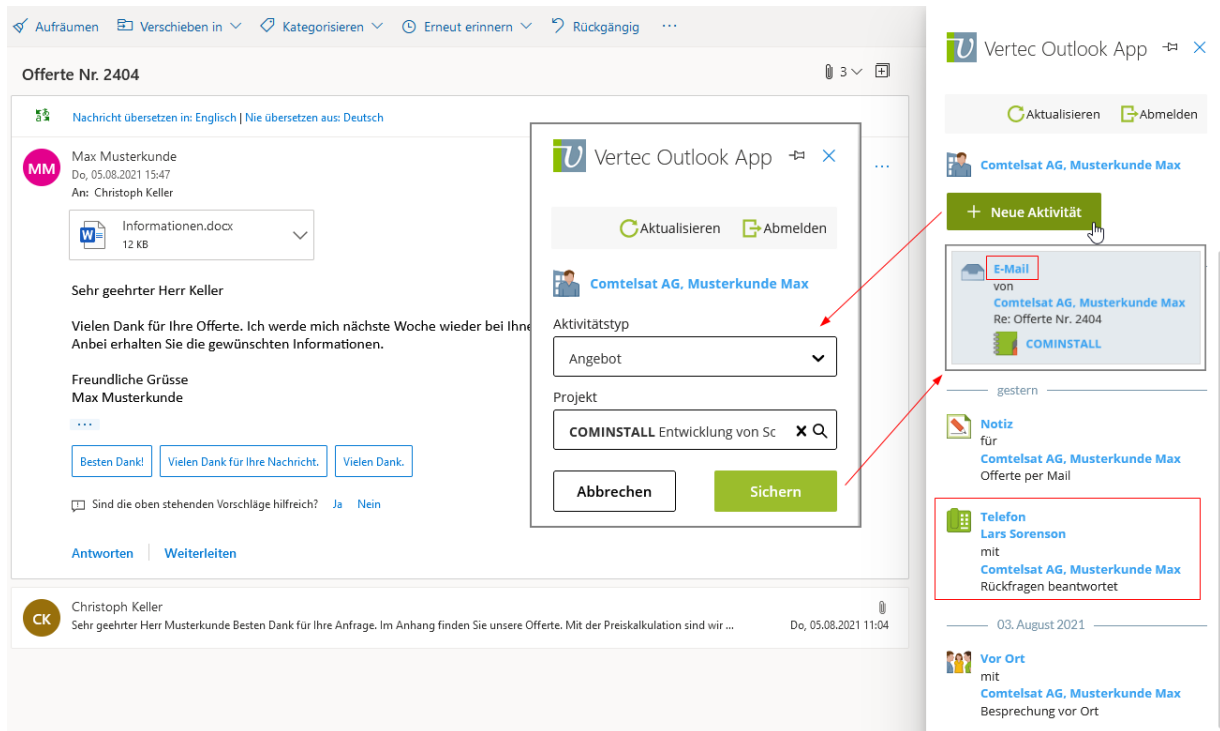


Abbildung 10: Neue Aktivität hinzufügen

Mit dem Button **+ Neue Aktivität** kann ein Aktivitätstyp sowie das dazugehörige Projekt ausgewählt werden. Dadurch wird das E-Mail in der Aktivitätshistorie der Outlook App gespeichert und auch direkt in Ihrem Vertec als Aktivität angelegt. Auf der Aktivität werden auch die Attachments gespeichert, die bei Bedarf heruntergeladen werden können.

Mehrere Empfänger

Wenn Sie ein E-Mail an mehrere Empfänger schicken oder ein E-Mail mit mehreren Empfängern erhalten, kann über folgende Pfeiltasten in der Outlook App durch die verschiedenen Empfänger gescrollt werden:

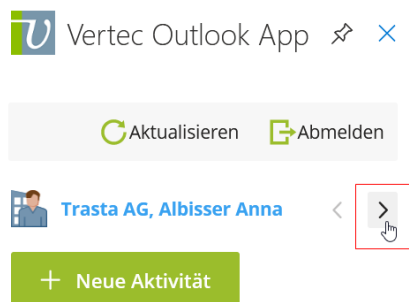


Abbildung 11: Pfeiltasten bei mehreren Empfängern

Sollte ein Empfänger noch nicht in Vertec angelegt sein, erscheint das weiter oben beschriebene Feld **+ Neue Adresse**, ansonsten die jeweilige Aktivitätshistorie mit dem Button **+ Neue Aktivität**.

Nicht unterstützte Objekte

Wird ein Objekt in Outlook angewählt, welches von Vertec nicht verarbeitet werden kann (z.B. eine Abwesenheitsnachricht), erscheint in der Outlook App die Meldung **Dieses Element in Outlook kann mit der Vertec Outlook App nicht verarbeitet werden.**

2.4 E-Mails verfassen

Auch beim Verfassen einer neuen E-Mail kann die Outlook App aufgerufen werden:

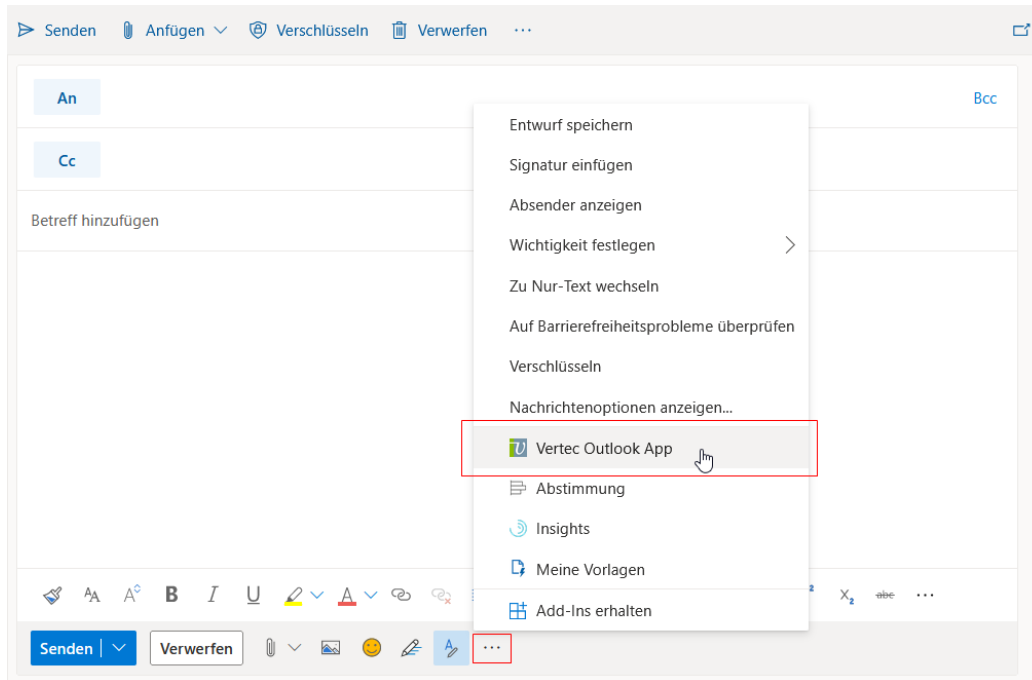


Abbildung 12: Outlook App aufrufen

Anschließend kann sie dazu genutzt werden, einen Kontakt in Vertec zu suchen:

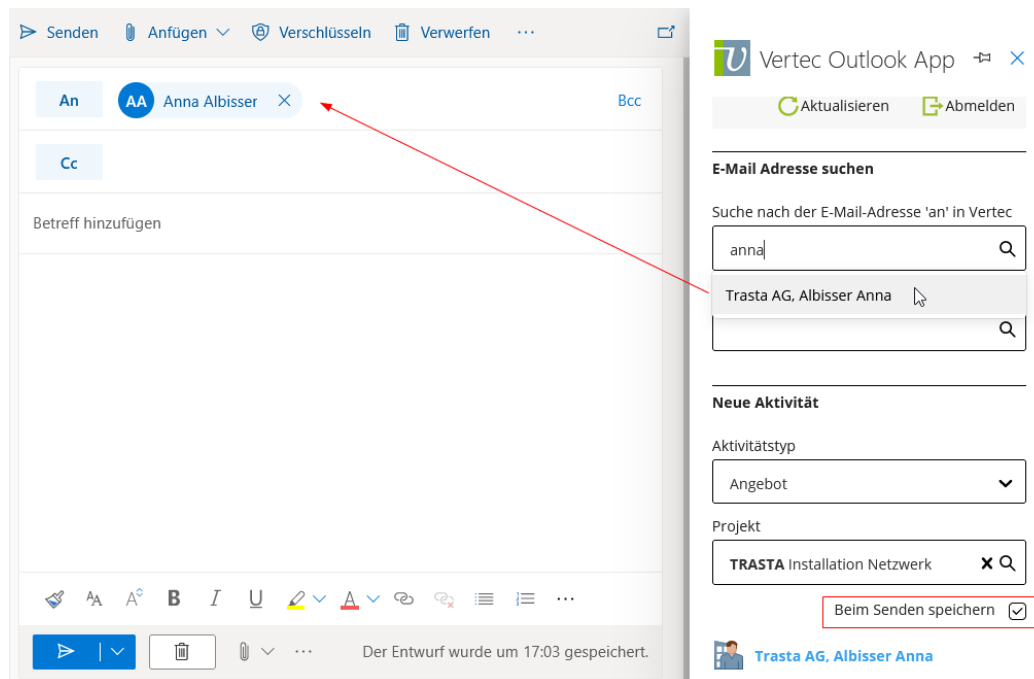


Abbildung 13: Kontakt über die Vertec-Suche

Im Suchfeld kann sowohl nach einem Namen als auch nach einem Projekt gesucht werden. Wurde der gewünschte Kontakt gefunden, kann er durch Klicken oder mithilfe der Enter-Taste ausgewählt werden. Die Ad-

resse wird dadurch direkt in das **An** bzw. **Cc** Feld übernommen und die Aktivitätshistorie des ausgewählten Kontaktes eingeblendet.

Auch der jeweilige **Aktivitätstyp** sowie das **Projekt** können hier ausgewählt werden. Es stehen alle Aktivitätstypen zur Auswahl, die in Ihrem Vertec gespeichert sind.

Durch die Checkbox **Beim Senden speichern** wird die E-Mail automatisch in Vertec angelegt. Dabei wird die Aktivität in Vertec mit der ausgewählten Adresse, dem Projekt und dem Aktivitätstyp verknüpft. Die Checkbox ist standardmässig **aktiv**, kann jedoch manuell deaktiviert werden.

Hinweis Wird die Outlook App beim Verfassen einer neuen E-Mail nicht eingeblendet, wird auch keine Aktivität in Vertec erzeugt.

2.5 Leistungen aufgrund von Kalendereinträgen erfassen

Neben E-Mails können auch **Kalendereinträge**, die in Outlook eingetragen werden, direkt mithilfe der Outlook App in Vertec als **Leistung** angelegt werden.

Beim Erstellen eines Kalendereintrags in Outlook kann über **weitere Optionen** die Outlook App aufgerufen werden. Durch den Button **+ Neue Leistung** lässt sich ein **Projekt**, eine **Phase** sowie eine **Tätigkeit** zuordnen. Es werden alle Tätigkeiten angezeigt, die in Ihrem Vertec hinterlegt sind. Mit dem Button **Sichern** wird die Leistung in der Sidebar angezeigt sowie in Vertec angelegt. Die erfasste Leistung ist als Link dargestellt und öffnet die Leistung in Ihrem Vertec.

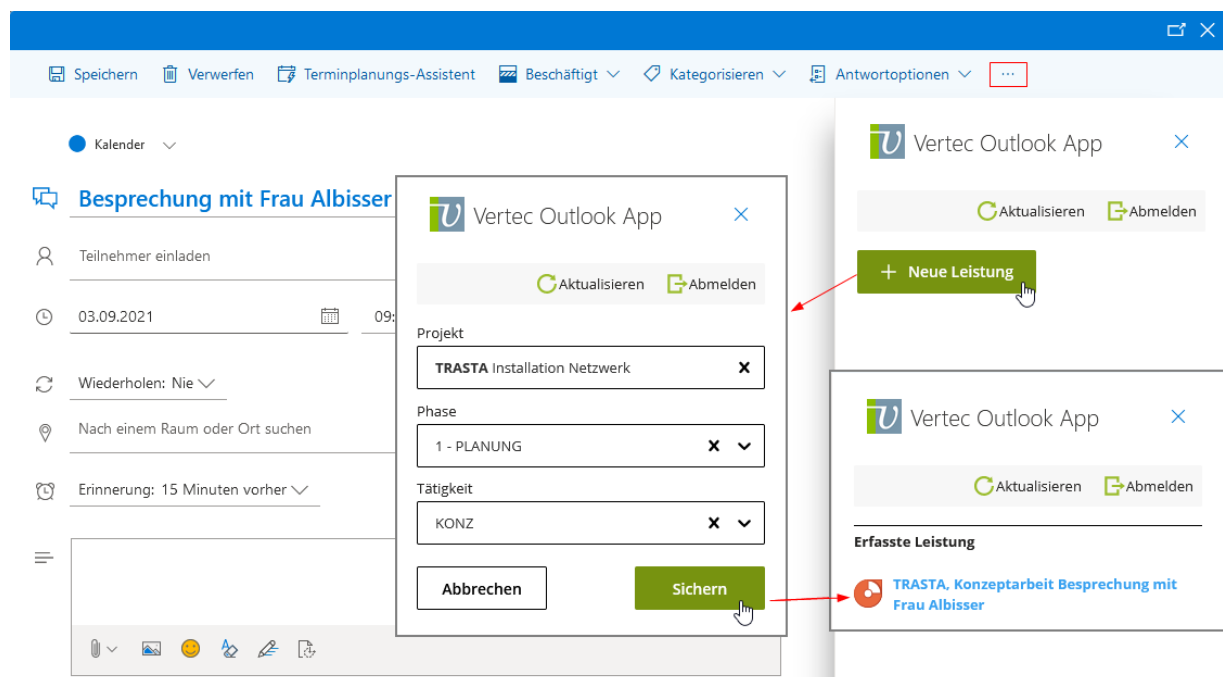


Abbildung 14: Neue Leistung hinzufügen

2.6 Die Aktivitätshistorie

In der Aktivitätshistorie werden jeweils 20 Aktivitäten geladen. Gelangt der Scrollbalken am unteren Ende der geladenen Liste an, werden weitere 20 Aktivitäten geladen. Die Darstellung der Aktivitäten erfolgt in zeitlich absteigender Folge und ist nach Tagen gruppiert. Die einzelnen Einträge können markiert werden, wobei die ausgewählte Aktivität durch einen grauen Hintergrund hervorgehoben wird. Auch alle Aktivitäten, die zur gleichen Konversation gehören, werden hervorgehoben dargestellt.

Mithilfe der Aufwärts-/Abwärts-Pfeiltasten kann zwischen den Aktivitäten navigiert werden. Die Tabnavigation ist blockiert. Falls die aktuell selektierte E-Mail nicht unter den ersten 20 ist, werden alle Aktivitäten bis zur selektierten E-Mail geladen.

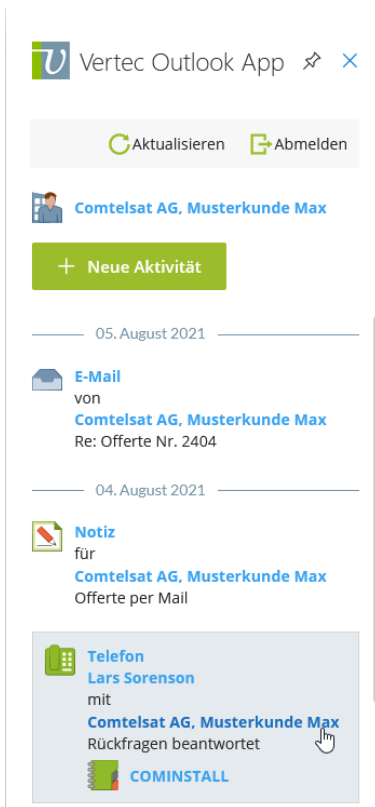


Abbildung 15: Aktivitätshistorie

Die Darstellung der Firma, des Kontaktes oder der Aktivität sind mit Links versehen. Beim Anklicken öffnet sich das jeweilige Objekt direkt in Vertec. Bei einer selektierten Aktivität ist auch das dazugehörige Projekt mit einem Link versehen.

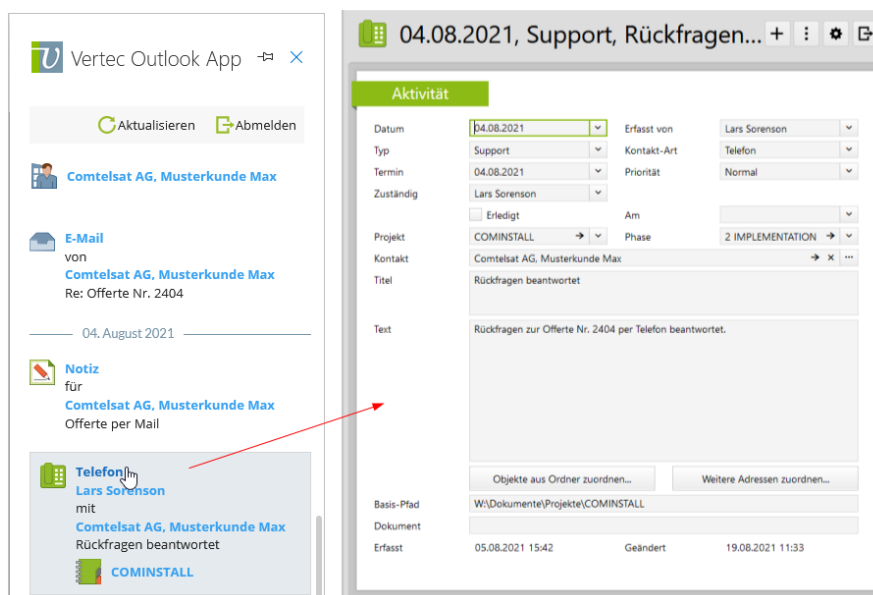
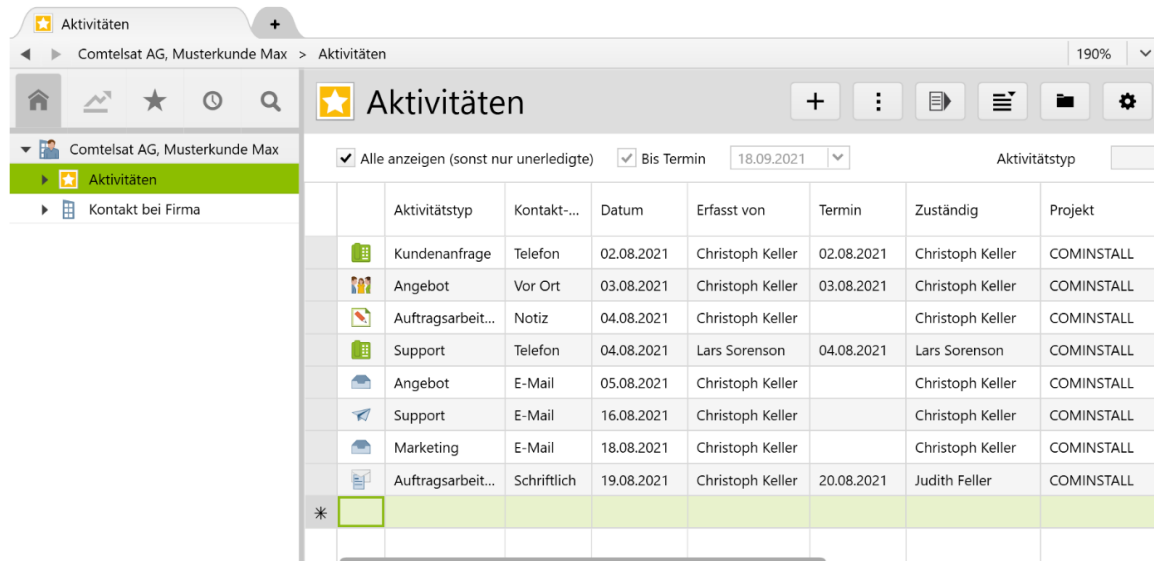


Abbildung 16: Link von der Outlook App ins Vertec

In Vertec wird die Aktivitätshistorie in einer Liste mit den dazugehörigen Icons dargestellt:












	Aktivitätstyp	Kontakt...	Datum	Erfasst von	Termin	Zuständig	Projekt
	Kundenanfrage	Telefon	02.08.2021	Christoph Keller	02.08.2021	Christoph Keller	COMINSTALL
	Angebot	Vor Ort	03.08.2021	Christoph Keller	03.08.2021	Christoph Keller	COMINSTALL
	Auftragsarbeit...	Notiz	04.08.2021	Christoph Keller		Christoph Keller	COMINSTALL
	Support	Telefon	04.08.2021	Lars Sorenson	04.08.2021	Lars Sorenson	COMINSTALL
	Angebot	E-Mail	05.08.2021	Christoph Keller		Christoph Keller	COMINSTALL
	Support	E-Mail	16.08.2021	Christoph Keller		Christoph Keller	COMINSTALL
	Marketing	E-Mail	18.08.2021	Christoph Keller		Christoph Keller	COMINSTALL
	Auftragsarbeit...	Schriftlich	19.08.2021	Christoph Keller	20.08.2021	Judith Feller	COMINSTALL
							

Abbildung 17: Liste der Aktivitäten mit den dazugehörigen Icons



Telefon



E-Mail eingehend



E-Mail ausgehend



Schriftlich



Notiz



Meeting im Haus oder vor Ort

Die entsprechenden Icons werden sowohl im Baum als auch in der Detailansicht einer Aktivität angezeigt. Ein Customizing des Icons via Klasseneinstellung für Aktivität übersteuert die eingebaute Icon Logik.

Neben den Icons gibt es ebenfalls die neue Spalte **Datum und Uhrzeit**, nach welcher die Liste standardmässig sortiert wird. Die Aktivitäten erhalten das Datum und die Uhrzeit, an welchem bzw. zu welcher die jeweiligen E-Mails erstellt oder erhalten wurden.

2.7 E-Mails in Vertec verwalten

Durch Klick auf den Link **E-Mail** in der Aktivitätshistorie ihres Outlook Apps öffnet sich die Aktivität direkt in Vertec.

