September 2021

Vertec Software Release 6.5

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

Table of contents

1 Highlights of version 6.5 and introduction

The focus of the major 6.5 release of Vertec lies with the new CRM functions: the new Outlook App and a new class of Vertec objects called Opportunities.

The cloud-enabled Outlook App allows for an easy to use and intuitive storage of Outlook e-mails as Vertec activities – directly out of Outlook. Moreover, the full history of activities performed in Vertec is visible in Outlook. In Outlook, it is possible to access the e-mail addresses of your Vertec contacts and immediately open new contacts in Vertec. Thus, the Vertec Outlook App really represents an integrated solution for a CRM with enhancing value.

The process from the first contact with a (potential) customer to the completion of the project is easily represented with Opportunities. Here you can input sales probabilities and keep track of the status of your current leads.

Also related to CRM, another major change in 6.5 is the replacement of the character sets which can be managed in Vertec. For historical reasons, the characters available in Vertec have been limited to the West-European (ANSI) set. With 6.5, Vertec opens to East-European and international languages (Unicode). In this way, Russian addresses specified in Cyrillic script, for instance, can also be used.

A new web interface for Vertec BI is also worth mentioning. It makes it possible for values computed in Vertec BI to be exchanged with and processed in third-party systems (Power BI for instance or Microsoft Excel).

Version 6.5 again presents many **highlights**. The most important ones are the following:

1.1 Backwards compatibility

For features where backwards compatibility constitutes an issue, detailed descriptions are given in the corresponding **Backwards compatibility** sections.

1.2 Pre-set features in Vertec 6.5

System setting for conflict check removed

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.21
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

The plug-in Konfliktcheck replaces the previous procedure to search for conflicts. For new installs, the system setting **Show information regarding job linkages (conflict search)** is no longer displayed. If the old search for conflicts was used, this system setting can be set to `False` to hide the corresponding menu items and standard fields.

Further information Information related to the new plug-in can be found in the article Plug-in: Konfliktcheck in the Online Knowledge Base under www.vertec.com/kb/plug-in-konfliktcheck/.

---

2 Vertec Outlook App

Line: Standard, Expert | Module: Services & CRM | Apps: Outlook App | Version: 6.5.0.1

The Vertec Outlook App displays CRM information in Outlook and brings support for daily tasks.

The Outlook App is a Vertec Cloud Client and can be used directly from the Vertec Cloud Server without further local app installation. It is a Specialized App and, accordingly, provides certain functions, however not the full function range of a full-featured app like the Web App or the Cloud App, for instance.

The Outlook App allows a tight integration of e-mails and calendar entries from Outlook in Vertec. The language adapts to the logged-in user. Before that, the language of the client computer is used.

The Outlook App covers the following subjects:

- Storing e-mails in an activity history and direct creation of an activity in Vertec
- Display of information from Vertec related to e-mails in Outlook
- Direct navigation to the relevant objects in Vertec
- Selection of contacts from Vertec when writing an e-mail
- Display of the activity history and of the e-mail exchange
- Transform calendar entries into services

The Vertec Outlook App is available in two editions: the **Windows Edition**, where the Outlook App runs in the local Windows Client of Outlook and the **Web Edition**, in which the Outlook App is installed in Office 365. Both installations are described below.

2.1 Installation of the Windows Edition

Procedure

1. Open your **Vertec App Portal**. You will the new **Outlook App there**.
2. Close Outlook if it is open.
3. Click on the button `Installer Download` and open the downloaded file.

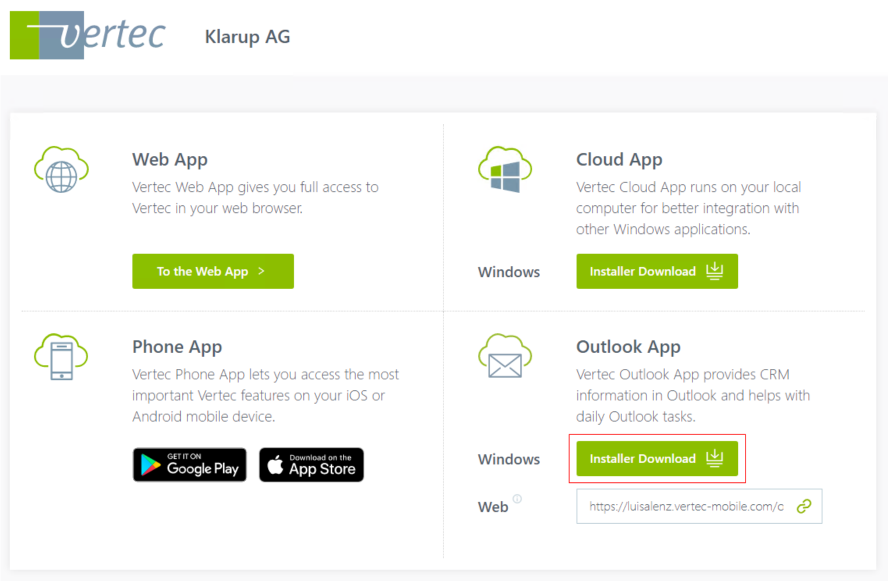


Figure 1: Installer Download in the Vertec App Portal

The Outlook App Installer is then automatically started. The installation of the Outlook App ends with the following message:

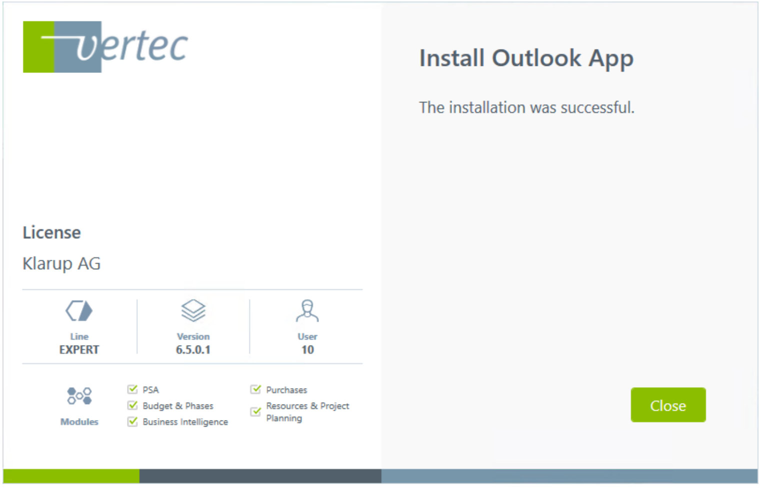


Figure 2: Successful installation

4. Select `Close`.
5. Open your Outlook client. The Vertec Outlook App is displayed as a sidebar.
6. Log in with your Vertec login.

The display of the Outlook App can be customized in the Menu bar via `Vertec` > `Anzeigen`.

In the Outlook App settings, it is further possible to specify whether Vertec should open in the Web App or in the Cloud/Desktop App:

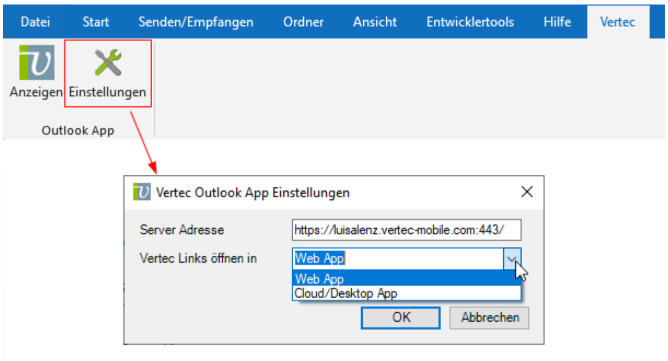


Figure 3: Outlook App displays and settings

If the settings are modified, the Outlook App is restarted.

2.2 Installation of the Web Edition

Procedure

1. Open Office 365, respectively Outlook, in the web browser.
2. Open your **Vertec App Portal**. You will now find the **Outlook App there**.
3. Copy the link under the Outlook App manually or with the following icon:

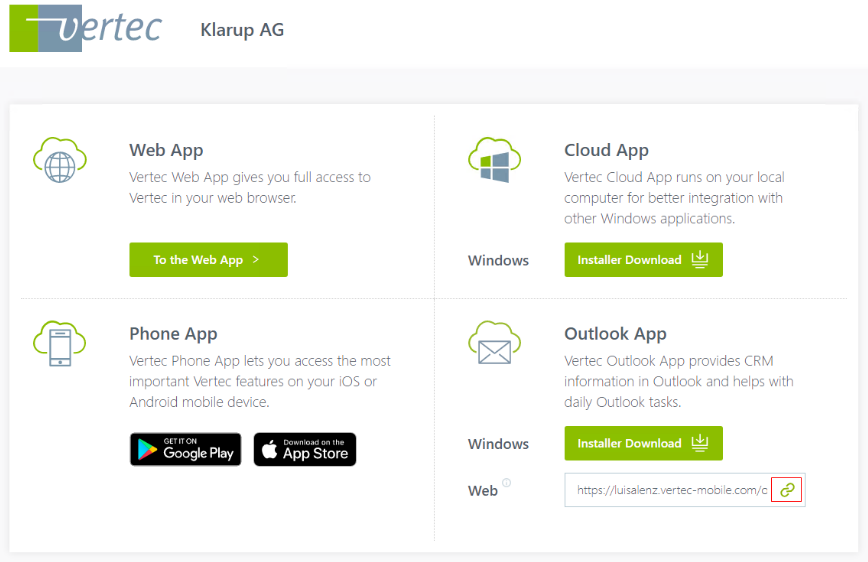


Figure 4: Copying the link in the new Vertec App Portal

4. Open any mail in Outlook.
5. Click on `More actions` and select `Get Add-ins`:

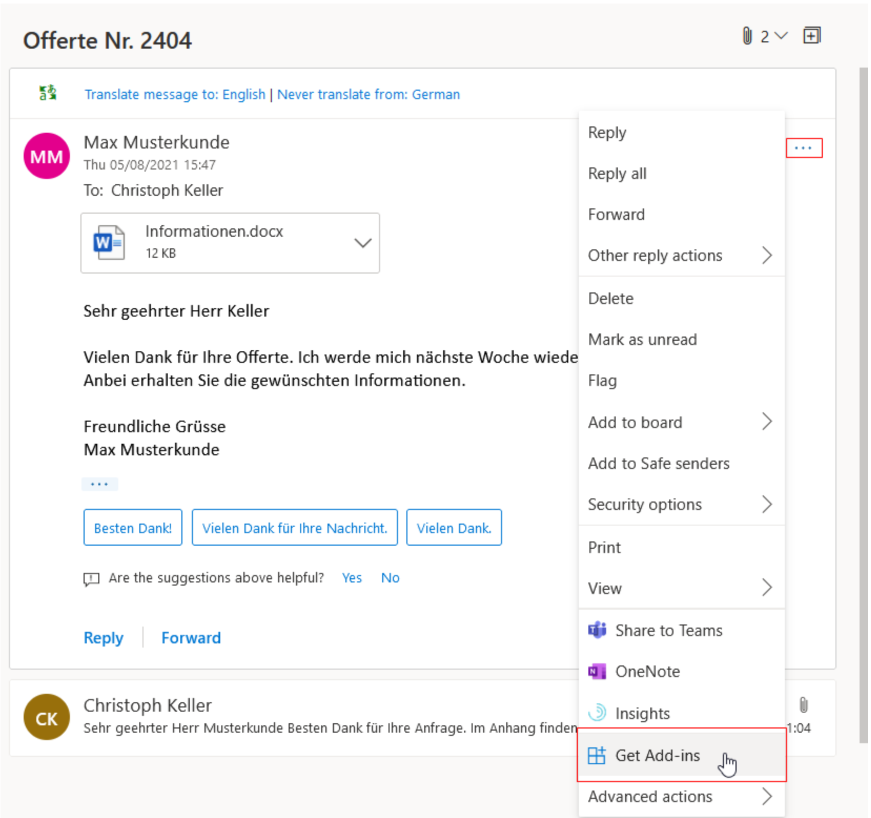


Figure 5: Get Add-Ins

6. In `My add-ins` click on `+ Add a custom add-in` and select `Add from URL`.

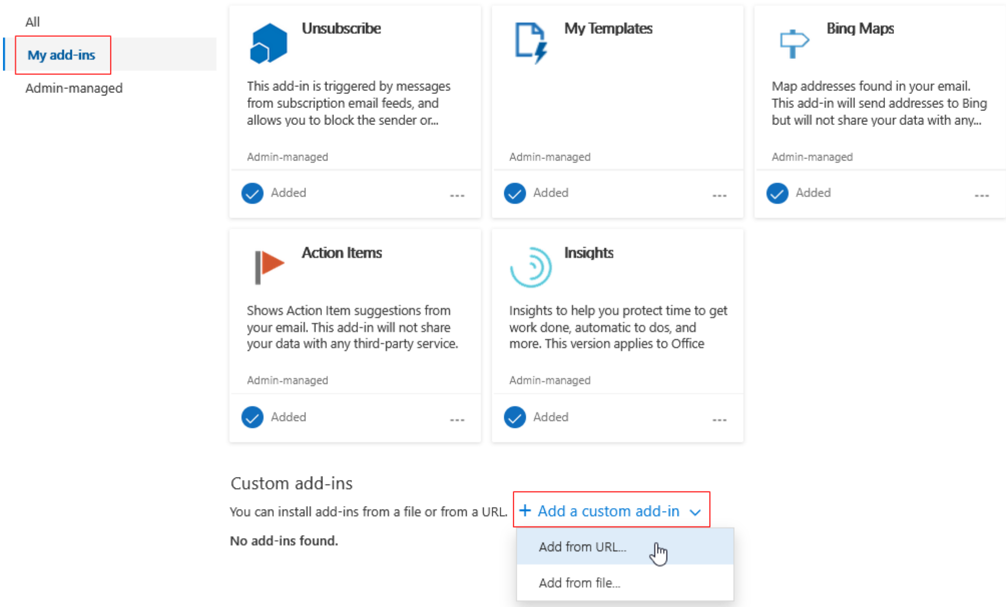


Figure 6: Add Add-In from URL

7. The link previously imported from the Vertec App Portal can be inserted and confirmed with `OK`.
8. Select `Install`.

The add-in is now successfully installed.

Allow third-party cookies

In the Web Edition, the Vertec Outlook App runs in the context of the Web Edition of Outlook. It is therefore a third-party application within the outlook.com application. The Outlook App uses its own cookies for authentication and session management. In view of the browser being used, these cookies are «third-party cookies», since they are not directly sent to the main page, but to a third-party page.

Many browsers offer the possibility to limit the use of such third-party cookies, since they are often used for «tracking» (advertising). If, however, the browser blocks the cookies of the Outlook App, it cannot function – third-party cookies must therefore be allowed for the outlook.com page.

Tip: Vertec Phone App and Outlook App are based on similar architectures. Certain settings related to the Web Edition of the Outlook App and third-party cookies will lead to changes for the Phone App as well. Old Phone App versions are therefore not compatible with the new server version. New Phone App versions however are backwards compatible with old server versions.

2.3 Process incoming e-mails

Once Outlook App has been successfully installed, it can be displayed as a sidebar.

Select any e-mail from Outlook. With `More actions`, the Outlook App may be called:

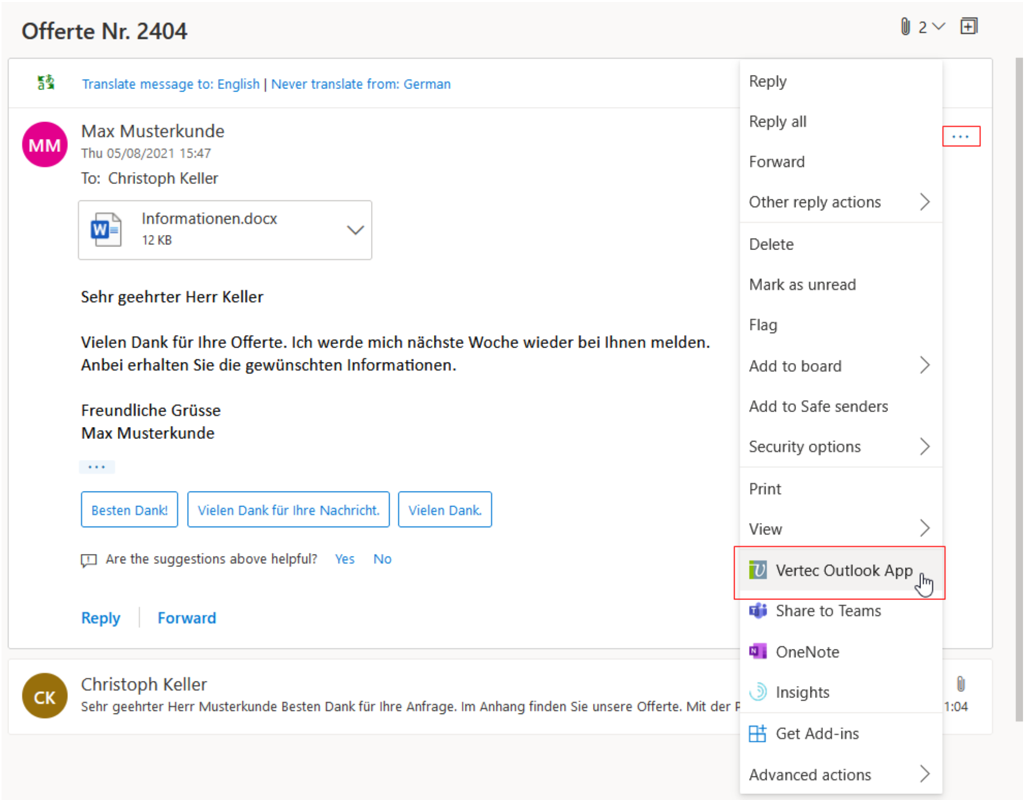


Figure 7: Calling the Outlook App

Log in with your Vertec login.

Outlook App can then be pinned or fixed as desired:

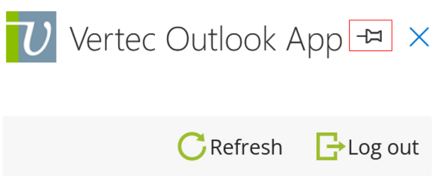


Figure 8: Pin the Outlook App

When you then select an incoming mail, two scenarios are possible: the author of the incoming e-mail already exists as a contact in Vertec or is still unknown. Both possibilities are described below:

Create a new contact

The selected e-mail comes from a person not yet registered in Vertec. With the button `+ New address` you have the possibility to create a new contact or register this person.

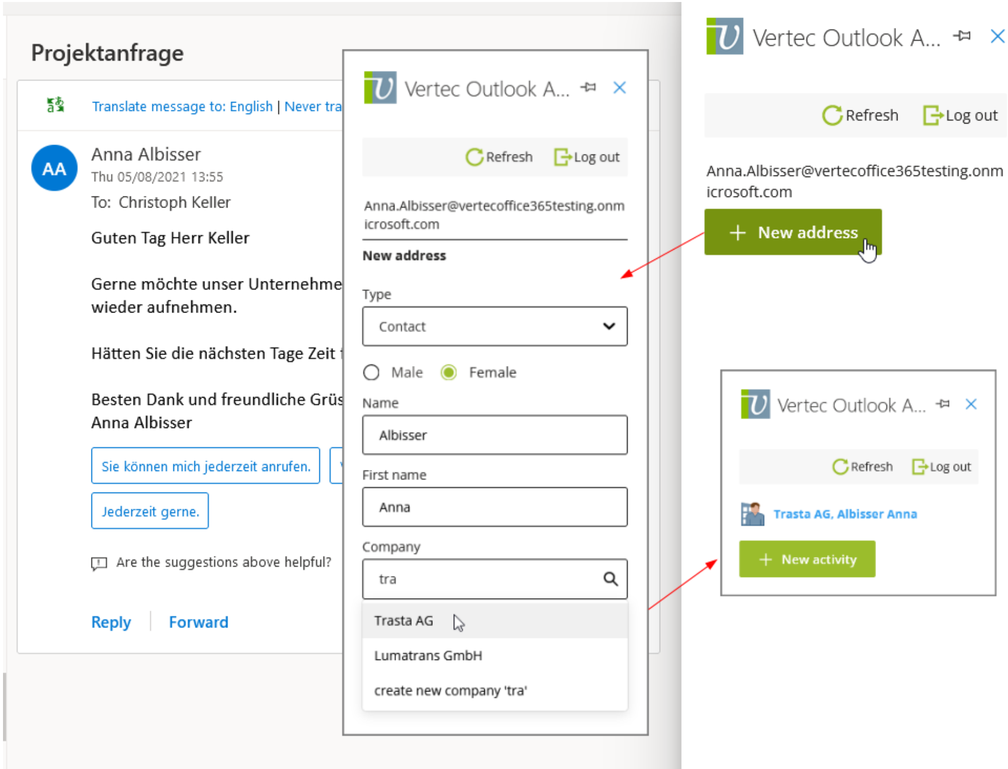


Figure 9: Add a new contact

With `Type` you can choose between **Contact** or **Person**. If contact is chosen, the field `Company` is displayed, otherwise only the fields `Name`, `First name` and the selection of the gender.

