February 2018

Vertec Software
Release 6.2

Schweiz +41 43 444 60 00
Deutschland +49 40 30 37 36 70

mail@vertec.com www.vertec.com

Table of contents

1 Introduction

These are the Release Notes for Vertec Version 6.2. Please consider the following points:

1.1 Vertec first set-up

For a first set-up, use the set-up program VertecSetup-6.2.0.exe, available for downloading. Detailed information on the set-up procedure in our Online Knowledge Base under www.vertec.com/de/kb/neuinstallation.

1.2 Vertec upgrade

To upgrade from an existing Vertec set-up, also use the set-up program. In the case of large customer systems with many customer-specific reports and list settings, we recommend to set-up a test installation first and test the release upgrade before doing it with the active system. Information on this subject can be found in the article Testinstallationen in the Online Knowledge Base under www.vertec.com/de/kb/testinstallation.

Backwards compatibility

For those features where backwards compatibility presents an issue, the subject is fully treated in the corresponding following articles under section Backwards compatibility:

Article	Page
2.2 Discontinued features in Vertec 6.2	6
2.3 Pre-announcement of Vertec 6.3	7
6.4 Change of control names on invoices	46
6.7 Custom dialogues: DatePicker with null value	48
6.14 Authorisation for SQL	58
6.15 Authorisation check for personal folders modified	60
8.2 Support of IBAN number on payment type	63
9.1 New implementation of Vertec-generated reports	64
9.2 Office templates changed to new Office formats	65
10.3 Abacus: Use country code from Vertec addresses	71

1.3 First start-up after the upgrade

The first start-up of Vertec after the upgrade is important and part of the conversion process, as certain changes to data, structures, etc. are only performed as this point. It is important that this first start-up runs through flawlessly. In no case should the first start-up process after an upgrade be interrupted.

1.4 Firebird version 3.0.2.

Vertec 6.2 comes with the latest Firebird version 3.0.2. The Vertec set-up automatically performs the upgrade after asking for confirmation.

Additional information: additional information about Firebird can be found in the Online Knowledge Base under www.vertec.com/de/kb/firebird.

2 Versions

2.1 Highlights of Version 6.2

Version 6.2 again presents several highlights. The most interesting ones are shortly presented here:

Article	Page
3 Vertec Phone App	9
4 Extended Office reports Office reports	21
5 Config Sets	33
6.1 Tags on user entries	43
6.2 Key values on user entries	44
8.1 Support of pay-in slips with QR code	62

2.2 Discontinued features in Vertec 6.2

Use of Classic App is discontinued

With Release 6.2, Classic App is discontinued. Most features which until now existed only in the Classic App are now provided by the new apps. The following features are definitely discontinued:

- Calendar: The calendar view of activities no longer exists. Calendar values (from-to, duration and whole day) can be entered with activities as before and the activities therefore matched with appointments in Outlook. More information on this subject can be found in the Online Knowledge Base under www.vertec.com/de/kb/kalender.
- vtcForms Dialogues: vtcForms Dialogues are discontinued as well. Instead, the new Python Dialogues are available. More information on this subject can be found in the Online Knowledge Base under www.vertec.com/de/kb/pythondialoge.

Use of the Classic Web input is discontinued

Along with Classic App disappears Classic Web Input. No more support is provided for this feature from now on. Vertec Web App is available to input data via internet browser.

Please check that all processes that you have so far performed with Classic Web Input can now be performed with Web App. Should this not be the case, please contact your Vertec Partner.

Use of iPhone App is discontinued

Along with Classic App, the iPhone App is discontinued in Version 6.2. Phone App is now available (see chapter 3).

The XML server can now only be operated through cloud server service

The XML server is now part of the cloud server service. The former XML server part of Vertec Service is discontinued. Authentication and session handling have been changed and extended. No other changes to the XML server have occurred. More information on this subject can be found in the Online Knowledge Base under www.vertec.com/de/kb/xmlreferenz.

Discontinued system settings and functionalities

The following system settings which only appeared in the Classic App in Vertec 6.0 have been discontinued:

Property	Section
Tree-view as default display	General
Detailed view displayed as tree for projects and users	General
Show folder list (further info)	General
Show favourites in search dialogs	General
Display services-text in weekly table	Classic Web
Support OpenID authentication	Classic Web
Display monthly validation	Classic Web
Use only external addresses	Addresses

More information on this subject can be found in the Online Knowledge Base under www.vertec.com/de/kb/legacy.

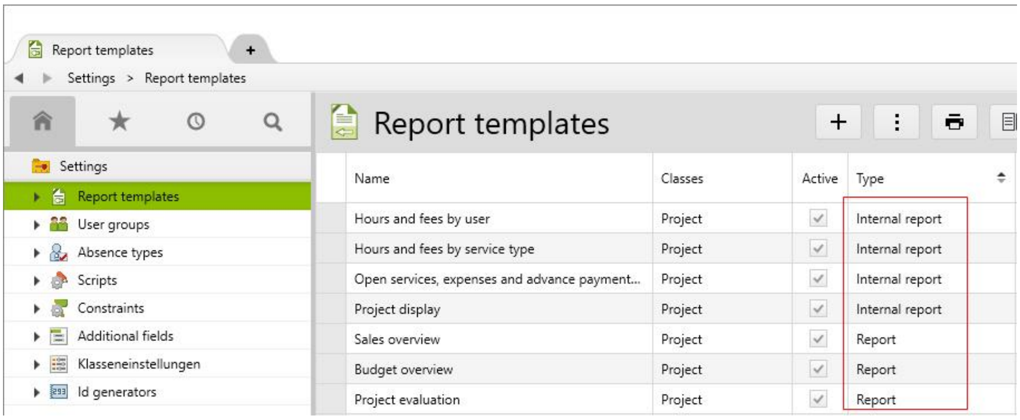
2.3 Pre-announcement of Vertec 6.3

In Vertec Version 6.3, the old reporting systems Internal Report and Report-Designer Reports will be discontinued.

For this reason, Vertec introduces with version 6.2 the Extended office reports (see chapter 4) which replace the above mentioned systems.

It is important that in the period before Vertec Version 6.3, you check your reports and change those running under the outdated systems over to the new system.

In Settings > Report templates this involves all reports of the types `Report` or `Internal Report`:



Report templates

Name	Classes	Active	Type
Hours and fees by user	Project	✓	Internal report
Hours and fees by service type	Project	✓	Internal report
Open services, expenses and advance payment...	Project	✓	Internal report
Project display	Project	✓	Internal report
Sales overview	Project	✓	Report
Budget overview	Project	✓	Report
Project evaluation	Project	✓	Report

Figure 1: Legacy reports in Report templates

Please check this list and determine which of these reports you still need after version 6.3.

Vertec already provides some of these reports as extended reports. They are listed in chapter 4.

Extended office reports provided by Vertec are registered with status inactive in Vertec 6.2. In Vertec 6.3, they will automatically be activated and the old reports will be deleted. If you are going to use the new reports "as is" without customization, you'll be able to continue working with the same functionalities after this change. Otherwise please adapt the new reports to your requirements beforehand.

Should you have questions, please contact your Vertec partner.