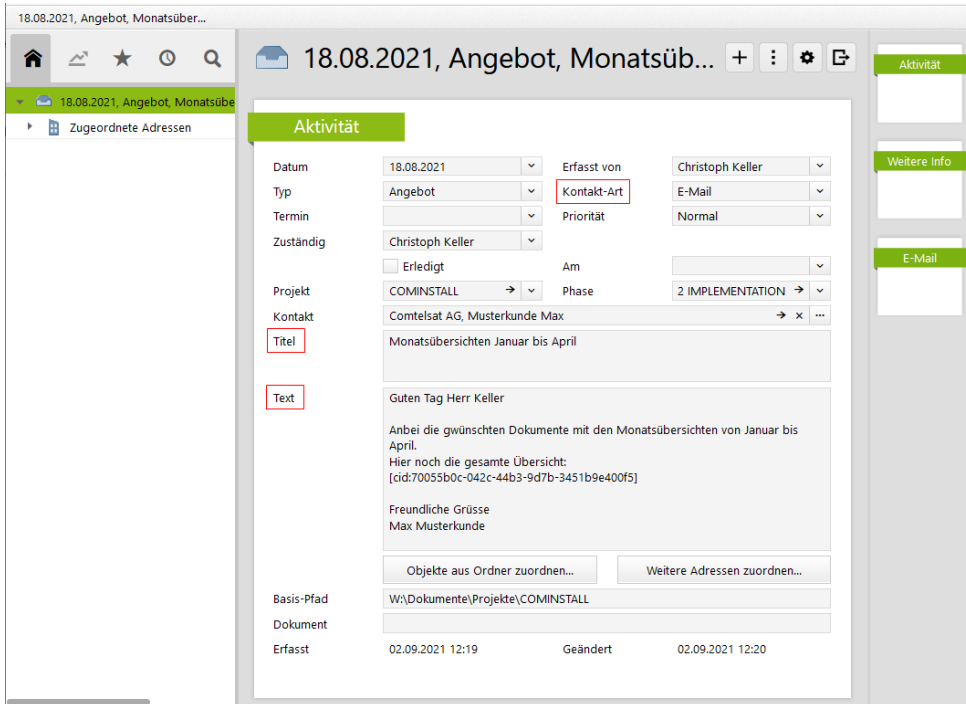


Abbildung 18: Kontakt-Art E-Mail

Viele Felder werden automatisch ausgefüllt, der Rest kann manuell bearbeitet werden. Im Feld **Titel** steht der Betreff der E-Mail und im Feld **Text** wird der Inhalt der E-Mail als Klartext dargestellt.

Bei Kontakt-Art **E-Mail** wurde auf der Seite **Weitere Info** der Abschnitt E-Mail entfernt und stattdessen eine neue Seite **E-Mail** angelegt. Hier werden die Felder **Betreff**, **Von** und **An** automatisch ausgefüllt sowie die Anhänge und der Inhalt der E-Mail möglichst originalgetreu dargestellt.

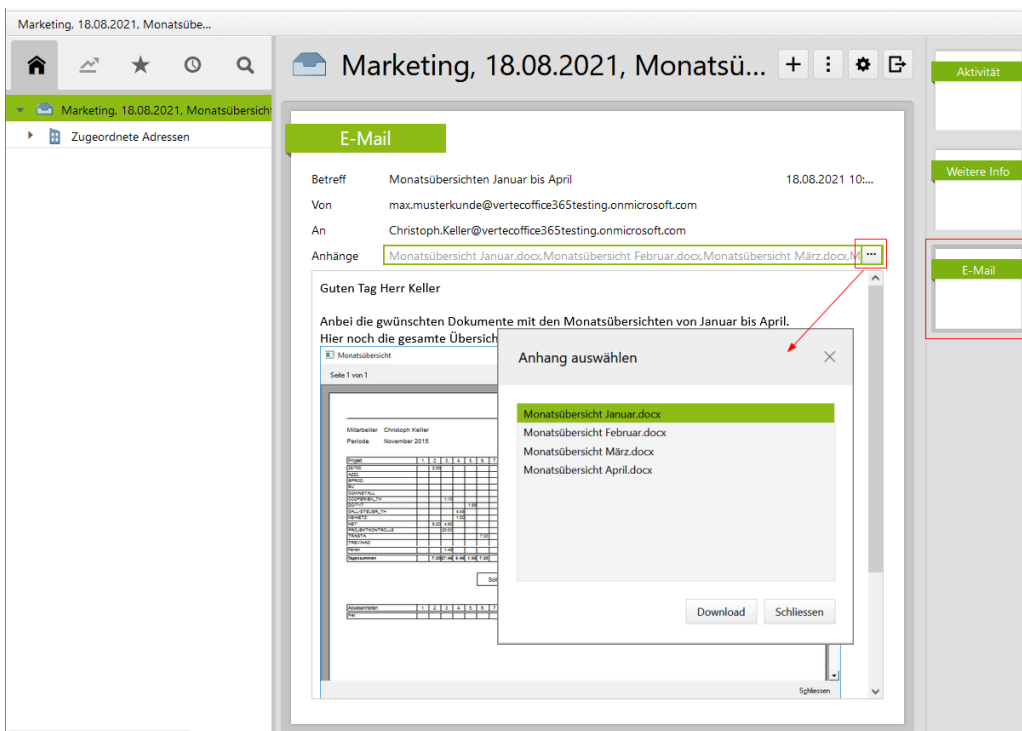


Abbildung 19: Neue Seite E-Mail auf Aktivität

Die Anhänge können bei Bedarf direkt über die drei Punkte heruntergeladen werden. Dabei öffnet sich der Dialog **Anhang auswählen** mit einer Liste aller Anhänge. Mit Doppelklick auf einen Anhang kann dieser in der Web App entweder **geöffnet** oder im Browser **heruntergeladen**, bzw. im Download Ordner gespeichert werden. In der Desktop und Cloud App kann der Anhang **geöffnet** oder **gespeichert** werden. Als Default Speicherpfad wird **C:\Users\name\AppData\Local\Temp** verwendet.

Von Vertec ins Outlook

Umgekehrt kann über den Button **Aktionen > E-Mail öffnen** die E-Mail direkt in ihrem Outlook geöffnet werden. In der Web App wird dabei eine **.eml** Datei heruntergeladen, die anschliessend ebenfalls in Outlook geöffnet werden kann.

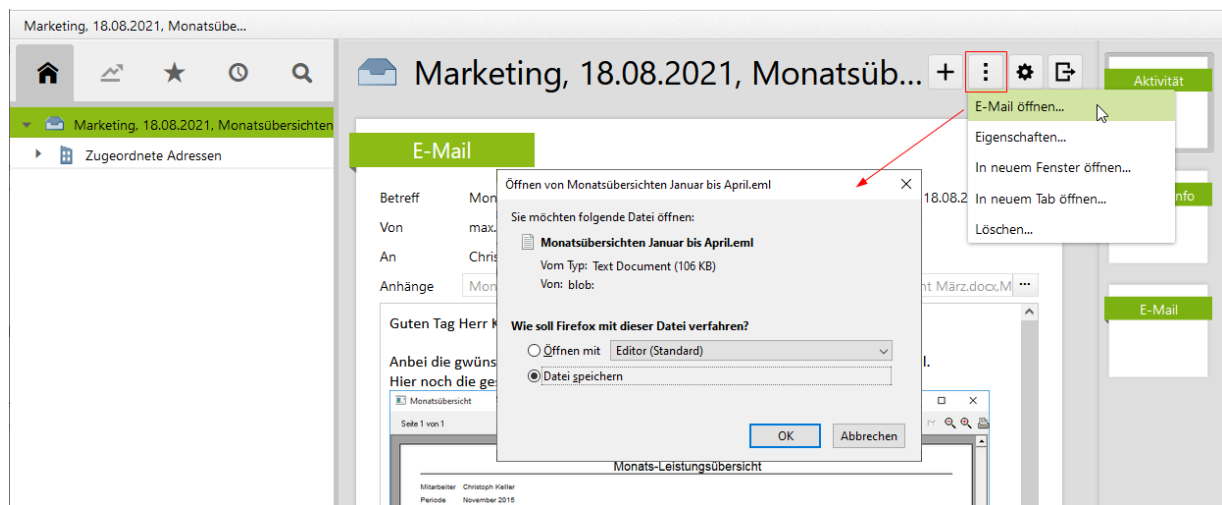


Abbildung 20: E-Mail in Outlook öffnen

Bei Bedarf kann ein **E-Mail Button** mit folgender Python Methode angelegt werden:

```
# e-mail content als File mit Endung .eml an den Client senden
vtcapp.sendfile(argobject.content, "email.eml")
```

2.8 Technische Informationen

Authentisierung

Obwohl es in der Outlook App keine Checkbox für **Angemeldet bleiben** gibt, nutzt die Outlook App dieses Feature (wie auch die Phone App). Nach einer erfolgreichen Authentisierung bleibt der Nutzer angemeldet.

CSRF Token für Web API

Die Outlook App verwendet ein serverseitiges Web API, welches Veränderungen an der Vertec Datenbank via POST Requests zulässt, z.B. beim Erfassen von Aktivitäten oder Leistungen. Da gleichzeitig die Authentisierung des Clients via ein Session Cookie sichergestellt wird, besteht die Möglichkeit von cross site request forgery (CSRF) Angriffen. Um dies zu verhindern, wird neben dem Session Cookie bei jedem Login ein **CSRF Token** generiert.

Session Type Identifizierung

Der Operator **AppType** (`TimSession.allInstances->first.appType`) liefert den Client Type **Outlook** zurück.

Automatisches Update

Die Outlook App Windows Edition prüft jeweils beim Start, ob der Cloud Server eine neue Version bereithält. Sollte das der Fall sein, wird zur Installation des Updates aufgefordert, wofür Outlook beendet werden muss.

Properties

- Neues Property `OutlookAppSaveOnSend` für die Funktion **Beim Senden speichern**, die das automatische Anlegen der E-Mails in Vertec ermöglicht.
- Die Properties `AktivitaetstypEmail`, `EmailOrdner` und `HandleSendEmail` enthalten am Ende die Bezeichnung (**Outlook Addin**), um das alte COM Outlook Addin von der Outlook App zu differenzieren.

Neue Felder auf Aktivitäten für die E-Mail Unterstützung

Die neue Outlook App basiert darauf, dass pro E-Mail Nachricht eine Aktivität in Vertec angelegt wird. Es ist neu möglich, Aktivitäten in einer Liste nach Datum und Zeitpunkt sortiert darzustellen, um E-Mails vom gleichen Tag in korrekter Reihenfolge anzuzeigen.

Dafür gibt es zusätzliche **Member** auf der Klasse Aktivität:

- `DateAndTime` vom Typ **DateTime**
- `Content` vom Type **BLOB**
- `MessageId` vom Typ **string**
- `ConversationId` vom Typ **string**

Neues Member

Um eine Leistung anhand eines Outlook Termins erfassen zu können, wurde das neue Member `Leistung.AppointmentId` eingeführt.

Outlook App deaktivieren

Im Vertec.ini-File kann die Outlook App deaktiviert werden, in dem in der [Cloudserver] Section `Outlook App Server` auf `False` gesetzt wird. Das sieht dann so aus:

 Vertec Outlook App ↗ ✕



Vertec Outlook App

Outlook App ist deaktiviert

Abbildung 21: Deaktivierte Outlook App

2.9 Security

Die E-Mails aus Outlook werden absolut unverändert nach Vertec übernommen, inkl. der Attachements. Enthalten diese, z.B. in einem Word-Attachment, Malware, wird diese unverändert nach Vertec übernommen und kann durch andere User heruntergeladen werden – ähnlich wie es bei einem Abspeichern auf einem Filesystem oder dem Weiterleiten einer E-Mail wäre. Vertec macht keinerlei Versuche, solche Malware zu entdecken.

3 Opportunitäten

Linie: Expert | Module: Leistung & CRM | Apps: Full-featured | Version: 6.5.0.1

Mit Opportunitäten gibt es eine unkomplizierte und leichtgewichtige Möglichkeit, den Verkaufsprozess in Vertec abzubilden.

Zu jedem Adresseintrag in Vertec können Opportunitäten erfasst werden. Die Opportunität stellt eine Verkaufschance dar und ermöglicht jederzeit einen Überblick über den aktuellen Status des geplanten Verkaufes, inkl. Wahrscheinlichkeit, Betrag und dem Datum des geplanten Abschlusses.

Die Erfassung von Opportunitäten ist auch ohne Anlage eines Projektes möglich und schliesst damit die Lücke zwischen der ersten Kontaktaufnahme mit einem potentiellen Kunden und der Erfassung eines Projektes.

3.1 Linktypen aktivieren

Um Opportunitäten zu verwenden, müssen zunächst die entsprechenden Linktypen aktiviert werden. Dies kann manuell oder durch die Verwendung des **Plug-ins Opportunitäten**, das auf der Knowledge Base zum [Download](#) zur Verfügung steht, durchgeführt werden. Mit dem Plug-in werden die Linktypen aktiviert und Ordner für die Auswertung von Opportunitäten erstellt.

Um die Linktypen manuell zu aktivieren, müssen im Ordner **Linktypen** die folgenden vier Einträge herausgesucht werden:

- Aktivitäten - Opportunität
- Kunde - Opportunitäten
- Projekt - Opportunitäten
- Verkäufer – Opportunitäten

Auf jedem dieser Einträge muss das Häkchen in der **Aktiv** Checkbox gesetzt werden.

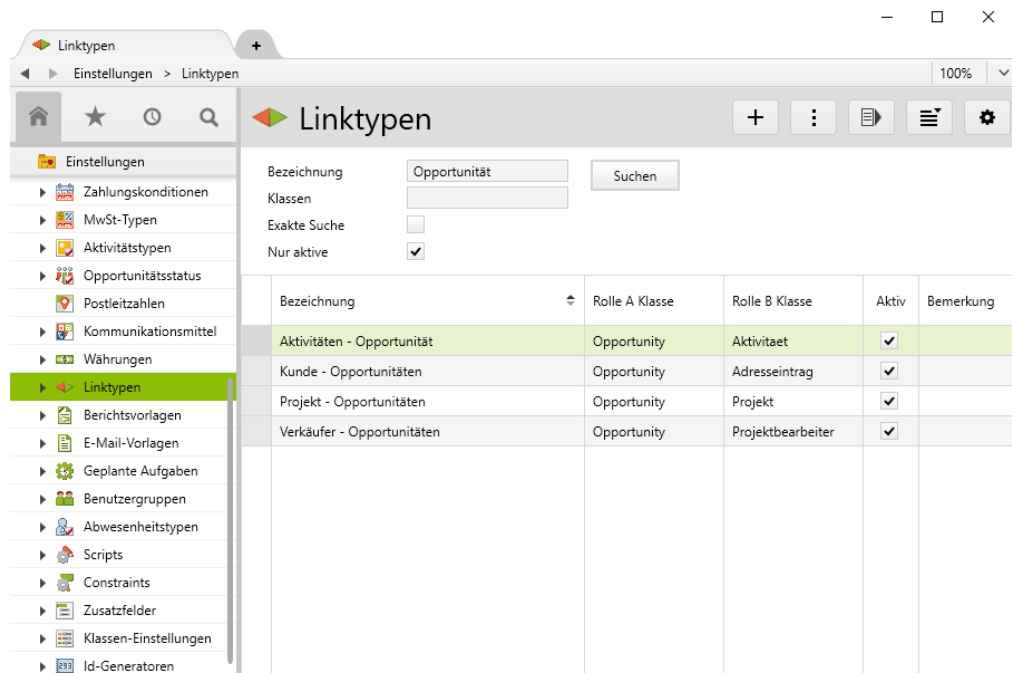


Abbildung 22: Linktypen zu Opportunitäten

3.2 Opportunitäten erfassen

Sind die Linktypen aktiviert, so können Opportunitäten erfasst werden. Öffnen Sie dazu zunächst den Adresseintrag, zu dem Sie eine Verkaufschance hinterlegen möchten. Im Normalfall ist dies eine Firma. Im Neu-Menü finden Sie den Eintrag **Opportunität**.

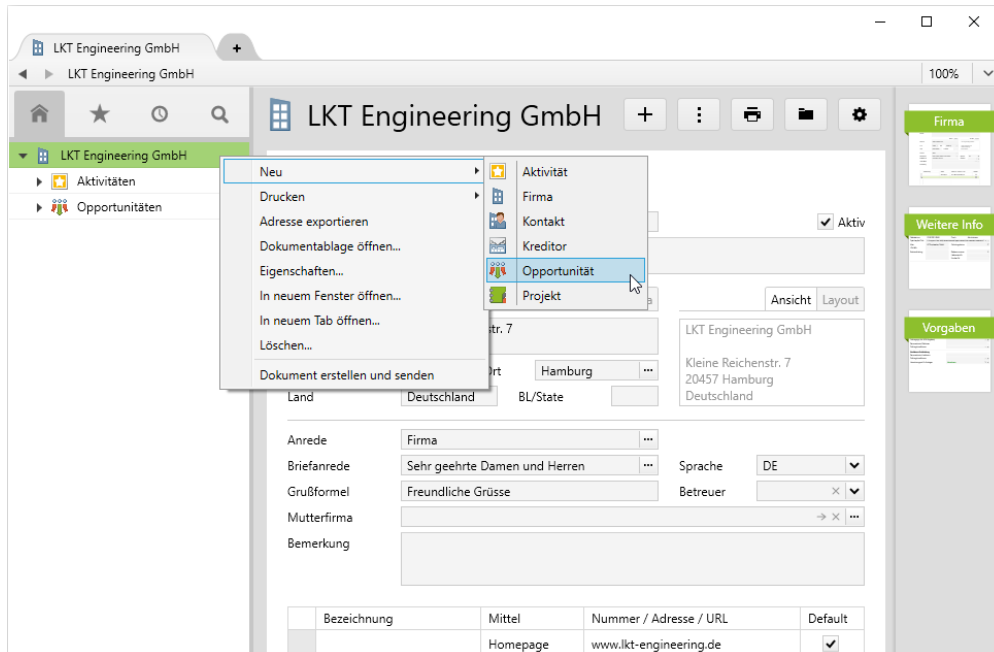


Abbildung 23: Neue Opportunität anlegen aus Firma

Ein Klick auf den Eintrag **Opportunität** öffnet ein neues Fenster mit der Detailansicht der neuen Opportunität:

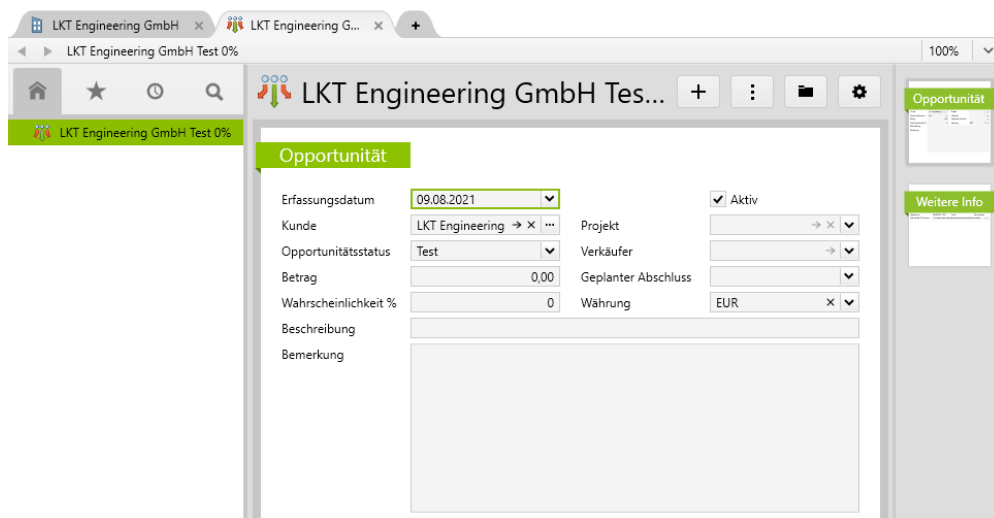


Abbildung 24: Detailansicht neue Opportunität

Nun können folgende Felder ausgefüllt werden:

Erfassungsdatum	Bereits vorausgefüllt: das aktuelle Tagesdatum.
Aktiv	Gibt an, ob die Opportunität noch aktiv ist. Gewonnene oder verlorene Opportunitäten werden automatisch auf inaktiv gesetzt. Mehr dazu im Abschnitt 3.3 Opportunitätsstatus.

Kunde	Bereits vorausgefüllt: der ausgewählte Adresseintrag.
Projekt	Wird automatisch gefüllt, wenn zu der Opportunität ein Projekt angelegt wurde. Wenn Sie eine Opportunität zu einem bestehenden Projekt hinzufügen möchten geht das ebenfalls. Die Auswahl ist eingeschränkt auf die Projekte des Kunden.
Opportunitätsstatus	Der Status der Opportunität. Mehr dazu im Abschnitt 3.3 Opportunitätsstatus.
Verkäufer	Bereits vorausgefüllt: der aktuell angemeldete User. Hier wird eingetragen, wer für den Verkauf zuständig ist. Insbesondere kann so ausgewertet werden, welcher Verkäufer welches Auftragsvolumen generiert hat. Zur Auswahl stehen alle aktiven Projektbearbeiter in Vertec.
Betrag	Das geschätzte Volumen des Auftrages.
Geplanter Abschluss	Das Datum des geplanten Projektabschlusses. Der Eintrag in diesem Feld ist insbesondere wichtig, wenn zeitliche Auswertungen erstellt werden sollen.
Wahrscheinlichkeit %	Eintrittswahrscheinlichkeit eines Projektabschlusses in Prozent. Hier können Sie detailliert erfassen, für wie wahrscheinlich Sie es halten, dass aus der Opportunität ein Projektauftrag wird. Zusammen mit dem Betrag kann so ein gewichteter Betrag ermittelt werden. Dazu wird der eingegebene Betrag mit der Wahrscheinlichkeit multipliziert. Beispiel: Bei einem Betrag von 10'000 CHF und einer Wahrscheinlichkeit von 50 % ergibt sich ein gewichteter Betrag von 5'000 CHF. Steigt die Wahrscheinlichkeit auf 80 %, so ergibt sich ein gewichteter Betrag von 8'000 CHF.
Währung	Die Währung, in der das resultierende Projekt abgerechnet werden soll. Die hier eingetragene Währung wird automatisch in ein aus der Opportunität neu angelegtes Projekt übernommen.
Beschreibung	Freitext als Beschreibung der Opportunität.
Bemerkung	Freitext als weitergehende Bemerkung.

Sie finden die Opportunitäten sowohl unter dem eingetragenen **Kunden**, als auch bei dem **Verkäufer**.

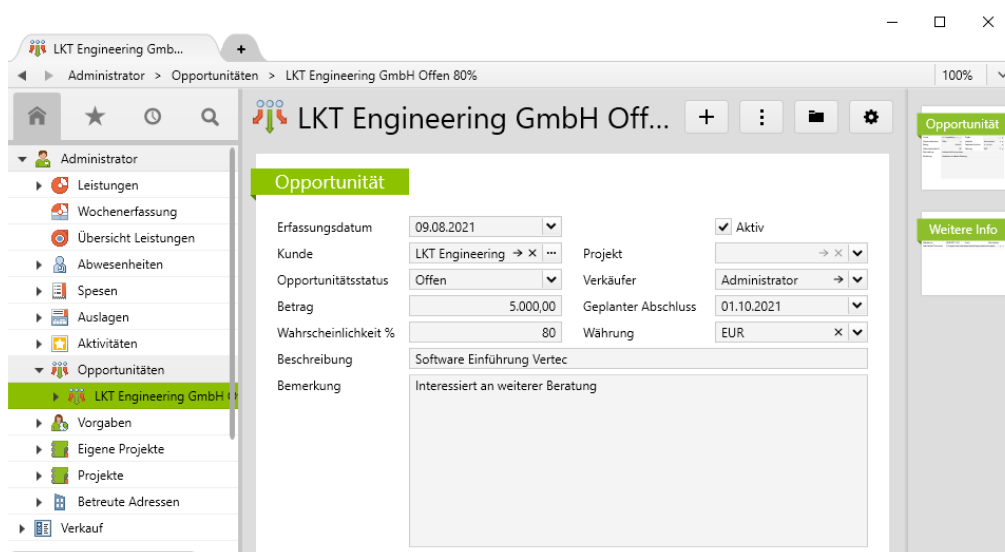


Abbildung 25: Link Opportunitäten unter Verkäufer

3.3 Opportunitätsstatus

Auf Opportunitäten lässt sich ein **Opportunitätsstatus** auswählen. Die Opportunitätsstatus werden im gleichnamigen Ordner in den **Einstellungen** gepflegt. Dafür sind Administratorrechte erforderlich.