If no address entry is found corresponding to the selected e-mail, a new entry can be created. The name may be input manually and confirmed with Enter. A message then appears in the field `Company` saying: «create new company with 'XYZ'» and the company is automatically added in Vertec.

With the button `Save` the contact is directly created in Vertec. In the Outlook App, it is displayed with the icon and a link showing company and name which, if you click on it, directly opens the contact in Vertec. At the same time, the button changes from `+ New address` to `+ New activity`.

Create a new activity

The contact of the selected e-mail is already known and stored in Vertec. In this case, Outlook App will display the contact and company at the top, followed by the existing Activity history. This history also includes the activities of colleagues in your company having dealt with this contact, as, in this example, the phone conversation between Mister Samplecustomer and Lars Sorenson.

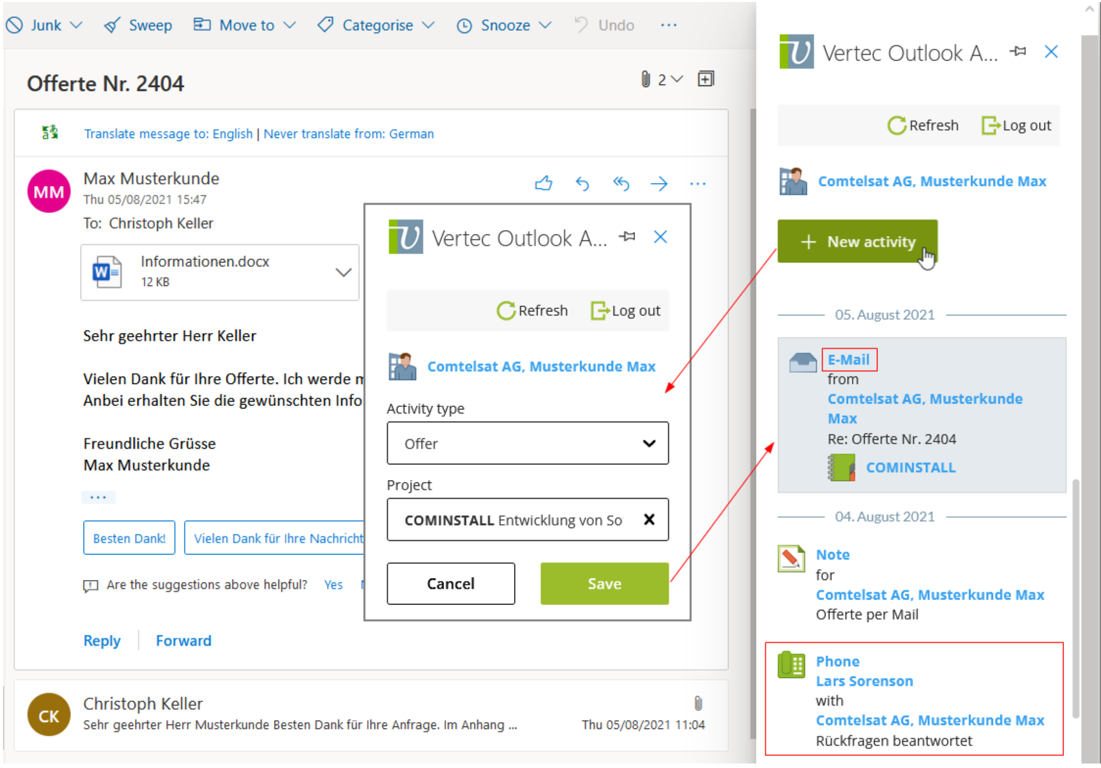


Figure 10: Create a new activity

With the button `+ New activity`, an activity type and the corresponding project can be selected. The e-mail is then stored in the history of activities of the corresponding project and as a new activity in Vertec. Attachments are also stored with the activity and can be downloaded if desired.

Multiple recipients

If you send an e-mail to several recipients or receive an e-mail with several recipients, you can use the arrow buttons to scroll through the various recipients:

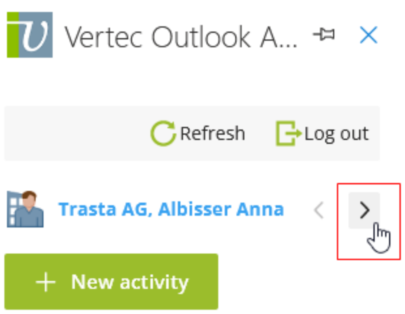


Figure 11: Arrow buttons to scroll through recipients

If a recipient is not yet registered in Vertec, the above-described field `+ New address` appears. Otherwise, the history of activities is displayed with the button `+ New activity`.

Not supported objects

If an object is selected in Outlook which cannot be processed in Vertec (an absence message for instance), the error message `This item in Outlook cannot be processed with the Vertec Outlook App` appears in the Vertec Outlook App.

2.4 Create E-Mails

The Outlook App can also be opened when creating a new e-mail:

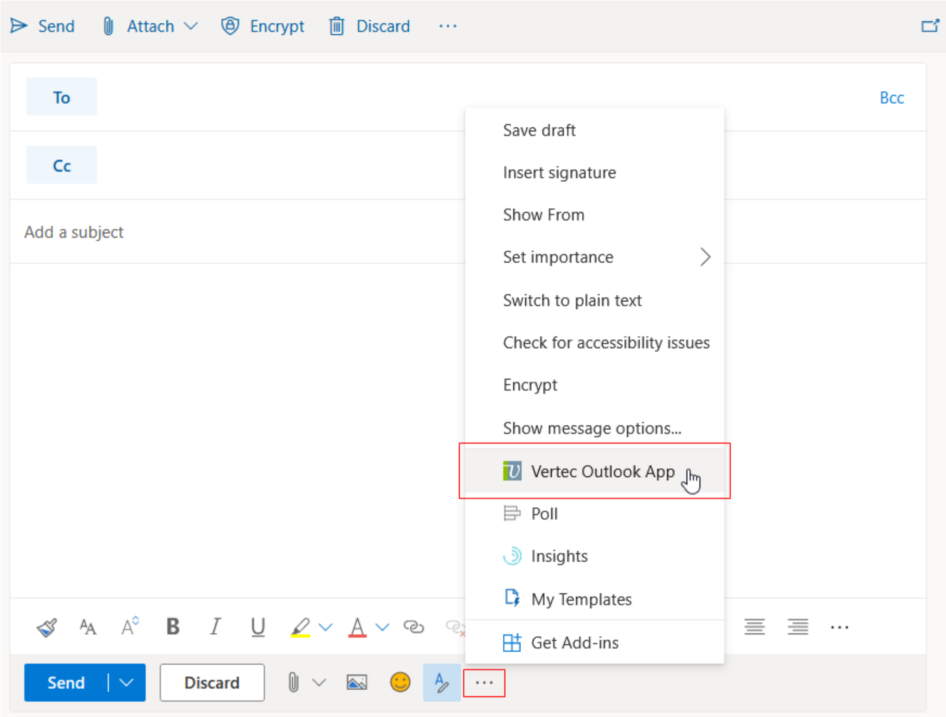


Figure 12: Open the Outlook App

It can then be used to find a contact in Vertec:

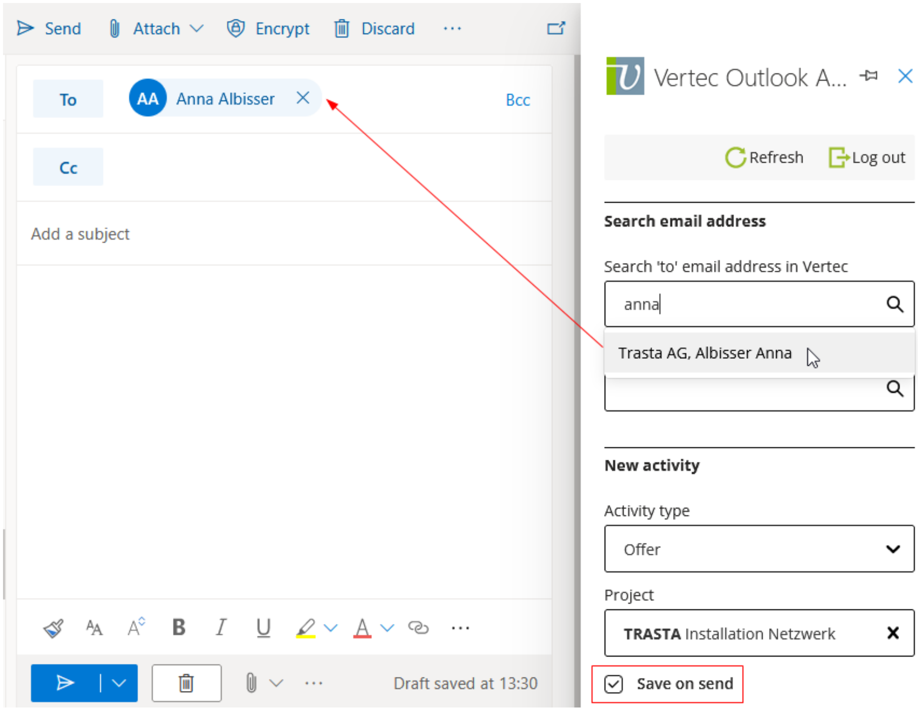


Figure 13: Find a contact with the Vertec search

The search field allows to search for a name as well as for a project. When the desired contact is found, it can be selected with a click or with the Enter key. The address is then directly copied into the `To` or `Cc` field and the activity history of the selected contact inserted.

The corresponding activity type and the project can also be selected here. All activity types stored in Vertec can be selected.

When the box `Save on send` is checked, the e-mail is automatically stored in Vertec. The activity in Vertec is then linked to the selected address, project and activity type. The checkbox is activated by default, can however be deactivated manually.

Tip: IF the Outlook App is not displayed when a new e-mail is created, no activity is generated in Vertec.

2.5 Input services based on calendar entries

Besides e-mails, calendar entries in Outlook can be stored as services in Vertec.

When creating a calendar entry in Outlook, the Outlook App can be called through `More options`. With the button `+ New service` a project, a phase or an activity can be attached. All activities stored in Vertec are displayed. Clicking on the button `Save` will display the service in the sidebar and store it in Vertec. The created service is represented as a link and the service is opened in Vertec.

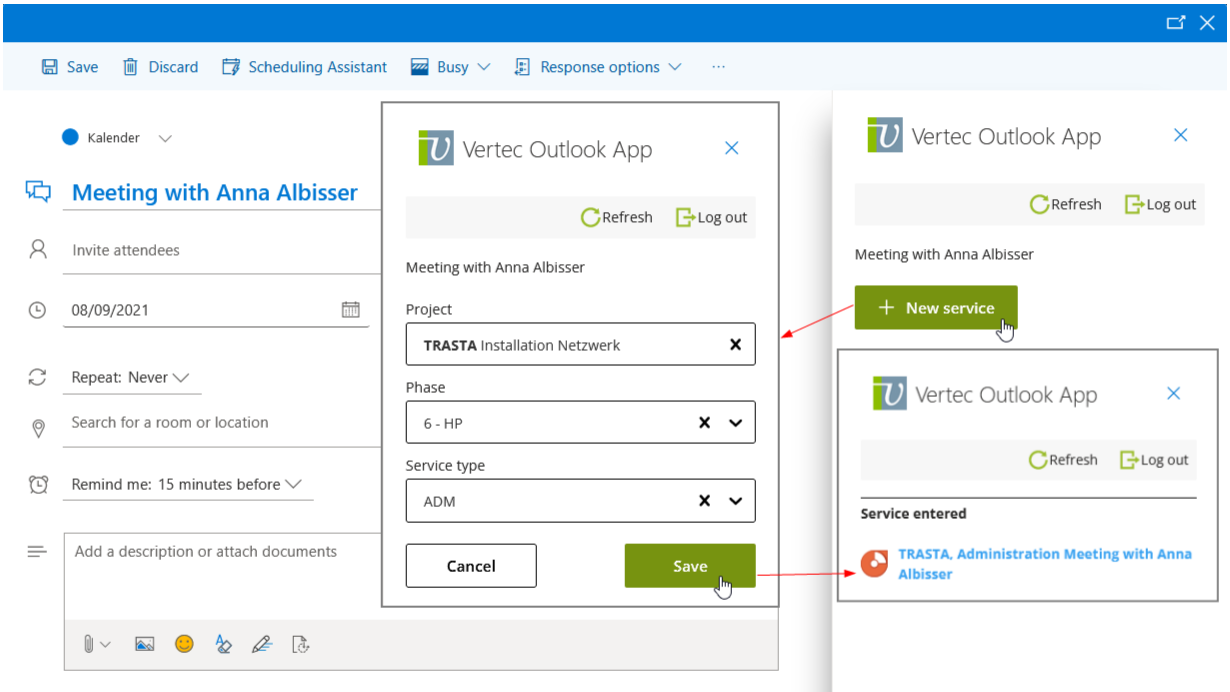


Figure 14: Create a new service

2.6 Activity history

At first, 20 activities are loaded into the activity history. If the scrollbar reaches the bottom of the list, 20 more activities are loaded. Activities are displayed in descending order of time and grouped by day. Individual entries can be marked, the selected activity will then be highlighted with a grey background. All activities belonging to a same conversation can be highlighted as well.

Use the up/down arrows to navigate through the activities. The navigation with tabs is blocked. If the selected e-mail is not within the 20 first, all activities up to the selected e-mail are loaded.

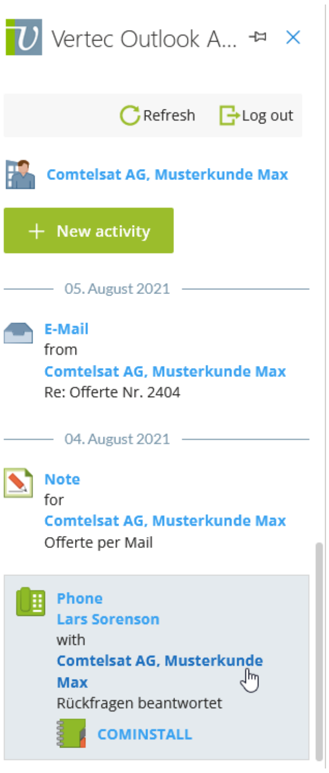


Figure 15: History of activities

Links are provided for the company, the contact or the activity. Clicking on those links will directly open the respective object in Vertec. When an activity is selected, the corresponding project is also provided with a link.

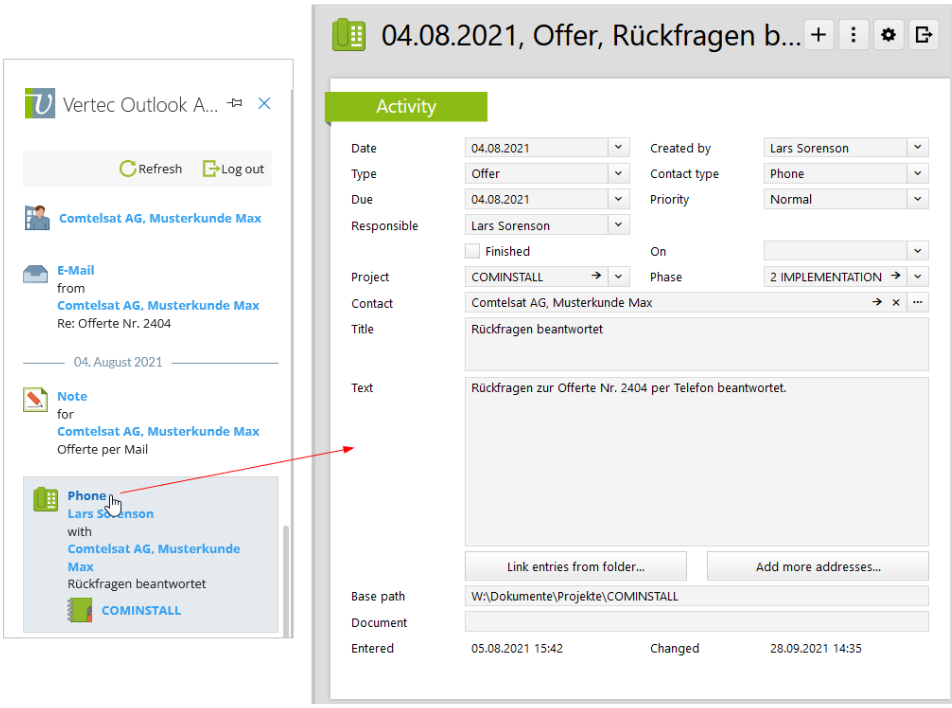
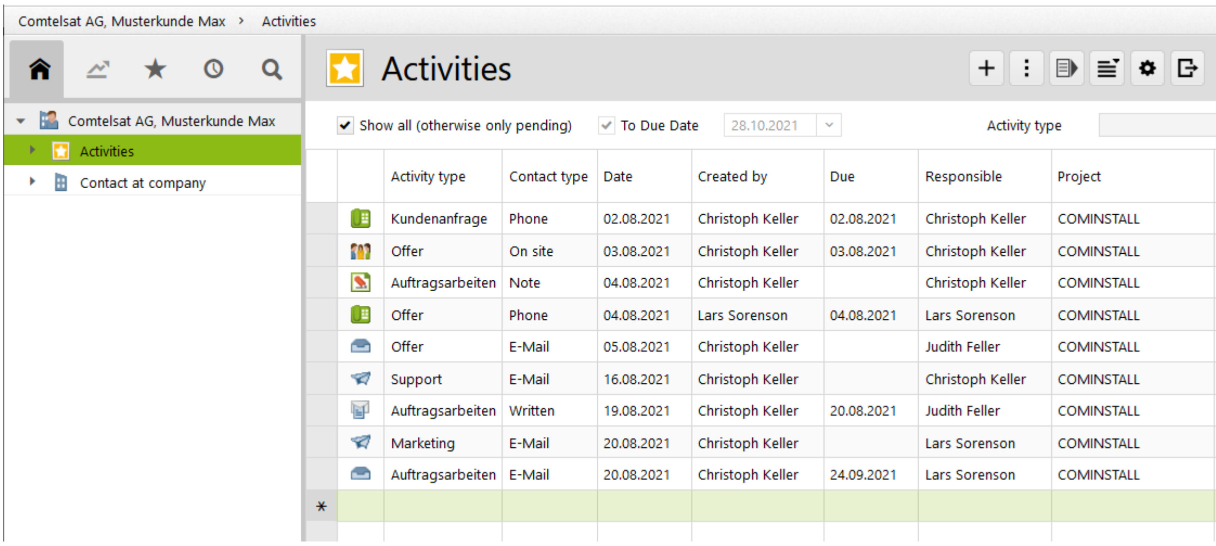


Figure 16: Link from the Outlook App to Vertec

In Vertec, the activity history is displayed in a list with the related icons:



Activity type	Contact type	Date	Created by	Due	Responsible	Project
Kundenanfrage	Phone	02.08.2021	Christoph Keller	02.08.2021	Christoph Keller	COMINSTALL
Offer	On site	03.08.2021	Christoph Keller	03.08.2021	Christoph Keller	COMINSTALL
Auftragsarbeiten	Note	04.08.2021	Christoph Keller		Christoph Keller	COMINSTALL
Offer	Phone	04.08.2021	Lars Sorenson	04.08.2021	Lars Sorenson	COMINSTALL
Offer	E-Mail	05.08.2021	Christoph Keller		Judith Feller	COMINSTALL
Support	E-Mail	16.08.2021	Christoph Keller		Christoph Keller	COMINSTALL
Auftragsarbeiten	Written	19.08.2021	Christoph Keller	20.08.2021	Judith Feller	COMINSTALL
Marketing	E-Mail	20.08.2021	Christoph Keller		Lars Sorenson	COMINSTALL
Auftragsarbeiten	E-Mail	20.08.2021	Christoph Keller	24.09.2021	Lars Sorenson	COMINSTALL

Figure 17: List of activities with the related icons

	Phone
	Incoming e-mail
	Outgoing e-mail
	Written
	Note
	Meeting in house or on site

The related icons are displayed in the tree and in the detailed view of an activity. Customizing of an icon with class settings for an activity overrides the built-in icon logic.

Next to the icon there is also a new column **Date and time** by which the list is sorted by default. Activities are given the date and time when the related e-mail was created or received.

2.7 Manage e-mails in Vertec

A click on the e-mail link in the activity history of the Outlook App directly opens the activity in Vertec.