3 Vertec Phone App

Line: Standard, Expert | Module: Services & CRM | Version 6.2

3.1 Installation and configuration

With Version 6.2, Vertec provides the new App for iPhone and Android. They are ready to be downloaded in:

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

The corresponding links are also available on the Vertec portal for cloud clients (see www.vertec.com/de/kb/cloudportal).

Like Web App and Cloud App, the Vertec Phone App is a cloud client, yet not "full-featured" as the former ones, but specialized.

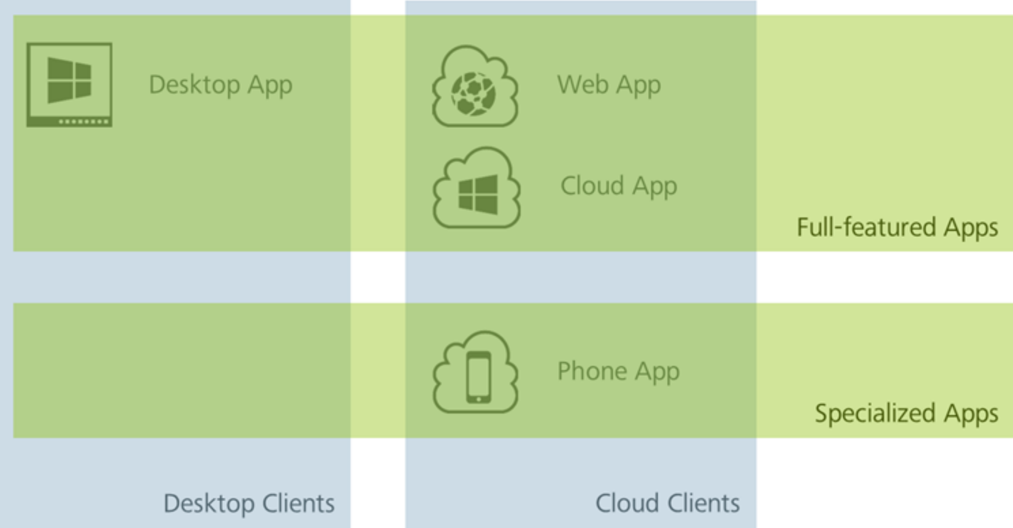


Figure 2: The Phone App in the Vertec application landscape

This means that the Phone App is based on the Cloud Server as well, but has its own specialized interface and a reduced functionality.

Just as with Cloud App, the access occurs through the Vertec web-access service, which allows a simple and SSL-encoded access from the internet.

You can also operate the access to Vertec through the normal web server, putting your own server on the internet. In that case, you must yourself take care of the SSL-configuration and certificate and parameterize the firewall to ensure the safety of your network.

Please note that everyone has the possibility to register to your web server over the internet if URL and username are known. It is therefore particularly important to use safe passwords for your users!

More information Detailed information on this topic in the article Vertec via Webaccess under www.vertec.com/de/kb/webaccess.

3.2 Remember me

The Phone App supports the Vertec Remember me option. Upon login, a token is automatically generated. Its lifetime is defined by the `Token Lifetime` parameter in the Vertec.ini file.

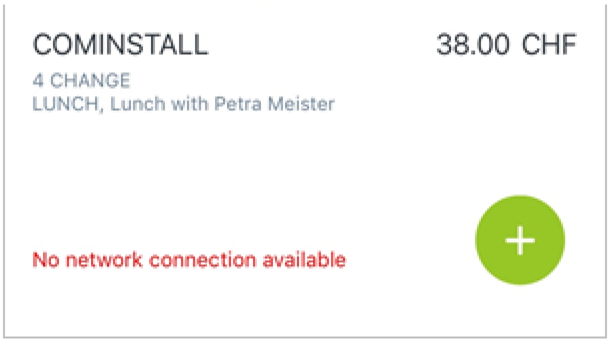
Remember me More information on the option Remember me can be found in the Login article under www.vertec.com/de/kb/verteclogin#angemeldet-bleiben.

Logging out of the Phone App removes the token entry from the AuthToken table.

Should the Phone App lose connection to the server, the same token is reused if the reconnect happens within the `Token Lifetime`.

3.3 Phone App session timeout

If the Phone App is unable to establish the communication, a corresponding message appears.



COMINSTALL 38.00 CHF

4 CHANGE

LUNCH, Lunch with Petra Meister

No network connection available

Figure 3: No network access

The user can however continue to work normally as long as the session stays alive. The duration of the session is defined (in minutes) in the Vertec.ini-File under `Phone App Session Timeout`:

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

This value must be larger than 0 and smaller than 2147483647.

If no value is set, 60 [minutes] is used.

Please note Be careful about token lifetime expiration (see 3.2): Use a sufficient session time to avoid that the user is constantly obliged to reconnect himself.

3.4 Minimal requirements

The Vertec Phone App supports the operating systems Android from version 4.0.3 and iOS from version 8.

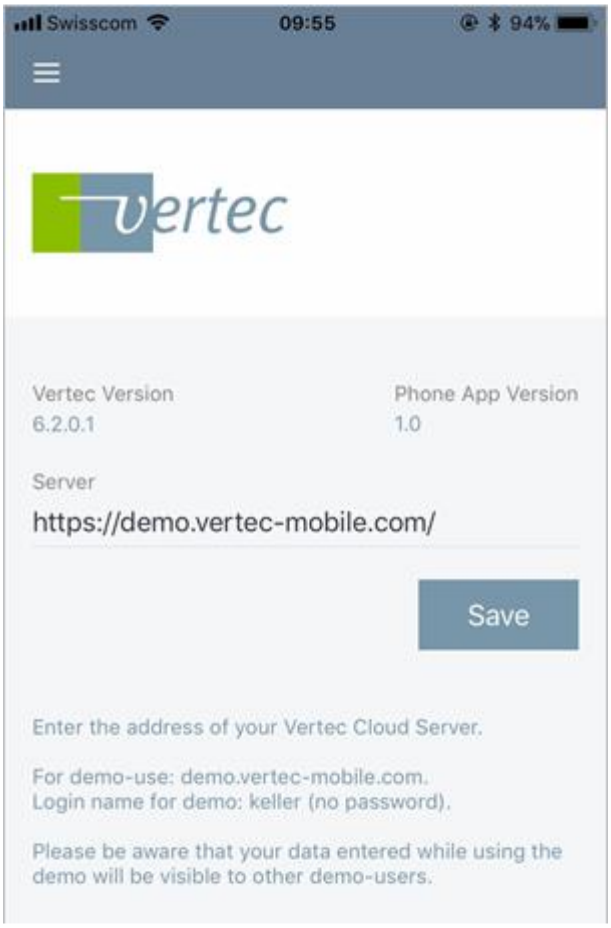
3.5 Login

A click on the Vertec icon starts the application.

Figure 4: the Phone App icon

The language of the Vertec Phone App conforms to the system language of the mobile device from which it is called.

When the Phone App is started for the first time, the Phone App settings screen appears. Specify here the path to the server.