Es sind bereits drei Opportunitätsstatus angelegt:

- Offen
- Gewonnen
- Verloren

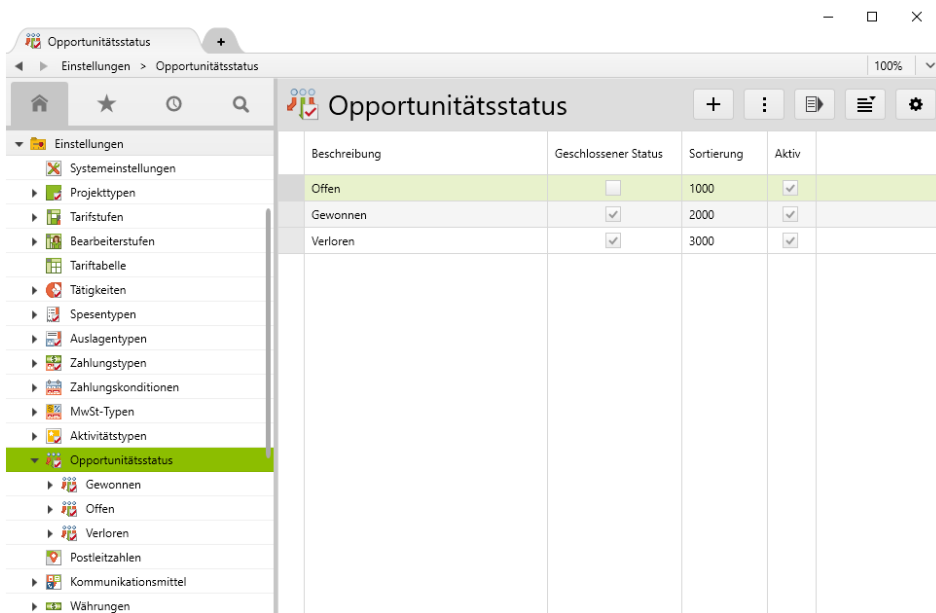


Abbildung 26: Opportunitätsstatus

Über das Neu-Menü können zudem bei Bedarf weitere Status angelegt werden, um den Prozess detaillierter zu beschreiben.

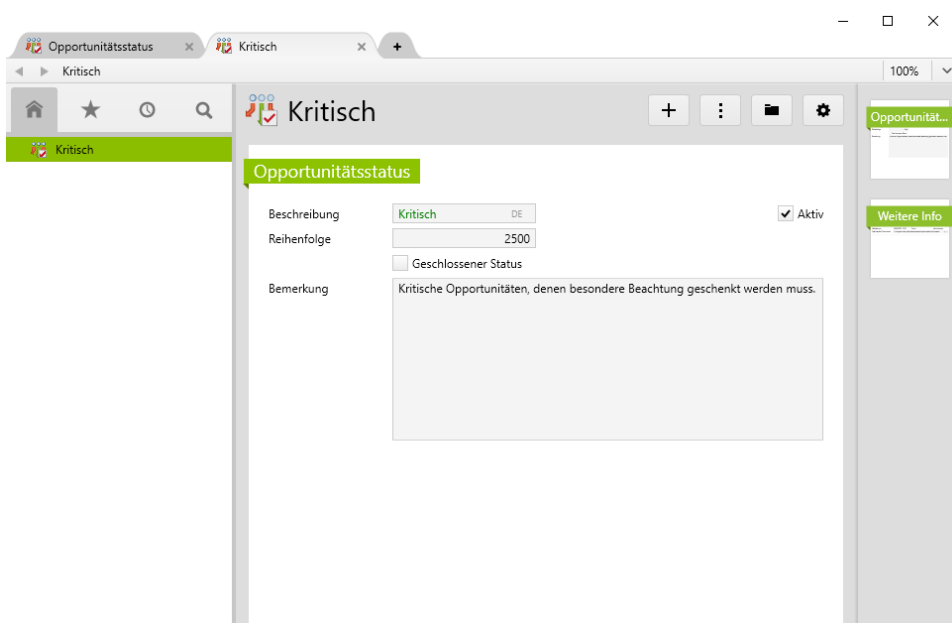


Abbildung 27: Detailansicht Opportunitätsstatus

Ein Opportunitätsstatus enthält folgende Felder:

Beschreibung	Freitext als Beschreibung des Opportunitätsstatus.
Reihenfolge	Gibt an, an welcher Reihenfolge in der Combobox auf Opportunitäten der Status zu sehen ist.
Geschlossener Status	Gibt an, ob mit diesem Status die Opportunität abgeschlossen ist. Bei den bereits angelegten Opportunitätsstatus gilt das für die Status Gewonnen und Verloren . Beim Setzen eines geschlossenen Status auf einer Opportunität, wird die Opportunität automatisch deaktiviert.
Bemerkung	Freitext als weitergehende Bemerkung

3.4 Anlegen eines Projekts aus einer Opportunität

Aus jeder Opportunität kann ein Projekt angelegt werden. Dazu wählen Sie auf der Opportunität im Neu-Menü den Eintrag **Projekt**.

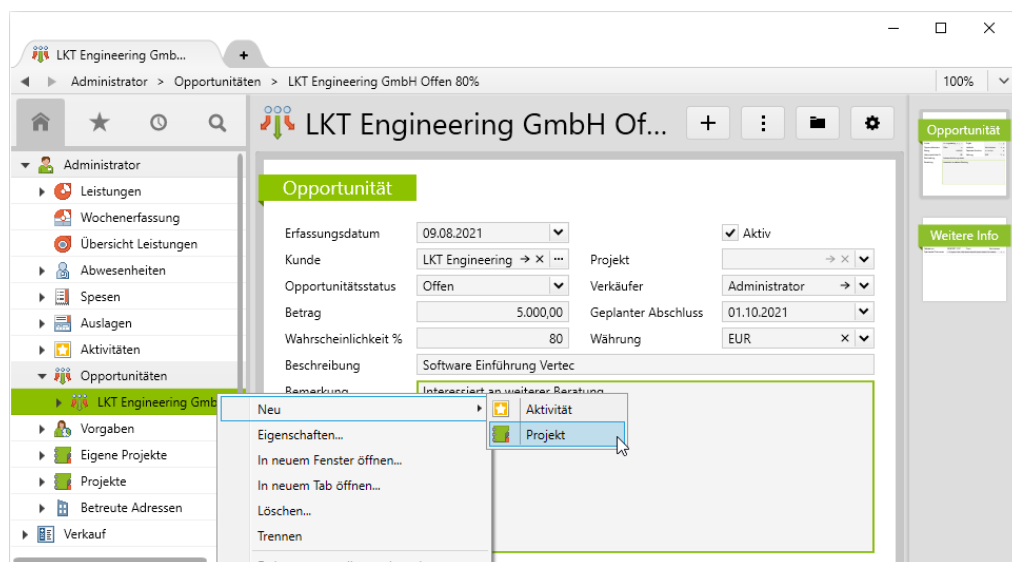


Abbildung 28: Neues Projekt anlegen aus Opportunität

Es öffnet sich die Detailansicht des neu angelegten Projektes. Folgende Felder werden aus der Opportunität übernommen:

Kunde	Der Kunde der Opportunität wird auch als Kunde des Projektes eingetragen. Sollten Sie als Kunde der Opportunität einen Kontakt gewählt haben, so ist es sinnvoll, im Projekt stattdessen die Firma des Kontaktes einzutragen.
Währung	Die Währung der Opportunität wird als Währung auf dem Projekt eingetragen, falls sie von der in den Systemeinstellungen hinterlegten Leitwährung abweicht.

Die **Opportunitäten** zu einem Projekt finden Sie unter dem entsprechenden Link im Navigationsbereich.

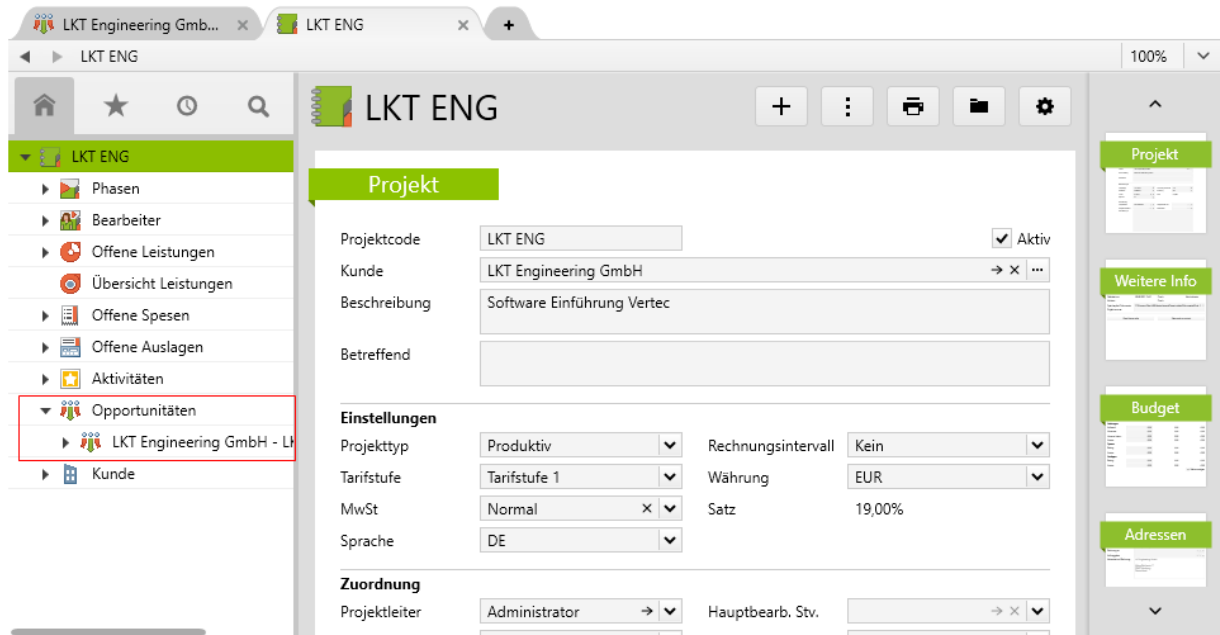


Abbildung 29: Link Opportunitäten unter Projekt

3.5 Aktivitäten

Um die Kommunikation mit dem Kunden zu dokumentieren, können zu Opportunitäten Aktivitäten angelegt werden. Dazu klicken Sie auf der Opportunität im Neu-Menü auf den Eintrag **Aktivität**.

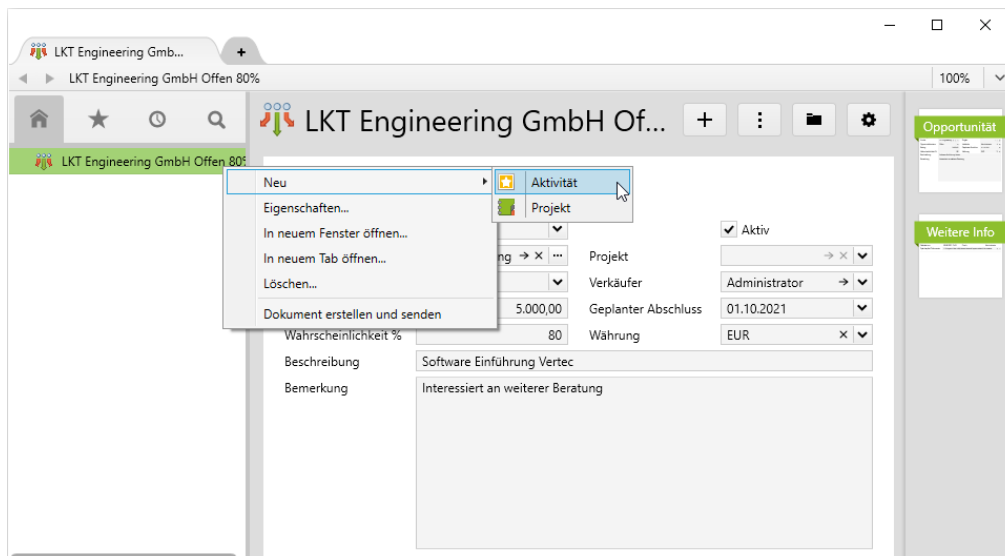


Abbildung 30: Aktivität zu Opportunität anlegen

Es öffnet sich eine neue Aktivität, die sie wie gewohnt bearbeiten können.

Um eine bestehende Aktivität einer Opportunität zuzuweisen, wählen Sie auf der Aktivität im Feld **Opportunität** den gewünschten Eintrag aus. Das Feld ist nur vorhanden, wenn das **Plug-in Opportunitäten** verwendet wird. Andernfalls können Sie die Aktivität via Drag and Drop zu einer Opportunität zuweisen.

3.6 Berechtigungen

Alle Standardbenutzer dürfen Opportunitäten erfassen und eigene Opportunitäten bearbeiten und löschen. Auf Opportunitäten anderer Benutzer haben Standardbenutzer keinen Zugriff.

Benutzer, die der Benutzergruppe **Projektadministratoren** oder **Adressadministratoren** zugeordnet sind, dürfen Opportunitäten aller Benutzer lesen, ändern und löschen.

Möchte weiteren Benutzergruppen Zugriff auf die Opportunitäten gewährt werden, so können in der entsprechenden Benutzergruppe unter **Berechtigungen** detaillierte Berechtigungen für Opportunitäten vergeben werden.

4 Fakturierung

4.1 Nur aktive Zahlungstypen in der ComboBox

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.10

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Die Zahlungstyp ComboBox (**PaymentTypeComboBox**) bringt nur noch **aktive** Zahlungstypen zur Auswahl. Dies kommt an folgenden Stellen zum Tragen: Seite **Vorgaben** auf Adressen, Seite **Buchhaltung** auf Projekt und Projektyp sowie bei **Zahlungen**.

4.2 Erfassen von neuen Kreditoren via QR-Code

Linie: Standard, Expert | Module: Leistung & CRM, Fremdkosten | Version: 6.4.0.14

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Ein Kreditor kann samt Adresseintrag (Lieferant) direkt aufgrund eines QR-Codes angelegt werden:

Nach dem Eröffnen eines Kreditors, kann auf den Button **Beleg importieren...** geklickt werden. Es erscheint ein Dateiauswahl-Dialog, in dem Sie den Rechnungsbeleg auswählen können. Es werden die Formate PDF, PNG oder JPG File unterstützt.

Enthält der importierte Beleg einen **Swiss-QR-Code**, werden die folgenden Angaben des Kreditors automatisch gesetzt:

- **Konto:** QR-IBAN-Nummer
- **Referenz:** Referenznummer
- **Währung**
- **Betrag**
- **Lieferant:** Gibt es in Vertec bereits eine Firma mit exaktem Namen und PLZ, wird diese Firma zugeordnet. Falls keine passende Firma gefunden wird, erscheint ein Dialog, welcher fragt, ob ein Lieferant angelegt werden soll.
Falls die Berechtigung, Firmen anzulegen, fehlt, erscheint eine Fehlermeldung. In diesem Fall können QR-Codes nur eingelesen werden, wenn die Adressen bereits im System vorhanden sind.
Wenn der Kreditor bereits einen Lieferanten zugeordnet hat und die Suche einen anderen ergibt, erscheint ein Dialog, der einen Fehler meldet. Der aktuelle Lieferant wird nicht überschrieben.
- **Externe Nummer:** Rechnungsnummer, falls im QR-Code encodiert
- **Fälligkeitsdatum,** falls im QR-Code encodiert
- **Datum:** Datum der Rechnung, falls im QR-Code encodiert

Wird ein Beleg importiert, obwohl bereits ein Beleg vorhanden ist, erscheint ein Dialog, welcher fragt, ob der bestehende Beleg überschrieben werden soll. Falls ja, werden bei einem Beleg mit QR-Code alle Angaben gemäss obiger Auflistung neu gesetzt.

Ist genau eine Auslage vorhanden, wird gleichzeitig noch der Betrag in die Auslage geschrieben und die Anzahl auf null gesetzt.

Wird ein Beleg ohne QR-Code importiert, wird der Beleg gespeichert, aber keine Angaben auf dem Kreditor gesetzt.

4.3 Unterstützung von Rechnungsdatum in QR-Codes bei Debitoren

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.14

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Ist das Rechnungsdatum oder die Zahlungsfrist einer Rechnung bekannt, wird die Zusatzinformation im QR-Code kodiert und im Zahlteil der Rechnung als sichtbare Zeichenkette ausgegeben. Die Rechnungsvorlagen

Rechnung mit Leistungsliste, Rechnung mit Bearbeitersummen und Rechnung mit Phasensummen geben die entsprechenden Informationen als Zusatzinformationen aus:

Vor der Einzahlung abzutrennen


Empfangsschein Konto / Zahlbar an CH6100700114600022365 Klarup AG Kasernenstrasse 29 3014 Bern Referenz 93 27300 00000 00000 01501 00049 Zahlbar durch Comtelsat AG Dachsweg 12 2500 Biel/Bienne <table border="0"> <tr> <td>Währung</td> <td>Betrag</td> </tr> <tr> <td>CHF</td> <td>8 326.80</td> </tr> </table> <div style="text-align: right;">Annahmestelle</div>	Währung	Betrag	CHF	8 326.80	Zahlteil  <table border="0"> <tr> <td>Währung</td> <td>Betrag</td> </tr> <tr> <td>CHF</td> <td>8 326.80</td> </tr> </table>	Währung	Betrag	CHF	8 326.80	Konto / Zahlbar an CH6100700114600022365 Klarup AG Kasernenstrasse 29 3014 Bern Referenz 93 27300 00000 00000 01501 00049 <div style="border: 1px solid red; padding: 2px;"> Zusätzliche Informationen //S1/10/15010004/11/150102/40/0:10 </div> Zahlbar durch Comtelsat AG Dachsweg 12 2500 Biel/Bienne
Währung	Betrag									
CHF	8 326.80									
Währung	Betrag									
CHF	8 326.80									

Abbildung 31: Zusätzliche Informationen auf QR-Rechnung

4.4 Rechnungsberichte mit fixierter Adresse von Rechnung

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.16

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Unsere Rechnungsberichte wurden so umgestellt, dass sie die fixierte Rechnungsadresse verwenden.

4.5 Schreibzugriff auf verr Attribute nicht mehr möglich

Linie: Standard, Expert | Module: Leistung & CRM, Budget & Teilprojekt, Fremdkosten | Version: 6.4.0.21

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Auf verschiedenen Objekten gibt es sogenannte **verr** Attribute, welche bei der Verrechnung des Objekts geschrieben werden und gewisse Werte fixieren. Bei diesen wurde der Schreibzugriff nun generell verweigert. Dies betrifft folgende Attribute:

- ProjektEintrag
 - VERRERTRAGSKONTO
 - VERRKOSTENSTELLE
 - VERRMWSTSATZ
 - VERRMWSTCODE
 - VERRMINUTENEXT
 - VERRWERTEXT
- Vorschuss
 - VERRVORSCHUSSKONTO
- Rechnung
 - VERRLEISTMWSTSATZ
 - VERRSPEENMWSTSATZ
 - VERRVORSCHUSSEFFEKTIV
 - VERRLEISTMWSTCODE
 - VERRSPEENMWSTCODE
 - VERRVORSCHUSSEFFEKTIVMITMWST
 - VERRTOTAL

- VERRAUSLAGENEXT
- VERRSPESENEXT
- VERRLEISTWERTEXT
- VERRRECHNUNGSADRESSTEXT
- VERRRECHNUNGSBRIEFANREDE
- VERRRECHNUNGSGRUSSTEXT
- VERRIBAN
- VERRESRTEILNEHMER
- VERRESRREFERENZ27
- VERRESRPKSTELLEN
- VERRESRPLUS
- VERRBESR
- VERRBESRBANK
- VERRBESRKUNDE
- VERRCOMPANYNAME
- VERRCOMPANYORT
- VERRCOMPANYPLZ
- VERRCOMPANYADRESSE
- VERRCOMPANYLAND
- VERRCOMPANYADRESSTEXT
- VERRCOMPANYMWSTNR
- VERRPROJEKTLITER
- VERRHB

Schreibzugriff auf ein VERR-Attribut ergibt folgenden Fehler: `AccessDeniedError: write access denied on Objekt.Attribut.`

Ausnahme: Im Systemcontext sowie wenn Vertec mit dem Parameter `/SUPER` gestartet wird, lassen sich die Attribute weiterhin schreiben.

4.6 QR-Code Schweizerkreuz Dimensionen verbessert

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.22

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Die Berechnung des QR-Codes wurde so geändert, dass bei Skalierung des gesamten Codes auf 46 mm das CH-Kreuz auf 7 mm kommt.

Der bestehende QR-Code wurde dafür geändert. Das bedeutet, dass es auch bei bereits bestehenden Verwendungen zu Unterschieden kommen kann.

4.7 QR-Zahlteil in Rechnungsreports Postfinance-konform

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.22

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Der QR-Zahlteil in den mitgelieferten Rechnungsreports wurde so angepasst, dass sie Postfinance-konform sind. Dies betrifft die Reports `Rechnung mit Leistungsliste`, `Rechnung mit Bearbeitersummen` und `Rechnung mit Phasensummen`.

4.8 Zahlungstyp auf Rechnung festlegen

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.5.0.1

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Der Zahlungstyp kann direkt auf der Rechnung und somit unabhängig vom Projekt festgelegt werden. Sobald die Rechnung verrechnet wird, kann der Zahlungstyp nicht mehr gesetzt werden.

Der Zahlungstyp auf Rechnung wird standardmässig nicht auf der Oberfläche angezeigt. Ein Zugriff kann via OCL oder Python über die Eigenschaft `xZahlungstyp` der Rechnung erfolgen.

4.9 Persistente Attribute auf Rechnungen

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.5.0.1

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Eine Rechnung in Vertec hat sogenannte **Totalen Objekte**, welche Leistungen, Spesen und Auslagen gruppiert nach MWST und Buchungskonten zusammenfassen.

Die meisten Zahlen auf einer Rechnung berechnen sich aufgrund dieser Totale.

Bei offenen Rechnungen werden die Totale laufend neu berechnet, so dass sich die Zahlen auf der Rechnung sofort an geändert Detaildaten anpassen.

Bei verrechneten Rechnungen werde die Totale anstelle der Projekteinträge (Leistungen, Spesen etc.) aus der Datenbank geladen.

Um eine Performance-Verbesserung beim Laden von Rechnungsdaten zu erzielen, werden mit Vertec Version 6.5 die wichtigsten Zahlen der Rechnung, insbesondere die, welche in Standard-Listen dargestellt werden, bei verrechneten Rechnungen direkt in einem Attribut (`verr`-Attribut) in der Datenbank gespeichert. Damit werden die zum Laden der Totale notwendigen SQL-Abfragen eingespart.

Es gibt daher neu die folgenden Attribute auf einer Rechnung:

- `verrAuslagenExt`
- `verrSpesenExt`
- `verrLeistWertExt`

Bei der Abfrage der Attribute

- `auslagenExt`
- `spesenExt`
- `leistWertExt`

auf einer verrechneten Rechnung werden die entsprechenden `verr`-Attribute herangezogen. Dieselbe Logik wird schon bei weiteren Derived Attributes auf Rechnungen verwendet, z.B. `vorschussEffektiv`.

Bei neu erstellten Rechnungen werden die Attribute beim Verrechnen gesetzt. Bei bestehenden Rechnungen werden wie bisher die entsprechenden Werte zur Laufzeit berechnet. Sie haben aber die Möglichkeit mit einem auf der Knowledge Base zur Verfügung stehendem [Script](#) in allen bestehenden Rechnungen die entsprechenden persistenten Attribute nachzutragen. So können Sie auch auf vorhandenen Rechnungen von der Performance-Optimierung profitieren.