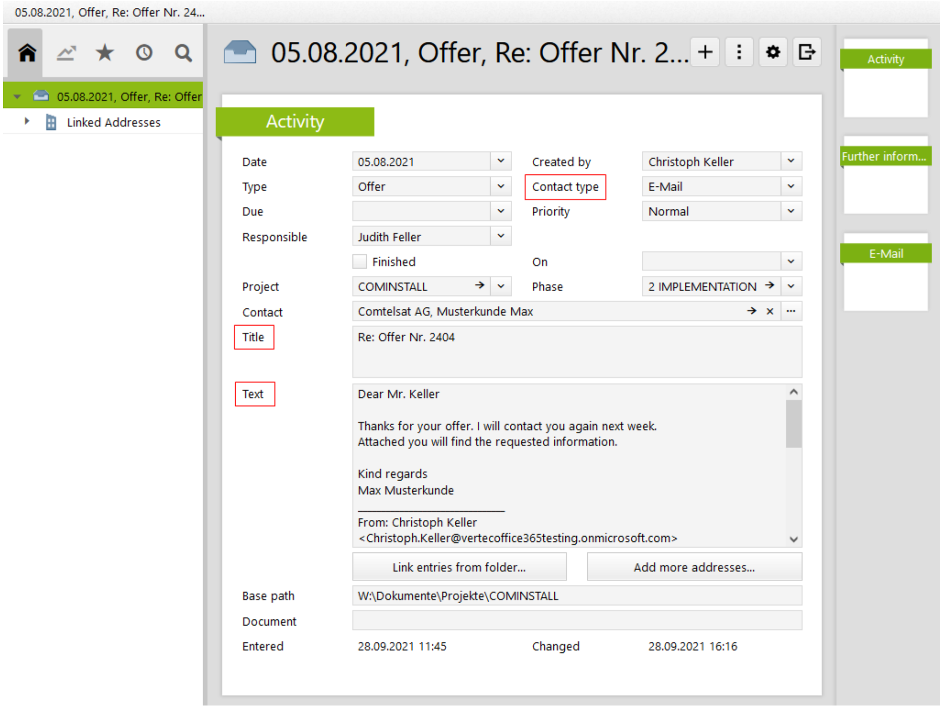


Figure 18: Contact type e-mail

Many fields are completed automatically, the rest can be completed manually. The field `Title` holds the subject of the e-mail and the field `Text` the content of the mail in plain text.

If the contact type is `E-Mail`, the e-mail section has been removed from the page Further information and a new page E-Mail has been created. Here, the fields `Subject`, `From` and `To` have been automatically completed and the attachments and the content of the e-mail are displayed as close to the original as possible.

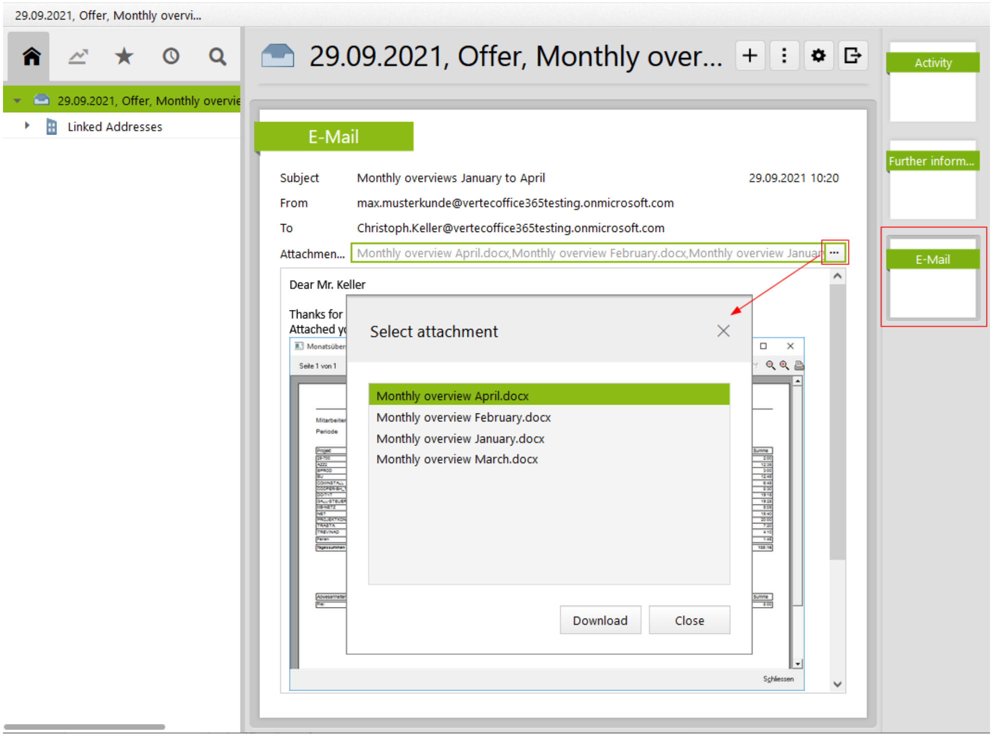


Figure 19: New E-mail page on Activity

If necessary, the attachments can be downloaded with a click on the three points. This opens the dialogue `Select attachment` with a list of all attachments. With a double click on an attachment, it can be either opened in the Web App or downloaded in the browser or stored in the download folder. The default storage path is `C:\Users\name\AppData\Local\Temp`.

From Vertec to Outlook

Inversely, the button `Actions` > `Open E-mail` allows to open the e-mail in Outlook. In the Web App, an `.eml` file is downloaded which can then be opened in Outlook.

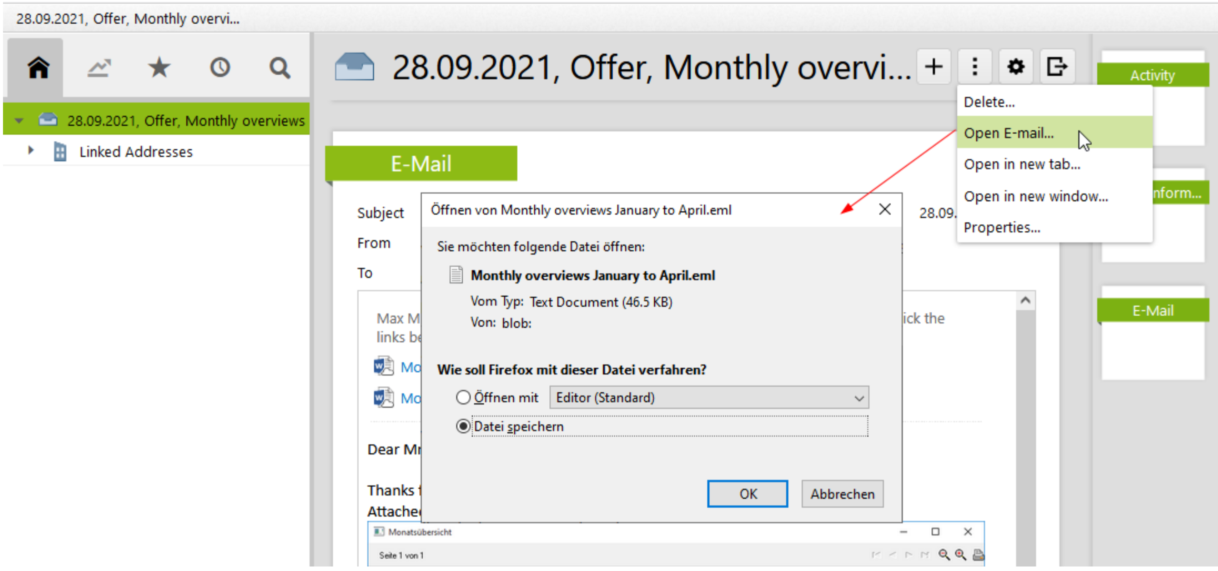


Figure 20: Open an e-mail in Outlook

If desired, an e-mail button can be created with the following Python method:

```
# send e-mail content to client as a file with ending.eml

vtcapp.sendfile(argobject.content, "email.eml")
```

2.8 Technical information

Authentication

Even though there is no checkbox `Remain connected` in the Outlook App, the Outlook App (just as the Phone App) uses this feature. After a successful authentication, the user remains connected.

CSRF Token for the Web API

The Outlook App uses an API on the server side which allows changes to the Vertec database via POST requests, for instance when inputting activities or services. Since the authentication of the client via a session cookie is established at the same time, the possibility of a cross site request forgery (CSRF) attack exists. To prevent this, a CSRF Token is generated with the session cookie at login.

Session type identification

The operator **AppType** (`TimSession.allInstances->first.appType`) returns the Outlook client type.

Automatic update

Whenever it is started the Outlook App Windows Edition checks if a new version is available in the Cloud Server. If this is the case, the user is prompted to install the update, which requires that Outlook be terminated.

Properties

– New property `OutlookAppSaveOnSend` for the function **Save on send**, which allows the automatic storage of e-mails in Vertec.
– The designation **(Outlook Addin) is added at the end of the** properties `AktivitaetstypEmail`, `EmailOrdner` and `HandleSendEmail` to distinguish the old COM Outlook Addin from the Outlook App.

New fields on activities for e-mail support

The new Outlook App bases on the schema of an activity being stored in Vertec for each e-mail. It is now possible to display activities in a list sorted by date and time, therefore to display e-mails with the same date in correct sequence.

For this reason, additional members to the class Activity have been created:

– `DateAndTime` of type **DateTime**
– `Content` of type **BLOB**
– `MessageId` of type **string**
– `ConversationId` of type **string**

New member

To allow the input of a service by means of an Outlook appointment, the new member `Leistung.AppointmentId` has been introduced.

Deactivate the Outlook App

The Outlook App can be deactivated in the Vertec.ini-File by setting `Outlook App Server` to `False` in the [Cloudserver] Section.

2.9 Security

E-mails are taken over from Outlook absolutely unchanged, including attachments. If these attachments contain malware, for instance in a Word file, this malware is taken over as received and can be downloaded by other users – just as if it were stored in the file system or sent to others in an e-mail. Vertec does not attempt to detect such malware.

3 Opportunities

Line: Expert | Modules: Services & CRM | Apps: Full-featured | Version: 6.5.0.1

Opportunities offer a simple and lightweight possibility to represent the sales process in Vertec.

Opportunities can be specified for each address entry in Vertec. An opportunity shows the possibility of a sale and gives at all times a perspective on the present status of the planned sale, including the probability, the amount and the date of the planned closing of the sale.

The input of opportunities is also possible without creating a project, thus closing the gap between the first contact with a potential customer and the opening of a project.

3.1 Activate link types

To use opportunities, the corresponding link types must first be activated. This can be done manually or by using the **plug-in Opportunities**, available in the Knowledge Base for Downloading. With this plug-in, the link types are activated and a folder for the evaluation of opportunities is created.

To activate the link types manually, the four following entries must be found in the folder `Link types`:

– Activities - Opportunities
– Client - Opportunities
– Project - Opportunities
– Seller – Opportunities

The checkbox `Active` must be set on each of these entries.

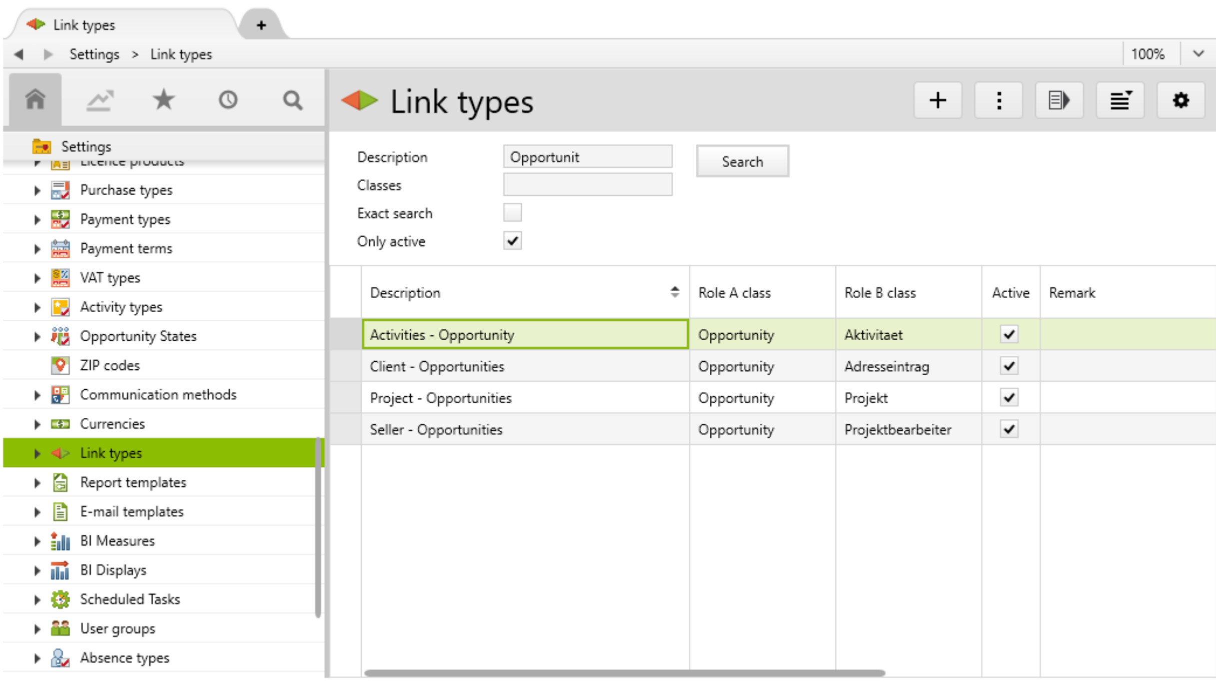


Figure 21: Link types for opportunities

3.2 Input opportunities

When the link types are activated, opportunities may be input. Open the address entry for which you want to input the possibility of a sale. Usually this is a company. In the New menu you will find the entry `Opportunity`.

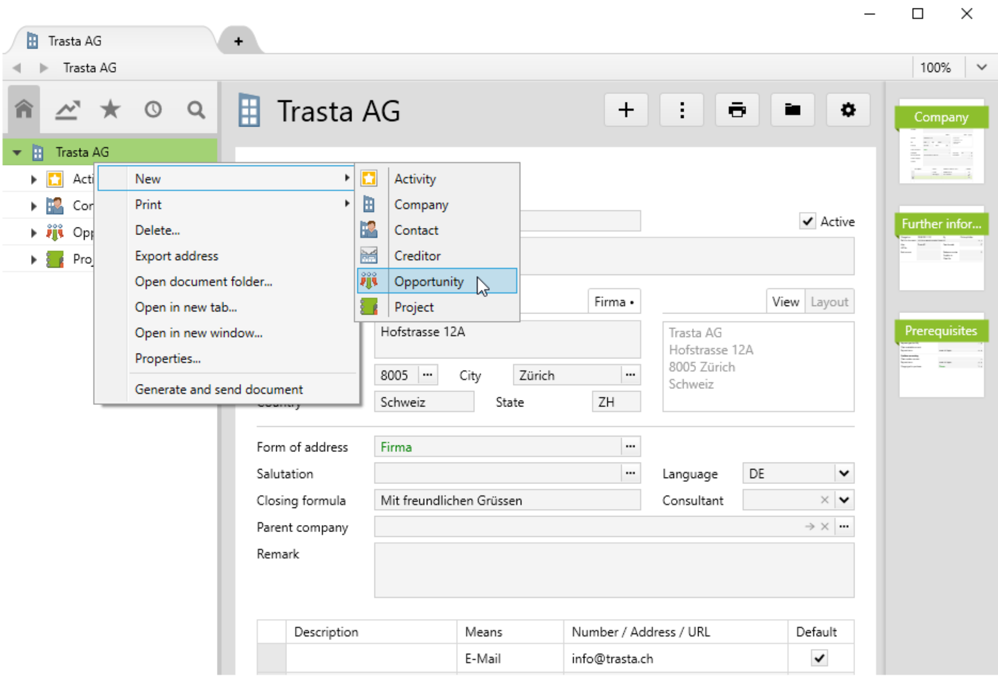


Figure 22: Create a new opportunity for a company

A click on `Opportunity` opens a window with the detail view of the new opportunity.:

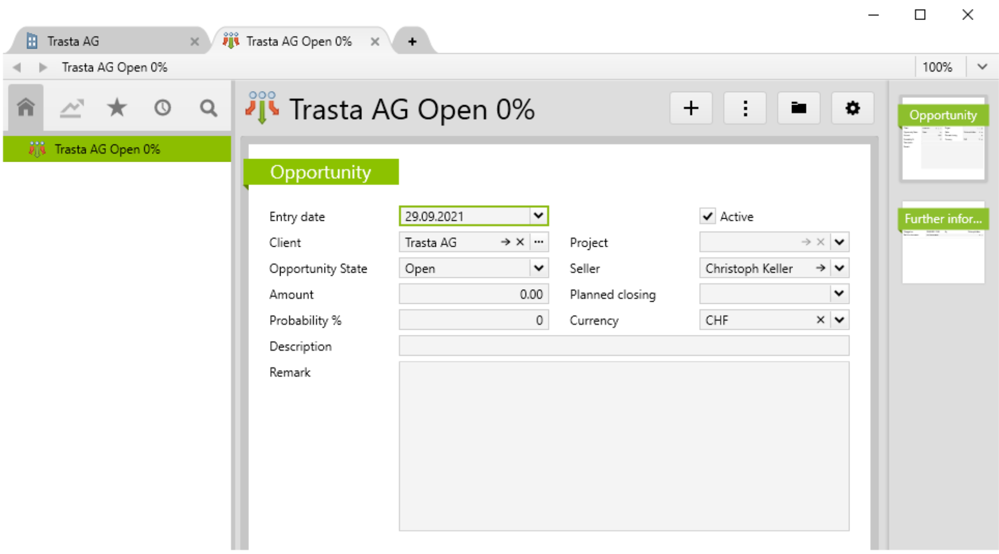


Figure 23: Detail view of a new opportunity

The following fields can now be specified:

Entry date	Pre-set: todays date.
Active	Indicates if the opportunity is still active. Won or lost opportunities are automatically set to inactive. More on this subject in section 3.3 Opportunity status.

Client	Pre-set: the selected address entry.
Project	Automatically set if a project has been created for the opportunity. Adding an opportunity to an existing project is also possible. The selection is limited to the projects of the customer.
Opportunity State	The status of the opportunity. More on this subject in section 3.3 Opportunity State.
Seller	Pre-set: the logged-in user. The person responsible for the sale. It is thus in particular possible to compute the sales volume generated by each salesperson. All users active in Vertec can be specified here.
Amount	The estimated amount of the sale.
Planned closing	The date of the planned end date of the project. The value in this field is particularly important if computations over time periods are performed.
Probability %	Probability, expressed as percentage, that the opportunity will turn into an order. With the amount a weighted amount can be calculated: multiply the entered amount with the probability. Example: With an amount of 10'000 CHF and a probability of 50 % the weighted amount is 5'000 CHF. If the probability rises to 80 %, the weighted amount becomes 8'000 CHF.
Currency	The currency in which the resulting project will be settled. The currency specified here will automatically be taken over from the opportunity to a new project.
Description	Free text to describe the opportunity.
Remark	Additional free text for remarks.

Opportunities are linked to the Client and to the Seller.

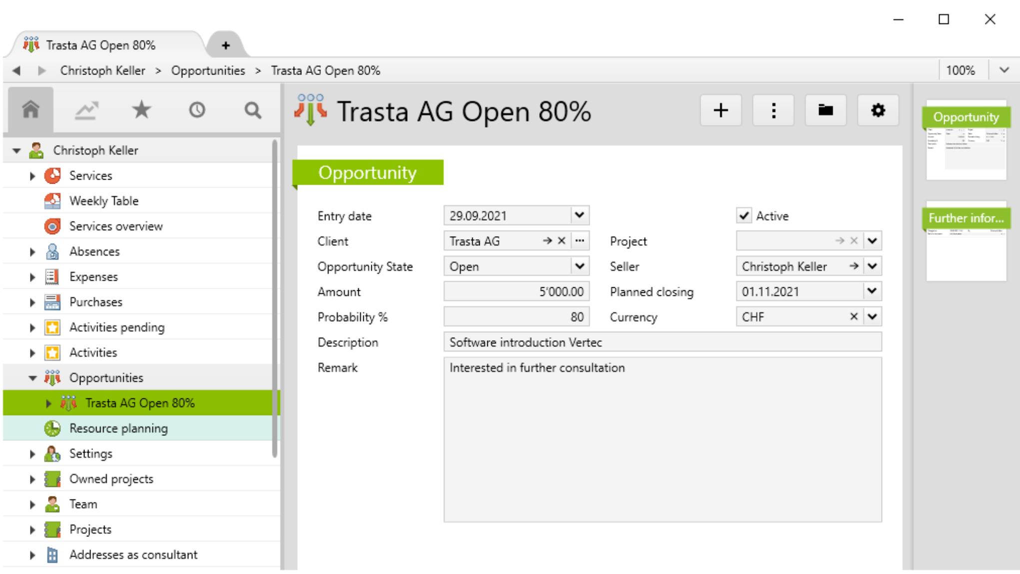


Figure 24: Link opportunity to sales person

3.3 Opportunity Status

An `Opportunity State` can be defined for opportunities. Opportunity status values are managed in a folder with the same name in the `Settings`. Administrator rights are required.