Vertec Version
6.2.0.1

Phone App Version
1.0

Server
https://demo.vertec-mobile.com/

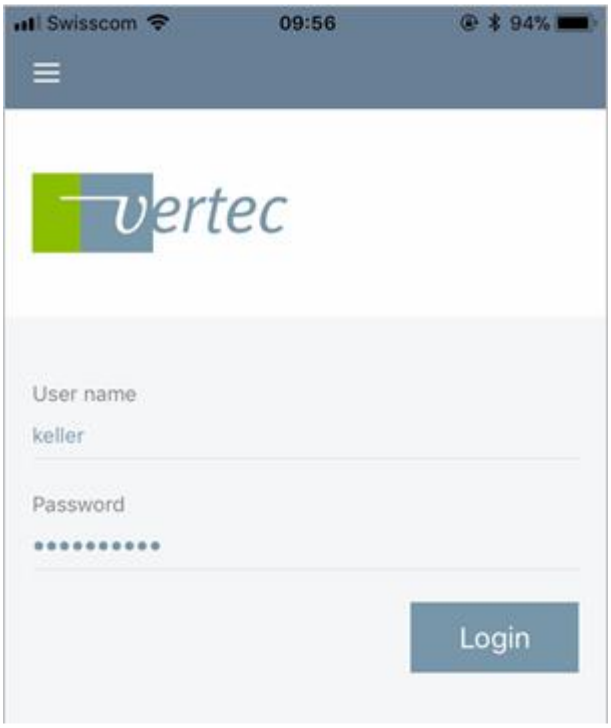
Enter the address of your Vertec Cloud Server.

For demo-use: demo.vertec-mobile.com.
Login name for demo: keller (no password).

Please be aware that your data entered while using the demo will be visible to other demo-users.

Figure 5: Specifying the server path in the Phone App settings

Then, and upon following starts the login dialogue appears:



User name

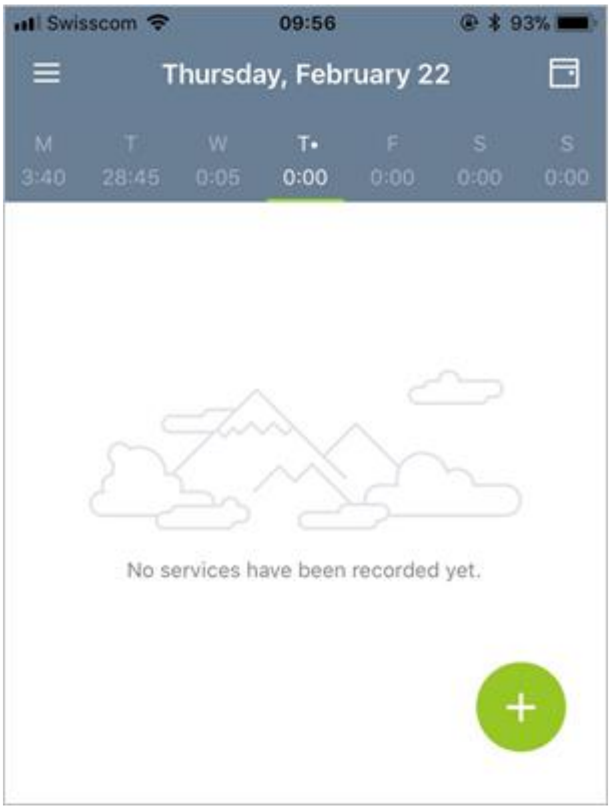
keller

Password

•••••••••••

Figure 6: Der Phone App login dialogue

The Phone App starts by default with the list of the services of the present day.



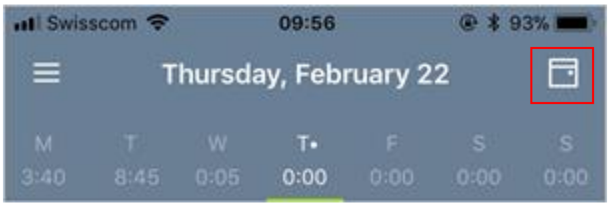
Thursday, February 22

M	T	W	T•	F	S	S
3:40	28:45	0:05	0:00	0:00	0:00	0:00

No services have been recorded yet.

Figure 7: Phone App list of services

The dot next to the day indicates the present date. By clicking on the icon at the top right it is always possible to navigate back to the present date.



Swisscom 09:56 93%

Thursday, February 22

M	T	W	T•	F	S	S
3:40	8:45	0:05	0:00	0:00	0:00	0:00

Figure 8: Navigating to the present date

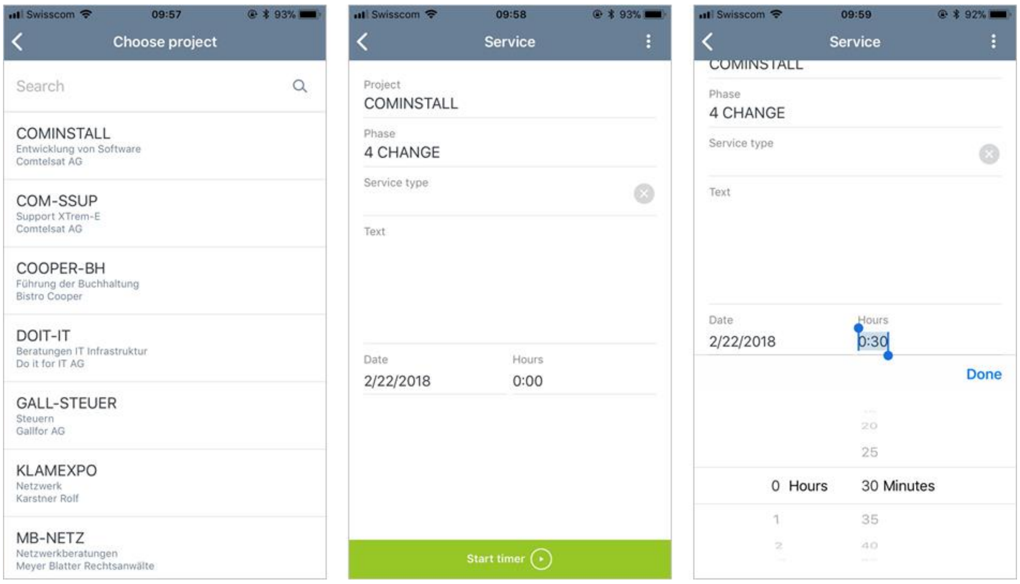
3.6 Registering services

Clicking on the + button creates a new service line. The list of projects appears to allow the selection of the project.

Shown are all projects for which the logged-in user has registered services during the last month and for which he/she is allowed to register services.

Using the magnifying glass allows to search for more projects.

As soon as the project is selected, the detailed mask for service input appears. It is now possible either to start the timer or to input the hours worked as well as other information.