5 Dokumente und Reports

5.1 Testdaten und Ausführen Button auf EOBs entfernt

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.4

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Das Feld **Testdaten** und der Button **Bericht ausführen** bei der Bericht-Definition von erweiterten Office-Berichten wurde entfernt, da dies immer wieder zu Problemen geführt hat, z.B. bei Verwendung von Link-Containern statt Ordnern.

5.2 Upload von Dokumenten auf einer Aktivität in der Web App

Linie: Expert | Module: Leistung & CRM | Version: 6.4.0.8

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Auch in der Web App können nun Dokumente auf einer Aktivität hochgeladen werden:

- Im Fileauswahl-Dialog auf einer Aktivität mit DMS-Dokumentenpfad kann ein lokales File ausgewählt werden. Dieses wird anschliessend ins DMS hochgeladen. Der Ordnerpfad entspricht dem Dokumentenpfad der Aktivität, der Filename dem ausgewählten File.
- `vtcapp.requestfilefromclient` kann auch in der Web App verwendet werden.
- Bei beiden Varianten erscheint ein Dialog, wo das File ausgewählt werden kann.

5.3 Speichern von PDF Belegen in der Phone App

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.9

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Mit der Phone App können PDF Belege für Spesen via **Senden an** in Vertec gespeichert werden. Die Phone App öffnet sich dann direkt in der Spesenansicht mit dem angehängten PDF.

5.4 Berichte ohne Filesystem

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.10

Betriebsart: On-Premises | Apps: Full-featured

Die Berichtsvorlagen befinden sich nicht mehr auf dem Filesystem, sondern direkt in der **Vertec Datenbank**. Die Standardberichte werden von Vertec als Ressourcen mitgeliefert. Bestehende kundenspezifische Berichtsvorlagen werden beim Update vom Filesystem in die Datenbank importiert.

- Berichtsvorlagen kommen alle entweder aus den Ressourcen (standardmässig von Vertec mitgeliefert) oder aus der Datenbank (durch Kunden hochgeladen). Dies gilt für alle Arten von Berichten, also für Office-generierte, Vertec-generierte und erweiterte Office-Berichte (Word und Excel).
- Das `VertecReport.dotm` Dokument, welches den Office Berichtsgenerator enthält, bleibt auf dem Filesystem erhalten. Dieses liegt entweder im Vertec Installationsverzeichnis oder im Reports-Verzeichnis. Löscht man den Reports-Ordner und die Datei war nur dort verfügbar oder freigegeben, kann dies zur Nicht-gefunden-Fehlermeldung führen. In diesem Fall sollte die Datei vorgängig ins Vertec Installationsverzeichnis verschoben werden.
- Bestehende kundenspezifische Berichtsvorlagen werden beim Update automatisch vom Filesystem in die Datenbank importiert.
- Die Systemeinstellung Vorlagenpfad für Word Berichte ist nicht mehr vorhanden. Scripts, welche darauf zugreifen, müssen angepasst werden.
- Der Menüpunkt **Drucken > Vorlage auswählen...** wurde entfernt.

5.5 Rechnungsreports ohne Briefanrede

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.19

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Die Briefanrede wurde aus den folgenden Vorlagen entfernt:

- Rechnung mit Leistungsliste (**InvoiceWithList**)
- Rechnung mit Bearbeitersummen (**InvoiceWithUserSums**)
- Rechnung mit Phasensummen (**InvoiceWithPhaseSums**)

Im Code existiert das Feld jedoch weiterhin und hat somit keine Auswirkungen auf kundenspezifische Vorlagen.

5.6 Umstellung Stunden/Minuten Darstellung in Excel EOB auf Stunden Dezimal

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.22

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Die Stundendarstellung in Excel EOBs wurde vom bisher verwendeten HH:MM Format auf Stunden mit 2 Dezimalstellen gestellt.

Hinweis zur Systemeinstellung **Anzeige Minuten**: Die Option **Stunden Minuten** wird ignoriert und als **Stunden Dezimal** ausgegeben. Die Option Minuten wird hingegen wie gewohnt als **Minuten** dargestellt.

5.7 Beschrieb statt Code bei EOB Monatsübersicht mit Sollzeiten

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.10

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Der Erweiterte Office-Bericht **Monatsübersicht mit Sollzeiten** zeigt bei den Abwesenheiten neu die Übersetzung des Beschriebs anstelle des Codes an.

5.8 Übergabe von Aktivität beim Speichern von Dokumenten in Addins

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.23

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Neu übergeben auch die Office-, Explorer- und Outlook-Addins beim Speichern von Dokumenten in ein DMS wie z.B. SharePoint die Aktivität als Objekt. Damit können die entsprechenden Metadaten geschrieben werden.

6 Controlling / BI

6.1 BI API

Linie: Standard, Expert | Module: Leistung & CRM, Business Intelligence | Version: 6.4.0.14

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Ab Vertec 6.4.0.14 gibt es ein API (Application Programming Interface) zur Abfrage von BI Daten aus externen Applikationen.

Dafür gibt es in der Konfigurationsdatei **Vertec.ini-File** den Parameter **BI API**:

```
[CloudServer]
BI_API=True
```

Standard ist **True**.

- Das BI API kann nur verwendet werden, wenn das Business Intelligence Modul lizenziert ist. Anderenfalls wird ein Fehler zurückgegeben.
- Das BI API kann nur nach erfolgter Authentifizierung verwendet werden. Anderenfalls wird ein Fehler zurückgegeben.

Der Aufruf erfolgt über **/api/bi**, welches folgende Endpoints zur Verfügung stellt:

/measures

Der GET-Endpoint **/measures** gibt alle zur Verfügung stehenden Kennzahlen zurück.

Es kann ein optionaler Parameter **f** angegeben werden:

- **f=csv** (**/api/bi/measures?f=csv**) liefert das Resultat als CSV-Datei.
 - **f=json** (**/api/bi/measures?f=json**) liefert das Resultat in Json formatiert.
- Der Parameter kann statt als Query-String auch als Accept Header angegeben werden. Ist beides vorhanden, wird der Query-String verwendet.

Die Rückgabe enthält folgende Felder:

- Name
- Description
- Dimensions (string[] der Klassennamen)
- Helptext
- Unit

Die Rückgabewerte erscheinen übersetzt in der Sprache, die der Benutzer in der Web App eingestellt hat.

Beispiel Abfrage

```
/api/bi/measures
```

Liefert folgendes Resultat zurück:

```
[{"Name": "PercentageOfCompletion", "Description": "Fertigstellungsgrad", "Unit": "Percentage", "Dimensions": [{"Projekt": "Projektbearbeiter", "ProjektPhase": "ProjektPhase"}, {"Name": "RateCharged", "Description": "Stundensatz verrechenbar", "Unit": "Rate", "Dimensions": [{"Projekt": "Projektbearbeiter", "ProjektPhase": "ProjektPhase"}, {"Name": "Advances", "Description": "Bestand Vorschüsse", "Unit": "Money", "Dimensions": [{"Projekt": "Projektbearbeiter", "ProjektPhase": "ProjektPhase"}, {"Name": "FeesCommenced", "Description": "Angefangene Arbeiten", "Unit": "Money", "Dimensions": [{"Projekt": "Projektbearbeiter", "ProjektPhase": "ProjektPhase"}, {"Name": "MinutesBudgetTotal", "Description": "Aufwand Budget Total", "Unit": "Hours", "Dimensions": [{"Projekt": "Projektbearbeiter", "ProjektPhase": "ProjektPhase"}, {"Name": "MinutesBudgetRest", "Description": "Aufwand Budget Rest", "Unit": "Hours", "Dimensions": [{"Projekt": "Projektbearbeiter", "ProjektPhase": "ProjektPhase"}, {"Name": "MinutesCumulated", "Description": "Aufwand Ist", "Unit": "Hours", "Dimensions": [{"Projekt": "Projektbearbeiter", "ProjektPhase": "ProjektPhase"}, {"Name": "FeesOrderbacklog", "Description": "Auftragsbestand Honorar", "Unit": "Money", "Dimensions": [{"Projekt": "Projektbearbeiter", "ProjektPhase": "ProjektPhase"}, {"Name": "FeesOrderintake", "Description": "Auftragseingang Honorar", "Unit": "Money", "Dimensions": [{"Projekt": "Projektbearbeiter", "ProjektPhase": "ProjektPhase"}, {"Name": "VacationBalanceDelimited", "Description": "Feriensaldo abgegrenzt", "Unit": "Hours", "Dimensions": [{"Projekt": "Projektbearbeiter", "ProjektPhase": "ProjektPhase"}, {"Name": "VacationBalance", "Description": "Feriensaldo", "Unit": "Hours", "Dimensions": [{"Projekt": "Projektbearbeiter", "ProjektPhase": "ProjektPhase"}, {"Name": "OvertimeBalance", "Description": "Überzeitsaldo", "Unit": "Hours", "Dimensions": [{"Projekt": "Projektbearbeiter", "ProjektPhase": "ProjektPhase"}, {"Name": "WorkingTime", "Description": "Arbeitszeit", "Unit": "Hours", "Dimensions": [{"Projekt": "Projektbearbeiter", "ProjektPhase": "ProjektPhase"}, {"Name": "Fte", "Description": "Vollzeitzellen", "Unit": "Percentage", "Dimensions": [{"Projekt": "Projektbearbeiter", "ProjektPhase": "ProjektPhase"}, {"Name": "TurnoverOutlays", "Description": "Umsatz Auslagen", "Unit": "Money", "Dimensions": [{"Projekt": "Projektbearbeiter", "ProjektPhase": "ProjektPhase"}, {"Name": "TurnoverExpenses", "Description": "Umsatz Spesen", "Unit": "Money", "Dimensions": [{"Projekt": "Projektbearbeiter", "ProjektPhase": "ProjektPhase"}, {"Name": "MarginContributed", "Description": "Deckungsbeitrag (DB)", "Unit": "Money", "Dimensions": [{"Projekt": "Projektbearbeiter", "ProjektPhase": "ProjektPhase"}, {"Name": "FeesExtWrittenOff", "Description": "Honorar extern verrechnet - Kosten", "Unit": "Money", "Dimensions": [{"Projekt": "Projektbearbeiter", "ProjektPhase": "ProjektPhase"}, {"Name": "FeesExtCharged", "Description": "Honorar intern verrechnet - Kosten", "Unit": "Money", "Dimensions": [{"Projekt": "Projektbearbeiter", "ProjektPhase": "ProjektPhase"}, {"Name": "FeesExtOpen", "Description": "Honorar extern offen", "Unit": "Money", "Dimensions": [{"Projekt": "Projektbearbeiter", "ProjektPhase": "ProjektPhase"}, {"Name": "FeesExtClosed", "Description": "Honorar intern geschlossen", "Unit": "Money", "Dimensions": [{"Projekt": "Projektbearbeiter", "ProjektPhase": "ProjektPhase"}, {"Name": "Costs", "Description": "Kosten", "Unit": "Money", "Dimensions": [{"Projekt": "Projektbearbeiter", "ProjektPhase": "ProjektPhase"}, {"Name": "MinutesExt", "Description": "Aufwand extern", "Unit": "Hours", "Dimensions": [{"Projekt": "Projektbearbeiter", "ProjektPhase": "ProjektPhase"}, {"Name": "MinutesInt", "Description": "Aufwand intern", "Unit": "Hours", "Dimensions": [{"Projekt": "Projektbearbeiter", "ProjektPhase": "ProjektPhase"}, {"Name": "FeesInt", "Description": "Honorar intern", "Unit": "Money", "Dimensions": [{"Projekt": "Projektbearbeiter", "ProjektPhase": "ProjektPhase"}, {"Name": "FeesExt", "Description": "Honorar extern", "Unit": "Money", "Dimensions": [{"Projekt": "Projektbearbeiter", "ProjektPhase": "ProjektPhase"}]}
```

/getdata

Der GET-Endpoint `/getdata` gibt die BI Daten zurück. Er nimmt folgende Parameter via Query-String entgegen:

measures: Kommaseparierte Liste der internen Kennzahlennamen (z.B. `"FeesExt, MinInt"`). Obligatorisch.

dimension: Liste der Dimensionen, nach denen das Ergebnis gruppiert ist (z.B. `"Projekt, Projektbearbeiter"`). Obligatorisch, case-insensitive.

- Als Pseudo-Dimension kann auch `"Month"` angegeben werden. Stichtageswerte wie `OvertimeBalance` werden nur akzeptiert für Zeitreihen. Bei diesen Werten muss also `dimension=month` angegeben werden, andernfalls erscheint eine Fehlermeldung.
- Sind verschiedene Rolle vorhanden, kann der Zugriff über ein Suffix [Index 0 .. 4] erfolgen:
 - `Projekt` funktioniert wie bisher und wählt die Dimension Projekt ohne Rolle.
 - `Projekt_1` wählt auf der Kennzahl die Dimension Projekt auf Index 1. Dies muss dann eine Rolle auf Projekt sein.
 - Beispiel bei Abfrage von `FeesExt` (Honorar extern):

```
dimension0 = Projekt
dimension1 = Projektbearbeiter_1
dimension2 = Projektbearbeiter_2
dimension3 = ProjektPhase
dimension4 = Taetigkeit
dimension5 = Month
```

- Werden mehrere Kennzahlen mit unterschiedlichem "Dimensionsaufbau" angefordert und die angeforderten Dimensionen können nicht über alle Kennzahlen geliefert werden, erscheint eine entsprechende Fehlermeldung. In einem solchen Szenario müssen die Kennzahlen mit gleicher Dimensionsstruktur in einer Query gebündelt und mehrere dieser Queries abgesetzt werden, um alle Daten zu extrahieren.

startDate: Startdatum im ISO 8601 Format (2020-01-01), optional.

endDate: Enddatum im ISO 8601 Format (2020-12-31), optional.

selectExpression: OCL-Expression für die Selektion, optional.

useDescriptions: True oder False. Bestimmt, ob die Spaltenbezeichnungen im Klartext und übersetzt geliefert werden oder als intern technische Bezeichnungen, die sprachunabhängig sind. Optional, Standard: False.

Der Endpoint liefert Ergebnisse in folgendem Format:

- Header (Spalten-Namen)
 - **Name** der angefragten Kennzahl (Interner Name). Falls der Parameter `useDescriptions = true` ist, wird statt des Internen Namens die Klartext-Bezeichnung der Kennzahl verwendet und gemäss aktueller Sprache der Session übersetzt zurückgeliefert.
- Datenzeile
 - Dimensionswerte
 - Kennzahl Werte

Die BI Berechtigungen werden angewendet.

Beispiel Abfrage FTE für das erste Quartal 2020

```
api/bi/getdata?measures=fte&dimension0=Projektbearbeiter&useDescriptions=true&dimension1=month&startDate=2020-01-01&endDate=2020-03-01
```

Rückgabe:

```
[{"Bearbeiter": "Administrator", "Monat": "2020-01-01", "Vollzeitstellen": 1.0}, {"Bearbeiter": "Administrator", "Monat": "2020-02-01", "Vollzeitstellen": 1.0}, {"Bearbeiter": "Administrator", "Monat": "2020-03-01", "Vollzeitstellen": 1.0}, {"Bearbeiter": "Christoph Keller", "Monat": "2020-01-01", "Vollzeitstellen": 1.0}, {"Bearbeiter": "Christoph Keller", "Monat": "2020-02-01", "Vollzeitstellen": 1.0}, {"Bearbeiter": "Christoph Keller", "Monat": "2020-03-01", "Vollzeitstellen": 1.0}, {"Bearbeiter": "Judith Feller", "Monat": "2020-01-01", "Vollzeitstellen": 1.0}, {"Bearbeiter": "Judith Feller", "Monat": "2020-02-01", "Vollzeitstellen": 1.0}, {"Bearbeiter": "Judith Feller", "Monat": "2020-03-01", "Vollzeitstellen": 1.0}, {"Bearbeiter": "Paul Thomson", "Monat": "2020-01-01", "Vollzeitstellen": 1.0}, {"Bearbeiter": "Paul Thomson", "Monat": "2020-02-01", "Vollzeitstellen": 1.0}, {"Bearbeiter": "Paul Thomson", "Monat": "2020-03-01", "Vollzeitstellen": 1.0}, {"Bearbeiter": "Werner Schmid", "Monat": "2020-01-01", "Vollzeitstellen": 1.0}, {"Bearbeiter": "Werner Schmid", "Monat": "2020-02-01", "Vollzeitstellen": 1.0}, {"Bearbeiter": "Werner Schmid", "Monat": "2020-03-01", "Vollzeitstellen": 1.0}, {"Bearbeiter": "Lars Sorenson", "Monat": "2020-01-01", "Vollzeitstellen": 0.5}, {"Bearbeiter": "Lars Sorenson", "Monat": "2020-02-01", "Vollzeitstellen": 0.5}, {"Bearbeiter": "Lars Sorenson", "Monat": "2020-03-01", "Vollzeitstellen": 0.5}]
```

Unterstützung für HTTP Basic Authentication

Für die Abfragen aus externer BI Software wird HTTP Basic Auth mit Username und Passwort unterstützt.

- Die gestartete Session hat einen Standard Timeout von 5 Minuten. Dieser kann in der [Cloud Server] Section im **Vertec.ini-File** mit `XML Session Timeout = 5` angepasst werden (das BI API und der XML Server teilen sich die Session Timeout Einstellung).
- Nach Ablauf des Timeout wird die Session beendet und der Prozess verschwindet

- Trifft während dieser Zeit eine weitere Anfrage mit Basic Authentication und demselben Benutzer ein, wird die Session wiederverwendet.

6.2 Speichern von BI Ansichten in Favoriten

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.9

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

BI Ansichten können als Favoriten gespeichert werden. Diese werden beim Aufruf mit allen Parametern dargestellt. Als Favoriten gespeicherte BI Anzeigen erhalten eine Bezeichnung nach dem Muster **Node (BI) : Displayname**, also beispielsweise **Projekt (BI) : Bearbeiter**. So können die BI Anzeigen in den Favoriten auf einen Blick identifiziert werden.

6.3 Erweiterte BI URL

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.9

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Die URL Unterstützung für die Business Intelligence (BI) Anzeigen wurde um folgende Eigenschaften erweitert:

- Datumsbereich
- Ausgewählte Anzeige
- Ausgewählte Kennzahlen inkl. Sortierung und Farbe

Eine URL sieht dann beispielsweise so aus:

```
vertec://bi-view/Stammdaten-303/Projekte+aktiv-306/?interval=2020-01~2020-06&display=DimensionDisplayProjektKunden&measure0=FeesExt&color0=3&measure1=FeesInt&color1=5&sort_measure=FeesInt&sort_order=Desc
```

6.4 Berechtigungen für BI Kennzahlen

Linie: Standard, Expert | Module: Leistung & CRM, Business Intelligence | Version: 6.4.0.14

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Berechtigungen können neu auch pro BI Kennzahl vergeben werden.

Dabei gilt:

- Das BI Daten Recht kann sowohl auf Klassen als auch auf Objekten vergeben werden.
- Sobald die Berechtigung auf BI Daten eingestellt und die entsprechende Klasse ausgewählt wird (möchte man eine Berechtigung auf allen Klassen setzen, hier **UserEintrag** wählen), kann man bei **Feldname** eine Kennzahl auswählen.
- Im Feld **Expressions** können wie bisher auf einer ausgewählten Klasse entsprechende Ausdrücke eingegeben werden.
- Damit der BI Button aktiv wird, muss der Benutzer auf der Auswahl mindestens eine Kennzahl freigeschaltet haben. Für ein einzelnes Objekt muss also eine Kennzahl verfügbar sein, für eine Liste von Objekten muss jedes Objekt mindestens eine Kennzahl haben.
- Gibt es im BI bei der Auswahl von Objekten zwar auf jedem Objekt mindestens eine Kennzahl, aber keine Kennzahl, für die auf allen Objekten eine Berechtigung gesetzt ist, erscheint die Anzeige **No data available**.
- Besteht für ein ausgewähltes Objekt im BI keine Berechtigung für irgendeine Kennzahl, erscheint die Anzeige **Zugriff verweigert**.

Im Rahmen dieser Neuerung wurde auch der OCL Operator **hasBiDataRight** angepasst. Er erwartet neu als Argument einen String mit dem internen Namen der Kennzahl.

Beispiel:

`projekt->hasBiDataRight('MinutesExt')` prüft, ob der aktuell eingeloggte Benutzer auf dem angegebenen Projekt das Recht auf die MinutesExt Kennzahl hat.

Rückwärtskompatibilität

Dies ist nicht rückwärtskompatibel. Die entsprechenden Aufrufe müssen angepasst werden.

7 Customizing

7.1 Liste mit offenen und verrechneten Leistungen für Leistungserfassung

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.19

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Leistungen wurden bisher unter **Offene Leistungen** eingetragen und nach dem Verrechnen nicht mehr angezeigt. Ab dieser Version heisst der Ordner auf dem Bearbeiter **Leistungen**. Wenn eine Leistung verrechnet wird, bleibt sie in der Liste unter Leistungen sichtbar, jedoch können keine Änderungen mehr vorgenommen werden und das Timer-Symbol verschwindet. Die Container für Spesen und Auslagen auf dem Bearbeiter verhalten sich analog.

7.2 Unterstützung für fixierte Spalten in Listeneinstellungen

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.8

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Die ersten Spalten in den Listen können fixiert werden, indem in den Listeneinstellungen ein Häkchen bei **fixiert** gesetzt wird.

- Die fixierten Spalten bleiben beim Scrollen links stehen.
- Die fixierten Spalten sind durch eine Trennlinie gekennzeichnet und bleiben beim Scrollen stehen.
- Es werden immer die ersten aufeinanderfolgenden fixierten Spalten berücksichtigt. Wird eine Spalte, deren vorangehende Spalten nicht alle fixiert sind, als fixiert konfiguriert, wird diese Einstellung ignoriert.