The following values for opportunity status are already defined:

– Open
– Won
– Lost

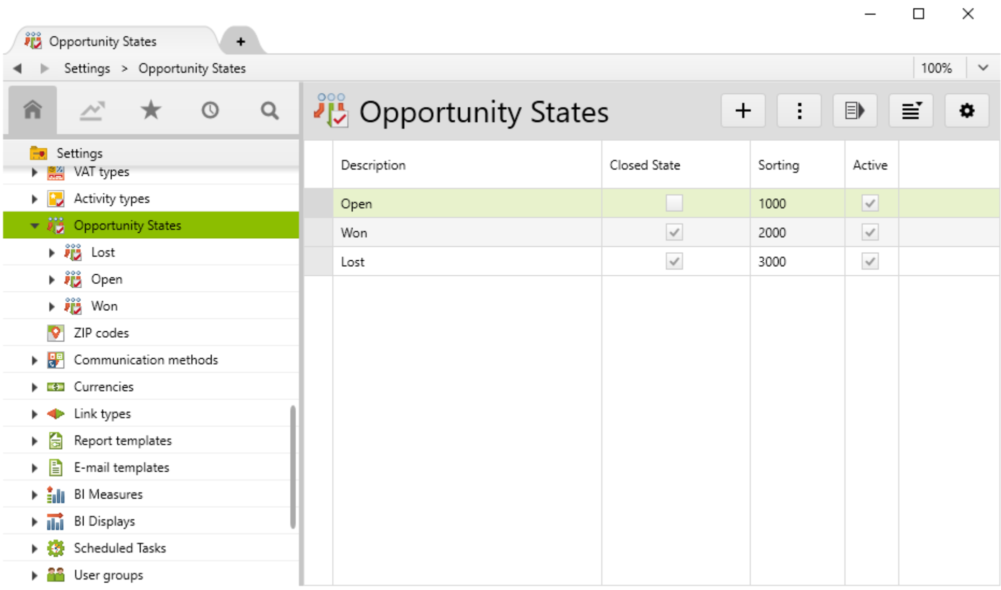


Figure 25: Opportunity status values

With the New menu, more status values can be defined if the process should be described more precisely.

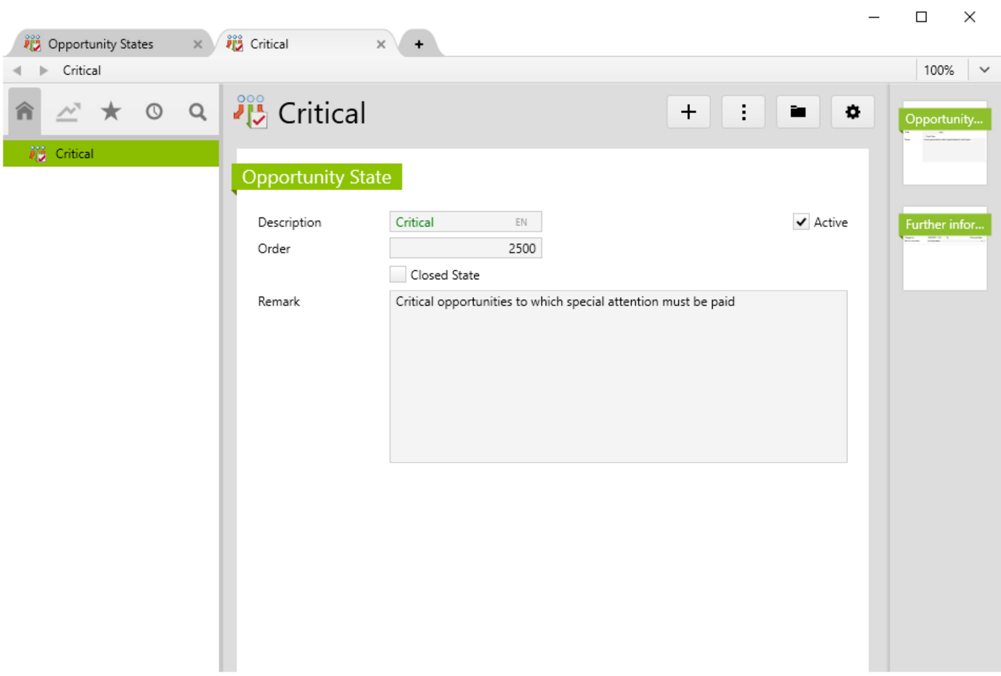


Figure 26: Detail view Opportunity status

An Opportunity status is described by the following fields:

Description	Free text to describe the Opportunity status.
Order	Specifies in which sequence in the combo box Opportunities this value will appear.
Closed State	Indicates if with this status the opportunity is closed. For the already defined values this is the case for the values `Won` and `Lost`. If the status of an opportunity is set to Closed State, the opportunity will automatically be deactivated.
Remarks	Additional free text for remarks.

3.4 Create a project from an opportunity

A project can be created from every opportunity. Select `Project` in the New menu of the opportunity.

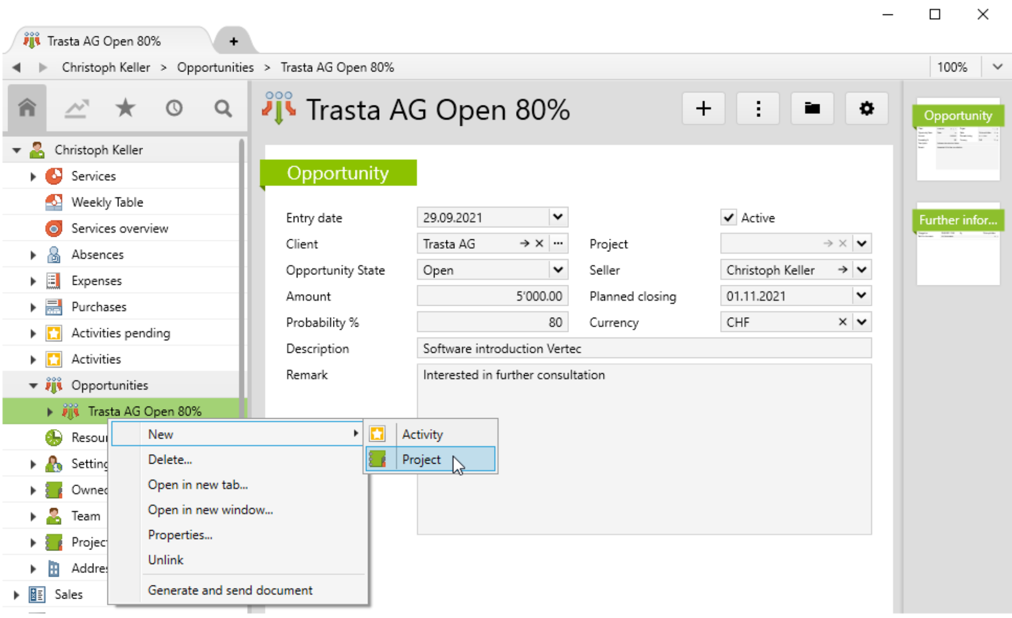


Figure 27: Create a new project on an opportunity

The detail view of the project created is then displayed. The following fields are taken over from the opportunity:

Customer	The customer of the opportunity is set as customer of the project. If you have inserted a contact as customer in the opportunity, it would be reasonable now to rather enter the company name for the project.
Currency	The currency of the opportunity is inserted as currency for the project, if it differs from the basic currency stored in the system settings.

Opportunities to a project are found under the corresponding link in the navigation domain.

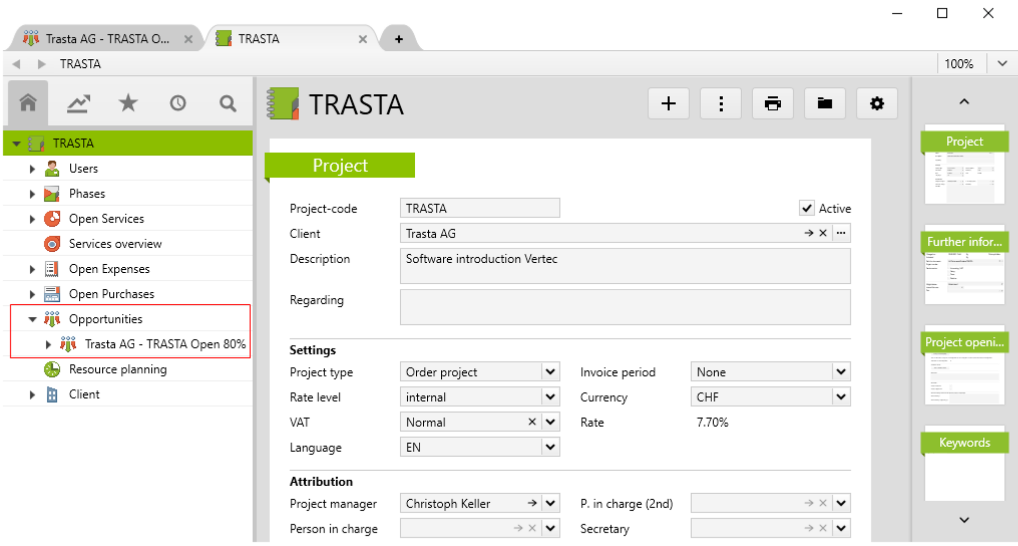


Figure 28: Link to opportunities in the project

3.5 Activities

In order to document the communication with the customer, activities can be attached to opportunities. For this purpose, click on `Activity` in the New menu.

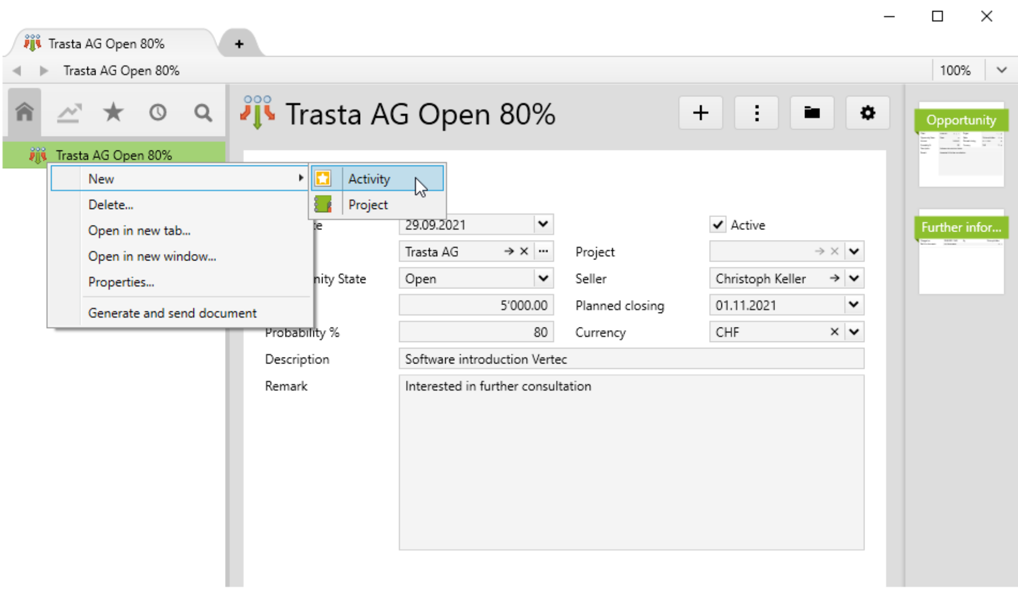


Figure 29: Create an activity linked to an opportunity

A new activity is opened which can be managed as usual.

To link an existing activity to an opportunity, select in the activity in field `Opportunity` the desired entry. The field is only present if the **plug-in Opportunities** is used. An activity can also be linked to an opportunity by drag and drop.

3.6 Access rights

All standard users can create opportunities and can manage and delete their own opportunities. Standard users have no access to opportunities of other users.

Users belonging to the user groups **Project administrator** or **Address administrator** can access, modify and delete opportunities of all users.

If one wishes to give other groups access to the opportunities, detailed access rights to opportunities can be attributed to this group in `Authorizations`.

4 Invoicing

4.1 Only active payment types in the combo box

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.10
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

The payment type ComboBox (`PaymentTypeComboBox`) now only shows active payment types for selection. This is relevant in the following places: Page **Prerequisites** on addresses, page **Accounting** on project and project type, as well as on **Payments**.

4.2 Inputting new creditors via the QR code

Line: Standard, Expert | Module: Services & CRM, Purchases | Version: 6.4.0.14
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

A creditor (supplier) with full address can directly be created out of a QR code:

Open a creditor and click on the button `Import receipt…`. A dialog for the file selection appears where you can select the invoice document. Formats PDF, PNG or JPG files are supported.

If the imported document includes a **Swiss-QR-Code**, the following information about the creditor is automatically set:

- **Account**: QR-IBAN number
- **Reference**: Reference number
- **Currency**
- **Amount**
- **Supplier**: If a company with the exact name and postal code is found in Vertec, this company is assigned. If no matching company is found, the user is asked whether a new supplier should be created.
  If the user is not allowed to open a new company an error message appears. In this case, QR codes can only be read if the address already exists in the system.
  If the creditor is already linked to a supplier and the search yields another supplier, a dialog appears with an error message. The existing supplier is not overwritten.
- **External number**: Invoice number, if encoded in the QR code.
- **Payment deadline: if encoded in the** QR code.
- **Date**: Date of the invoice, **if encoded in the** QR code.

If a document is imported although a document already exists, a dialogue appears asking whether the existing document should be overwritten. If the answer is yes and the document has a QR code, all fields are set anew according to the list above.

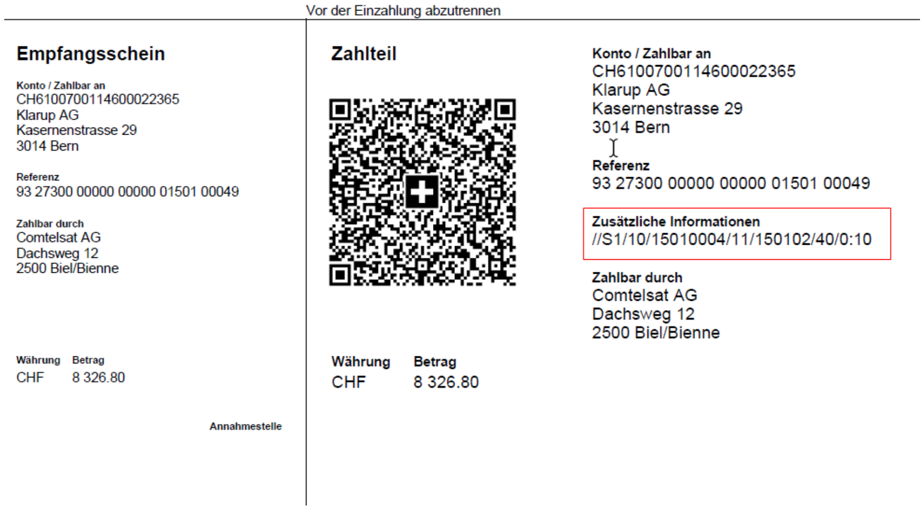
If exactly one purchase is present, the amount is written into the purchase at the same time and the count is set to zero.

If a document without QR code is imported, the document is stored but no values will be set on the creditor.

4.3 Support of the invoice date in QR codes for debtors

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.14
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

If the invoice date or the payment deadline of an invoice is known, this information is encoded into the QR code and displayed in readable form on the invoice. The invoice templates `Invoice with services list`, `Invoice with user sums` and `Invoice with phase sums` display the corresponding information under Further information:



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

Währung	Betrag
CHF	8 326.80

Annahmestelle

Zahlteil

Währung	Betrag
CHF	8 326.80

Konto / Zahlbar an
CH6100700114600022365
Klarup AG
Kasernenstrasse 29
3014 Bern

Referenz
93 27300 00000 00000 01501 00049

Zusätzliche Informationen
//S1/10/15010004/11/150102/40/0:10

Zahlbar durch
Comtelsat AG
Dachsweg 12
2500 Biel/Bienne

Figure 30: Additional information on QR invoice

4.4 Invoice reports with fixed address of the invoice

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.16
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

Invoice reports have been changed to use the fixed invoice.

4.5 Write access on verr attributes no longer possible

Line: Standard, Expert | Module: Services & CRM, Budget & Phases, Purchases | Version: 6.4.0.21
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

Various objects have so-called `verr` attributes which are filled with fixed values when the object is passed to account. Write access to these attributes is now blocked.

This concerns the following attributes:

- Project entry
  - VERRERTRAGSKONTO
  - VERRKOSTENSTELLE
  - VERRMWSTSATZ
  - VERRMWSTCODE
  - VERRMINUTENEXT
  - VERRWERTEXT
- Advance payment
  - VERRVORSCHUSSKONTO
- Invoice
  - VERRLEISTMWSTSATZ
  - VERRSPESENMWSTSATZ
  - VERRVORSCHUSSEFFEKTIV
  - VERRLEISTMWSTCODE
  - VERRSPESENMWSTCODE
  - VERRVORSCHUSSEFFEKTIVMITMWST
  - VERRTOTAL
  - VERRAUSLAGENEXT
  - VERRSPESENEXT

- VERRLEISTWERTEXT
- VERRRECHNUNGSADRESSTEXT
- VERRRECHNUNGSBRIEFANREDE
- VERRRECHNUNGSGRUSSFORMEL
- VERRIBAN
- VERRESRTEILNEHMER
- VERRESRREFERENZ27
- VERRESRPKSTELLEN
- VERRESRPLUS
- VERRBESR
- VERRBESRBANK
- VERRBESRKUNDE
- VERRCOMPANYNAME
- VERRCOMPANYORT
- VERRCOMPANYPLZ
- VERRCOMPANYADRESSE
- VERRCOMPANYLAND
- VERRCOMPANYADRESSTEXT
- VERRCOMPANYMWSTNR
- VERRPROJEKTLEITER
- VERRHB

Attempts to overwrite the values of a VERR attribute produces the following error message: `AccessDeniedError: write access denied on Object.Attribute`.

Exception: In the system context and if Vertec has been started with the parameter `/SUPER`, the attributes can still be overwritten.

4.6 Dimensions of the Swiss cross on the QR Code improved

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.22
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

The computation of the QR code has been changed so that if the full code is scaled to 46 mm, the size of the CH cross will be 7 mm.

The existing QR code has been changed for this purpose. Therefore, differences may appear even for existing uses.

4.7 QR value part in invoice reports now Postfinance compliant

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.22
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

The QR value part of a payment in the provided invoice reports has been changed to be Postfinance compliant. This applies to the reports `Invoice with services list`, `Invoice with user sums` and `Invoice with phase sums`.

4.8 Specify the payment type on the invoice

Line: Standard, Expert | Module: Services & CRM | Version: 6.5.0.1
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

The payment type can be directly specified on the invoice independently of the project. As soon as the invoice has been passed to account, the payment type can no longer be changed.

By default, the payment type of the invoice is not displayed. It can be accessed with OCL or Python to the field `xZahlungstyp` of the invoice.

4.9 Persistent Attributes in Invoices

Line: Standard, Expert | Module: Services & CRM | Version: 6.5.0.1
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

In Vertec, an invoice has so called **Totals objects**, which hold totals of services, expenses and purchases grouped and summed by VAT and account.

Most values displayed on an invoice are computed using these totals.

When invoices are still open, the totals are recomputed permanently and the values on the invoice are immediately updated to take changes into account.

When invoices have been booked, these totals, not the project entries (services, expenses, etc.), are loaded from the database.

From Vertec version 6.5 on, to improve performance when loading invoice data, the most important values of the invoice, particularly those displayed in standard reports, are directly stored in an attribute (`verr` attribute) in the database for booked invoices. This avoids the necessary SQL queries to load the totals.

Consequently, an invoice has the following new attributes:

– verrAuslagenExt
– verrSpesenExt
– verrLeistWertExt

When the attributes

– auslagenExt
– spesenExt
– leistWertExt

are queried for a booked invoice, the corresponding `verr` attributes are loaded. This logic is already used for other derived attributes on invoices, `vorschussEffektiv` for example.