Swisscom 09:57 93%

Choose project

Search

COMINSTALL
Entwicklung von Software
Comtelsat AG

COM-SSUP
Support XTrem-E
Comtelsat AG

COOPER-BH
Führung der Buchhaltung
Bistro Cooper

DOIT-IT
Beratungen IT Infrastruktur
Do it for IT AG

GALL-STEUER
Steuern
Gallfor AG

KLAMEXPO
Netzwerk
Karstner Rolf

MB-NETZ
Netzwerkberatungen
Meyer Blatter Rechtsanwälte

Swisscom 09:58 93%

Service

Project: COMINSTALL

Phase: 4 CHANGE

Service type

Text

Date: 2/22/2018

Hours: 0:00

Start timer

Swisscom 09:59 92%

Service

COMINSTALL

Phase: 4 CHANGE

Service type

Text

Date: 2/22/2018

Hours: 0:30

Done

	20
	25
0 Hours	30 Minutes
1	35
2	40

Figure 9: Inputting a service

A service for which the timer is started appears in the list with the green timer symbol:

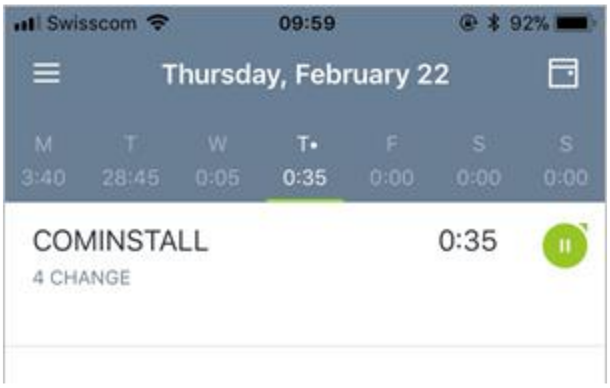


Figure 10: Service with timer started

Clicking on the timer symbol stops the timer.

3.7 Menu

All other features are available in the Phone App Menu which opens be clicking on the Menu symbol:

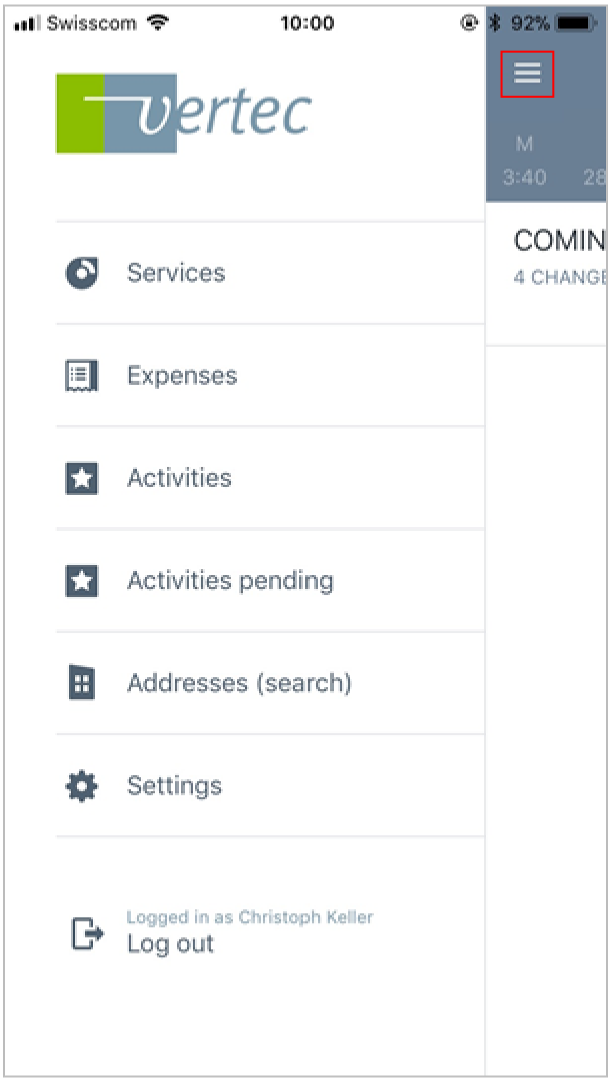
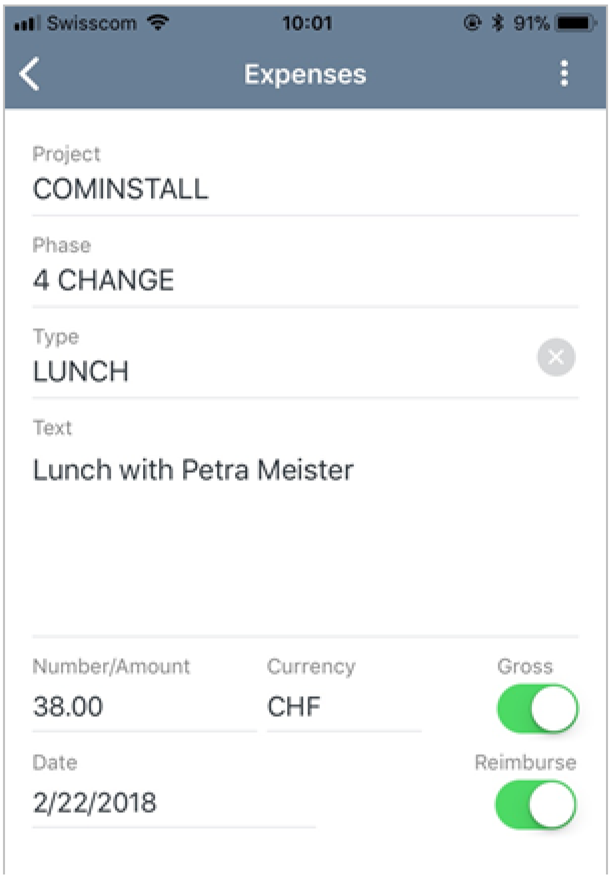


Figure 11: The Phone App menu

3.8 Registering expenses

The input of expenses happens like the input of services. Buttons are available for gross values and refund options:



Swisscom 10:01 91%

Expenses

Project
COMINSTALL

Phase
4 CHANGE

Type
LUNCH

Text
Lunch with Petra Meister

Number/Amount 38.00

Currency CHF

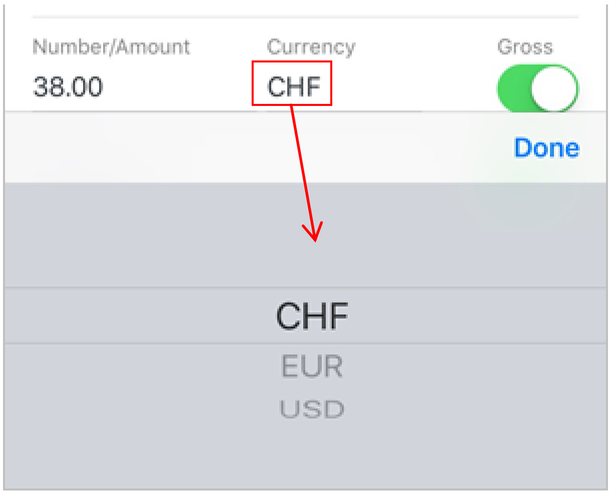
Gross

Date 2/22/2018

Reimburse

Figure 12: Expenditure details

If necessary, the currency can be changed (Line Expert):