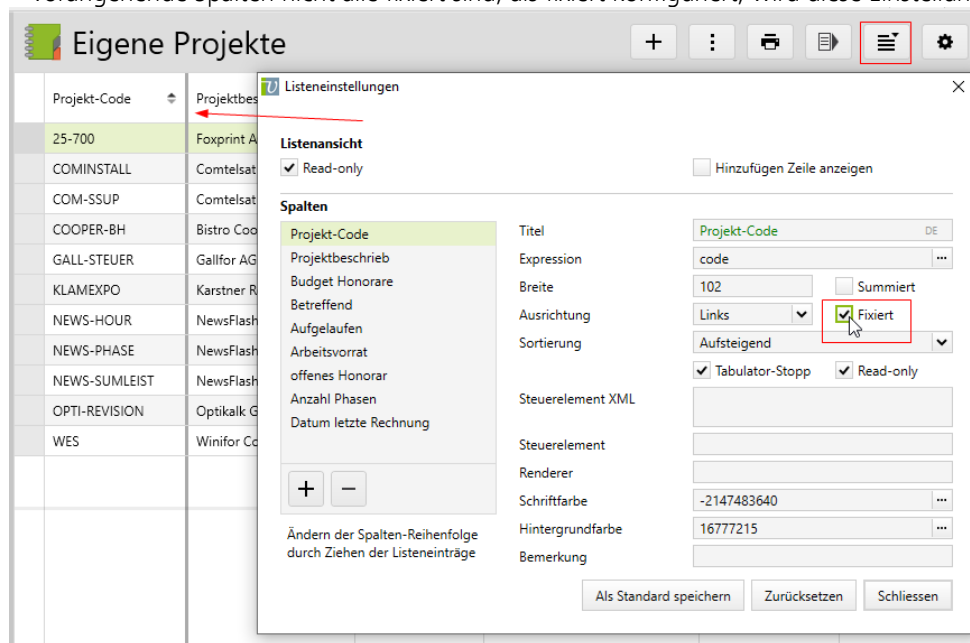


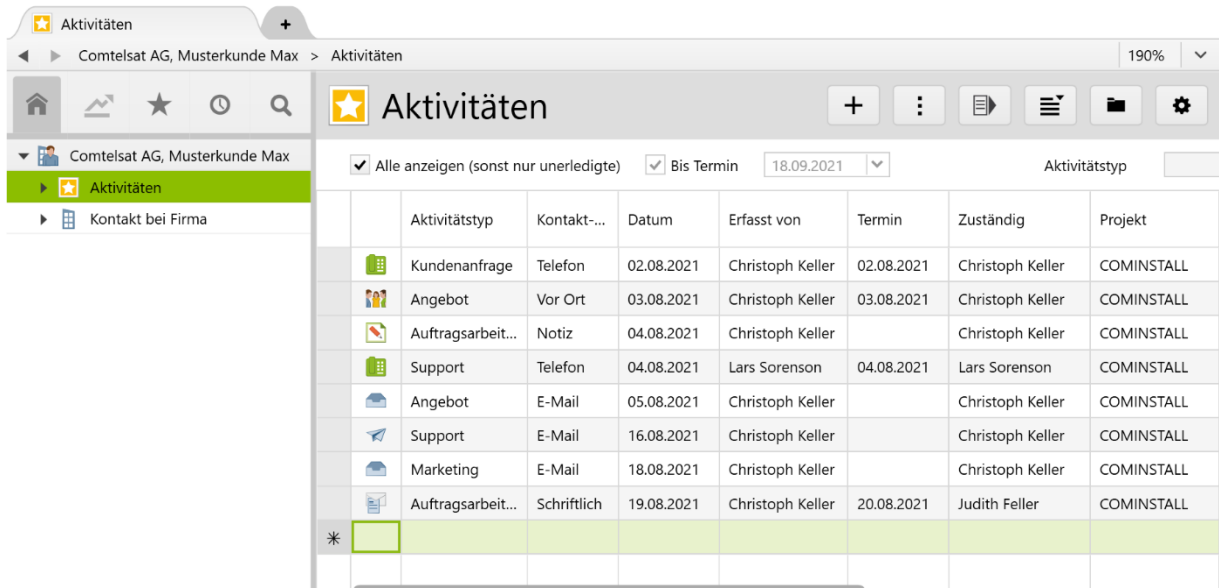
Abbildung 32: Fixierte Spalte

7.3 Icon Darstellung in Listen

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.5.0.1

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

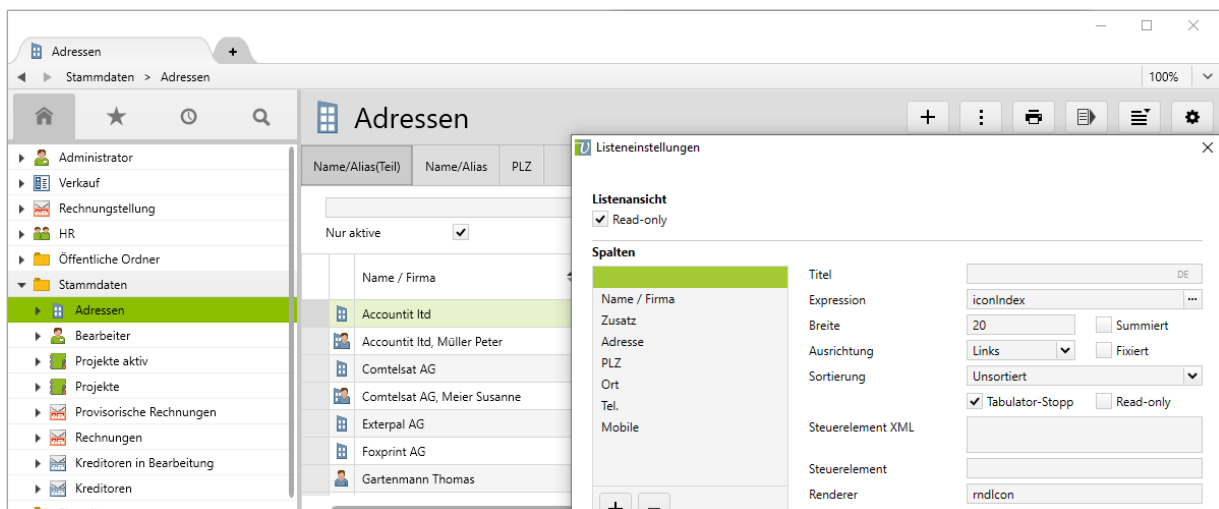
In der Aktivitätsliste gibt es neu eine Spalte, in welcher die Icons der dazugehörigen Aktivitätstypen dargestellt werden:



	Aktivitätstyp	Kontakt...	Datum	Erfasst von	Termin	Zuständig	Projekt
	Kundenanfrage	Telefon	02.08.2021	Christoph Keller	02.08.2021	Christoph Keller	COMINSTALL
	Angebot	Vor Ort	03.08.2021	Christoph Keller	03.08.2021	Christoph Keller	COMINSTALL
	Auftragsarbeit...	Notiz	04.08.2021	Christoph Keller		Christoph Keller	COMINSTALL
	Support	Telefon	04.08.2021	Lars Sorenson	04.08.2021	Lars Sorenson	COMINSTALL
	Angebot	E-Mail	05.08.2021	Christoph Keller		Christoph Keller	COMINSTALL
	Support	E-Mail	16.08.2021	Christoph Keller		Christoph Keller	COMINSTALL
	Marketing	E-Mail	18.08.2021	Christoph Keller		Christoph Keller	COMINSTALL
	Auftragsarbeit...	Schriftlich	19.08.2021	Christoph Keller	20.08.2021	Judith Feller	COMINSTALL
*							

Abbildung 33: Icon Darstellung in Listen

Auch anderen Listen ist es möglich, Icons in Spalten darzustellen. Dies macht insbesondere Sinn, wenn in einer Liste Objekte verschiedener Klassen dargestellt werden. Als Spaltenexpression trägt man in diesem Fall `iconindex` ein, zusätzlich muss der Renderer `rndIcon` angegeben werden.



Name/Alias(Teil)	Name/Alias	PLZ
Nur aktive <input checked="" type="checkbox"/>		
Name / Firma		
Accountit Ltd		
Accountit Ltd, Müller Peter		
Comtelsat AG		
Comtelsat AG, Meier Susanne		
Exterpal AG		
Foxprint AG		
Gartenmann Thomas		

Listeneinstellungen	
Listenansicht <input checked="" type="checkbox"/> Read-only	
Spalten	
Name / Firma	Titel
Zusatz	Expression
Adresse	Breite
PLZ	Ausrichtung
Ort	Sortierung
Tel.	Steuerelement XML
Mobile	Steuerelement
	Renderer

Abbildung 34: Listeneinstellungen zur Icon Darstellung in Listen

7.4 Icon bei IX-Ordern

Linie: Expert | Module: Leistung & CRM | Version: 6.4.0.5

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Ist ein Ordner als **Inklusiv/Exklusiv** markiert, erschien immer das IX-Icon, auch wenn explizit ein Icon-Index angegeben war. Nun wird, wenn vorhanden, das angegebene Icon verwendet, und nur bei `-1` (keine Angabe) das IX-Icon.

7.5 Icon Index für Subordner auf Ordnern

Linie: Expert | Module: Leistung & CRM | Version: 6.4.0.6

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Ordner erhalten die Eigenschaft **Icon für Unterordner**. Wird dort ein Icon Index eingesetzt, erhält jeder neu erstellte Subordner automatisch dieses Icon.

7.6 Klassenspezifische Sortierung von Einträgen im Baum

Linie: Standard, Expert | Module: Leistung und CRM | Version: 6.5.0.1

Betriebsart: Cloud Abo / On-Premises | Apps: Alle

Die Sortierung im Baum kann folgendermassen klassenspezifisch festgelegt werden:

- **Aktivitäten** werden im Baum nach Datum und Uhrzeit absteigend sortiert. Das Datum erscheint in den Spalten vor der Uhrzeit.
- **Projekteinträge** (Leistungen, Spesen, Auslagen) werden nach Datum aufsteigend sortiert. Innerhalb des gleichen Datums wird zusätzlich die Objekt-Id berücksichtigt.
- **Projektphasen** werden aufsteigend nach Projektcode und **orderidx** sortiert.
- **Rechnungen** werden nach Datum und Nummer aufsteigend sortiert.
- **Kreditoren** werden nach Datum und Nummer aufsteigend sortiert.
- **Bearbeitervorgaben** werden nach Datum aufsteigend sortiert. Innerhalb des gleichen Datums wird zusätzlich die Objekt-Id berücksichtigt.
- **Offerten** werden nach Datum und Nummer sortiert.
- **Opportunitäten** werden nach Datum sortiert. Innerhalb des gleichen Datums wird zusätzlich die Objekt-Id berücksichtigt.
- **Abwesenheiten** werden nach Datum sortiert. Innerhalb des gleichen Datums wird zusätzlich die Objekt-Id berücksichtigt.

Die Sortierung kann via **Klasseneinstellung** für jede Klasse individuell mittels einer **OCL Expression** definiert werden.

7.7 Python Methode zum Erstellen einer E-Mail in Outlook

Linie: Expert | Module: Leistung & CRM | Version: 6.4.0.4

Betriebsart: Cloud Abo / On-Premises | Apps: Desktop App, Cloud App

Es gibt in Python neu eine Methode **vtcapp.createoutlookmail()**, welche es ermöglicht, auf dem Client via Outlook eine E-Mail Nachricht zu erstellen.

Voraussetzung ist ein installierter Outlook Windows Client.

Die Syntax ist wie folgt:

```
createoutlookmail(to, subject, body, [cc, bcc, attachments, show])
```

- **to**: String. Empfängeradressen, mit **;** getrennt, falls mehrere.
- **subject**: String. Betreff der E-Mail.
- **body**: String. Der eigentliche E-Mail-Text. Kann HTML oder plaintext sein. HTML wird als solches erkannt und dann in Form einer Outlook HTML Mail erstellt.
- **cc**: String, Optional, Keyword. Cc-Adressen, mit **;** getrennt, falls mehrere.
- **bcc**: String, Optional, Keyword. Bcc-Adressen, mit **;** getrennt, falls mehrere.
- **attachments**: Liste von Tuples, Optional, Keyword. Liste von Attachments als Tuple von zwei Strings [(Name des Files, Inhalt)]

- **show**: Boolean, Optional, Keyword. Bei `true` wird die E-Mail erstellt und angezeigt, bei `false` wird die E-Mail im "Entwürfe"-Ordner gespeichert und nicht angezeigt. Standard, also wenn nichts anderes angegeben wird, ist `true`.

Keyword bedeutet, dass die optionalen Werte mit dem Parameter als Keyword, z.B. `CC="abc@vertec.com"` angegeben werden können. So wird keine bestimmte Anzahl Kommas als Platzhalter für nicht angegebene optionalen Werte benötigt.

Weitere Informationen Eine Reihe von Beispielen finden Sie im Artikel [Vertec Python Funktionen](#) auf der Online Knowledge Base unter www.vertec.com/kb/pythonfunktionen/.

7.8 Bemerkungsfeld in Listeneinstellungen

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.4

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

In den Listeneinstellungen gibt es neu pro Spalte ein Bemerkungsfeld:

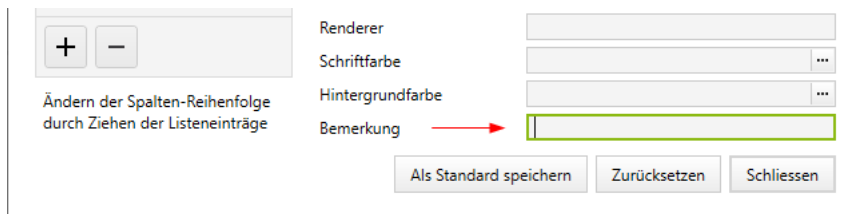


Abbildung 35: Bemerkungsfeld in den Listeneinstellungen

Dieses kann z.B. für die interne Dokumentation der entsprechenden Spalte eingesetzt werden.

7.9 Python Methode executereport liefert Bericht-Output als Byte-String

Linie: Expert | Module: Leistung & CRM | Version: 6.4.0.8

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Eine bessere Unterstützung für Berichte ohne Zugriff auf das Filesystem in Python Scripts: Die Python Methode `executereport` liefert den Bericht-Output als Byte-String. Dieser kann weiterverarbeitet und beispielsweise als Attachment an eine E-Mail gehängt werden via `vtcapp.createoutlookmail()`.

7.10 Anlegen von Systemeinstellungen mittels Config Set

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.10

Betriebsart: Cloud Abo / On-Premises | Apps: Desktop

Bisher konnten Systemeinstellungen per Config Set zwar verändert, nicht aber neu angelegt werden. Nun können Systemeinstellungen als normale Objekte in einem Config Set übergeben werden, mit Angabe all ihrer Eigenschaften.

- Will man **bestehende Systemeinstellungen verändern**, verwendet man die `system-settings` und gibt die entsprechenden Zeilen in der Config Set Builder Oberfläche an.
- Will man **neue Systemeinstellungen anlegen**, fügt man diese als Objekt direkt ins XML ein, nachdem man das Config Set das erste Mal exportiert hat. Systemeinstellungen haben keine Eintrag Id und werden darum über ihren Namen (`PropertyName`) identifiziert.

Die Systemeinstellungen werden direkt als Objekte eingefügt. Wird ein so importiertes Config Set wieder exportiert, bleibt das Systemeinstellungs-Objekt als Objekt im XML File bestehen. Die manuelle Eingabe muss also nur einmal erfolgen.

Weitere Informationen Ein Beispiel dazu und alle weiteren Informationen finden Sie im Artikel Config Set XML in der Online Knowledge Base unter <https://www.vertec.com/kb/config-set-xml/#new-system-settings>.

7.11 vtcapp.executefile() unterstützt Commandline Parameter

Linie: Expert | Module: Leistung & CRM | Version: 6.4.0.10

Betriebsart: Cloud Abo / On-Premises | Apps: Desktop

Die Python Methode `vtcapp.executefile()` unterstützt nun auch Commandline Parameter. Dafür können optional Argumente übergeben werden. Der bisherige, einfache Fall funktioniert weiterhin.

Das Parameterparsing ist wie folgt:

- Im `command` findet kein Parsing statt. Leerzeichen werden als Teil des Pfades interpretiert.
- Im `argumentString` findet das Parsing statt, dort trennen Leerzeichen die Argumente, ausser sie befinden sich innerhalb von doppelten Anführungszeichen "".

Beispiel:

```
vtcapp.executefile('notepad++.exe', r'-lpython "C:\Dokumente\Mein Script.py"')
```

7.12 Verbesserung der Anzeigebedingung für Container im BI Baum

Linie: Standard, Expert | Module: Leistung & CRM, Business Intelligence | Version: 6.4.0.14

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Die Kriterien, wer welche Ordner im BI Baum sieht, wurden angepasst. Welche Objekte und Ordner in der BI Baumansicht zur Verfügung stehen, werden wie folgt ermittelt:

Zuerst wird ermittelt, ob das **BI Daten Recht** auf der Klasse vorhanden ist:

- Ein Bearbeiter ohne generelles BI Daten Recht auf der Inhaltsklasse und ohne Projektleiter-, Projekt-Administrator oder Administrator-Recht, sieht keinen Container (z.B. "Alle Adressen") im BI Baum.
- Insbesondere impliziert das **Adressenadministrator-Recht** kein BI Daten Recht. Ein Bearbeiter, welcher Adressenadministrator-Recht hat, hat dadurch nicht BI Daten Recht auf Adresseinträgen. Das bedeutet: um sinnvolle BI Daten zu sehen, sollte der Benutzer neben dem BI Daten Recht auf Adresseintrag auch noch **Projekt-Supervisoren-Rechte** oder ähnliches haben.
- Sobald ein Bearbeiter das **Projektleiter-Recht** hat, sieht er alle Projekt- und Phasen Container.
- Ein Bearbeiter mit **Teamleiter-Recht** sieht alle Bearbeiter-, Projekt- und Phasen-Container.
- Ein Bearbeiter mit dem **ProjektAdmin-Recht** sieht alle Bearbeiter-, Projekt-, Phasen-, Adresseintrag- und Tätigkeit-Container.
- Ein **Administrator** sieht alle Container, für die eine BI Ansicht definiert und aktiv ist.

In einem zweiten Schritt werden alle explizit vergebenen Rechte auf dem Benutzer abgearbeitet. Hier interessieren nur die Rechte vom Typ **BI Daten** auf den übereinstimmenden Klassen (Vererbung wird berücksichtigt, d.h. ein Recht auf Adresseintrag hat Auswirkungen auf eine Abfrage auf Firma).

- Wenn das Recht erlaubt wird, dann wird das BI Daten Recht erteilt. Es spielt keine Rolle, was man als `Feldnamen` (Kennzahl) oder als `Expression` eingetragen hat.
- Wenn das Recht verweigert wird müssen `Feldname` und `Expression` leer sein (da diese Angaben auf Objektebene wirken, die grundsätzliche Berechtigung für die Anzeige im Baum aber auf Klassenebene entschieden werden muss). Wird das Recht verweigert und es sind `Feldname` oder `Expression` gesetzt, dann wird diese Berechtigung für die Anzeige ignoriert.

Diese Ermittlung ist wie üblich bei Rechten aufeinander aufbauend. Das heisst, wenn grundsätzlich das BI Daten Recht auf Einträge vergeben wird, dann aber ein Recht die BI Daten der Projekte verweigert, hat der Benutzer nachher kein Recht auf Projekte.

Wichtig: Bei der Beschreibung hier handelt es sich nur um die Ermittlung, was überhaupt im Baum angezeigt und vom Benutzer ausgewählt werden kann, und nicht um die eigentliche "Berechtigung" auf den Daten.

7.13 Python Stub Files für builtin Modules

Linie: Standard, Expert | Module: Leistung & CRM, Fremdkosten | Version: 6.4.0.15

Betriebsart: On-Premises | Apps: Full-featured

Code Editoren (z.B. VSCode) haben mittlerweile gute static Analysis Fähigkeiten, welche es erlauben, viele Fehler bereits vor der Ausführung zu erkennen. Ein Problem sind dabei aber die eingebauten Python Module, die auf dem Filesystem nicht zur Verfügung stehen. Damit diese für den Type Checker verfügbar sind, liefern wir neu Stub Files der eingebauten Python Module `vtcapp`, `vtcextensions` und `ziputils` mit.

Die Stub Files befinden sich im Unterordner `PythonStubs` des Vertec Installationsverzeichnis oder können in der Online Knowledge Base unter <https://www.vertec.com/kb/pythonstubfiles/> heruntergeladen werden.

7.14 Checkbox für Stichwort-Ordner und einfache Ordner in Listen

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.19

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Neu können Stichwort Zuordnungen direkt in Listen dargestellt und bearbeitet werden. Verarbeitet werden können Stichwort-Ordner oder normale Ordnerstrukturen, welche nicht als IX-Ordner markiert sind. In der Listenspalte werden die entsprechenden Subordner (die Stichwörter) dargestellt und die Einträge können per Klick in die Checkbox zugeordnet bzw. entfernt werden. Dies geschieht mittels Renderer `rndKeywordFolder`.

7.15 LinkRollen in GetLinks sind via Eintrag Id identifizierbar

Linie: Expert | Module: Leistung & CRM | Version: 6.4.0.19

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Die GetLinks Operatoren und Methoden wurden so erweitert, dass die zu verwendende LinkRolle nicht nur durch ihre Bezeichnung (wie bisher), sondern auch über die **Eintrag Id** des Linktyps identifiziert werden kann.

Als Rolle wird somit die Eintrag Id des Linktyps, gefolgt von `_A` für die Rolle A oder `_B` für die Rolle B, je nachdem, welche Seite das Target im Linktyp hat, angegeben:

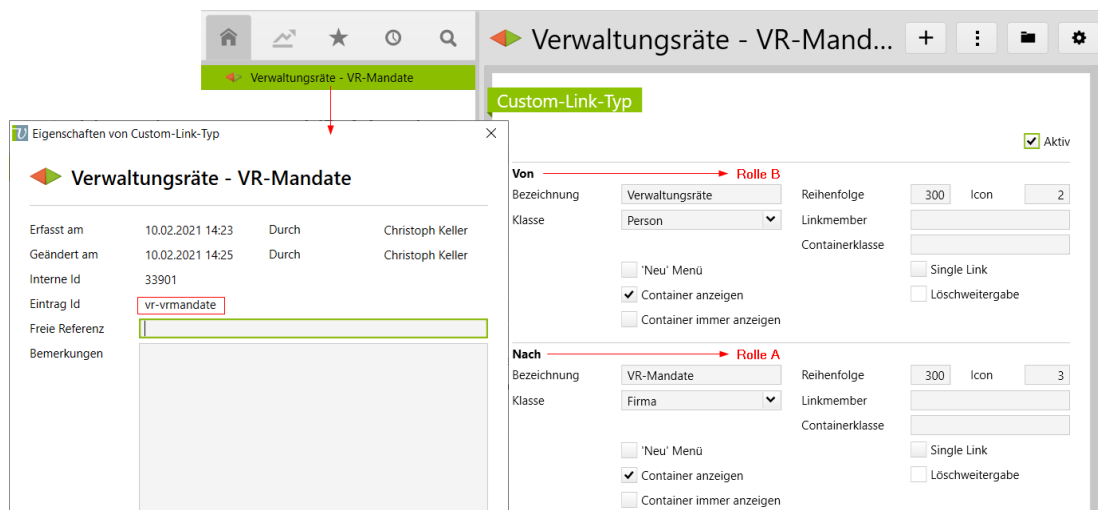


Abbildung 36: Neue Opportunität anlegen aus Firma

In Python sieht das wie folgt aus:

Objekt: Person	<code>person = argobject</code>
Target: Firma	<code>firma = vtcapp.getobjectbyid(2995)</code> <code>person.linkto(firma, "vr-vrmandate_A")</code>
Objekt: Person	<code>person = argobject</code>
Target: Firma	<code>firma = vtcapp.getobjectbyid(2995)</code> <code>firma.linkto(person, "vr-vrmandate_B")</code>

7.16 OCL Translate Operator

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.20

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Analog zu Python gibt es nun auch einen Translate Operator für OCL. Dieser ermöglicht es, in einer OCL Expression einen String-Wert via `.translate` in die aktuelle Vertec Oberflächensprache zu übersetzen.