For newly created invoices, these attributes are computed during the booking process. For existing invoices, these values are as before computed at execution time. However, you have the possibility with a script stored in the Knowledge Base to compute these attributes for all existing invoices. You will thus profit from the performance optimization for existing invoices as well.

5 Documents and Reports

5.1 Test-data and Execute button removed for extended Office reports

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.4
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

The field **Test-data** and the button **Execute report** have been removed from the report definition of extended Office reports since this has often caused problems, for instance when using link containers instead of folders.

5.2 Upload of documents to an activity in the Web App

Line: Expert | Module: Services & CRM | Version: 6.4.0.8
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

In the Web App, documents can now also be uploaded to an activity:

- In the file selection dialogue for an activity with a DMS document path, a local file can be selected. This will then be uploaded into the DMS. The file path corresponds to the document path of the activity, the file name to the name of the selected file.
- `vtcapp.requestfilefromclient` can be used in the Web App as well.
- In both variants, a dialog allows to select the file.

5.3 Storing PDF receipts in the Phone App

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.9
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

In the Phone App, PDF receipts for expenses can be stored in Vertec with `Send to`. The Phone App then directly opens in the Expense view with the appended PDF.

5.4 Reports without file system

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.10
Operation mode: On-Premises | Apps: Full-featured

Report templates are no longer stored in the file system but directly in the database. The standard reports are provided by Vertec as resources. Existing customer-specific report templates are imported from the file system into the database during the Update process.

- Report templates come either from the resources (provided as a standard by Vertec) or from the database (uploaded by the customer). This is valid for all types of reports, Office generated, Vertec generated and extended Office reports (Word and Excel).
- The `VertecReport.dotm` document, which holds the Office report generator, remains on the file system. It is located in the Vertec Installation folder or in the Reports directory. If the Reports folder is erased and this file was only available or made available there, a Not found message can appear. To avoid this, the file should first be moved to the Installation folder.
- Existing customer-specific report templates are imported from the file system into the database during the Update process.
- The system setting Template path for Word reports no longer exists. Scripts accessing it must be adapted.
- The menu position `Print > Select template...` has been removed.

5.5 Invoice reports without letterhead.

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.19
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

The letterhead has been removed from the following templates:

- `Invoice with services list`
- `Invoice with user sums`
- `Invoice with phase sums`

The field still exists in the code but has no effect on user specific templates.

5.6 Conversion from hours/minutes to hours decimal in Excel extended Office reports

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.22
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

In Excel extended Office reports, the display format of times has been changed from the previously used HH:MM format to hours with 2 decimal places.

Tip on the system setting `Display minutes`: The option `Hours Minutes` is ignored and displayed as `Hours Decimal`. The option `Minutes` however is displayed as minutes as before.

5.7 Description instead of Code for extended Office report Monthly overview with target times

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.10
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

The extended Office report **Monthly overview with target times** displays the description text instead of the code for absences.

5.8 Transmission of activity when storing documents in add-ins.

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.23
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

Office, Explorer und Outlook add-ins can now hand over the activity as an object when documents are stored in a DMS, SharePoint for instance. In this way, the corresponding metadata can be stored.

6 Controlling / BI

6.1 BI API

Line: Standard, Expert | Module: Services & CRM, Business Intelligence | Version: 6.4.0.14
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

As from Vertec 6.4.0.14, there is an API (Application Programming Interface) to query BI data from external applications.

This API is enabled with the parameter BI API in the **Vertec.ini** configuration file:

```
[CloudServer]
BI API=True
```

Default is `True`.

- The BI API can only be used if the Business Intelligence module is licensed. Otherwise, an error message occurs.
- The BI API can only be used after successful authentication. Otherwise, an error message occurs.
- The API is called with `/api/bi`, with the following endpoints available:

/measures
The GET endpoint `/measures` returns all available key values.

An optional parameter `f` can be specified:

- `f=csv` (`/api/bi/measures?f=csv`) returns the result as a CSV file.
- `f=json` (`/api/bi/measures?f=json`) returns the result in Json format.
  The parameter can also be specified as accept header instead of query string. If both are present, the query string is used.

Following fields are returned:

- Name
- Description
- Dimensions (string[] class name)
- Help text
- Unit

The returned values are provided in the language specified by the user in the Web App.

Example of a query

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

Returns the following values:

[{"Name":"PercentageOfCompletion","Description":"Fertigstellungsgrad","Unit":"Percentage","Dimensions":["Projekt","Projektbearbeiter","ProjektPhase"],"Helptext":"Aufwand Ist / Aufwand Budget Total"},
{"Name":"RateChargeable","Description":"Stundensatz verrechenbar","Unit":"Rate","Dimensions":["Projekt","Projektbearbeiter","Projektbearbeiter","ProjektPhase","Taetigkeit"],"Helptext":"Honorar extern / Aufwand intern"},
{"Name":"Advances","Description":"Bestand Vorschüsse","Unit":"Money","Dimensions":["Projekt"],"Helptext":"Nettobetrag aller Vorschüsse - Geleistete Vorschüsse der bis Ende Monat verrechneten Rechnungen (Valutadatum)."},
{"Name":"FeesCommenced","Description":"Angefangene Arbeiten","Unit":"Money","Dimensions":["Projekt"],"Helptext":"Honorar extern der Leistungen von produktiven Projekten, welche sich per Ende Monat nicht auf einer
Rechnung befinden oder deren Rechnung erst nachher verrechnet wurde."},{"Name":"MinutesBudgetTotal","Description":"Aufwand Budget Total","Unit":"Hours","Dimensions":
["Projekt","Projektbearbeiter","ProjektPhase"],"Helptext":"Budget Aufwand von erteilten und noch nicht abgeschlossenen Phasen"},{"Name":"MinutesBudgetRest","Description":"Aufwand Budget Rest","Unit":"Hours","Dimensions":
["Projekt","Projektbearbeiter","ProjektPhase"],"Helptext":"Aufwand Budget Total - Aufwand Ist"},{"Name":"MinutesCumulated","Description":"Aufwand Ist","Unit":"Hours","Dimensions":
["Projekt","Projektbearbeiter","ProjektPhase"],"Helptext":"Kumulierter Aufwand intern von erteilten und noch nicht abgeschlossenen Phasen"},{"Name":"FeesOrderbacklog","Description":"Auftragsbestand
Honorar","Unit":"Money","Dimensions":["Projekt","Projektbearbeiter","ProjektPhase"],"Helptext":"Budget Honorare - kumuliertes Honorar extern. Nur erteilte und noch nicht abgeschlossene Phasen"},
{"Name":"FeesOrderintake","Description":"Auftragseingang Honorar","Unit":"Money","Dimensions":["Projekt","Projektbearbeiter","ProjektPhase"],"Helptext":"Budget Honorare der neu erteilten Phasen oder Änderungen dieses
Budgetwerts während dieses Monats."},{"Name":"VacationBalanceDelimited","Description":"Feriensaldo abgegrenzt","Unit":"Hours","Dimensions":["Projektbearbeiter"],"Helptext":"Abgegrenzter Feriensaldo per Ende Monat"},
{"Name":"VacationBalance","Description":"Feriensaldo","Unit":"Hours","Dimensions":["Projektbearbeiter"],"Helptext":"Feriensaldo per Ende Monat"},
{"Name":"OvertimeBalance","Description":"Überzeitsaldo","Unit":"Hours","Dimensions":["Projektbearbeiter"],"Helptext":"Überzeitsaldo per Ende Monat"},
{"Name":"WorkingTime","Description":"Arbeitszeit","Unit":"Hours","Dimensions":["Projektbearbeiter"],"Helptext":null},{"Name":"SetTime","Description":"Sollzeit","Unit":"Hours","Dimensions":
["Projektbearbeiter"],"Helptext":null},{"Name":"Fte","Description":"Vollzeitstellen","Unit":"Percentage","Dimensions":["Projektbearbeiter"],"Helptext":"Beschäftigungsgrad"},{"Name":"TurnoverOutlays","Description":"Umsatz
Auslagen","Unit":"Money","Dimensions":["Projekt","Projektbearbeiter"],"Helptext":"Auslagen verrechnet (Valutadatum der Rechnung)"},{"Name":"TurnoverExpenses","Description":"Umsatz Spesen","Unit":"Money","Dimensions":
["Projekt","Projektbearbeiter"],"Helptext":"Spesen verrechnet (Valutadatum der Rechnung)"},{"Name":"TurnoverServices","Description":"Umsatz Leistungen","Unit":"Money","Dimensions":
["Projekt","Projektbearbeiter"],"Helptext":"Total nach Rabatt verrechnet (Valutadatum der Rechnung)"},{"Name":"MarginContributed","Description":"Deckungsbeitrag (DB)","Unit":"Money","Dimensions":
["Projekt","Projektbearbeiter","Projektbearbeiter","ProjektPhase","Taetigkeit"],"Helptext":"Honorar extern verrechnet - Kosten"},{"Name":"FeesExtWrittenoff","Description":"Honorar extern
abgeschrieben","Unit":"Money","Dimensions":["Projekt","Projektbearbeiter","Projektbearbeiter","ProjektPhase","Taetigkeit"],"Helptext":"Honorar intern verrechnet - Honorar extern verrechnet"},
{"Name":"FeesExtOpen","Description":"Honorar extern offen","Unit":"Money","Dimensions":["Projekt","Projektbearbeiter","Projektbearbeiter","ProjektPhase","Taetigkeit"],"Helptext":null},
{"Name":"FeesExtCharged","Description":"Honorar extern verrechnet","Unit":"Money","Dimensions":["Projekt","Projektbearbeiter","Projektbearbeiter","ProjektPhase","Taetigkeit"],"Helptext":null},
{"Name":"Costs","Description":"Kosten","Unit":"Money","Dimensions":["Projekt","Projektbearbeiter","Projektbearbeiter","ProjektPhase","Taetigkeit"],"Helptext":null},{"Name":"MinutesInt","Description":"Aufwand
intern","Unit":"Hours","Dimensions":["Projekt","Projektbearbeiter","Projektbearbeiter","ProjektPhase","Taetigkeit"],"Helptext":null},{"Name":"MinutesExt","Description":"Aufwand extern","Unit":"Hours","Dimensions":
["Projekt","Projektbearbeiter","Projektbearbeiter","ProjektPhase","Taetigkeit"],"Helptext":null},{"Name":"FeesInt","Description":"Honorar intern","Unit":"Money","Dimensions":
["Projekt","Projektbearbeiter","Projektbearbeiter","ProjektPhase","Taetigkeit"],"Helptext":null},{"Name":"FeesExt","Description":"Honorar extern","Unit":"Money","Dimensions":
["Projekt","Projektbearbeiter","Projektbearbeiter","ProjektPhase","Taetigkeit"],"Helptext":null}]

/getdata

The GET endpoint `/getdata` returns the BI data. It accepts the following parameters as query string:

measures: List of the internal key values separated by commas (e.g., `"FeesExt,MinInt"`). Required.

dimension: List of the dimensions by which the result is grouped (e.g., `"Projekt,Projektbearbeiter"`. Required, case-insensitive.

- `"Month"` may also be specified as pseudo-dimension. Reference date values such as OvertimeBalance are only accepted for time series. For such values, the dimension=month must be specified, otherwise an error message appears.
- If several roles are present, the access can be made using a suffix [Index 0 .. 4]:
  - `Project` functions as before and selects the dimension Projekt without role.
  - `Project_1` selects on the key value the dimension Projekt at index 1. This must be a role on the project.
  - Example for a query of `FeesExt` (External fees):

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

  - If several key values with different "dimension structures" are queried and the requested dimensions cannot be provided for all key values, an error message appears. In such a scenario, key values with the same dimension structure must be bundled into a query and several queries must be specified to extract all the data.

startDate: Start date in ISO 8601 format (2020-01-01), optional.

endDate: End date in ISO 8601 format (2020-12-31), optional.

selectExpression: OCL-Expression for the selection, optional.

useDescriptions: True or False. Determines if the column headings are provided in plain language and translated, or with their internal names, independently of the language. Optional, default: False.

The End point yields the results in following format:

- Header (column names)
  - **Name of the requested key value (internal name). If parameter** `useDescriptions = true`, the plain language designation of the key value is returned, translated into the actual language of the session, instead of the internal name.

- Data row
  - Dimension value
  - Value of key value

The BI authorizations are applied.

Example: FTE query for Q1 2020

```
api/bi/getdata?measures=fte&dimension0=Projektbearbeiter&useDescriptions=true&dimen
sion1=month&startDate=2020-01-01&endDate=2020-03-01
```

Result:


```
[{"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}]
```


Support of HTTP basic authentication

HTTP basic auth with user name and password is provided for queries on external BI software.

- The started session has a default timeout of 5 minutes. This can be changed in the [Cloud Server] section of the **Vertec.ini-File** with `XML Session Timeout = 5` (BI API and XML Server share the same timeout value).
- When timeout is reached, the session is closed and the process disappears.

– If during this time, another query with basic authentication and the same user occurs, the existing session is reused.

6.2 Storing BI views as favourites

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.9
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

BI views can be stored as favourites. When are selected, they are shown with all parameters. BI views stored as favourites have a designation in the form of `Node (BI): Displayname`, for instance `Project (BI): Projectmember`. In this way, BI views can be easily identified in the favourites.

6.3 Extended BI URL

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.9
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

URL support for Business Intelligence (BI) displays has been extended as follows:

– Date range
– Selected display
– Selected key values including sorting and colours

A URL will for instance look like this:

```
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 Authorizations for BI key values

Line: Standard, Expert | Module: Services & CRM, Business Intelligence | Version: 6.4.0.14
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

Authorizations can now be assigned to individual BI key values.

The following applies:

– The rights on BI data can be assigned on classes as well as on objects.
– As soon as the rights on BI data have been specified and the corresponding class is selected (if an access right to all classes is desired, use `UserEintrag`), a key value can be selected at **Field name**.
– In the field **Expressions**, the corresponding printouts can, as before, be specified for a selected class.
– The BI Button becomes active if the user has access to at least one key value of the selection. For a single object, at least one key value must be available; for a list of objects, every object must at least have one key value.
– If, when selecting objects, every object has at least one key value, but not a key value for which authorization exists on all objects, the display **No data available** appears.
– If, for a selected object, no authorization for any key value exists, the display **Access denied** appears.

In this context, the OCL operator `hasBiDataRight` has also been adapted. It now expects as argument a string with the internal name of the key value.

Example:

`projekt->hasBiDataRight('MinutesExt')` checks if the logged-in user has access rights to the key value MinutesEx for the given project.

Backwards compatibility
This feature is not backwards compatible. All calls to this function must be adapted.

7 Customizing

7.1 List with open and booked services for the input of services

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.19
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

Up to now, services were stored under **Open services** and no longer displayed after booking. From this version on, the folder for the project member is called **Services**. When a service is booked, it remains visible in the list of services but changes can no longer be made and the timer symbol vanishes. The same thing applies to the user containers for expenses and purchases.

7.2 Support for fixed columns in list settings

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.8
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

The first columns of lists can be fixed by checking the box `fixed` in the list settings.

- The fixed columns remain unchanged on the left during scrolling.
- The fixed columns are marked with a separation line and remain unchanged during scrolling.
- The first consecutive fixed columns are taken into account. If a column is marked as fixed and not all preceding columns are fixed, the setting is ignored.

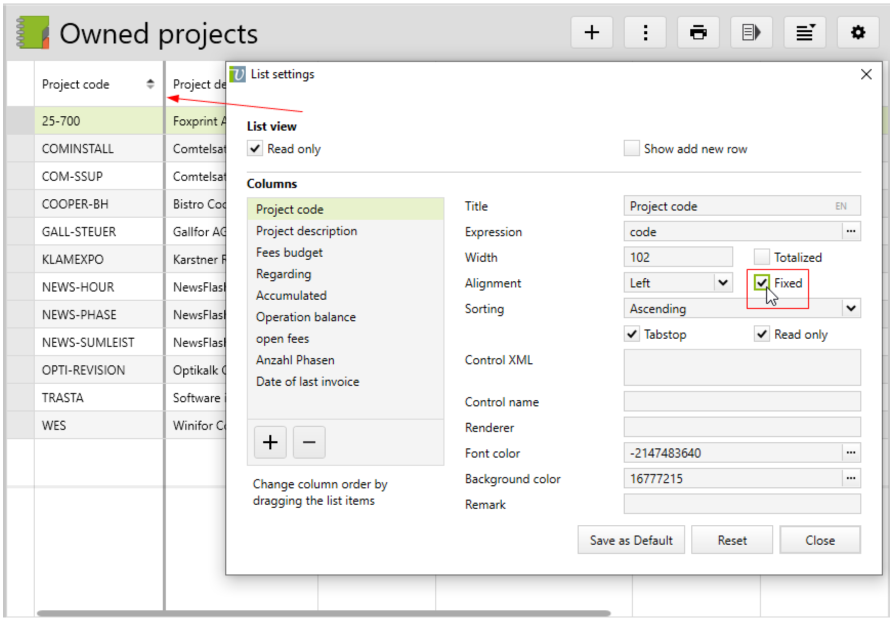


Figure 31: Fixed columns

7.3 Icon representation in lists

Line: Standard, Expert | Module: Services & CRM | Version: 6.5.0.1
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

In the Activity list, there is a new column in which the icons of the corresponding activity types are displayed:

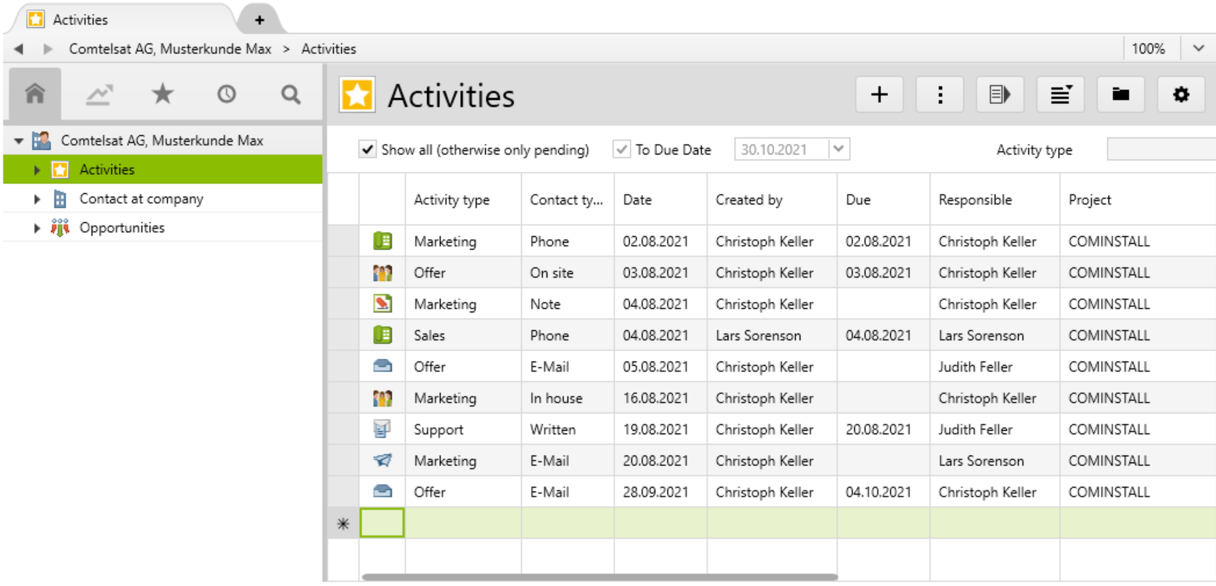


Figure 32: Icon representation in lists

Icons can be displayed in columns in other lists as well. This makes particularly sense when objects of different classes are represented in a list. Use in this case `iconindex` as column expression. In addition, the renderer `rndIcon` must be specified.

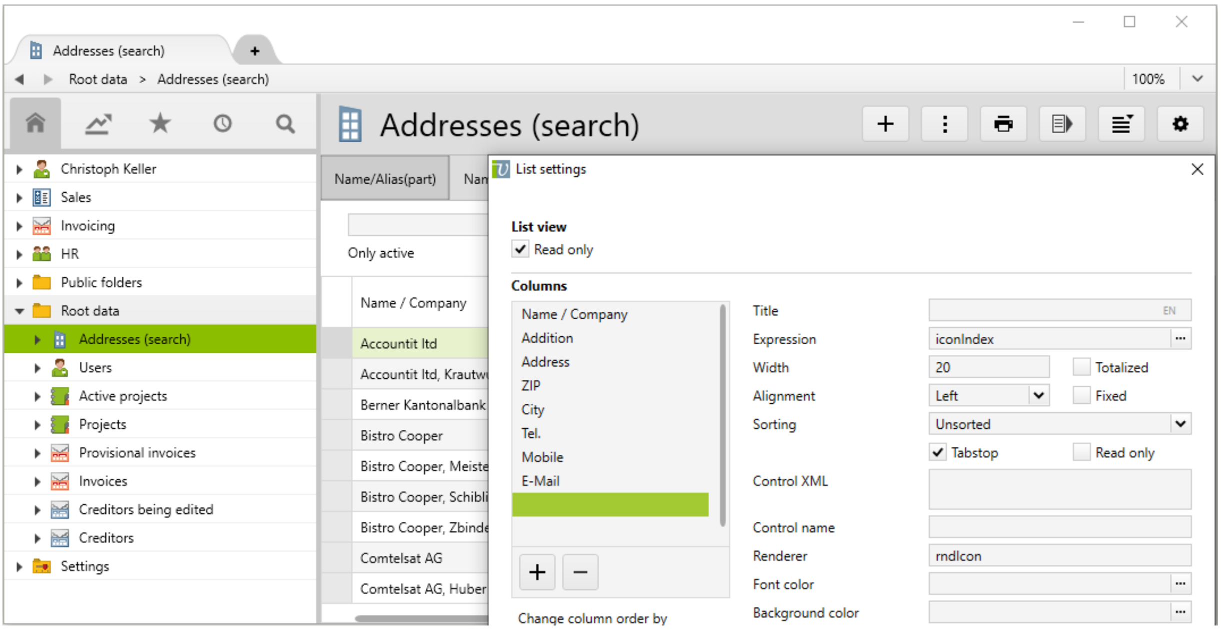


Figure 33: List settings for icon display in lists

7.4 Icon for IX folders

Line: Expert | Module: Services & CRM | Version: 6.4.0.5
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

If a folder is marked **Inclusive/Exclusive**, the IX Icon always appeared, even when an icon index had explicitly been specified. From now on, the specified icon is used, the IX-Icon only in the case of `-1` (no value).

7.5 Icon index for subfolders on folders

Line: Expert | Module: Services & CRM | Version: 6.4.0.6
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

Folders receive the property **Icon for sub folder**. If an icon index is set there, every new subfolder created automatically receives this icon.

7.6 Class-specific sorting of entries in the tree

Line: Standard, Expert | Module: Services & CRM | Version: 6.5.0.1
Operation mode: Cloud Abo / On-Premises | Apps: All

Sorting in the tree can be set to be class-specific in the following way:

- **Activities** are sorted in the tree in descending order of date and time. The date column appears before the time.
- **Project entries** (services, expenses, purchases) are sorted by date in ascending order. Within the same date, the object id is also taken into account.
- **Project phases** are sorted in ascending order by project code and `orderidx`.
- **Invoices** are sorted by date and number in ascending order.
- **Creditors** are sorted by date and number in ascending order.
- **User Settings** are sorted by date in ascending order. Within the same date, the object id is also taken into account.
- **Offers** are sorted by date and number.
- **Opportunities** are sorted by date. Within the same date, the object id is also taken into account.
- **Absences** are sorted by date. Within the same date, the object id is also taken into account.

Sorting can be specified individually for each class by means of an OCL expression in **Class settings**.

7.7 Python method for creating an e-mail in Outlook

Line: Expert | Module: Services & CRM | Version: 6.4.0.4
Operation mode: Cloud Abo / On-Premises | Apps: Desktop App, Cloud App

A new method `vtcapp.createoutlookmail()` is now available in Python to generate via Outlook an e-mail on the client.

Precondition: an installed Windows Outlook client.

The syntax is follows:

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

- **to**: string. Recipient(s) of the mail, separated by `;` if there is more than one.
- **subject**: string. Subject of the e-mail.
- **body**: string. The text of the e-mail. Can be HTML or plain text. HTML is recognized as such and will then be generated as an Outlook HTML mail.
- **cc**: string, optional, keyword. Cc-addresses, separated by `;` if there is more than one.
- **bcc**: string, optional, keyword. Bcc-addresses, separated by `;` if there is more than one.
- **attachments**: list of tuples, optional, keyword. List of attachments as tuples of two strings [(name of the file, contents)]
- **show**: boolean, optional, keyword. If `true`, the e-mail is created and displayed, if `false`, the e-mail is stored in the folder "Drafts" and not displayed. Default is `true` if not specified otherwise.

  Keyword means that the optional values can be specified with the parameter as keyword, e.g., `CC="abc@vertec.com"` This avoids to specify commas as placeholders for unused optional values.

Further information Examples can be found in the article Vertec Python Funktionen in the Online Knowledge Base under www.vertec.com/kb/pythonfunktionen/.

7.8 Comment field in list settings

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.4
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

In the list settings there is now a Remark field for every column:

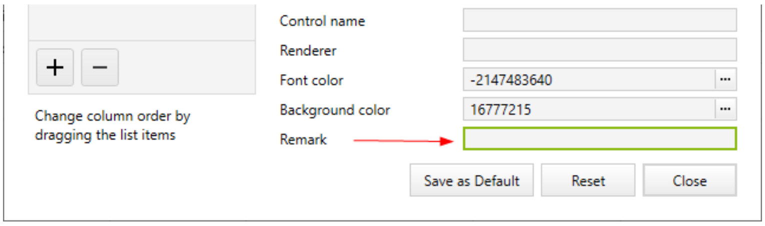


Figure 34: Remark field in the list settings

This field can for instance be used for the internal documentation of the corresponding column.

7.9 Python method executereport produces report output as byte string.

Line: Expert | Module: Services & CRM | Version: 6.4.0.8
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

Better support in Python scripts for reports without access to the file system: The Python method executereport generates the report output as a byte string. The output can be further processed and, for instance, be used as attachment to an e-mail via `vtcapp.createoutlookmail()`.

7.10 Create system settings using Config Set

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.10
Operation mode: Cloud Abo / On-Premises | Apps: Desktop

Until now, system settings could be changed using Config Set but could not be created. System settings can now be passed as normal objects to Config Set with their properties.

- **To** change existing system settings, use `system-settings` and specify the corresponding lines in the Config Set Builder interface.
- To create new system settings, insert these objects directly into XML after having exported the Config Set first. System settings do not possess an Entry Id and are thus identified by their name (`PropertyName`).
  The system settings are directly inserted as objects. If such an imported config set is later exported, the system settings object remains as an object in the XML file. The manual input must therefore be performed only once.

Further information An example of this procedure and further information are available in the article Config Set XML in the Online Knowledge Base under https://www.vertec.com/ch/kb/config-set-xml/#new-system-settings.

7.11 vtcapp.executefile() supports command line parameters

Line: Expert | Module: Services & CRM | Version: 6.4.0.10
Operation mode: Cloud Abo / On-Premises | Apps: Desktop

The Python method `vtcapp.executefile()` now also supports command line parameters. Optional arguments may be passed. The existing simple case still works.

Parameter parsing is as follows:

– No parsing occurs in `command`. Spaces are interpreted as being a part of the path.
– The parsing occurs in `argumentString`, spaces separate arguments unless they lie within double quotes "".
Example:

```
vtcapp.executefile('notepad++.exe', r'-lpython "C:\Documents\My script.py"')
```

7.12 Improvement of display conditions for containers in the BI tree

Line: Standard, Expert | Module: Services & CRM, Business Intelligence | Version: 6.4.0.14
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

The criteria defining who sees which folder in the BI tree have been adapted. The objects and folders available in the BI tree view are determined as follows:

First, it is checked if the BI data access right on the class exists:

– A project member without general BI data access right to the contents class and without project manager, project administrator or administrator rights will not see any container in the BI tree ("All addresses", for example).
– In particular, the **Address administrator right** does not imply any BI data right. A user with address administrator right has not, because of that, BI data access right on address entries. This means: to be able to see the necessary BI data, the user should, in addition to the BI data right for entering addresses, also have **Project supervisor right** or similar.
– A user with **Project manager right** will see all project and phase containers.
– A user with **Team manager right** **will see** all team member, project and phase containers.
– A user with **Project admin right** will see all team member, project, phase, address entry and activity containers.
– An **Administrator** will see all defined containers active for a BI view.

In a second step, all rights explicitly given the user are checked. Of interest here are only the rights of type BI data on the matching classes (inheritance is taken into account, meaning that the access right to an address has consequences for a query on company.

– If the access right exists, the BI data right is granted, regardless of what has been specified as `Field name` (key value) or `Expression`.
– If the access is denied, Field name and Expression must be empty (since these values act at the level of the object, but the basic authorization for the display of the tree must be decided at the level of the class). If the access is denied and Field name or Expression are set, then the authorization for the display is ignored.

The check, as always with rights, is done level upon level. Meaning that if, basically, the BI data right for entries has been granted, but the right to the BI data of the projects has been denied, the user will have no access to the projects.

Important: the description above only covers how what can be displayed and selected from the tree is determined, not the actual "authorization" on the data.

7.13 Python stub files for built-in modules

Line: Standard, Expert | Module: Services & CRM, Purchases | Version: 6.4.0.15
Operation mode: On-Premises | Apps: Full-featured

Code editors (VSCode for instance) today have good static capabilities for analysis which make it possible to detect many mistakes before execution. Built-in Python modules not available in the file system however constitute a problem. To make them available to the Type Checker, stub files of the built-in Python modules `vtcapp`, `vtcextensions` and `ziputils` are now provided with Vertec.

The stub files are located in the sub-folder **PythonStubs** of the Vertec installation folder or can be downloaded from the Online Knowledge Base under https://www.vertec.com/kb/pythonstubfiles/.

7.14 Checkbox for keyword folders and ordinary folders in lists

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.19
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

Keyword attributions can now be directly represented and managed in lists. This applies to keyword folders and ordinary folder structures not marked as IX folders. In the column of the list, the corresponding subfolders (the keywords) are displayed and the contents may be specified or deleted with a click on the checkbox. This is done with the renderer `rndKeywordFolder`.

7.15 Link roles in GetLinks can be identified by means of entry Id

Line: Expert | Module: Services & CRM | Version: 6.4.0.19
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

GetLinks operators and methods have been enhanced in such a way that the link role to be used can be identified not only by its designation (as before) but also by the **Entry Id** of the link type.

Specify for the role the Entry Id of the link type, followed by `_A` for role A or `_B` role B, depending on the side of the target in the link type:

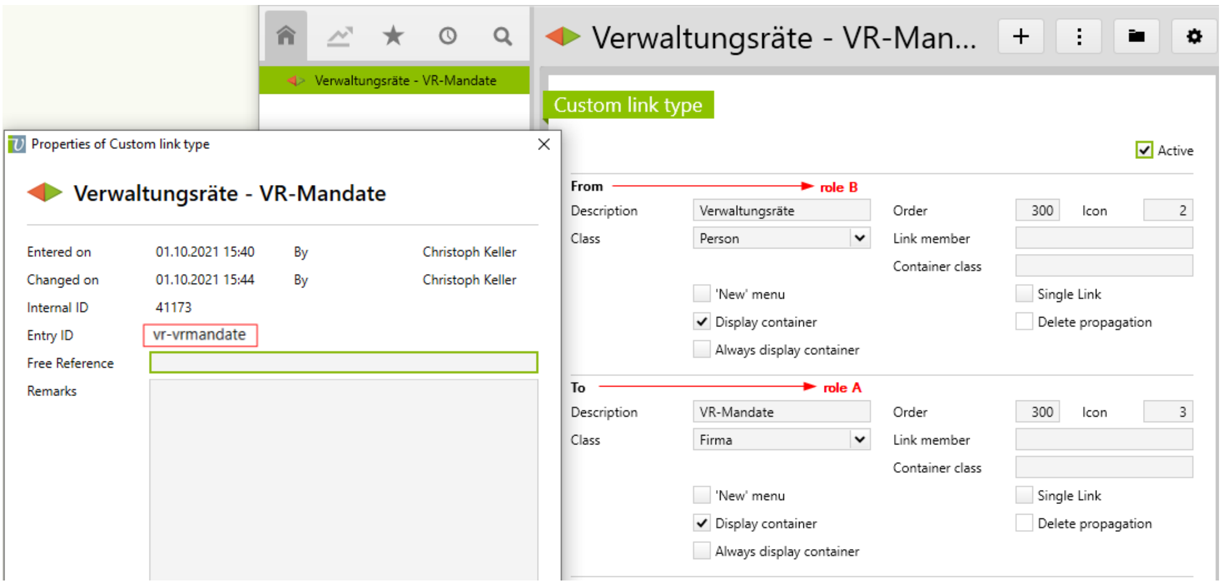


Figure 35: Create a new opportunity from a company

In Python this looks as follows:

Object: Person
Target: Firma
	`person = argobject`
	`firma = vtcapp.getobjectbyid(2995)`
	`person.linkto(firma, "vr-vrmandate_A")`

Object: Person
Target: Firma
	`person = argobject`
	`firma = vtcapp.getobjectbyid(2995)`
	`firma.linkto(person, "vr-vrmandate_B")`

7.16 OCL translate operator

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.20
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

As with Python, there is now a Translate Operator for OCL. It allows with `.translate` to translate a string value in an OCL expression into the actual Vertec interface language.

Example:

`if projects->size > 0 then 'Client'.translate else 'Address'.translate endif` on an address list produced for German, the following result appears in column Status:


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

Figure 36: Example of the translate operator

7.17 Access to container and parent object with OCL expressions in lists

Line: Expert | Module: Services & CRM | Version: 6.4.0.21
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

It is now possible for OCL expressions in lists to access with `varContainer` the container (folder or link container) and with `varParent` the object to which the container belongs (parent object of the link container). In addition to the column expressions in the list, `ListExpression` and `GhostRowListExpression` of a control have access to these context variables.

7.18 Selection dialogues callable from Python

Line: Expert | Module: Services & CRM | Version: 6.4.0.21
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

The various search and selection dialogues for addresses, projects, phases and the tree can now be called directly from Python. The following new methods of the `vtcapp` module are available for that:

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

The methods support the following optional keyword arguments:

- **selectaddress(string title, string classname, string filter)**
  - title: Dialog title (Default: select an address)
  - classname: Address class in which the search is performed. e.g., `Company` (Default: address entry)
  - filter: additional SQL filter condition (Default: empty), e.g.: only for active objects.
    The search dialogue for addresses uses the standard search definition according to system settings.
- **selectphase(string title, string filter)**
  - title: Dialog title (Default: select phase)
  - filter: additional SQL filter condition (Default: empty), e.g.: only for active objects.
- **selectproject(string title, string filter)**
  - title: Dialog title (Default: select project)
  - filter: additional SQL filter condition (Default: empty), e.g.: only for active objects.
- **selectobjectintree(string title, list entrylist, string browsefilter, string selectfilter)**
  - title: Dialog title (Default: select object)
  - entrylist: List of Vertec objects represented in the dialogue as root elements.
  - browsefilter: comma-delimited filter for object types (e.g.: `folder`, `expressionfolder`)
  - selectfilter: the select filter determines on which entries (classes and their subclasses) one can confirm the dialogue and therefore “select“ them; comma-delimited as well.

Sample code:

```
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 method returning the activity

Line: Expert | Module: Services & CRM | Version: 6.4.0.22
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

There is a new Python method `executereport2()` which returns the report as well as the activity.

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

The returned value of the method is a tuple (reportdoc, activity) where

– **reportdoc** is the content of the resulting report document (byte string).
– **activity** is the applied activity object or none.
The method is called by assigning the returned value to 2 variables:

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

Further information Further information is available in the article Vertec Python Funktionen in the Online Knowledge Base under www.vertec.com/kb/pythonfunktionen/.

7.20 Python function for payment settlement

Line: Expert | Module: Services & CRM | Version: 6.4.0.22
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

The new Python function `vtcapp.syncpayments()` performs a global payment settlement. If no debtor interface with payment settlement is installed, an error occurs.

The function can for instance be registered as a planned task.

7.21 Script editor opens only with administrator rights

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.23
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

The script editor is only displayed when the logged-in user owns Administrator rights. Otherwise, an error message is displayed.

7.22 Python update

Line: Standard, Expert | Module: Services & CRM | Version: 6.5.0.1
Operation mode: On-Premises | Apps: Full-featured

From Vertec 6.5 on, the Python Version used changes from 2.7.16 to version 2.7.18. A detailed description of the changes is available in the corresponding Release Notes.

7.23 Activity types are multi-language enabled and active

Line: Standard, Expert | Module: Services & CRM | Version: 6.5.0.1
Operation mode: Cloud Abo / On-Premises | Apps: Alle

Activity types still have the normal text field `Description`, which is now however displayed as code on the interface. For reasons of backwards compatibility, the field name is still **Description**.

At the beginning, there is now the text field `Code` and the check box `Active`. For the standard string representation of activity types, the new multi-language field `Description` is used.

Existing activity codes receive in the field `Code` the contents of the designation and are `Active`.

7.24 Translation of configuration data in the business logic

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.9
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

To make the standard configuration objects available in all languages, Vertec automatically translates the string representation to the desired interface language.

Since up to now Vertec did this on the interface level at runtime, problems sometimes occurred with user data if by chance they bore the same name. Thus PM, for instance, was displayed as PL in a German Vertec because PM was interpreted by the translation mechanism to mean "project manager".

For this reason, the translation of configuration objects was moved to the business logic. Are translated exactly the string representations which have an entry Id. The string representations of other objects are not translated. If one wishes to display the translated string representation of an object, it is done with the expression `self`.

7.25 Support of multiple class settings

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.17
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

Multiple class settings can exist in parallel for a class. This is principally relevant for plug-ins to avoid conflicts between imported class settings and existing customer-specific adaptations. Class settings are taken into account as follows:

- The order of preference in case of different class settings is determined by sorting of the internal Id. Higher Ids correspond to newer objects.
- The latest class settings are used for the following properties:
  - Document path
  - Icon index
  - Display class name
  - Standard display
  - Sort order in the tree
- For standard list settings, the oldest class settings are always used. The button `Save as Default` in the list-settings editor always stores values in the oldest class settings for this reason.
- If customizing of pages has occurred for many class settings, they are all combined to a page definition. If identical elements are defined, the latest class settings are applied.

7.26 Restriction of rights on absences

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.22
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

Users belonging to the user group `Standard users` can now only see their own absences. The user group `Project supervisors` however has read and write rights on all absences of other users. Group absences can be read by all user groups but not be modified.

7.27 String representation of project-entry types now contain a translated description

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.22
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

The string representation of services, expenses and purchases now contain a translated description of the activity, expense type or purchase type, instead of the code.

7.28 String representation of absences now contains a translated description

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.22
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

The string representation of absences now contains the description instead of the code.

8 Extensions

8.1 TAPI interface also functions without CallerIdName

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.9
Operation mode: On-Premises | Apps: Desktop

The Vertec TAPI interface handles incoming calls via TAPI and can display them in Vertec, also for providers not delivering a `CallerIdName` field.

8.2 Handing over an activity in the SharePoint online DMS Extension

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.10
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

The method `StoreDocument(self, vertecFilePath, content, activity)` hands over the activity as an object. In this way, the corresponding meta data can be stored in SharePoint.

To make it possible for the meta data to be written, the used DMS-interface must be overwritten, respectively completed, as follows. The example is based on the standard Vertec SharePoint DMS Extension.

A script with the following text is created:

```
#
import DmsSharePoint
class MyDmsSharePointExtension(DmsSharePoint.DmsSharePointExtension):
    __doc__ = "Extension of the DMS extension with meta data"
#    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()
```

In this way, the method `getExampleMetaDataDict` located at the interface is activated and returns sample meta data. The corresponding code is located in the standard interface:

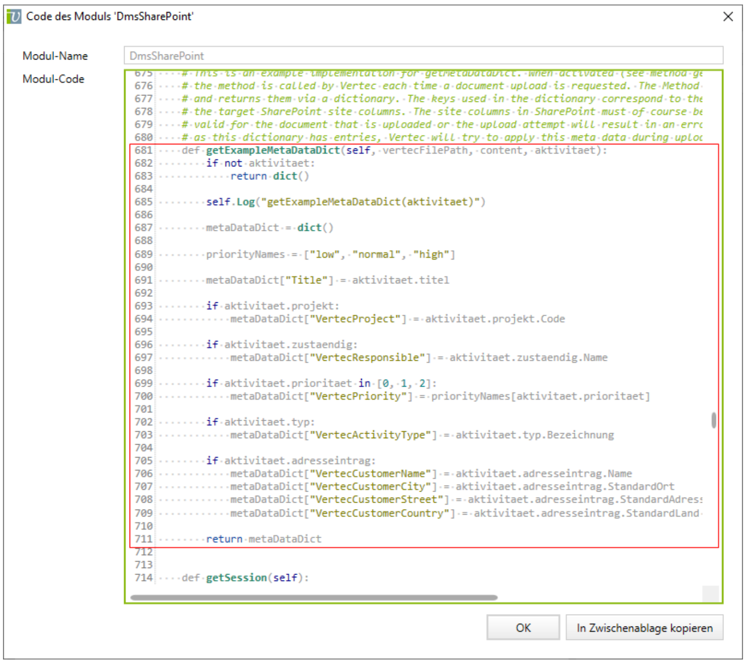


Figure 37: Code in the standard interface

The transmitted fields must correspond to the meta data schema of your SharePoint. In the above generated script, the method is overwritten for this reason as well:

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

instead of the following line

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

returning directly the corresponding values.