Number/Amount 38.00

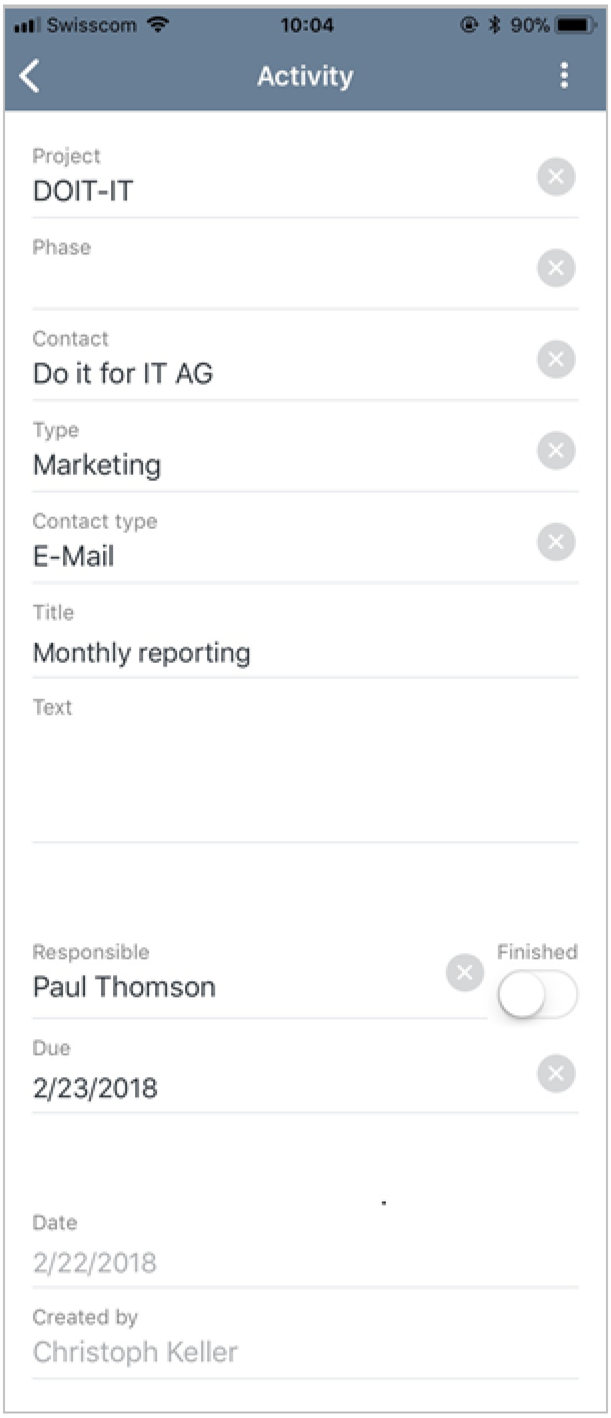
Currency CHF

Gross

Figure 13: Specifying the currency

3.9 Registering activities

Menu item Activities allows the input of activities.



Project
DOIT-IT

Phase

Contact
Do it for IT AG

Type
Marketing

Contact type
E-Mail

Title
Monthly reporting

Text

Responsible
Paul Thomson

Finished

Due
2/23/2018

Date
2/22/2018

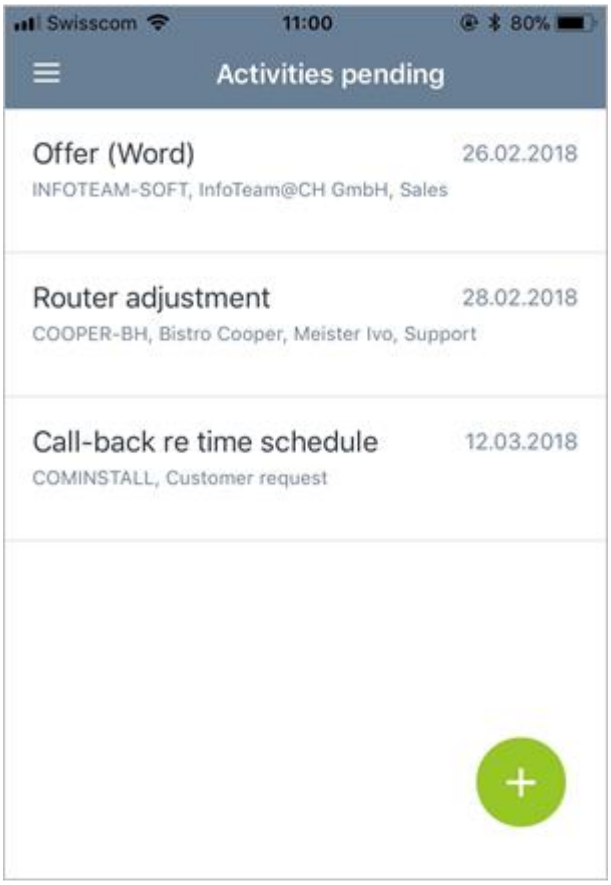
Created by
Christoph Keller

Figure 14: A Phone App activity (complete view including scrolling)

In the upper part, the details of the activity are entered. In the lower part, a person in charge and a date can be input. The activity thus becomes a pending activity and appears to the person in charge under Activities pending.

3.10 Activities pending

The list of pending activities shows all activities of the logged-in user which have not been taken care of:



Swisscom 11:00 80%

Activities pending

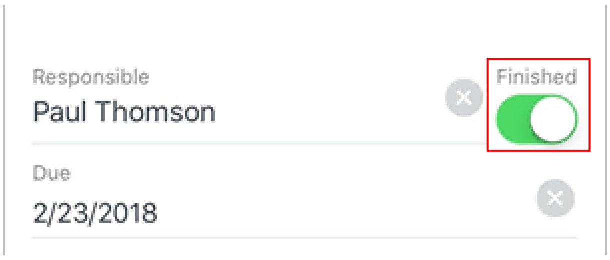
Offer (Word) 26.02.2018
INFOTEAM-SOFT, InfoTeam@CH GmbH, Sales

Router adjustment 28.02.2018
COOPER-BH, Bistro Cooper, Meister Ivo, Support

Call-back re time schedule 12.03.2018
COMINSTALL, Customer request

Figure 15: List of pending activities

When an activity is completed, the activity can be set to `Finished` in the detail view. It then disappears from the list of pending activities.



Responsible: Paul Thomson

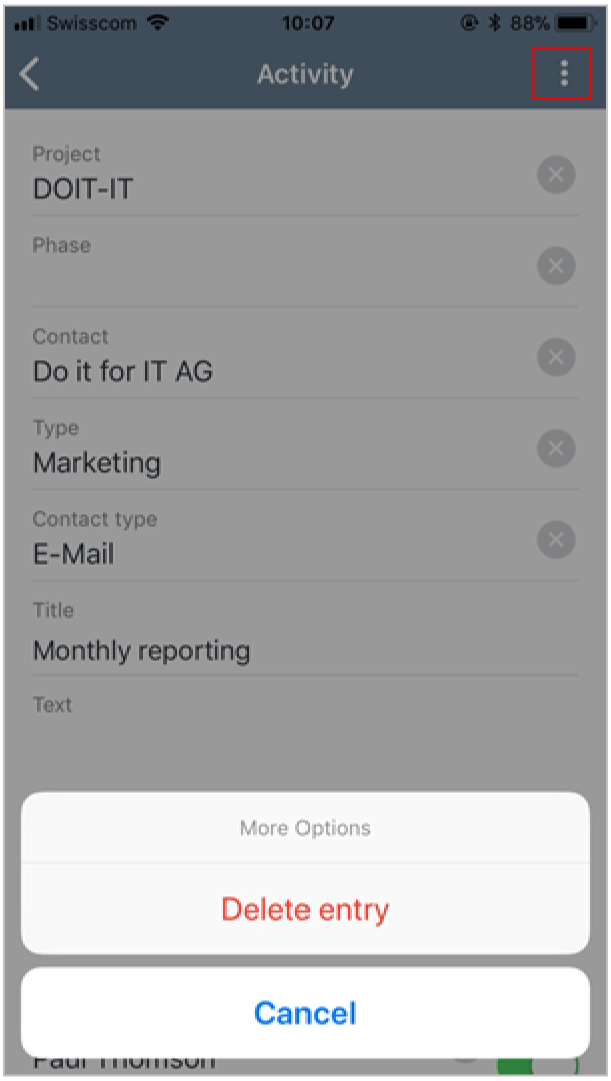
Finished

Due: 2/23/2018

Figure 16: An activity is finished

3.11 Deleting an entry

In the detail view, the context menu can be displayed by clicking on the button with the three dots.