Beispiel:

`if projekte->size > 0 then 'Client'.translate else 'Address'.translate endif` auf einer Adressliste zeigt in der Spalte folgendes Resultat bei Deutsch:

Betreute Adressen						
Name / Firma	Adresse	PLZ	Ort	Telefon	Status	
Gallfor AG	Sebastianweg 1	5400	Baden	052 544 58 98	Kunde	
Huber Thomas	Zur Linde 12	8005	Zürich	044 231 84 96	Adresse	
Hutform AG, Keller Verena	Zwielstrasse 76a	8005	Zürich	044 555 66 77	Kunde	
Hutform AG, Meier Lukas	Zwielstrasse 76a	8005	Zürich	044 555 66 77	Adresse	
Meyer Blatter Rechtsanwälte	Rennweg 34	8001	Zürich		Kunde	
Monnier Pierre	Gallastrasse 4	1200	Genève	021 483 59 78	Adresse	
Netallnet AG	Kellerstrasse 5	6004	Luzern	041 750 89 89	Kunde	
NewsFlash AG	Oberlunig 13	5001	Aarau	052 222 222 11	Kunde	
SBB		3000	Bern		Adresse	
Westlox AG	Baarerstrasse 31	6302	Zug		Kunde	
Winet Petra	Rudolfweg 45	6002	Luzern	041 862 71 65	Adresse	
Zolak AG, Zbinden Urs	Technoparkstrasse 1	8005	Zürich		Adresse	

Abbildung 37: Beispiel des `.translate` Operators

7.17 Zugriff auf Container und Parent-Objekt in OCL Expressions in Listen

Linie: Expert | Module: Leistung & CRM | Version: 6.4.0.21

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Neu kann man in OCL Expressions in Listen über `varContainer` auf den Container (Ordner oder LinkContainer) und über `varParent` auf das Objekt, dem der Container gehört (Parent-Objekt des Link-Containers), zugreifen. Neben den Spalten-Expressions in der Liste haben auch die `ListExpression` und `GhostRowListExpression` eines Controls Zugriff auf diese Context-Variablen.

7.18 Auswahldialoge aus Python aufrufbar

Linie: Expert | Module: Leistung & CRM | Version: 6.4.0.21

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Die verschiedenen Such- und Auswahldialoge für Adressen, Projekte, Phasen und den Baum können nun direkt aus Python heraus aufgerufen werden. Dafür gibt es folgende neuen Methoden auf dem `vtcapp` Modul:

- `selectaddress()`
- `selectphase()`
- `selectproject()`
- `selectobjectintree()`

Die Methoden unterstützen die folgenden optionalen Keyword Argumente:

- **`selectaddress(string title, string classname, string filter)`**
 - title: Dialog Titel (Default: Adressen auswählen)
 - classname: Adressklasse, in der gesucht wird, z.B. `Firma`. (Default: Adresseintrag)
 - filter: Zusätzliche SQL Filter Bedingung (Default: leer), z.B. nur auf aktive Objekte
Der Suchdialog für Adressen verwendet die Standard-Suchdefinition gemäss Systemeinstellungen.
- **`selectphase(string title, string filter)`**
 - title: Dialog Titel (Default: Phasen auswählen)
 - filter: Zusätzliche SQL Filter Bedingung (Default: leer), z.B. nur auf aktive Objekte
- **`selectproject(string title, string filter)`**
 - title: Dialog Titel (Default: Projekte auswählen)
 - filter: Zusätzliche SQL Filter Bedingung (Default: leer), z.B. nur auf aktive Objekte
- **`selectobjectintree(string title, list entrylist, string browsefilter, string selectfilter)`**
 - title: Dialog Titel (Default: Objekte auswählen)
 - entrylist: Liste von Vertec Objekten, welche im Dialog als Root Elemente dargestellt werden.
 - browsefilter: Komma-delimited Filter für Objekttypen (z.B. `Ordner` , `Expressionordner`)
 - selectfilter: Der Selectfilter bestimmt, auf welchen Einträgen (Klassen und deren Subklassen) man den Dialog bestätigen kann und somit "auswählt", auch Komma-delimited.

Beispielcode:

```
adresse= vtcapp.selectaddress(title="Adressauswahl", classname="Kontakt",
filter="vorname like '%S%'")

projekt = vtcapp.selectproject(title="Projektauswahl", filter="code like '%S%'")

phase= vtcapp.selectphase(title="Phasenauswahl", filter="code like '%S%'")

Kontakte = argobject.subordner
Kontaktordner = list(Kontakte)
Kontaktordner.append(argobject)
selected=vtcapp.selectobjectintree("Objekt wählen",Kontaktordner, "Ordner,
Kontakt", selectfilter="Kontakt")
```

7.19 Python Executereport Methode mit Rückgabe von Aktivität

Linie: Expert | Module: Leistung & CRM | Version: 6.4.0.22

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Es gibt eine neue Python Methode `executereport2()`, welche als Rückgabewert den Bericht sowie die Aktivität zurückgibt.

```
executereport2(rootObj, optarg, berichtObj, [saveAs, showDialog, doPrint, showApp]): (reportDoc, activity)
```

Der Rückgabewert der Methode ist ein Tuple (reportdoc, activity) wobei

- **reportdoc** der Inhalt des resultierenden Report Dokuments ist (Byte-String).
- **activity** das angelegte Aktivitäts-Objekt ist oder None

Der Aufruf der Methode erfolgt mit Zuordnung des Rückgabewertes zu 2 Variablen:

```
reportdoc, activity = vtcapp.executereport2(argobject, None, reportdef)
```

Weitere Informationen Weitere Informationen finden Sie im Artikel Vertec Python Funktionen in der Online Knowledge Base unter www.vertec.com/kb/pythonfunktionen/.

7.20 Python Funktion für Zahlungsabgleich

Linie: Expert | Module: Leistung & CRM | Version: 6.4.0.22

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Die neue Python Funktion `vtcapp.syncpayments()` führt einen globalen Zahlungsabgleich durch. Falls keine Debitorenschnittstelle mit Zahlungsabgleich installiert ist, meldet die Funktion einen Fehler.

Die Funktion kann beispielsweise als geplante Aufgabe registriert werden.

7.21 Script Editor öffnet sich nur mit Administratorrechten

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.23

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Der Script Editor wird nur angezeigt, wenn der eingeloggte Benutzer Administratorrechte hat. Ansonsten erscheint eine Fehlermeldung.

7.22 Python Update

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.5.0.1

Betriebsart: On-Premises | Apps: Full-featured

Mit Vertec 6.5 wird die in Vertec verwendete Python Version angepasst von der Version 2.7.16 auf Version 2.7.18. Eine ausführliche Beschreibung der Änderungen findet sich in den entsprechenden [Release Notes](#).

7.23 Aktivitätstypen sind Multilanguage-fähig und aktiv

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.5.0.1

Betriebsart: Cloud Abo / On-Premises | Apps: Alle

Aktivitätstypen haben weiterhin das normale Textfeld **Bezeichnung**, welches auf der Oberfläche jedoch neu als **Code** geschrieben wird. Aus Gründen der Rückwärtskompatibilität bleibt der Feldname **Bezeichnung**.

Neu befindet sich zu Beginn das Textfeld **Code** sowie die Checkbox **Aktiv**. Als Standard String-Darstellung der Aktivitätstypen wird das neue Multi-Language String Feld **Bezeichnung** verwendet.

Bestehende Aktivitätstypen erhalten im Feld **Code** den Inhalt der Bezeichnung und sind **Aktiv**.

7.24 Übersetzung von Konfigurationsdaten in der Businesslogik

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.9

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Um die Standard-Konfigurationsobjekte in allen Sprachen zur Verfügung stellen zu können, übersetzt Vertec die String-Darstellung automatisch in die ausgewählte Oberflächensprache.

Da Vertec dies bisher zu Laufzeit auf der Oberfläche gemacht hat, führte das teilweise zu Problemen mit Nutzdaten, wenn diese zufälligerweise gleich hießen. So wurde beispielsweise **PM** im Baum in einem deutschsprachigen Vertec immer als **PL** angezeigt (weil PM vom Übersetzungsmechanismus als "project manager" interpretiert und übersetzt wird).

Aus diesem Grund wurde die Übersetzung der Konfigurations-Objekte in die Businesslogik verschoben. Es werden genau diejenigen String-Representations von Objekten übersetzt, die eine Eintrag Id haben. Die String-Representations aller anderen Objekte werden nicht übersetzt. Möchte man in einer Liste die übersetzte String-Representation eines Objektes anzeigen, so geht dies über die Expression **self**.

7.25 Unterstützung von mehrfachen Klasseneinstellungen

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.17

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Pro Klasse können mehrere Klasseneinstellungen parallel geführt werden. Dies ist vor allem für Plug-ins relevant, um Konflikte zwischen importierten Klasshareposeneinstellungen und bereits bestehenden kundenspezifischen Anpassungen zu vermeiden. Die Klasseneinstellungen werden wie folgt berücksichtigt:

- Die Rangfolge bei verschiedenen anwendbaren Klasseneinstellungen bestimmt sich durch Sortierung nach der **Internen Id**. Höhere Ids entsprechen neueren Objekten.
- Bei folgenden Eigenschaften wird der Wert der neuesten Klasseneinstellungen verwendet:
 - Dokumentenpfad
 - Icon Index
 - Anzeige Klassenname
 - Standardanzeige
 - Sortierung im Baum
- Bei den (Standard-)Listeneinstellungen werden immer die ältesten Klasseneinstellungen angewandt. Der Button **Als Standard speichern** im Listeneinstellungs-Editor schreibt deshalb immer in die ältesten Klasseneinstellungen.
- Gibt es auf mehreren Klasseneinstellungen **Customizing von Seiten**, werden alle zu einer Page-Definition kombiniert. Sind dabei gleiche Elemente definiert, wird jeweils das der neuesten Klasseneinstellungen angewandt.

7.26 Berechtigungen auf Abwesenheiten eingeschränkt

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.22

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Benutzer, die zur Benutzergruppe **Standardbenutzer** gehören, können nur noch die eigenen Abwesenheiten einsehen. Die Benutzergruppe **Projekt-Supervisoren** hat hingegen Lese- und Schreibrechte auf allen Abwesenheiten anderer Benutzer. Gruppenabwesenheiten können von allen Benutzergruppen gelesen, aber nicht modifiziert werden.

7.27 String-Darstellung von Projekteintrag-Typen enthält übersetzte Bezeichnung

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.22

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Die String-Darstellung von Leistungen, Spesen und Auslagen enthält an Stelle des **Codes** die übersetzte Bezeichnung der **Tätigkeit**, des **Spesentyps** oder des **Auslagentyps**.

7.28 String-Darstellung von Abwesenheit enthält übersetzte Bezeichnung

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.22

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Die String-Darstellung von Abwesenheiten enthält die **Bezeichnung** anstelle des **Codes**.

8 Schnittstellen

8.1 TAPI Extension funktioniert auch ohne CallerIdName

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.9

Betriebsart: On-Premises | Apps: Desktop

Die **Vertec TAPI Schnittstelle** behandelt eingehende Anrufe via TAPI auch mit Providern, welche kein **CallerIdName** Feld liefern, und kann sie in Vertec anzeigen.

8.2 Übergabe von Aktivität in der SharePoint Online DMS Schnittstelle

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.10

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Die Methode `StoreDocument(self, vertecFilePath, content, aktivitaet)` übergibt die Aktivität als Objekt. Damit können im SharePoint die entsprechenden Metadaten geschrieben werden.

Damit Metadaten geschrieben werden, muss die verwendete DMS-Schnittstelle wie folgt überschrieben bzw. ergänzt werden. Das Beispiel bezieht sich auf die Standard **Vertec SharePoint DMS Schnittstelle**.

Es wird ein Script erstellt mit folgendem Text:

```
#
import DmsSharePoint

class MyDmsSharePointExtension(DmsSharePoint.DmsSharePointExtension):
    __doc__ = "Erweiterung DMS-Extension mit Metadaten"
    # pass

    # Override this method in a subclass to be able to define meta data during
    document upload.

    # This method is called by Vertec for each document that is uploaded. If the
    returned

    # dictionary contains entries, Vertec interprets the keys as site column name
    and the values

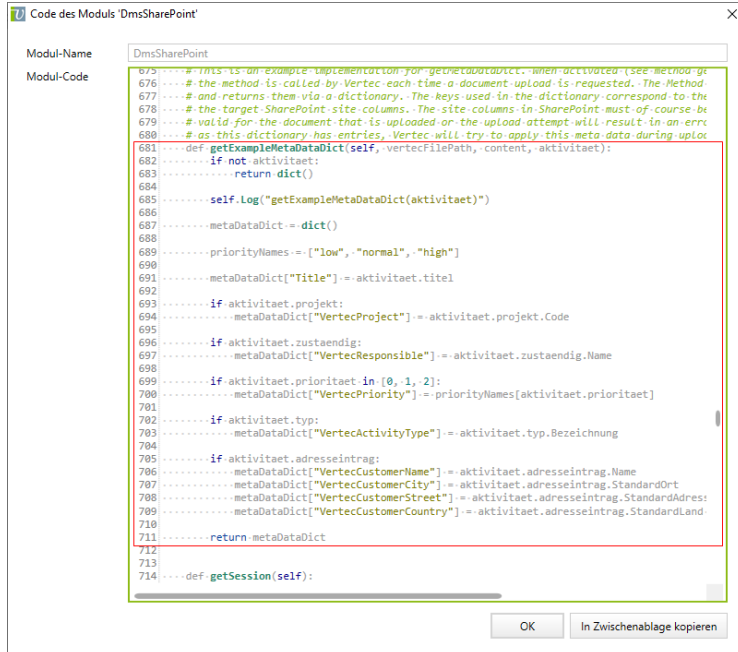
    # as site column value and writes meta data accordingly during document upload.
    def getMetaDataDict(self, vertecFilePath, content, aktivitaet):
        self.Log("getMetaDataDict(...) + (" with aktivitaet" if aktivitaet else "
without aktivitaet")

        # override the extension and uncomment this line to activate the meta data
        example

        return self.getExampleMetaDataDict(vertecFilePath, content, aktivitaet)

        # empty dict => Vertec does not try to write meta data to SharePoint
        return dict()
```

Damit wird die Methode `getExampleMetaDataDict` aktiviert, die sich in der Standardschnittstelle befindet und Beispiel-Metadaten zurückgibt. Der entsprechende Code befindet sich in der Standardschnittstelle:



```

671 ... # this is an example implementation for getExampleMetaDataDict. When activated (see Method get
672 ... # the method is called by Vertec each time a document upload is requested. The Method
673 ... # and returns them via a dictionary. The keys used in the dictionary correspond to the
674 ... # the target SharePoint site columns. The site columns in SharePoint must of course be
675 ... # valid for the document that is uploaded or the upload attempt will result in an error
676 ... # as this dictionary has entries, Vertec will try to apply this meta data during upload
677 ...
678 ... def getExampleMetaDataDict(self, vertecFilePath, content, aktivitaet):
679 ...     if not aktivitaet:
680 ...         return dict()
681 ...     self.Log("getExampleMetaDataDict(aktivitaet)")
682 ...     metaDataDict = dict()
683 ...     priorityNames = ["low", "normal", "high"]
684 ...     metaDataDict["Title"] = aktivitaet.titel
685 ...     if aktivitaet.projekt:
686 ...         metaDataDict["VertecProject"] = aktivitaet.projekt.Code
687 ...     if aktivitaet.zustandig:
688 ...         metaDataDict["VertecResponsible"] = aktivitaet.zustandig.Name
689 ...     if aktivitaet.prioritaet in [0, 1, 2]:
690 ...         metaDataDict["VertecPriority"] = priorityNames[aktivitaet.prioritaet]
691 ...     if aktivitaet.typ:
692 ...         metaDataDict["VertecActivityType"] = aktivitaet.typ.Bezeichnung
693 ...     if aktivitaet.adresseintrag:
694 ...         metaDataDict["VertecCustomerName"] = aktivitaet.adresseintrag.Name
695 ...         metaDataDict["VertecCustomerCity"] = aktivitaet.adresseintrag.StandardOrt
696 ...         metaDataDict["VertecCustomerStreet"] = aktivitaet.adresseintrag.StandardAdresse
697 ...         metaDataDict["VertecCustomerCountry"] = aktivitaet.adresseintrag.StandardLand
698 ...     return metaDataDict
699 ...
700 ... def getSession(self):

```

Abbildung 38: Code in der Standardschnittstelle

Die übergebenen Felder müssen zum Metadaten-Schema Ihres SharePoint passen. Diese Methode wird deshalb im oben erzeugten Script ebenfalls überschieben:

```
def getExampleMetaDataDict(vertecFilePath, content, aktivitaet):
```

oder statt der folgenden Zeile

```
return self.getExampleMetaDataDict(vertecFilePath, content, aktivitaet)
```

direkt die entsprechenden Werte zurückgegeben.

Dann wird ein Extension-Eintrag angelegt mit `<Name des Scripts>.<Name der Methode>`.

Die Standard SharePoint Online DMS Schnittstelle wird daraufhin deaktiviert und die hier neu erzeugte installiert.

8.3 Cloudfähige Abacus Web Debitoren Schnittstelle

Linie: Standard, Expert | Module: Leistung & CRM, Business Intelligence | Version: 6.4.0.15

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Die **Abacus Web Debitoren Schnittstelle** wurde komplett neu gemacht und ist cloudfähig.

Weitere Informationen Weitere Informationen zur neuen Schnittstelle finden Sie im Artikel Abacus Web Debitoren Schnittstelle in unserer Online Knowledge Base unter www.vertec.com/kb/abacuswebdebitoren.

8.4 Zeep auf Whitelist für Restrict Scripting

Linie: Standard, Expert | Module: Leistung & CRM, Business Intelligence | Version: 6.4.0.15

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Die für die Abacus Web Debitoren Schnittstelle verwendete Webservice Library **Zeep** wurde auf die whitelist für das **Restricted Scripting** gesetzt.

8.5 Abacus Web Kreditoren Schnittstelle ist cloudfähig

Linie: Standard, Expert | Module: Leistung & CRM, Fremdkosten | Version: 6.4.0.17

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Die **Abacus Web Kreditoren Schnittstelle** wurde komplett neu gemacht und ist cloudfähig.

Weitere Informationen Weitere Informationen zur neuen Schnittstelle finden Sie im Artikel Abacus Web Kreditoren Schnittstelle in unserer Online Knowledge Base unter www.vertec.com/kb/abacus-web-kreditoren-schnittstelle/.

8.6 Unterstützung für ZUGFeRD 2.1

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.16

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Vertec unterstützt nun den **ZUGFeRD 2.1 Standard**, welcher die Unterschiede zwischen ZUGFeRD 2.0 und Factur-X sowie dem Profil XRechnung behebt.

Die Abwärtskompatibilität ist gewährleistet. Bereits angepasste Reports funktionieren nach wie vor und erzeugen wie bisher ein ZUGFeRD PDF mit Version 2.0.1 bzw. mit dem angegebenen XMP-Schema. Möchte man für bereits angepasste Reports den neuen ZUGFeRD Standard verwenden, muss der Code entsprechend angepasst werden. Der Vertec Standard-Bericht **Rechnung mit Leistungsliste** wird mit dem neuen Standard ausgeliefert.

Weitere Informationen Weitere Informationen finden Sie im Artikel Rechnungen nach ZUGFeRD Standard (X-Rechnung) in unserer Online Knowledge Base unter www.vertec.com/kb/zugferd/.

8.7 Bereinigung Adressenbehandlung bei cloudfähigen Schnittstellen

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.17

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Die Behandlung von **Rechnungsadressen** und **Personenkonti** bei listenbasierten (cloudfähigen) Buchhaltungsschnittstellen wurde angepasst:

Es gibt eine neue (derived) Eigenschaft **buchungsadresse** auf der Rechnung, welche immer den Adresseintrag liefert, der beim Buchen verwendet werden soll. Die Eigenschaft **rechnung.personenkonto** liefert dann immer das Personenkonto von **rechnung.buchungsadresse**. Diese entspricht dabei **rechnung.rechnungsadresse**, ausser wenn dies ein Kontakt ist. Dann wird stattdessen die Firma des Kontakts verwendet. Falls leer, wird eine neue Adresse geschrieben und die Personenkonto-Nummer auf **rechnung.buchungsadresse** eingetragen.

Bei Verwendung von **rechnung.buchungsadresse** wird das Festlegen eines Personenkonto auf Projekt nicht mehr unterstützt.

Betrifft folgende Schnittstellen:

- Abacus Web Debitoren Schnittstelle
- BMD Debitoren Schnittstelle
- DATEV Debitoren Schnittstelle
- Run my Accounts Debitoren Schnittstelle

8.8 Einlesen von Zahlungsdaten für die cloudfähige DATEV Debitoren Schnittstelle

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.19
Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Die bestehende Funktion **DATEV Zahlungsimport** wurde durch **Zahlungsabgleich** ersetzt.

Im Menü **Einstellungen** kann der **Zahlungsabgleich** aufgerufen werden. Es erscheint ein File-Auswahldialog, in dem ein Zahlungsdaten-File ausgewählt werden kann. Eine Zahlung darf nicht doppelt importiert werden.

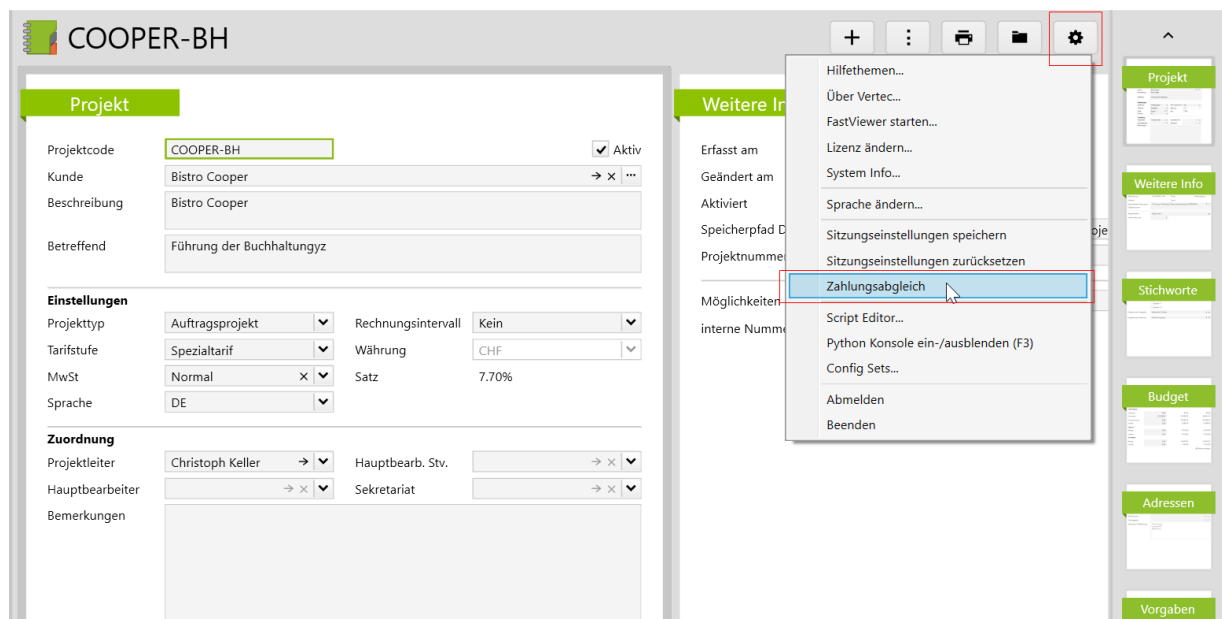


Abbildung 39: Zahlungsabgleich aufrufen

Hinweis Falls Sie eine frühere DATEV-Schnittstelle mit verwenden: Die bestehende Funktion DATEV Zahlungsimport muss manuell entfernt werden, indem das Skript **DATEV Zahlungsimport** gelöscht wird.

8.9 Topal Solutions Kreditoren Schnittstelle mit QR-Rechnung

Linie: Standard, Expert | Module: Leistung & CRM, Fremdkosten | Version: 6.4.0.19
Betriebsart: On-Premises | Apps: Full-featured

Die **Topal Solutions Kreditoren Schnittstelle** wurde so erweitert, dass ein in Vertec via QR-Code erfasster Kreditor über die Topal Solutions Kreditoren Schnittstelle gebucht werden kann.