An extension entry is then established with `<Name of the script>.<Name of the method>`.

As a result, the standard SharePoint Online DMS Extension is deactivated and the new one generated here is installed.

8.3 Cloud-enabled Abacus Web Debtor Extension

Line: Standard, Expert | Module: Services & CRM, Business Intelligence | Version: 6.4.0.15
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

The **Abacus Web Debtor Extension** has been completely rewritten and is now cloud enabled

---

Further information Further information relating to the new interface can be found in the article Abacus Web Debtor Extension in the Online Knowledge Base under www.vertec.com/kb/abacuswebdebitoren.

8.4 Zeep on whitelist for restrict scripting

Line: Standard, Expert | Module: Services & CRM, Business Intelligence | Version: 6.4.0.15
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

The web service library **Zeep**, used for the Abacus accounts receivable web interface, has been put on the whitelist for **restricted scripting**.

8.5 Abacus Web Creditor Extension is cloud enabled

Line: Standard, Expert | Module: Services & CRM, Purchases | Version: 6.4.0.17
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

The **Abacus Web Creditor Extension** has been completely rewritten and is now web enabled.

Further information Further information on the new interface is found in the article Abacus Web Kreditoren Schnittstelle in the Online Knowledge Base under www.vertec.com/kb/abacus-web-kreditoren-schnittstelle/.

8.6 Support for ZUGFeRD 2.1

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.16
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

Vertec now supports the **ZUGFeRD 2.1standard**, which removes the differences between ZUGFeRD 2.0 and Factur-X as well as Profil XRechnung.

Descending compatibility is ensured. Reports which have already been adapted still function and produce as before a ZUGFeRD PDF with version 2.0.1, respectively with the specified XMP-Schema. If one wishes to use the new ZUGFeRD standard for reports already adapted, the code must be adapted accordingly. The Vertec standard report **Invoice with services list** is provided with the new standard.

Further information Further information is found in the article Rechnungen nach ZUGFeRD Standard (X-Rechnung) in the Online Knowledge Base under www.vertec.com/kb/zugferd/.

8.7 Clean up of address handling for cloud-enabled interfaces

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.17
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

The handling of invoice addresses and personal accounts has been adapted for list-based (cloud-enabled) accounting interfaces:

There is a new (derived) property `buchungsadresse` on the invoice, which in every case produces the address to be used when the booking occurs. The property `rechnung.personenkonto` then always produces the personal account of `rechnung.buchungsadresse`. This corresponds to `rechnung.rechnungsadresse`, except for a contact. In that case, the company of the contact is used. In this case, the company of the contact is used. If empty, a new address is created and the personal account number is written into `rechnung.buchungsadresse`.

If `rechnung.buchungsadresse` is used, the setting of a personal account on a project is no longer supported.

This concerns the following interfaces:

- Abacus Web Debtor Extension

- BMD Debtor Extension
- DATEV Debtor Extension
- Run my Accounts Debtor Extension

8.8 Input of payment data for the cloud-enabled DATEV Debtor Extension

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.19
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

The existing function **DATEV payment import** has been replaced by Payment synchronization.

In the `Settings` menu, the function `Payment synchronization` can be called. A file selection dialogue is displayed in which a payment data file can be selected. A payment cannot be imported twice.

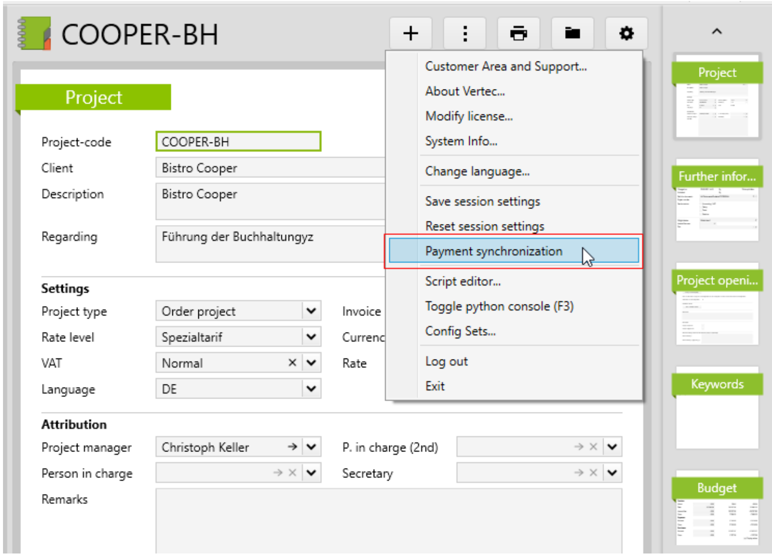


Figure 38: Calling the payment settlement

Tip: If a previous DATEV Extension is used: the existing DATEV payment import must be deleted manually by deleting the script `DATEV Zahlungsimport`.

8.9 Topal Solutions creditors interface with QR invoice

Line: Standard, Expert | Module: Services & CRM, Purchases | Version: 6.4.0.19
Operation mode: On-Premises | Apps: Full-featured

The **Topal Solutions creditors interface has been extended in such a way that a creditor opened in Vertec via** QR-Code can be booked with the Topal Solutions creditors interface.

- At booking time, a check is performed if a value is entered in the field `Kreditor.ESRTeilnehmner` (payment account).
- Topal checks whether a transaction method (technically `Payment method`) is already available in the creditor's address. If this is the case, this payment method is added to the Topal creditor object. If not, a new transaction method is established and the account number is set.
- Payment type is QR in the case of a QR IBAN, IBAN for a regular IBAN and ESR in all other cases.

8.10 Difference between ESR and QR debtor invoices

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.21
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

In Abacus 2020, the field `Reference type` has been added. When an accounting document is input, the difference is made between the reference types **ESR** and **QR**. To ensure proper assignment, Vertec transmits this reference type when booking. If the IBAN number is a QR-IBAN, `PaymentReferenceType=QR` transmitted, otherwise `PaymentReferenceType=ESR`.

8.11 Support of more than 200 sites in the SharePoint Extension

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.21
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

SharePoint Extensions can be used which support more than 200 sites.

8.12 Support of ESR invoices for the Abacus Web Creditor Extension

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.21
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

Up to now, the **Abacus Web Creditor Extension** handed over the payment account and the reference information only if the payment account was a QR IBAN. ESR invoices can now also be booked in Abacus. The external number (`kreditor.nummer`) is in Abacus stored into the field `Belegreferenz-Nr`.

8.13 SharePoint Online DMS Extension now Unicode compatible

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.21
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

The interface correctly processes all Unicode characters in file names, folder names, metadata fields and error messages which are generated by SharePoint itself. Internally, it uses the Python type `Unicode` and converts only to "external" if necessary.

8.14 Cloud enabled DATEV Creditor Extension

Line: Expert | Module: Services & CRM, Fremdkosten | Version: 6.4.0.23
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

The **DATEV Debtor Extension** is now cloud enabled. In the folder **Extensions, only the cloud-enabled** DATEV Extensions are shown, if none had been installed before. If a DATEV Extension is already installed, it stays.

8.15 Setting the EZ procedure for booking

Line: Expert | Module: Services & CRM | Version: 6.4.0.23
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

When the reference type is **QR**, the applicable QR procedure must be specified on the Abacus invoice for several QR procedures defined in Abacus. For this purpose, Vertec hands over the EZ procedure (`PaymentOrderProcedure`) based on the Vertec payment type. The assignment is done as follows:

- In Abacus, a matching **EZ procedure must be defined**, the payment account of which must correspond to the payment account (VESR member number for ESR, IBAN for QR).
- The code of the payment type must correspond to the number of the EZ procedure in Abacus. If the code is not numeric, no EZ procedure is transmitted.
- Vertec determines the payment type on the basis of the payment account. If an IBAN is defined, the payment type is determined by this IBAN. If a payment type is found, the EZ procedure is set in Abacus.

8.16 Improvement of the payment settlement in Sage 200

Line: Expert | Module: Services & CRM | Version: 6.4.0.24
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

Record numbers used for the payment settlement were ambivalent. This prevented importing different payments with the same date and amount. Starting with version 6.4.0.24 payments with same date, text and amount can be correctly imported with **Sage 200 Zahlungsabgleich**.

9 Performance

9.1 Performance improvement in lists without active members

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.5
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

Listen check if an object is `active` or `inactive` to display it with a grey bar in the first column. For objects without any `active` members, this check has been switched off.

9.2 Performance optimization when Vertec is started

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.9
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

When Vertec is started, settings objects such as `Rechttyp`, `Waehrung`, `MWSTTyp`, `Zahlungstyp`, `KommMittelTyp` and `AktivitaetsTyp` are no longer preloaded separately; all instances are now preloaded together.

9.3 Shadow attribute removed from invoice

Line: Standard, Expert | Module: Services & CRM, Budget & Phases | Version: 6.4.0.17
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

Shadow attributes, used to store into the database values which were actually derived, had the effect that for a read access to certain values a write access was performed as well. Two such shadow values existed on the invoice, which led to performance losses when loading longer invoice lists. For this reason, the two shadow attributes `Rechnung.SqlRabattBetrag` and `Rechnung.SqlValutadatum` are no longer stored and used.

9.4 More performant loading of the budget page of a project

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.17
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

The display of budget details has been optimized by preloading planned values.

10 Security

10.1 Restricted session process

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.12
Operation mode: Cloud Abo / On-Premises | Apps: Full-featured

Vertec.Cloudserver can be configured in such a way that session processes are started with restricted possibilities:

- The Vertec session process runs as «Low Integrity» process.
- The Vertec session process is not allowed to start further subprocesses.

This is done by a setting in the `[CloudServer]` section of the Vertec.ini file:

```
Restrict Session Process=True
```

If this setting is specified, one must be careful when using Firebird that the `ServerName` parameter in the database section holds a network path, not a file path. Example:

- Network path (valid): ServerName = `MYSERVER:C:\Programme\Vertec\DB\VertecDemo.fdb`
- File path (invalid): ServerName = `C:\Programme\Vertec\DB\VertecDemo.fdb`

The corresponding log files (**Vertec.Session.log** and **Vertec.TaskRunner.log**) can be found under `C:\WINDOWS\system32\config\systemprofile\AppData\LocalLow`.

11 Technical aspects

11.1 Unicode support

Up to version 6.4 included, texts in Vertec (addresses, texts on services, remarks, etc.) were stored limited to 1 byte encoding, using Windows-1252 or ANSI (they are the same, from now on we talk of ANSI). Vertec could thus exactly work with the characters defined in ANSI, see for instance https://de.wikipedia.org/wiki/Windows-1252. Other characters were replaced by `?`.

The ANSI character set with its 250 characters is sufficient for English and all West-European languages. Also sufficient for Vertec in most cases. More and more customers with business relations in Eastern Europe, for instance, wish to be able to represent repective addresses correctly in Vertec. For this reason, character encoding is changed to Unicode starting with Vertec 6.5.

From Vertec 6.5 on, texts are stored in Unicode. Vertec now supports all of approximately 65'000 characters of **BMP** ("Basic Multilingual Plan"), hence all characters of all relevant languages, including all Asiatic. Characters not part of BMP entered into Vertec are, (as before for non-supported characters) replaced by `?`, which should only happen for a few emojis.

When the database is converted from 6.4 to 6.5, all texts in Vertec are converted from **ANSI** to **Unicode**. The change is not visible on the interface. The size of the Vertec database becomes larger for this reason, in general by 20%-30%, which however should not have negative visible effects.

Backwards compatibility

The extension to Unicode is totally downwards compatible, but not necessarily forward compatible. This means that an existing Vertec installation will function as before, but if Unicode characters are processed (knowingly or unknowingly, for instance by copying a text), this will not necessarily be the case.

Here are some points for customer installations to which attention should be drawn. The risks are mostly related to customer-specific scripts (principally Python) and interfaces.

Python and Unicode

A Vertec string member (such as the text of a service, for instance) was up to now of type `str`, in 6.5 it is of type `unicode`. This ensures that existing Python code for copying, assembling, completing a string will still work in the same way. We have in addition «bent» the `str` module to Unicode because an analysis of Python code at customers has shown that many uses of `str()` exist to transform for instance strings to other strings, which would cause an error without this correction.

The default encoding in Python remains ANSI, meaning that the Python source code remains in A**NSI**. This ensures maximal backwards compatibility for existing code. This implies that non-ANSI characters cannot directly be used in string literals (there exist however alternatives in Python).

The transition from a Unicode string to default encoding, meaning from Unicode to ANSI, has been designed to be fault-tolerant: characters which cannot be translated do not cause an error but are replaced by `?`.

Encodings

The default encoding in ANSI has as a consequence that existing Python code which processes strings, for instance, and stores them into a file or sends them to a web service may have to be adapted. To ensure that non-ANSI characters, which may be present in Vertec starting with version 6.5, will arrive correctly, a Unicode encoding (UTF-8 for instance) must be selected – the opposite side however must also be able to read them. Without explicit encoding like, e.g.,

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

the implicit transformation to ANSI will happen with the above-mentioned loss of non- ANSI characters.

It is also necessary to deal with the encoding when data is input into Vertec, for instance by opening a text file, with the Vertec XML Server or with `vtcapp.requestfilefromclient()`: this was already the case, but with Vertec 6.5 and the support of Unicode, the requirement becomes more urgent because loss of non-ANSI characters can be avoided. If one inputs a text file encoded in UTF-8, for instance, it must be correctly decoded:

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

As a rule of thumb, remember that in Python

– `encode` always converts from a Unicode string to a byte string (**ä** needs for instance 2 bytes in UTF-8)
– `decode` always converts from a byte stream (encoded in UTF-8, for instance) to Unicode.

Customer-specific extensions

The «bending» of the `str` module to Unicode is not used for customer-specific extensions stored as Python files in the Extensions folder of the Vertec folder. This applies as well to modified extension code originally provided by Vertec. For these extensions, `str()` will transform a Unicode string to ANSI with the expected data loss for non-ANSI characters. Customers with such interfaces should inspect their Python code to detect such problems.

Unicode support for Additional fields

The additional field types `String` and `Text` could display text in Vertec 6.4. **String** is a field that is limited to a maximum of 255 characters, while the field **Text** is unlimited. The additional fields of type String are automatically converted to Unicode with 6.5, those of type Text are not.

The additional fields of type `Text` are stored in the database as **BLOB fields** (Binary Large Object). This means that the data is stored in the database as pure binary data and only interpreted as text in the Vertec business logic. Since additional fields of type `Image` are stored in the same database field, additional fields of type Text remain unchanged 8-byte texts in ANSI format in Vertec 6.5 due to reasons of downward compatibility.

With Vertec 6.5 there is the new additional field type **Unicode Text** instead:

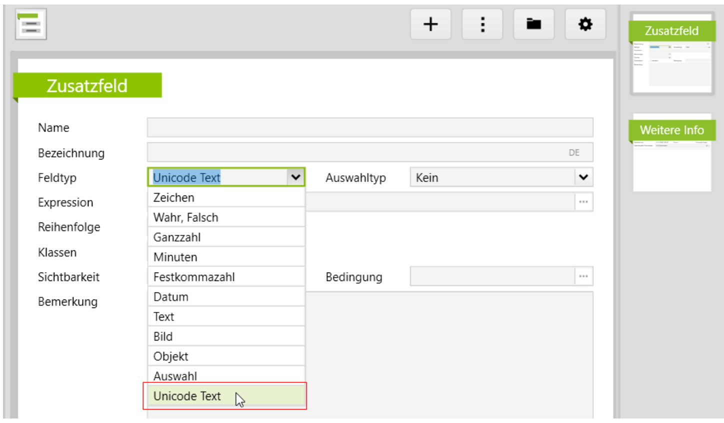


Abbildung 39: Additional field type Unicode Text

If you use additional fields of type Text, we strongly recommend to change them to Unicode Text immediately after the comvert. The DB convert to 6.5 automatically copies the texts from the Text field to the Unicode Text field in the correct format.

Attention: All accesses to the additional field (OCL, SQL) must also be converted:

– in SQL from `WERTBLOB` to `WERTTEXT`
– in OCL from `zusatzfeldblob('zusatzfeldname')` to `zusatzfeld('zusatzfeldname')`.

11.2 Redesign of the login dialogue and Vertec splash screens

Line: Standard, Expert | Module: Services & CRM | Version: 6.5.0.1
Operation mode: Cloud Abo / On-Premises | Apps: Alle

The user interface of the login dialogue and Vertec splash screens has been redesigned and adapted to the design of the web page and the customer portal.

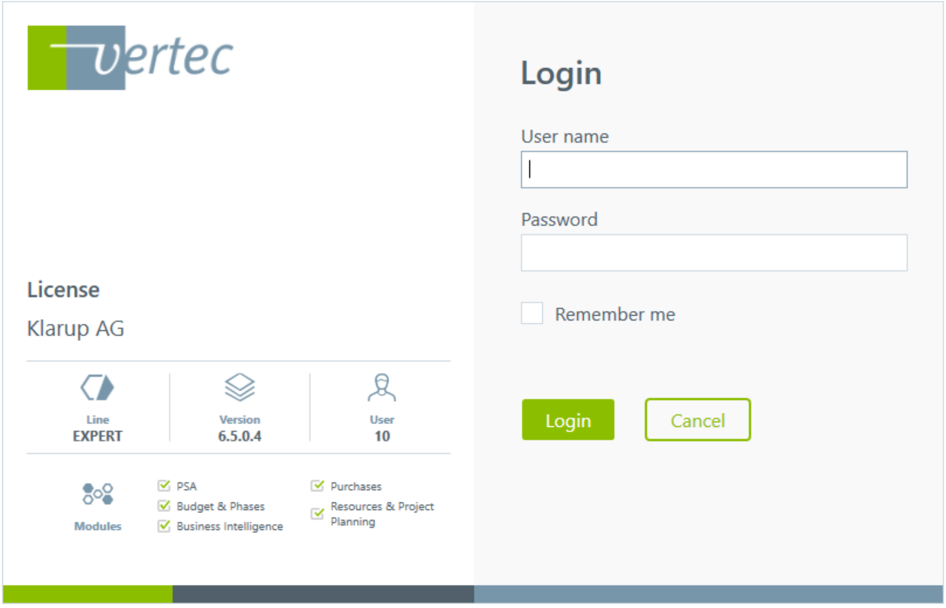


Figure 40: New design of the login dialogue

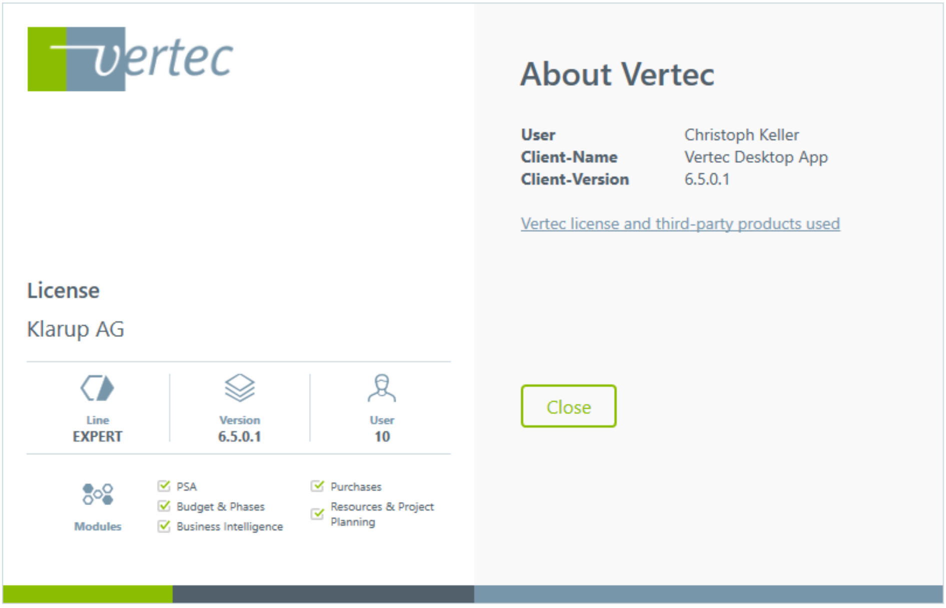


Figure 41: New design of a splash screen

In addition, the cloud installer `Vertec.Updater.exe` has been integrated into `Vertec.CloudInstaller.exe`.