Project
DOIT-IT

Phase

Contact
Do it for IT AG

Type
Marketing

Contact type
E-Mail

Title
Monthly reporting

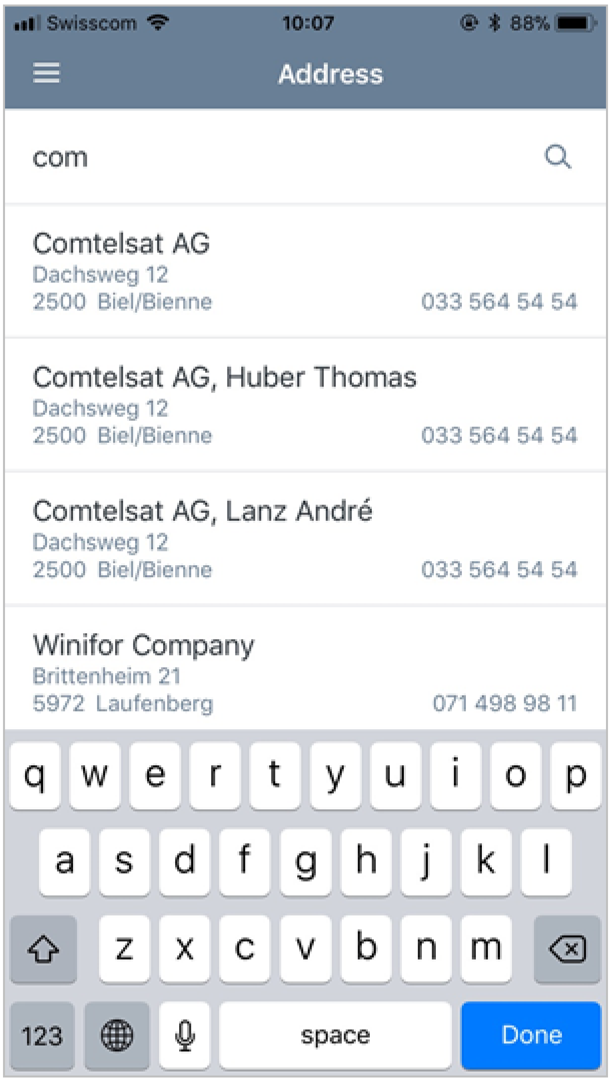
Text

Figure 17: Deleting a Phone App entry

To delete the entry, click on the `Delete` button. The entry is deleted without further confirmation.

3.12 Display addresses

New addresses cannot be entered in the Phone App, but existing addresses can be displayed. Clicking on the menu item `Addresses (search)` opens a search mask. Input the desired value and click on the magnifying glass.



com

Comtelsat AG
Dachsweg 12
2500 Biel/Bienne 033 564 54 54

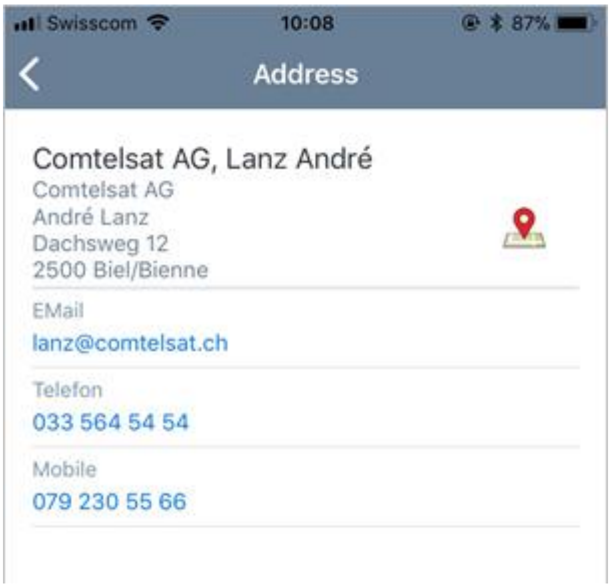
Comtelsat AG, Huber Thomas
Dachsweg 12
2500 Biel/Bienne 033 564 54 54

Comtelsat AG, Lanz André
Dachsweg 12
2500 Biel/Bienne 033 564 54 54

Winifor Company
Brittenheim 21
5972 Laufenberg 071 498 98 11

Figure 18: Searching for addresses in the Phone App

Clicking on an address opens the detail view.



Swisscom 10:08 87%

Address

Comtelsat AG, Lanz André
Comtelsat AG
André Lanz
Dachsweg 12
2500 Biel/Bienne

EMail
lanz@comtelsat.ch

Telefon
033 564 54 54

Mobile
079 230 55 66

Figure 19: A single address in the Phone App

From this screen you can directly place a phone call, send an email, open the website or open the address in the Map app of your smartphone.

4 Extended Office reports

Office reports, Word reports in particular, are an established part of Vertec and have proved their worth. Since the release of Vertec 6.1, Word reports can be generated in Vertec also and are therefore available in Web App as well. Word reports however are not well suited for complex calculations because initial dialogs are not available and complex logic is hard to specify in the reports.

Moreover there exist in Vertec two reporting systems which are not cloud-compatible and will be discontinued in Vertec 6.3 (see 2.3).

This is the reason why Vertec now provides a new cloud-compatible report generator based on Office: extended Office reports.

Extended Office reports combine a Word template (.docx file) describing the layout structure with a report definition in Python code to generate the contents.