- Beim Buchen wird geprüft, ob das Feld **Kreditor.ESRTeilnehmer** (Zahlungskonto) eine Angabe enthält.
- In Topal wird geprüft, ob für dieses Zahlungskonto bereits eine **Transaktionsmethode** (technisch **Paymentmethod**) auf der Kreditor-Adresse vorhanden ist. Falls ja, wird dem Topal Kreditor Objekt diese Zahlmethode hinzugefügt. Falls nicht, wird eine neue Transaktionsmethode angelegt und die Kontonummer eingesetzt.
- Zahlungsart ist **QR**, falls es eine QR-IBAN ist, **IBAN** falls es eine IBAN ist und **ESR** in allen anderen Fällen.

8.10 Unterscheidung von ESR und QR Debitor Rechnungen

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.21

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Mit Abacus 2020 wurde auf dem Beleg ein Feld **Referenztyp** eingeführt, welches die Referenztypen **ESR** und **QR** beim Einlesen von Belegen unterscheidet. Damit eine korrekte Zuweisung möglich ist, liefert Vertec beim Buchen den Referenztyp mit. Handelt es sich bei der verwendeten IBAN Nummer um eine QR-IBAN, dann wird als **PaymentReferenceType=QR** mitgegeben, ansonsten **PaymentReferenceType=ESR**.

8.11 Unterstützung von mehr als 200 Sites in SharePoint Schnittstelle

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.21

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Es können SharePoint Instanzen verwendet werden, die mehr als 200 Sites unterstützen.

8.12 Unterstützung von ESR Rechnungen in der Abacus Web Kreditoren Schnittstelle

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.21

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Bisher übertrug die **Abacus Web Kreditoren Schnittstelle** das Zahlungskonto und die Referenzangabe nur, wenn es sich beim Zahlungskonto um eine QR-IBAN handelte. Es können nun auch ESR Rechnungen in Abacus gebucht werden. Die Nummer extern (**kreditor.nummer**) wird in Abacus in das Feld **Belegreferenz-Nr.** geschrieben.

8.13 SharePoint Online DMS Schnittstelle Unicode-fähig

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.21

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Die Schnittstelle verarbeitet jegliche Unicode Zeichen in Dateinamen, Ordernamen, Metadaten-Feldern und Fehlermeldungen korrekt, welche von SharePoint selbst erzeugt werden. Sie verwendet intern den Python Type **Unicode** und konvertiert nur gegen "extern", falls notwendig.

8.14 Cloudfähige DATEV Kreditoren Schnittstelle

Linie: Expert | Module: Leistung & CRM, Fremdkosten | Version: 6.4.0.23

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Die **DATEV Kreditoren Schnittstelle** ist jetzt cloudfähig. Im Ordner **Extensions** erscheinen nur noch die cloudfähigen DATEV Schnittstellen, sofern zuvor keine installiert war. Wenn bereits eine DATEV Schnittstelle installiert ist, bleibt diese bestehen.

8.15 Setzen des EZ Verfahrens beim Buchen

Linie: Expert | Module: Leistung & CRM | Version: 6.4.0.23

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Beim Referenztyp **QR** und mehreren in Abacus definierten QR Verfahren muss zusätzlich auf der Abacus Rechnung das zur IBAN passende QR Verfahren angegeben werden. Daher liefert Vertec das EZ Verfahren (**PaymentOrderProcedure**) aufgrund des verwendeten Vertec Zahlungstyps mit. Die Zuordnung erfolgt auf folgender Basis:

- In Abacus muss ein passendes **EZ Verfahren** definiert sein, dessen Einzahlungskonto dem Zahlungskonto (VESR Teilnehmernummer bei ESR, IBAN bei QR) entsprechen muss.

- Der **Code** des Zahlungstyps in Vertec muss der Nummer des EZ Verfahrens in Abacus entsprechen. Falls der Code nicht numerisch ist, wird kein EZ Verfahren übergeben.
- Vertec eruiert den Zahlungstyp aufgrund des Zahlungskontos. Ist eine IBAN definiert, suchen wir den Zahlungstypen aufgrund dieser IBAN. Wird ein Zahlungstyp gefunden, wird in Abacus das EZ Verfahren gesetzt.

8.16 Verbesserung Sage 200 Zahlungsabgleich

Linie: Expert | Module: Leistung & CRM | Version: 6.4.0.24

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Die Belegnummern, die beim Zahlungsabgleich verwendet wurden, waren nicht eindeutig. Dadurch konnten nicht mehrere Zahlungen mit gleichem Datum und Betrag importiert werden. Ab Version 6.4.0.24 können Zahlungen mit gleichem Datum, Text und Betrag via **Sage 200 Zahlungsabgleich** korrekt importiert werden.

9 Performance

9.1 Performance-Verbesserung in Listen ohne aktiv Member

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.5

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Listen werden aus, ob ein Objekt **aktiv** oder **inaktiv** ist, um dies in der ersten Spalte entsprechend mit dem grauen Balken anzuzeigen. Bei Objekten, welche gar kein **aktiv** Member haben, wurde diese Überprüfung ausgeschaltet.

9.2 Performanceoptimierung beim Starten von Vertec

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.9

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Beim Starten von Vertec werden Einstellungsobjekte wie **Rechttyp**, **Waehrung**, **MWSTTyp**, **Zahlungstyp**, **KommMittelTyp** und **AktivitaetsTyp** nicht mehr einzeln, sondern alle Instanzen auf einmal vorgeladen.

9.3 Shadow Attribute bei der Rechnung entfernt

Linie: Standard, Expert | Module: Leistung & CRM, Budget & Teilprojekt | Version: 6.4.0.17

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Die Shadow Attribute, welche dazu dienten, Werte in der Datenbank zu speichern, die eigentlich derived sind, führten dazu, dass beim Lesezugriff auf gewisse Werte ein Schreibzugriff in die Datenbank erfolgte. Auf der Rechnung gab es zwei dieser Shadow Attribute, was zu Performance-Einbussen beim Laden langer Rechnungslisten führte. Deshalb werden die beiden Shadow Attribute **Rechnung.SqlRabattBetrag** und **Rechnung.SqlValutadatum** nicht mehr geschrieben und verwendet.

9.4 Performanteres Laden der Budget Seite auf Projekt

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.17

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Das Anzeigen der Budget-Details wurde durch Preloading der Planwerte optimiert.

10 Security

10.1 Restricted Session Prozess

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.12

Betriebsart: Cloud Abo / On-Premises | Apps: Full-featured

Der **Vertec.Cloudserver** kann so konfiguriert werden, dass **Session Prozesse** mit eingeschränkten Möglichkeiten gestartet werden:

- Der Vertec Session Prozess läuft als «Low Integrity» Prozess.
- Der Vertec Session Prozess darf keine weiteren Subprozesse starten.

Das geschieht durch eine Einstellung in der `[CloudServer]` Section im Vertec.ini-File:

```
Restrict Session Process=True
```

Ist diese Einstellung gesetzt, dann muss darauf geachtet werden, dass bei Verwendung von Firebird der `ServerName` Parameter in der Datenbank-Section einen Netzwerkpfad enthält, keinen Dateipfad. Als Beispiel:

- Netzwerkpfad (gültig): `ServerName = MYSERVER:C:\Programme\Vertec\DB\VertecDemo.fdb`
- Dateipfad (ungültig): `ServerName = C:\Programme\Vertec\DB\VertecDemo.fdb`

Die entsprechenden Logfiles (**Vertec.Session.log** und **Vertec.TaskRunner.log**) sind unter `C:\WINDOWS\system32\config\systemprofile\AppData\LocalLow` zu finden.

11 Technik

11.1 Unicode Unterstützung

Bis und mit Vertec 6.4 war die Datenhaltung von Texten in Vertec (z.B. Adressen, Leistungstexte, Bemerkungen) auf ein 1 Byte Encoding beschränkt, nämlich Windows-1252 oder ANSI (die beiden Begriffe meinen das gleiche, wir verwenden im folgenden ANSI als Begriff). Vertec konnte also genau die in **ANSI** definierten Zeichen verarbeiten, siehe z.B. <https://de.wikipedia.org/wiki/Windows-1252>. Andere Zeichen wurden durch `?` ersetzt.

Mit diesem 250 Zeichen umfassenden ANSI-Zeichensatz können Englisch und alle westeuropäischen Sprachen abgebildet werden, was in den meisten Fällen für die Anwendung von Vertec genügt. Immer mehr Kunden mit Geschäftsbeziehungen nach beispielsweise Osteuropa wünschten sich jedoch die Möglichkeit, auch diese Adressen korrekt in Vertec abbilden zu können. Aus diesem Grund erfolgt mit Vertec 6.5 die **Umstellung auf Unicode**.

Ab Vertec 6.5 werden Texte in Vertec in Unicode abgespeichert. Dementsprechend unterstützt Vertec alle rund 65'000 Zeichen der **BMP** ("Basic Multilingual Plane") und damit die Zeichen aller relevanten Sprachen, einschliesslich sämtlicher asiatischer Sprachen. Zeichen, die ausserhalb der BMP liegen und in Vertec eingetragen werden, werden (wie bereits bisher bei nicht unterstützten Zeichen) durch ein `?` ersetzt, was nur bei einigen wenigen Emojis der Fall sein dürfte.

Bei einem Datenbank Konvert von 6.4 auf 6.5 werden alle Texte in Vertec von **ANSI** auf **Unicode** konvertiert. Auf der Oberfläche bleibt diese Änderung unsichtbar. Die Vertec Datenbank wird dadurch in der Regel 20%-30% grösser, was jedoch kaum negativ auffallen dürfte.

Rückwärtskompatibilität

Die Erweiterung auf Unicode ist vollständig abwärtskompatibel, hingegen nicht unbedingt vorwärtskompatibel. Dies bedeutet, dass sich eine bestehende Vertec Installation bei gleicher Nutzung auch gleich verhält, sobald jedoch Unicode Zeichen in Vertec verarbeitet werden (bewusst oder unbewusst, z.B. über das Reinkopieren eines Textes), dies nicht unbedingt der Fall sein wird.

Im Folgenden werden einige Punkte aufgeführt, die bei Kundeninstallationen allenfalls Aufmerksamkeit benötigen. Die Risiken konzentrieren sich dabei auf kundenspezifische Scripts (vor allem Python) und Schnittstellen.

Python und Unicode

Ein Vertec String-Member (wie z.B. der Text einer Leistung) war bisher vom Typ `str`, in 6.5 vom Typ `unicode`. Das stellt sicher, dass bisheriger Python Code der Strings kopiert, zusammensetzt, ergänzt etc. immer noch gleich funktioniert. Das `str`-Modul wurde von uns zusätzlich auf Unicode «umgebogen», weil eine Analyse von bestehendem Python Code bei Kunden gezeigt hat, dass es viele Verwendungen von `str()` gibt für eine Umwandlung von z.B. auch Strings selber zu einem String existieren, die ohne diese Korrektur dann Fehler generieren würde.

Das Default Encoding in Python bleibt bei ANSI, d.h. der **Python Source Code** selber verbleibt in **ANSI**. Das sichert eine grösstmögliche Rückwärtskompatibilität bei bestehendem Code. Allerdings können nicht-ANSI Zeichen nicht direkt als Stringliterals angegeben werden (in Python gibt es jedoch Alternativen).

Der Übergang von einem Unicode String in das Default Encoding, also von Unicode nach ANSI, wurde fehlertolerant gestaltet: Zeichen, die nicht umgewandelt werden können, führen zu keinem Error, sondern werden durch `?` ersetzt.

Encodings

Das **Default Encoding** in ANSI bringt es mit sich, dass bestehender Python Code allenfalls angepasst werden muss, der z.B. Strings verarbeitet und in einem File speichert oder an einen Webservice sendet. Damit Nicht-ANSI Zeichen, die ab Version 6.5 in Vertec vorkommen können, korrekt ankommen, muss ein Unicode Encoding

(wie z.B. UTF-8) gewählt werden – die Gegenseite muss dieses aber auch lesen können. Ohne ein explizites Encoding wie zum Beispiel

```
string.encode("UTF-8")
```

kommt die implizite Umwandlung auf ANSI mit dem allfälligen, oben erwähnten, Datenverlust von nicht-ANSI Zeichen zum Tragen.

Auch beim Einlesen von Daten nach Vertec, sei es über das Öffnen einer Textdatei, über den Vertec XML Server oder einem `vtcapp.requestfilefromclient()` muss man sich ums Encoding kümmern: Dies war schon immer so, aber mit Vertec 6.5 und der Unicode Unterstützung wird die Anforderung dringender, weil Datenverluste von nicht-ANSI Zeichen unnötig sind. Liest man also z.B. ein Textfile ein, welches in UTF-8 encodiert ist, muss dieses korrekt dekodiert werden:

```
unicodestring = filecontent.decode("UTF-8")
```

Als Faustregel kann man sich merken, dass in Python

- `encode` immer von einem Unicode String zu einem Bytestream konvertiert (ä belegt z.B. 2 Bytes in UTF-8)
- `decode` immer von einem Bytestream (z.B. in UTF-8 encodiert) zu Unicode konvertiert.

Kundenspezifische Extensions

Das «Umbiegen» des `str`-Moduls auf Unicode wird nicht bei kundenspezifische Extensions gemacht, die als Python Files im Ordner Extensions im Vertec Verzeichnis liegen. Dies betrifft auch abgeänderter Extension-Code, der ursprünglich von Vertec geliefert wurde. Bei diesen Extensions führt ein `str()` dazu, dass ein Unicode String mit dem erwarteten Datenverlust bei nicht-ANSI Zeichen nach ANSI umgewandelt wird. Kunden mit solchen Schnittstellen sollten den Python Code auf mögliche Probleme untersuchen.

Unicode Unterstützung bei Zusatzfeldern

Die Zusatzfeldtypen `Zeichen` und `Text` konnten in Vertec 6.4 Text darstellen. Dabei ist **Zeichen** ein Feld, welches auf maximal 255 Zeichen beschränkt ist, während das Feld **Text** unbeschränkt ist. Die Zusatzfelder vom Typ Zeichen werden mit 6.5 automatisch auf Unicode umgestellt, diejenigen vom Typ Text nicht.

Die Zusatzfelder vom Typ `Text` werden in der Datenbank als **BLOB-Felder** (Binary Large Object) abgespeichert. Dies bedeutet, dass die Daten als reine Binärdaten in der Datenbank abgespeichert und erst in der Vertec Businesslogik als Texte interpretiert werden. Da die Zusatzfelder vom Typ `Bild` im gleichen Datenbankfeld gespeichert werden, bleiben Zusatzfelder vom Typ Text aus Gründen der Abwärtskompatibilität in Vertec 6.5 unveränderte 8-Byte Texte im ANSI Format.

Mit Vertec 6.5 gibt es stattdessen den neuen Zusatzfeldtyp **Unicode Text**:

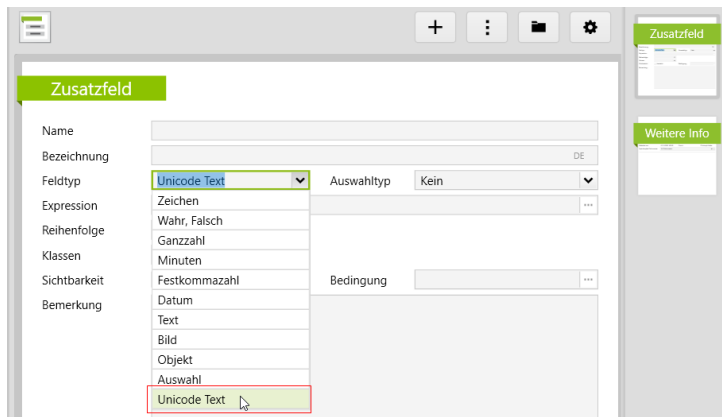


Abbildung 40: Zusatzfeldtyp Unicode Text

Falls Sie Zusatzfelder vom Typ Text verwenden, empfehlen wir dringend, diese unmittelbar nach dem Konvert auf Unicode Text umzustellen. Der DB Konvert auf 6.5 kopiert die Texte aus dem Feld Text automatisch im richtigen Format in das Feld Unicode Text.

Achtung: Alle Zugriffe auf das Zusatzfeld (OCL, SQL) müssen ebenfalls umgestellt werden:

- bei SQL von `WERTBLOB` auf `WERTTEXT`
- in OCL von `zusatzfeldblob('zusatzfeldname')` auf `zusatzfeld('zusatzfeldname')`.

11.2 Neugestaltung des Login-Dialogs und Vertec Splash-Screens

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.5.0.1

Betriebsart: Cloud Abo / On-Premises | Apps: Alle

Die Benutzeroberfläche des Login-Dialogs und Vertec Splash-Screens wurde neugestaltet und damit dem Design der Webseite und des Kundenportals angepasst.

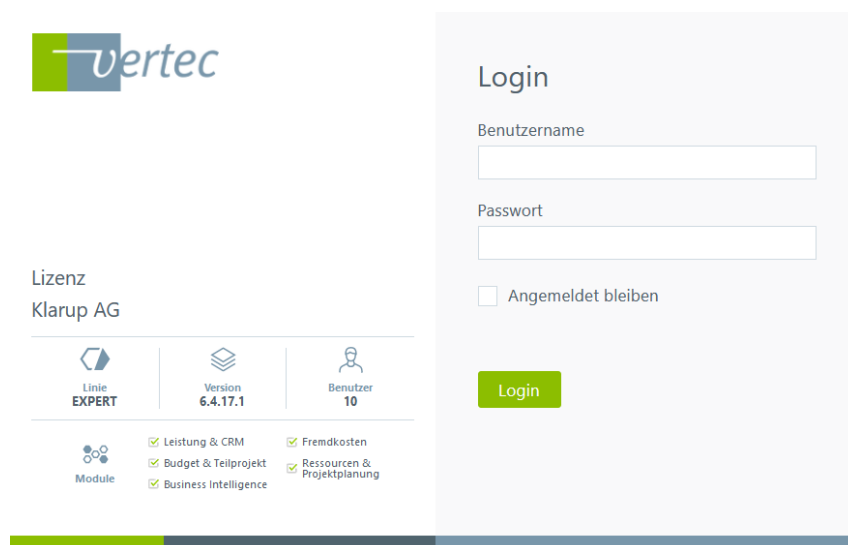


Abbildung 41: Neugestaltung Login-Dialog

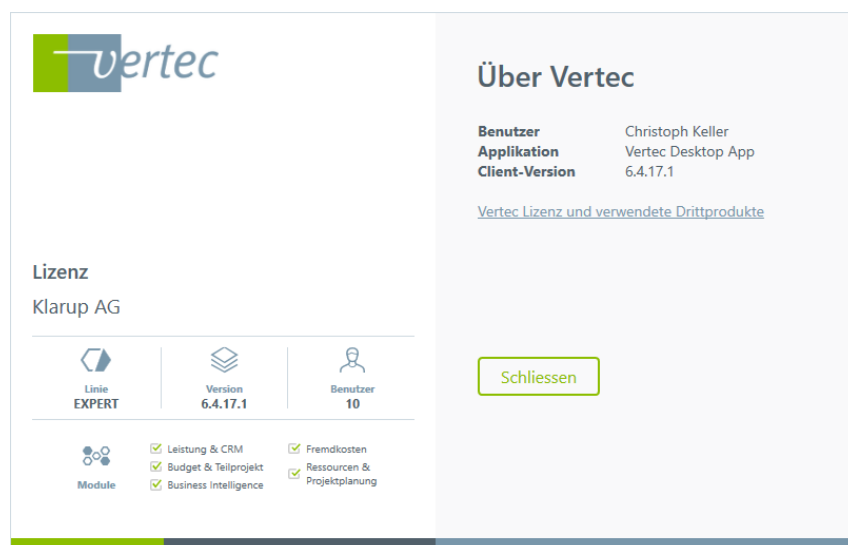


Abbildung 42: Neugestaltung Splash-Screen

Ausserdem wurde der Cloud Installer `Vertec.Updater.exe` integriert in `Vertec.CloudInstaller.exe`.

11.3 Unterstützung von mehreren Cloud App Installationen

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.5.0.1

Betriebsart: Cloud Abo / On-Premises | Apps: Cloud App

Es können verschiedene Cloud Apps von verschiedenen Servern auf einem Client installiert werden:

Bei einer Neuinstallation der **Cloud App** via Vertec App Portal, wird unter `%localappdata%\Vertec Cloud App` ein Unterordner für den Server-Namen angelegt, von dem aus die Installation erfolgt. Die Cloud App und das dazugehörige Vertec.ini werden dort abgelegt. Der Name des Unterordners bildet sich aufgrund des Server-Namens. Das ist besonders nützlich, wenn mit der Cloud App Zugriff auf mehrere Vertec Instanzen benötigt wird, wie beispielsweise auf eine Testumgebung.

11.4 Zusätzliches Logging

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.8

Betriebsart: On-Premises | Apps: Full-featured

Es gibt 2 neue Möglichkeiten, Debug Log Output aus Vertec anzuzeigen:

1. **Laden von Objekten aus der Datenbank:** Das Einsetzen des Wertes `DebugCategories = Vertec.PMappers` im Vertec.ini-File im Abschnitt **[Log]** ermöglicht es, das Laden von Objekten aus der Datenbank zu beobachten.
Die Persistence Mapper Logeinträge erscheinen nur, wenn sie via `DebugCategories` aktiviert werden (Kategorie DEBUG).
Hier ist zu beachten, dass diese Art von Logging nur temporär aktiviert werden sollte, da es durch das Schreiben der Logdaten in die Logdatei zu Performance-Einbussen kommen kann.
2. **Konvertierung, Initdata:** Log-Output beim Ausführen von versionsabhängiger Konvert Logik. Die Logeinträge erscheinen immer (Kategorie INFO).

11.5 Stoppen von hängenden TaskRunner Prozessen

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.8

Betriebsart: On-Premises | Apps: Full-featured

Es gibt Fälle, wo sich ein Vertec.TaskRunner Prozess nicht korrekt beendet. Diese Prozesse bleiben bestehen und blockieren Speicher auf dem Server. Deshalb kann im **Vertec.ini-File** unter `[Cloudserver]` nun unter `Task Scheduler Max Minutes` die maximale Laufzeit für einen geplanten Task angegeben werden. Standardwert ist 1200 (20h).

- Wenn ein Prozess länger als die maximale Zeit läuft, wird er vom Cloudserver automatisch beendet.
- Beim Beenden von Vertec.CloudServer werden auch alle gestarteten TaskRunner Prozesse beendet.

11.6 Log Output an Loggly Dienst senden

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.4.0.9

Betriebsart: On-Premises | Apps: Full-featured

Der Log Output kann aus den Server-basierten Vertec Prozessen (Vertec.CloudServer, Vertec.Session, Vertec.TaskRunner) sowie aus der Vertec Desktop App an den Cloud-basierten Logger Dienst **Loggly** gesendet werden.

Dafür müssen folgende Werte im Vertec.ini-File im Abschnitt **[Log]** gesetzt werden:

- **LogglyToken:** Hier wird das kundenspezifische Loggly Token eingegeben. Ist dieser Wert gesetzt, wird der Log Output an Loggly gesendet, und zwar an die angegebene LogglyUrl.
- **LogglyUrl:** Hier wird die gewünschte URL für Loggly angegeben. Falls nichts angegeben wird, wird der Output an <https://logs-01.loggly.com/> gesendet.