11.3 Support of Cloud App installations

Line: Standard, Expert | Module: Services & CRM | Version: 6.5.0.1
Operation mode: Cloud Abo / On-Premises | Apps: Cloud App

Several Cloud Apps of different server can be installed on client:

When **Cloud App** is newly installed Vertec App Portal, a subfolder for the server's name is created under `%localappdata%\Vertec Cloud App`,`from which the installation occurs.` The Cloud App and its Vertec.ini are stored there. The name of the subfolder is formed using the server name. This is particularly useful if access with the Cloud App to several Vertec instances is necessary, e.g. for a test environment.

11.4 Additional logging

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.8
Operation mode: On-Premises | Apps: Full-featured

There are 2 possibilities to display debug log output from Vertec:

1. **Loading of objects from the database. Using the value** `DebugCategories = Vertec.PMappers` in the Vertec.ini file in section [Log] makes it possible to observe the loading of objects from the database.
   Persistence mapper log entries only appear if they are activated via **DebugCategories** (category DEBUG).
   Note that this type of logging should only be activated temporarily, as the writing of log date into the log file may cause a loss of performance.
2. **Conversion, Initdata**: Log output when the version-dependent Convert Logic is performed. The log entries always appear (category INFO).

11.5 Terminating hanging TaskRunner processes

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.8
Operation mode: On-Premises | Apps: Full-featured

There are situations where a Vertec.TaskRunner process does not terminate correctly. Such hanging processes block memory on the server. For this reason, it is now possible in the **Vertec.ini file** under [Cloudserver] `Task Scheduler Max Minutes` to specify the maximum running time for a planned task. Default value is 1200 (20h).

- If a process runs longer than the maximum time, it is automatically terminated by the cloud server.
- When Vertec.CloudServer is terminated, all running TaskRunner processes are terminated as well.

11.6 Send the log output to the Loggly service

Line: Standard, Expert | Module: Services & CRM | Version: 6.4.0.9
Operation mode: On-Premises | Apps: Full-featured

Log outputs from the server-based Vertec processes (Vertec.CloudServer, Vertec.Session, Vertec.TaskRunner) as well as from the Vertec Desktop App can be sent to the cloud-based logger service **Loggly**.

For this purpose, the following values must be set in the Vertec.ini file in section [Log]:

- **LogglyToken**: The customer-specific Loggly token is set here. Once this value is set, the log output is sent to Loggly to the specified LogglyUrl.
- **LogglyUrl**: The desired Loggly URL is set here. If empty, the output is sent to https://logs-01.loggly.com/.

After setting these values, the `Vertec.CloudServer` must be restarted.

The following values are sent to Loggly:

- hostName: The name of the host computer on which the process is running.
- level: Log level of the information (Debug, Info, Warning, Error or Fatal)

– loggerName: Name of the affected logger (e.g. `Vertec.TaskRunner.TaskScheduling.TaskScheduler`)
– message: The actual log information
– process: Name of the affected process (e.g. `Vertec.TaskRunner`)
– timestamp: Date/time of the log entry in ISO-8601 format
The customer's name in the license information is transmitted as a tag if available when the log entry occurs (early in the start-up process, this information is not yet available).

11.7 Removal of the Fastviewer client

Line: Standard, Expert | Module: Services & CRM | Version: 6.5.0.1
Operation mode: Cloud Abo / On-Premises | Apps: Desktop App

The Fastviewer client has been removed from the Vertec installation folder and the Menu of the Desktop App.

In the Menu now appears the new option `Customer Area and Support`:

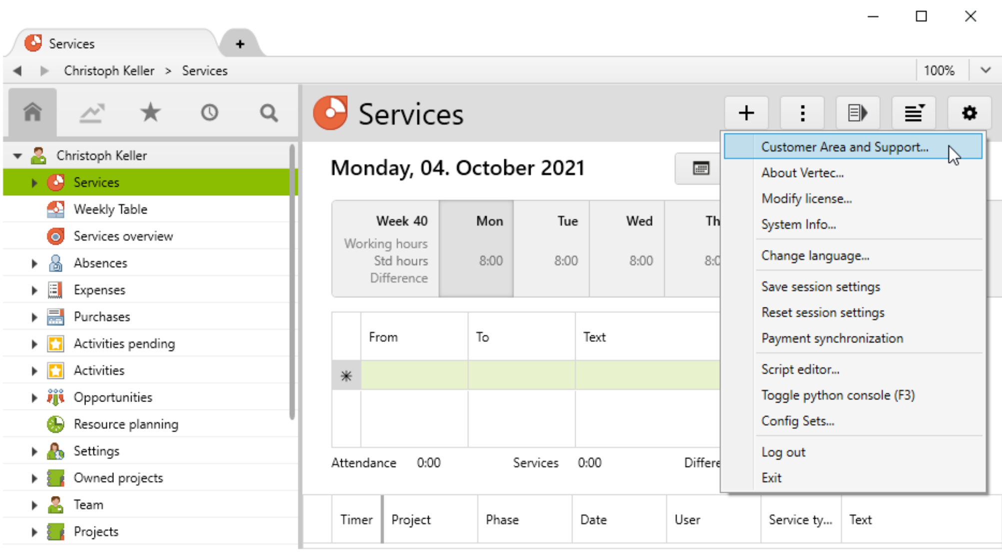


Figure 42: Menu option Customer Area and Support

It leads directly to the page of the Vertec Support where the Fastviewer can always be downloaded in the actual version.

11.8 Database convert

Line: Standard, Expert | Module: Services & CRM | Apps: Full-featured | Version: 6.5

Vertec 6.5 comes with a newly implemented DB convert. The DB convert is for instance necessary when the Vertec database is updated to a new version when a Vertec Update is desired.

With the new DB convert it is also possible to copy Firebird databases to a Microsoft SQL database and vice versa. A conversion is possible

– from MS SQL to MS SQL
– from MS SQL to Firebird
– from Firebird to Firebird
– from Firebird to MS SQL

In addition, the new DB convert offers the following improvements:

– A DTS component is no longer necessary for the MS SQL Server convert.
– The 32-bit `Vertec.DBConvert.exe` no longer exists for the MS SQL Server convert.

– The new DB convert is in some cases able to copy user-specific indexes which will then be available in the new database after the conversion.

Customized indexes

In Vertec databases, customized indexes can be created after careful analysis and consideration. In order for indexes to be preserved after a database conversion, they must begin with the prefix `CIX_`.

Customized indexes are created as follows:

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

The syntax for Firebird and SQL for creating simple indexes is similar, but when additional specifications are added there are differences.

– More information on creating indexes in **Firebird** databases can be found on [https://firebirdsql.org/ref-docs/langrefupd20-create-index.html](https://firebirdsql.org/refdocs/langrefupd20-create-index.html)
– More information on creating indexes in **SQL** databases can be found on https://docs.microsoft.com/de-de/sql/t-sql/statements/create-index-transact-sql?view=sql-server-ver15

Example:

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

Creating indexes can improve performance, but it also involves risks. For example, inserting or updating a record takes more time if an index also has to be processed, and an index does not always lead to a better result with a **SELECT**: it may be that the performance of a field with actually good selectivity but at the same time many values such as **«»** or **NULL** is even worse. Additionally, an index is only effective for a **=** comparison, not for a **LIKE**.

Before deciding to insert a new index, the effect on the entire application should be thoroughly tested in any case.

Automatic conversion with Firebird

If Firebird databases are used, the conversion can, as before, be performed automatically when Vertec is updated.

When Vertec is started with a new database version, the DB convert dialog is displayed:

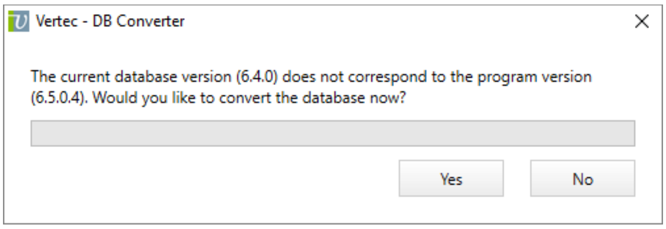


Figure 43: Asking whether the automatic database convert should be performed

A click on `yes` starts the automatic conversion. The progression is displayed in a status bar.

After the conversion process, the Vertec Desktop App must be restarted:

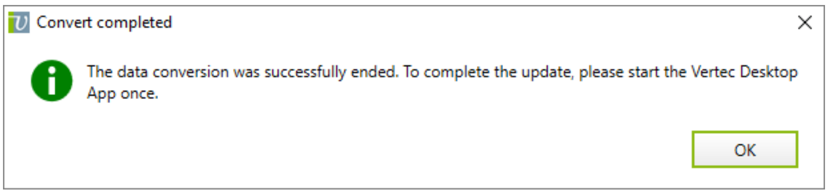


Figure 44: Confirmation of the automatic database convert

Manual conversion

If MS SQL Server is used as database server, the source and target databases must be created manually and the convert with DB convert must be performed manually.

Start Vertec with parameter `/DC`. The convert dialog is opened:

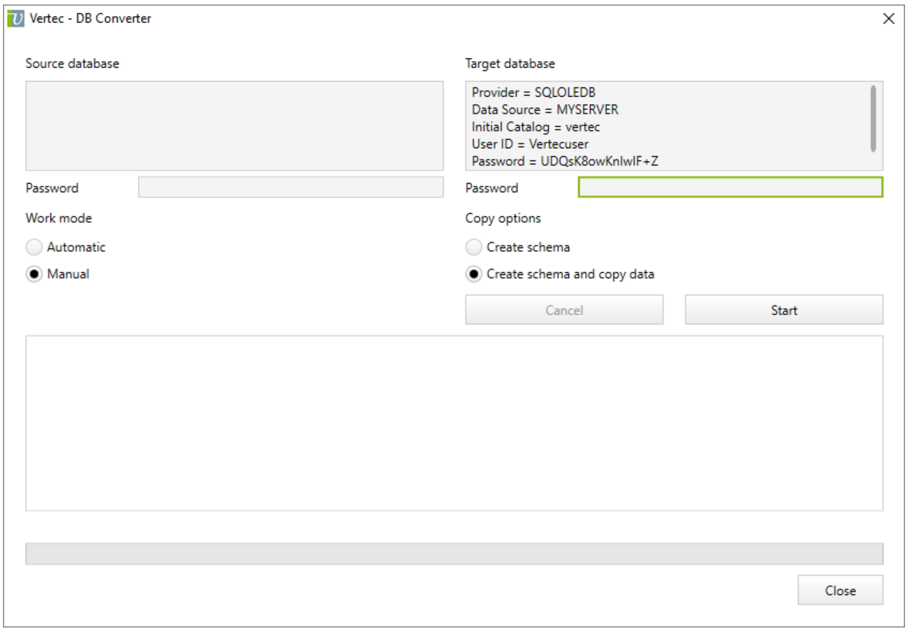


Figure 45: Manual database convert

With a manual convert, the source database is modified. A temporary copy of the source database should therefore be established. The exact procedure to do this is described in the [Knowledge Base](#).

Copying databases

Databases can be copied using DB convert. In this way it is also possible to copy a Firebird database to a Microsoft SQL database and vice versa.

The procedure is similar to the manual conversion:

1. Start Vertec with parameter `/DC`.
2. Select as source database the database to be copied.
3. Specify an empty Firebird or MS SQL database.

The following information is required:

Firebird

Server name	Identifies the server and the database.
User name	Identification for the database server.
Password	Encoded identification password for the database server, together with the u**ser name**. To encode the password, use the [Vertec DB-Tool](#).

Example:

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

Microsoft SQL Server

Provider	Driver for the SQL link
Data Source	Server name (or IP address) of the SQL Server.
Initial Catalog	Name of the database on the MS SQL Server.
User Name	Identification for the database server.
Password	Encoded password for identification at the database server, together with **User name**. To encode and the password, use the Vertec DB-Tool.

Example:

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

It is then possible to choose whether only the database schema is created or the data is also copied:

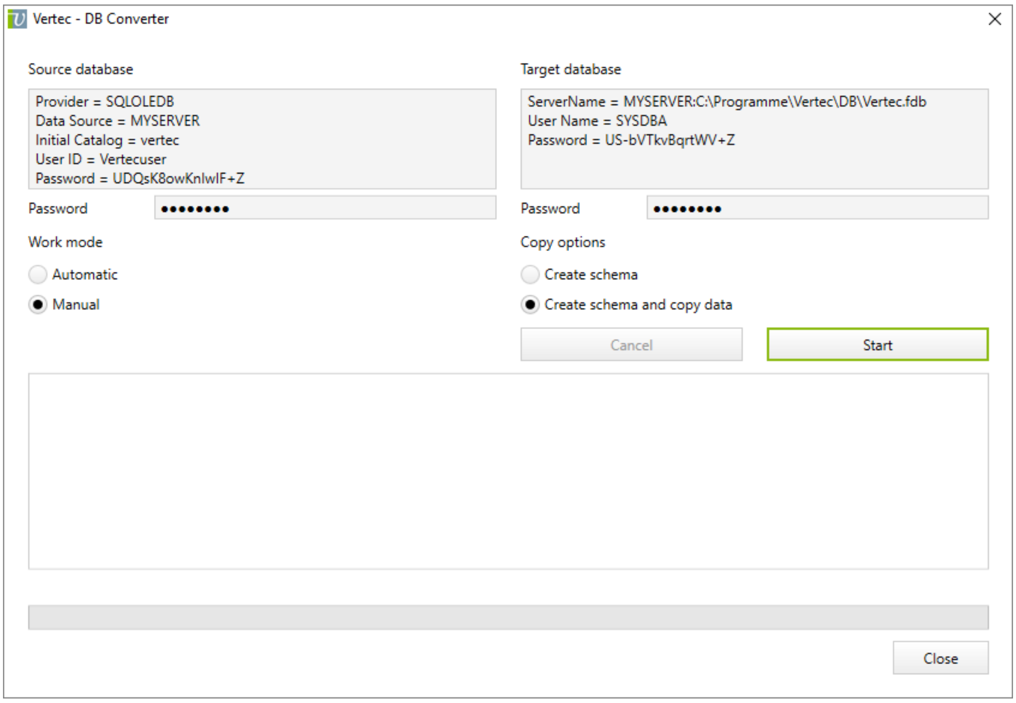


Figure 46: Copy a Microsoft SQL database to a Firebird DB

A click on `Start` starts the database convert. The progression is displayed in a status bar. When the convert is finished, Vertec must be restarted.

11.9 Update of IBQuery

Line: Standard, Expert | Module: Services & CRM | Version: 6.5.0.1
Operation mode: On-Premises | Apps: Full-featured

With Vertec 6.5, the most recent version of IBQuery is provided. If you use an on-premises installation of Vertec, you will find in your installation directory after the update the version 9.5.0 of IBQuery (`IBQ.exe`) Professional Version.

The details related to this new version are found in the Homepage of the editor.

12 Improvement of Firebird database support

Line: Standard, Expert | Module: Services & CRM | Apps: Full-featured | Version: 6.5

In Vertec 6.5, the support of Firebird databases has been improved. The following changes have been made:

12.1 Update to Firebird 4.0

With the update to Vertec 6.5, the update from Firebird 3.0 to Firebird 4.0 can also be made:

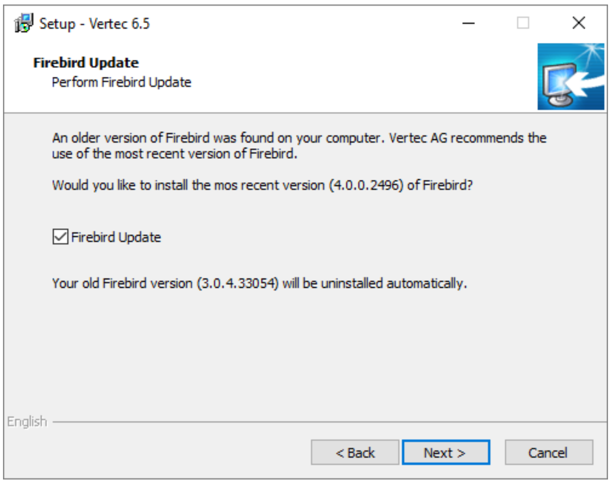


Figure 47: Request for Firebird update

If it is desired to execute the Firebird Update, set the check box and click on `Continue`. The Firebird server is then shut down and restarted.

Further information about Firebird 4.0 is available in the Firebird Release Notes.

12.2 Update of Firebird FIB components

The Delphi Firebird components were up to now stored in the Client Library gds32.dll of the installation folder. Starting with Vertec version 6.5, the actual Firebird client library **fbclient.dll** is used, the file gds32.dll is removed from the Vertec installation folder.

All Vertec executables needing to access Firebird DB (Session Process, Cloud Server, Task Runner and, particularly, the Vertec Desktop App) now use the standard library **fbclient.dll**, stored by the Firebird Setup into system32.

To use the Vertec Desktop App in the future, it is absolutely necessary to execute **Desktop App Setup** (up to now Client Setup).

12.3 Firebird Dialect 3

After the update to Vertec 6.5, **Dialect 3 is now used instead of** Dialect1. Dialect 1 is not supported by Firebird 4.0 This change has some consequences:

Double quotes

Text enclosed in double quotes is interpreted by Dialect 3 as a column, whereas in Dialect 1 double quotes were equivalent to single quotes. The following code will produce an error in Dialect 3:

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

The same is valid in string comparisons with **=** and **like**

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

To insure a best possible backwards compatibility, double quotes in the SQL code in SQL folders and in the function `getwithsql()` in Python scrip s are automatically replaced by single quotes at execution time.

Care should however be taken to exclusively use single quotes in SQL code in the future. For instance:

```
select projekt.creationdatetime from projekt

where  projekt.creationdatetime < 'today'
```

or

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

Integer division

Another change from Dialect 1 to Dialect 3 is the division of integers.

In Dialect 1, a division of 2 integers results in a float, not in Dialect 3.

Dialect 1: 1/3 = 0.33333

Dialect 3: 1/3 = 0

In the case of literals, simply write `1.0/3.0` to obtain 0.3333. But when dividing integer values in tables, a `CAST` must be performed.

Date transforms

More important are the differences for date fields. In Vertec 6.4, Firebird 3.0.x and Dialect 1, all date fields with or without time portion were of type `DATE`.

In Vertec 6.5, Firebird 4.0 und Dialect 3, date fields are of type `TIMESTAMP`. In Vertec, this is however noticeable only when performing explicit `CAST` operations, for example:

```
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` is without time portion in Dialect 3, `AS DATE cuts the time part` off if present. This is not relevant in the above example, but there also exist relevant date values with the time part in Vertec, for instance in `Budgetchange` objects, or if one wishes to query all new objects with `CreationDateTime`. On a `CAST ... AS DATE`, Firebird with Dialect 3 will issue a warning:

```
SQL warning code = 301

DATE data type is now called TIMESTAMP

SQL warning code = 301

DATE data type is now called TIMESTAMP
```

Backwards compatibility

The conversion to Firebird Dialect 3 is not backwards compatible if explicit `CAST` operations are used. Customer-specific code using the time portions should be modified from `DATE` to `TIMESTAMP`.

12.4 Firebird error messages

Error messages from the Firebird client library (via Delphi) mostly begin with the text `Can't format message`, because Firebird does not find the language resource. These error messages mostly appear when attempts to execute erroneous SQL queries in SQL folders are made.

These messages now contain a text produced by Vertec which makes more sense.

The file firebird.msg is created at setup time and stored in the system for this purpose. This concerns a temporary folder during installation, the Vertec Installation folder and two system folders. When the setup is over, the file must be present in the three following folders:

- Vertec installation folder
- %SYSTEMROOT%\System32
- %SYSTEMROOT%\SysWOW64.

13 Update of Vertec On-Premises installations

13.1 New install of Vertec

For a new install, use the program VertecSetup-6.5.0.exe, available as a download. More information related to the install is available in our Online Knowledge Base under www.vertec.com/kb/neuinstallation.

13.2 Before the update to 6.5

Please take note of the sections related to backwards compatibility: 11.1 about Unicode support und 12.3 about the use of Firebird Dialect 3.

13.3 Update of Vertec

Use the installation program as well for the update of a Vertec installation. For large customer installations with many customer-specific reports and lists we recommend to first create a test installation and to step through the release change before the update is performed on the active system. Further information can be found in the article **Testinstallationen** in our Online Knowledge Base under www.vertec.com/kb/testinstallation.

13.4 First start-up after the convert

After the data conversion, the Vertec Desktop App must be started. This first start-up is important and is a part of the conversion process, since several adaptations to the data and to structures, etc. are only performed at this time. It is important that this first start-up terminates without problem. In no case should the first start-up after a convert be interrupted.

Restart of the Cloud server

After the convert and the first start-up of the Vertec Desktop App, the Cloud server must be restarted. The update of Vertec is then complete.