Office reports provided by Vertec

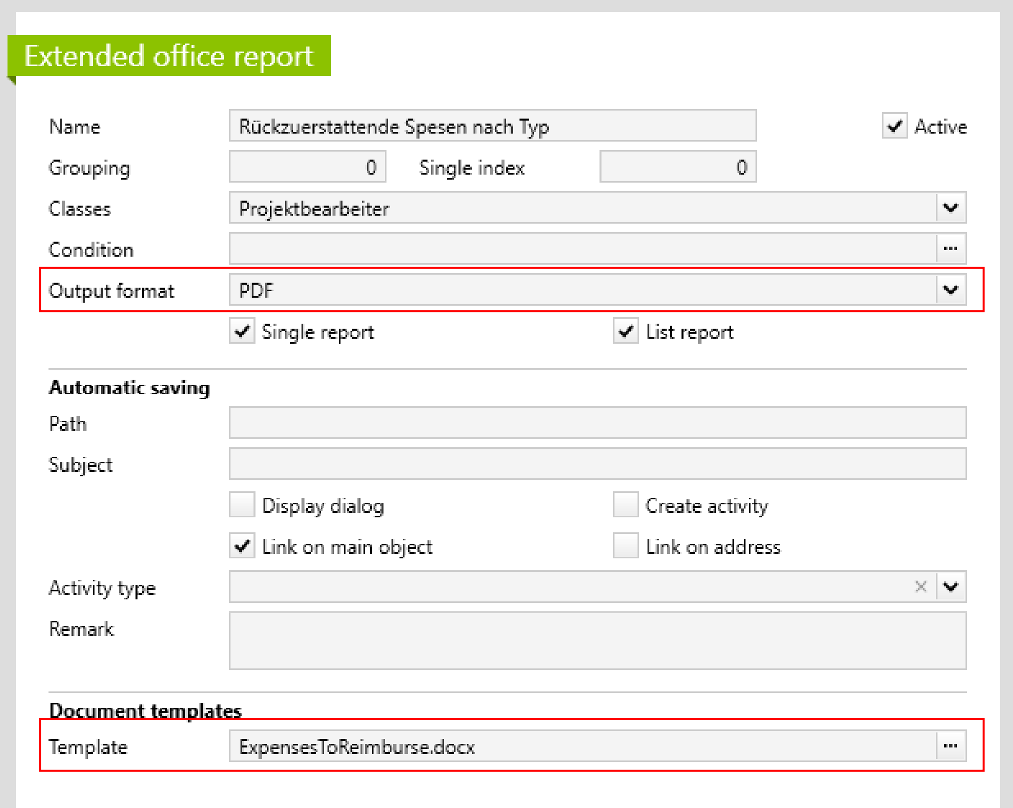
Vertec 6.2 comes with the extended Office reports listed below. They are stored in the standard report directory and registered in Vertec, yet remain inactive.

In Vertec 6.3, the existing report registrations will be deleted and the extended Office reports will be activated. Please ensure before this change that all reports that you need exist as extended Office reports. Please take note of chapter 2.3 on this subject.

Report	Registered on	Template document
Invoice proposal	Project/Mandate	InvoiceProposal.docx
Commenced work	Project/Mandate	CommencedWork.docx
Vacation account	Collaborator	VacationReport.docx
Recapitulation of bookings	Project/Mandate	RecapitulationOfBookings.docx
Expenses to reimburse by type	Collaborator	ExpensesToReimburse.docx
Overtime control	Collaborator	OvertimeReport.docx

4.1 Registration of extended Office reports

The following differences exist between the registration of existing Office reports and the registration of extended Office reports:



Extended office report

Name: Rückzuerstattende Spesen nach Typ ☑ Active
Grouping: 0 Single index: 0
Classes: Projektbearbeiter
Condition:
Output format: PDF
☑ Single report ☑ List report

Automatic saving
Path:
Subject:
□ Display dialog □ Create activity
☑ Link on main object □ Link on address
Activity type:
Remark:

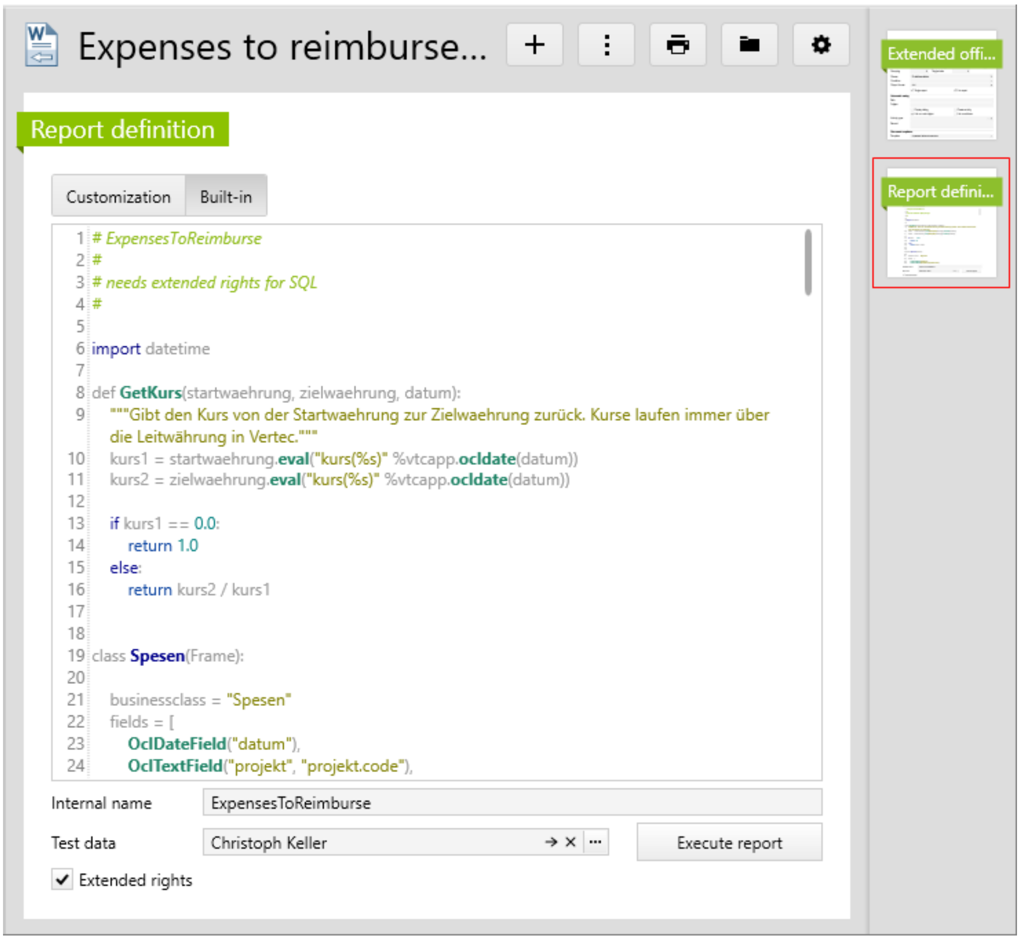
Document templates
Template: ExpensesToReimburse.docx

Figure 20: Registration of an extended Office report

- As extended Office reports are always generated by Vertec, the corresponding selection no longer exists. It is however still possible to choose between Word and PDF for the output format.
- Under Template specify the template document. Unlike existing Word reports where a `.dotx` document was used, this is a normal Word document with `.docx` ending.
- Only one single template remains. The list of templates for different languages as existed with Office reports no longer exists. Translations are performed by the normal Vertec translation mechanism (see 4.12).

4.2 Report definition

The Python code belonging to the template is input on the page Report definition.



```
# ExpensesToReimburse
#
# needs extended rights for SQL
#

import datetime

def GetKurs(startwaehrung, zielwaehrung, datum):
    """Gibt den Kurs von der Startwaehrung zur Zielwaehrung zurück. Kurse laufen immer über
    die Leitwährung in Vertec."""
    kurs1 = startwaehrung.eval("kurs(%s)" %vtcapp.ocldate(datum))
    kurs2 = zielwaehrung.eval("kurs(%s)" %vtcapp.ocldate(datum))

    if kurs1 == 0.0:
        return 1.0
    else:
        return kurs2 / kurs1


class Spesen(Frame):

    businessclass = "Spesen"
    fields = [
        OclDateField("datum"),
        OclTextField("projekt", "projekt.code"),
```

Internal name: ExpensesToReimburse

Test data: Christoph Keller

Execute report

☑ Extended rights

Figure 21: Definition of the contents of an extended Office report

The internal name designates the code provided by Vertec. It has the same name as the report and is displayed under tab Built-in.