Nach dem Setzen dieser Werte muss der `Vertec.CloudServer` neu gestartet werden.

Folgende Werte werden an Loggly gesendet:

- `hostName`: Hostname des Rechners, auf dem der Prozess läuft.
- `level`: Log-Level der Information (Debug, Info, Warning, Error oder Fatal)
- `loggerName`: Name des betroffenen Loggers (z.B. `Vertec.TaskRunner.TaskScheduling.TaskScheduler`)
- `message`: Die eigentliche Log-Information
- `process`: Name des betroffenen Prozesses (z.B. `Vertec.TaskRunner`)
- `timestamp`: Zeitpunkt des Logeintrags im ISO-8601 Format

Der Kundenname aus den Lizenzinformationen wird als Tag mitgeschickt, sofern er zum Zeitpunkt des Logeintrags verfügbar ist (sehr früh im Aufstartprozess sind diese Informationen noch nicht vorhanden).

11.7 Entfernung Fastviewer Client

Linie: Standard, Expert | Module: Leistung und CRM | Version: 6.5.0.1

Betriebsart: Cloud Abo / On-Premises | Apps: Desktop App

Der Fastviewer Client wurde aus dem Vertec Installationsordner und dem Menü in der Desktop App entfernt.

Im Menü erscheint neu der Punkt `Kundenbereich und Support`:

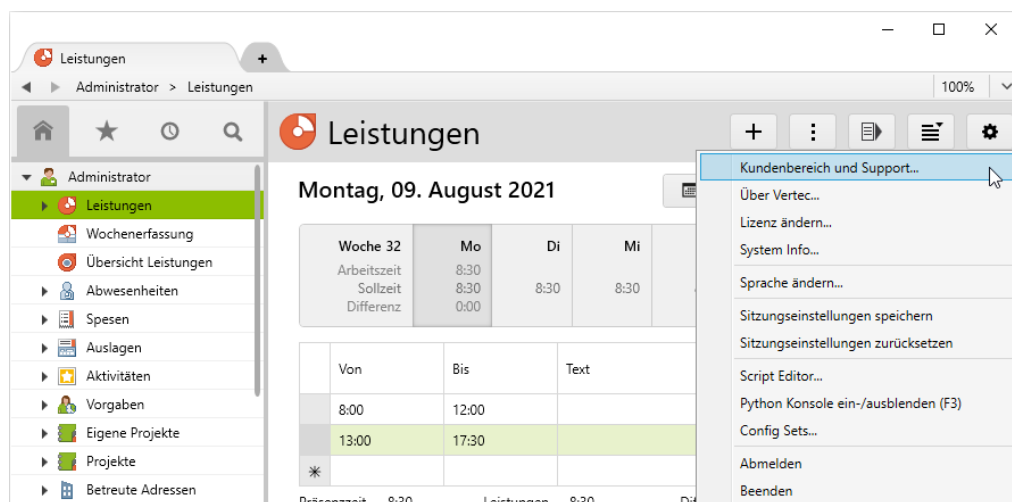


Abbildung 43: Menüpunkt Kundenbereich und Support

Dieser führt direkt auf die Seite des Vertec Supports, wo der Fastviewer immer in der aktuellen Version heruntergeladen werden kann.

11.8 Datenbank Konvert

Linie: Standard, Expert | Module: Leistung und CRM | Apps: Full-featured | Version: 6.5

Mit Vertec 6.5 wird ein neu implementierter DB Konvert ausgeliefert. Der DB Konvert wird z.B. benötigt, um die Vertec Datenbank auf eine neue Version anzupassen, wenn ein [Vertec Update](#) durchgeführt werden möchte.

Mit dem neuen DB Konvert ist es ausserdem möglich, Firebird Datenbanken in eine Microsoft SQL Datenbank zu kopieren und umgekehrt. Eine Konvertierung ist möglich

- von MS SQL nach MS SQL
- von MS SQL nach Firebird
- von Firebird nach Firebird

- von Firebird nach MS SQL

Der neue DB Konvert bietet ausserdem folgende Verbesserungen:

- Es braucht für den MS SQL Server Konvert keine DTS- Komponente mehr.
- Es gibt den 32-bit `Vertec.DBConvert.exe` nicht mehr für den MS SQL Server Konvert.
- Der neue DB Konvert ist in der Lage, eventuell vorhandene kundenspezifische Indizes zu kopieren, sie stehen also auch nach der Konvertierung in der neuen Datenbank zur Verfügung.

Kundenspezifische Indizes

In Vertec Datenbanken können nach sorgfältiger Analyse und Abwägung kundenspezifische Indizes angelegt werden. Damit Indizes nach einem Datenbank Konvert erhalten bleiben, müssen Sie mit dem Präfix `CIX_` beginnen.

Kundenspezifische Indizes werden wie folgt erstellt:

```
CREATE [UNIQUE] INDEX indexname ON tablename (column1, column2, column3)
```

Die Syntax für Firebird und SQL für die Erstellung einfacher Indizes ist ähnlich, wenn weitere Angaben hinzukommen gibt es jedoch Unterschiede.

- Weitere Informationen zur Erstellung von Indizes bei **Firebird** Datenbanken finden Sie unter <https://firebirdsql.org/refdocs/langrefupd20-create-index.html>
- Weitere Informationen zur Erstellung von Indizes bei **SQL Server** Datenbanken finden Sie unter <https://docs.microsoft.com/de-de/sql/t-sql/statements/create-index-transact-sql?view=sql-server-ver15>

Beispiel:

```
CREATE INDEX CIX_INVOICENUMBER ON RECHNUNG (NUMMER);
```

Das Erstellen von Indizes kann Performance-Verbesserungen bewirken, allerdings gibt es auch Risiken. So kostet das Einfügen oder Updaten eines Records mehr Zeit, wenn auch noch ein Index zusätzlich bearbeitet werden muss, und auch bei einem **SELECT** führt ein Index nicht immer zu einem besseren Resultat: so kann es sein, dass bei einem Feld mit eigentlich guter Selektivität aber gleichzeitig vielen Werten wie «» oder **NULL** die Performance sogar schlechter wird. Auch wirkt ein Index nur bei einem = Vergleich, nicht bei einem **LIKE**.

Vor dem Entscheid, einen neuen Index einzufügen, sollte man den Effekt auf die gesamte Applikation jedenfalls eingehend testen.

Automatische Konvertierung mit Firebird

Bei der Verwendung von **Firebird Datenbanken** kann die Konvertierung beim Update von Vertec wie bisher automatisch durchgeführt werden.

Beim Aufstarten von Vertec mit einer neuen Datenbank Version erscheint der DB Konvert Dialog:

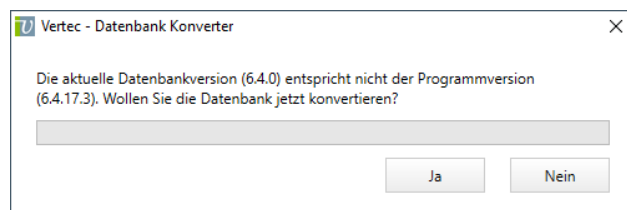


Abbildung 44: Abfrage automatischer Datenbank Konvert

Mit einem Klick auf **Ja** startet die automatische Konvertierung. Eine Statusleiste zeigt den Fortschritt der Konvertierung an.

Nach Beendigung des Vorganges muss die Vertec Desktop App einmal neu gestartet werden:

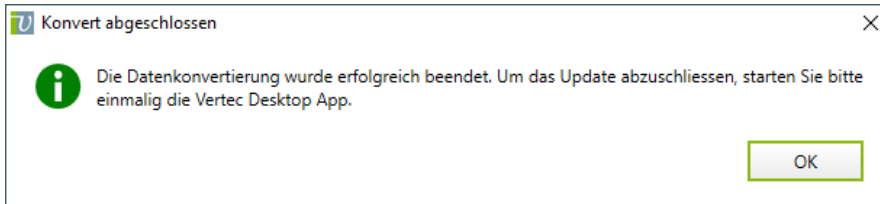


Abbildung 45: Bestätigung automatischer Datenbank Konvert

Manuelle Konvertierung

Wird ein MS SQL Server als Datenbank Server verwendet, muss die Quell- und die Ziel-Datenbank manuell erstellt und die Konvertierung mit dem DB Konvert manuell durchgeführt werden.

Starten Sie Vertec mit dem Parameter `/DC`. Es öffnet sich der Konvertierungs-Dialog:

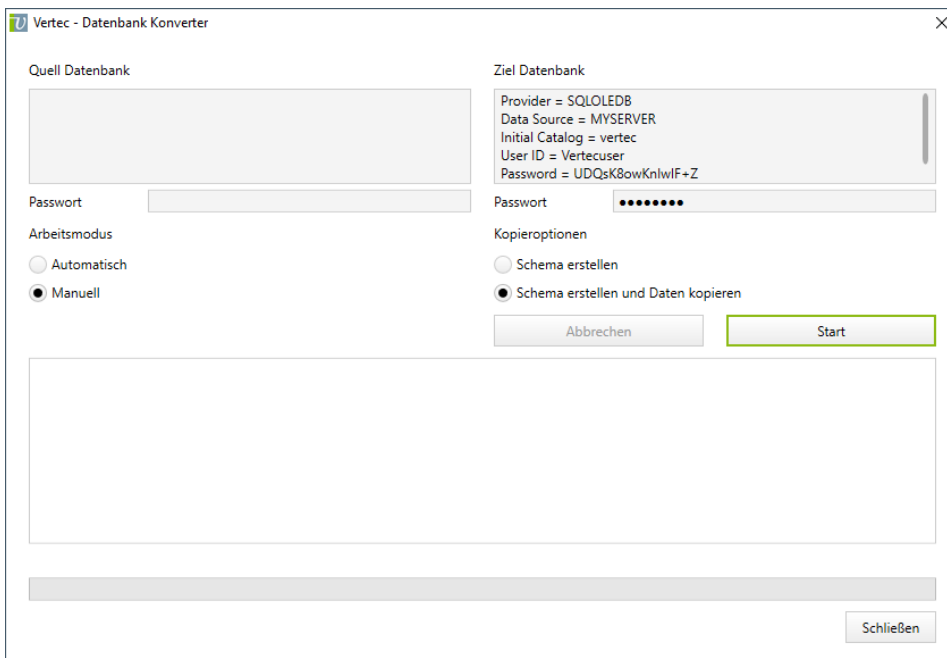


Abbildung 46: Manueller Datenbank Konvert

Bei der manuellen Konvertierung wird die Quell-Datenbank verändert. Daher sollte temporär eine Kopie der Quell-Datenbank angelegt werden. Das genaue Vorgehen hierzu ist auf der [Knowledge Base](#) beschrieben.

Kopieren von Datenbanken

Mit dem DB Konvert lassen sich Vertec Datenbanken kopieren. Dabei ist es auch möglich, eine Firebird Datenbank in eine Microsoft SQL Datenbank zu kopieren und umgekehrt.

Das Vorgehen entspricht der manuellen Konvertierung:

1. Starten Sie Vertec mit dem Parameter `/DC`.
2. Wählen Sie als Quell-Datenbank die Datenbank aus, die Sie kopieren möchten.
3. Tragen Sie als Ziel-Datenbank eine leere Firebird oder MS SQL Datenbank ein.

Folgende Angaben werden benötigt:

Firebird

Servername	Identifiziert den Server und die Datenbank.
------------	---

User Name	Identifikation für den Datenbankserver.
Password	Verschlüsseltes Passwort für die Identifikation am Datenbank Server, zusammen mit dem User Name . Um das Passwort verschlüsselt einzutragen, verwenden Sie das Vertec DB-Tool .

Beispiel:

```
ServerName = MYSERVER:C:\Programme\Vertec\DB\Vertec.fdb
User Name = SYSDBA
Password = US-bVTkvBqrtWV+Z
```

Microsoft SQL Server

Provider	Treiber für die SQL Verbindung
Data Source	Servename (oder auch IP-Adresse) des MS SQL Servers.
Initial Catalog	Name der Datenbank auf dem MS SQL Server.
User Name	Identifikation für den Datenbankserver.
Password	Verschlüsseltes Passwort für die Identifikation am Datenbankserver, zusammen mit dem User Name . Um das Passwort verschlüsselt hier einzutragen, verwenden Sie bitte das Vertec DB-Tool .

Beispiel:

```
Provider = SQLOLEDB
Data Source = MYSERVER
Initial Catalog = vertec
User ID = Vertecuser
Password = UDQsK8owKnIwIF+Z
```

Anschliessend kann ausgewählt werden, ob nur das **Datenbank Schema** erstellt werden möchte oder ob auch die **Daten** kopiert werden sollen:

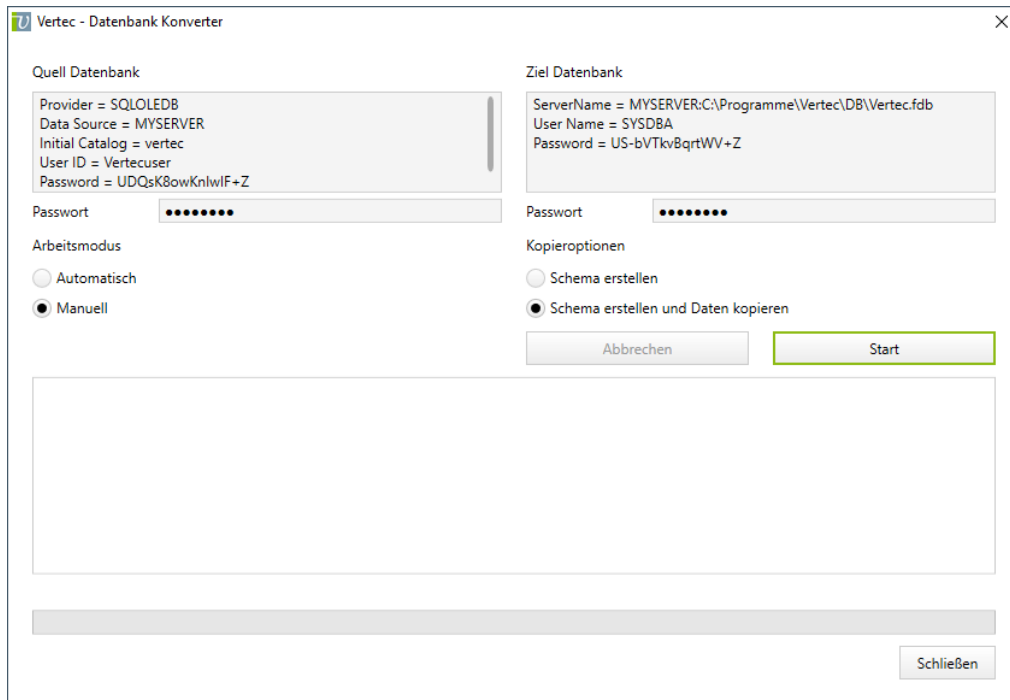


Abbildung 47: Kopieren einer Microsoft SQL Datenbank in eine Firebird Datenbank

Mit einem Klick auf **Start** beginnt der Datenbank Konvert. Der Fortschritt wird in einer Statusleiste angezeigt. Nach Beendigung der Konvertierung muss Vertec neu gestartet werden.

11.9 Update von IBQuery

Linie: Standard, Expert | Module: Leistung & CRM | Version: 6.5.0.1

Betriebsart: On-Premises | Apps: Full-featured

Mit Vertec 6.5 wird die aktuellste Version von IBQuery mitgeliefert. Wenn Sie eine On-Premises Installation von Vertec haben, finden Sie nach dem Update in Ihrem Installationsverzeichnis die Version 9.5.0 von IBQuery (**IBQ.exe**) in der Professional Version.

Die Details zur neuen Version finden Sie auf der [Homepage des Herstellers](#).

12 Verbesserungen Firebird Datenbank Unterstützung

Linie: Standard, Expert | Module: Leistung und CRM | Apps: Full-featured | Version: 6.5

Mit Vertec 6.5 wird die Unterstützung von Firebird Datenbanken verbessert. Dazu wurden folgende Punkte umgesetzt:

12.1 Update auf Firebird 4.0

Bei dem Update auf Vertec 6.5 kann das Update von Firebird 3.0 auf Firebird 4.0 durchgeführt werden. Dazu erscheint im Installationsprozess folgende Abfrage:

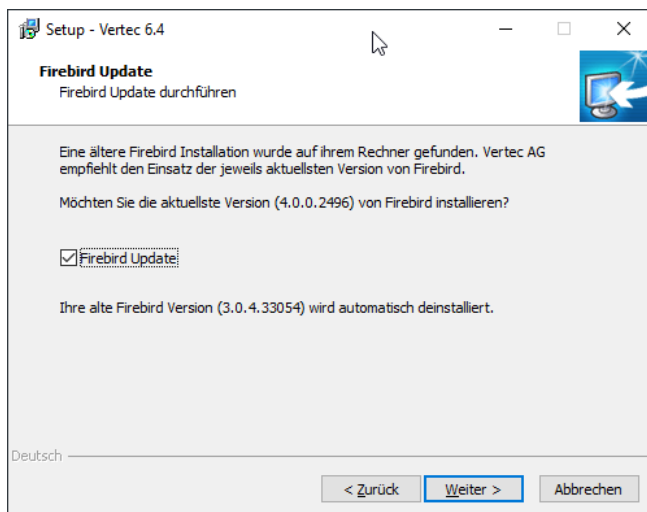


Abbildung 48: Abfrage Firebird Update

Wenn Sie das Firebird Update durchführen möchten, aktivieren Sie die entsprechende **Checkbox** und klicken Sie auf **Weiter**. Dabei wird der Firebird Server zunächst heruntergefahren und anschließend neu gestartet.

Weitere Informationen zu Firebird 4.0 finden Sie in den [Firebird Release Notes](#).

12.2 Update von Firebird FIB Komponenten

Die von Vertec verwendeten Delphi Firebird Komponenten wurden bisher in der Client Library gds32.dll im Installationsverzeichnis abgelegt. Ab Vertec Version 6.5 wird die aktuelle Firebird Client Library **fbclient.dll** verwendet, die Datei gds32.dll wird aus dem Vertec Installationsordner entfernt.

Alle Vertec Executables, die einen Zugriff zur Firebird DB brauchen (Session Prozess, Cloud Server, Task Runner und insbesondere die Vertec Desktop App), benutzen dann die Standard-Library **fbclient.dll**, die vom Firebird Setup ins system32 geschrieben wird.

Um in Zukunft die Vertec Desktop App zu verwenden, muss daher zwingend der **Desktop App Setup** (bisher Client Setup) ausgeführt werden.

12.3 Firebird Dialect 3

Mit dem Update auf Vertec 6.5 wird nicht mehr wie bisher der Dialect 1 verwendet, sondern der **Dialect 3**. Dialect 1 wird von Firebird 4.0 nicht unterstützt. Dieser Wechsel hat einige Auswirkungen:

Doppelte Anführungszeichen

In doppelte Anführungszeichen eingeschlossener Text wird im Dialect 3 als Spalte interpretiert, während sie im Dialect 1 synonym zu einfachen Anführungszeichen verwendet werden konnten. Der folgende Code erzeugt im Dialect 3 einen Fehler:

```
select projekt.creationdatetime from projekt
where projekt.creationdatetime < "today"
```

Dasselbe gilt für Abfragen in String-Vergleichen mit `=` und `like`

```
select code from projekt where code like "T%"
```

Um eine möglichst gute Rückwärtskompatibilität zu gewährleisten, werden doppelte Anführungszeichen im SQL-Code in SQL-Ordern und in der Funktion `getwithsql()` in Python-Scripts bei der Ausführung automatisch durch **einfache** Anführungszeichen ersetzt.

Es sollte jedoch darauf geachtet werden, in Zukunft nur noch einfach Anführungszeichen im SQL-Code zu verwenden, also zum Beispiel:

```
select projekt.creationdatetime from projekt
where projekt.creationdatetime < 'today'
```

oder

```
select code from projekt where code like 'T%'
```

Integer Division

Ein weiterer Punkt, der sich von Dialect 1 auf Dialect 3 ändert ist die Division von **Integern**.

In Dialect 1 ergibt eine Division von 2 Integern einen Float, in Dialect 3 nicht.

Dialect 1: $1/3 = 0.33333$

Dialect 3: $1/3 = 0$

Bei Literals kann man einfach `1.0/3.0` schreiben um 0.3333 zu erhalten, bei einer Division von Integer Werten in Tabellen muss man einen `CAST` machen.

Datumsumwandlungen

Vermutlich relevanter sind die Unterschiede bei den Datumsfeldern. In Vertec 6.4, Firebird 3.0.x und Dialect 1 waren alle Datumsfelder mit oder ohne Zeitanteil vom Typ `DATE`.

In Vertec 6.5, Firebird 4.0 und Dialect 3 sind die Datumsfelder vom Typ `TIMESTAMP`. Das merkt man in Vertec allerdings nur bei expliziten `CAST`-Operationen, zum Beispiel:

```
extract (month from datum) = extract (month from cast('TODAY' as date))-1) and
(extract (year from datum) = extract (year from cast('TODAY' as date)))
```

`DATE` ist in Dialect 3 ohne Zeitanteil, ein `AS DATE` schneidet einen solchen also ab, falls vorhanden. Das ist in diesem Beispiel nicht relevant, aber es gibt auch relevante Datumswerte in Vertec mit Zeitanteil, z.B. bei `Budgetchange` Objekten oder wenn man alle neu erstellen Objekte mit `CreationDateTime` abfragen will. Firebird gibt bei einem `CAST ... AS DATE` eine Warnung aus mit Dialect 3:

```
SQL warning code = 301
DATE data type is now called TIMESTAMP
SQL warning code = 301
DATE data type is now called TIMESTAMP
```

Rückwärtskompatibilität

Die Umstellung auf Firebird Dialect 3 ist bei der Verwendung von expliziten **CAST**-Operationen nicht rückwärtskompatibel. Kundenspezifischer Code, der Zeitanteile verwendet, muss von **DATE** auf **TIMESTAMP** angepasst werden.

12.4 Firebird Fehlermeldungen

Fehlermeldungen aus der Firebird Client-Library (via Delphi) beginnen meist mit dem Text **Can't format message**, weil die Sprachressource von Firebird nicht gefunden wird. Diese Fehlermeldungen erschienen insbesondere bei dem Versuch, fehlerhafte SQL-Abfragen in SQL-Ordnern auszuführen.

Neu enthalten diese Meldungen einen von Vertec erstellten sinnvolleren Text.

Beim Setup wird dazu die Datei **firebird.msg** erstellt und in Ihrem System abgelegt. Das betrifft neben einem temporären Ordner während der Installation auch den Vertec Installationsordner und zwei Systemordner. Die Datei muss nach erfolgreichem Setup daher in den folgenden drei Ordnern vorhanden sein:

- Vertec Installationsordner
- %SYSTEMROOT%\System32
- %SYSTEMROOT%\SysWOW64.

13 Update von Vertec On-Premises Installationen

13.1 Neuinstallation von Vertec

Für eine Neuinstallation verwenden Sie das zum Download verfügbare Installationsprogramm [VertecSetup-6.5.0.exe](#). Nähere Informationen zur Installation finden Sie in unserer Online Knowledge Base unter www.vertec.com/kb/neuinstallation.

13.2 Vor dem Update auf 6.5

Bitte beachten Sie dazu die Abschnitte zur Rückwärtskompatibilität in den Kapiteln 11.1 zur Unicode Unterstützung und 12.3 zur Verwendung von Firebird Dialect 3.

13.3 Update von Vertec

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

13.4 Das erste Aufstarten nach dem Konvert

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

Neustart des Cloud Servers

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