Under tab Customization, the code provided can be overwritten. If any code is present under Adapted, only that code is valid. To change something, it is therefore necessary to copy the complete code from Built-in and modify it under Customization.

Extended Office reports created or modified by the user have no Built-in code and no internal name, the code is directly inserted under Customization.

The structure of the Python code is described in detail in chapter 4.3.

To test the code, choose a Vertec object from Test data. A click on the `Execute Report` button then directly executes the report for this object.

If additional authorisations are required in the Python code, e.g. for a SQL access, the field `Extended rights` below the report definition must be set.

4.3 Structure of the Python code

The data logic of an extended Office report is based on a hierarchy of so-called frames. A frame can be compared to a table and consists of lines with data fields. The fields of a specific frame are specified by field definitions.

The Python code of an extended Office report mainly consists of the declaration of the frame, beginning with the main frame (`main_frame`) of the report.

A minimal report definition must include a frame declaration and the assignment of the `main_frame` variable:

```
# Example of a minimal collaborator report

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

main_frame = Bearbeiter
```

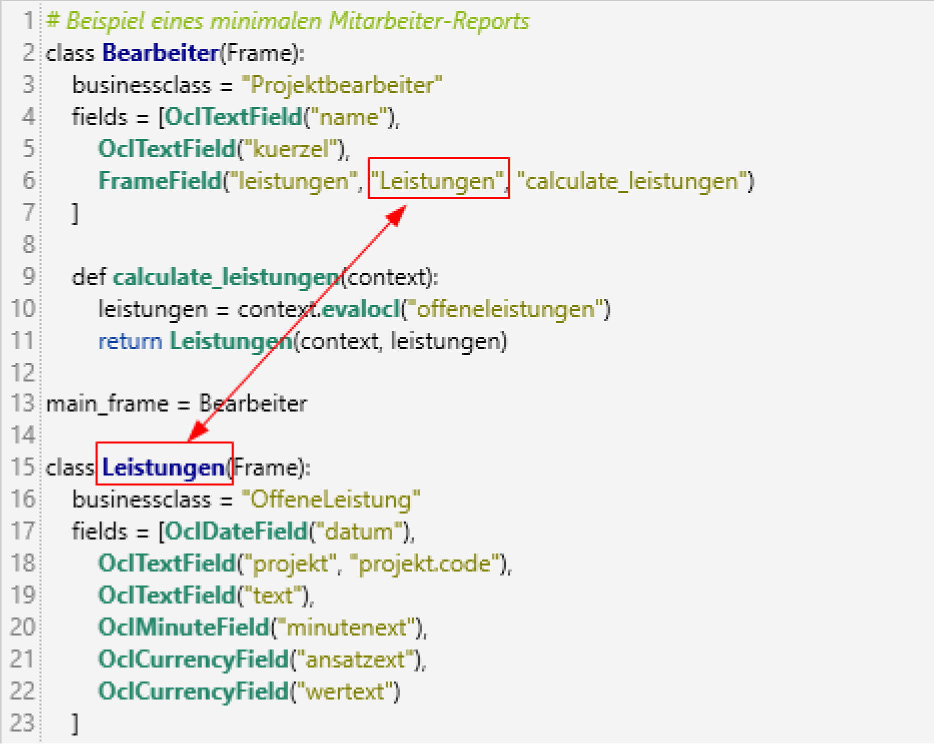
Here a frame is defined with 2 fields to be computed by OCL.

4.4 Field types

The following field types are available. Field types are defined by name and optional specifications for their computation. For each type (except FrameField) there is also a variant with prefix `Ocl` (see OCL fields below).

TextField	String value
CurrencyField	Fixed-point value, usually used for currency amounts. Format according to country setting.
IntegerField	Integer value
MinuteField	Minutes as an integer value. Format according to setting in Vertec.
BooleanField	True/false value
DateField	Date
DateTimeField	Date and time
ImageField	Picture
FrameField	The value is another frame. Frame fields are used to represent hierarchical data structures.

For frame fields, the frame type being represented must always be indicated as second parameter.



```
# Beispiel eines minimalen Mitarbeiter-Reports
class Bearbeiter(Frame):
    businessclass = "Projektbearbeiter"
    fields = [OclTextField("name"),
        OclTextField("kuerzel"),
        FrameField("leistungen", "Leistungen", "calculate_leistungen")
    ]

    def calculate_leistungen(context):
        leistungen = context.evalocl("offeneleistungen")
        return Leistungen(context, leistungen)

main_frame = Bearbeiter

class Leistungen(Frame):
    businessclass = "OffeneLeistung"
    fields = [OclDateField("datum"),
        OclTextField("projekt", "projekt.code"),
        OclTextField("text"),
        OclMinuteField("minutenext"),
        OclCurrencyField("ansatzext"),
        OclCurrencyField("wertext")
    ]
```

Figure 22: Specifying a frame field

The frame type can be given as a direct reference or as name of a declared frame. The reference by name allows placing the frame declaration anywhere in the code. When using the direct reference, the referenced frame declaration must occur earlier in the code.

Computed fields

The fields of a frame can be computed automatically. Computation is done with OCL or Python code.

OCL fields

An OCL Field (OclTextField, OclCurrencyField, OclBooleanField etc.) is created by specifying a field name. In the simplest case, the field name is also the OCL expression used for its computation:

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

Optionally, the OCL expression can be indicated as a separate second argument. This is particularly useful and meaningful in case of a longer expression.

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

OCL-fields are exclusively allowed on business object frames, i.e. frames which are attributed to a specific class with `businessclass=`.

Fields calculated manually or with Python code

Field types without Ocl prefix can be limited to a name. Without further indication, they represent fields which are not computed automatically. When the frame is evaluated, they can be filled with values in the code.

Optionally a Python function can be indicated as a second argument to compute the field. This function can be defined as a direct function reference or as name of a function (string) defined within the frame.

Computation of fields by means of code

When a field is computed by means of code, a function reference (direct reference or name of the function) is passed to the field.

The function used for the computation must include an argument and the computation context and must return the computed field value.

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

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

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

        return sum
```

Computation of frame fields

Fields of the type `FrameField` can also be computed. If the computation is performed with an OCL expression, the result of the expression must be a list of business objects corresponding to the frame declaration.

If the computation is performed with a Python method, the method must return a frame instance of the correct type.

In most cases, the frame instance is created in the code. For this purpose, the `add_row` method is provided which allows building the frame line by line. The fields defined within the frame are available as properties of the `Row` object.

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

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

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

        frame = DetailsFrame(context)

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

Calculated main frame

The main frame of a report is not a field value but can also be calculated by function or OCL.

This happens with `calculate_main_frame` in the report definition. The following variants are possible:

- `calculate_main_frame` is a function which is passed a context object and must return a frame instance of the correct type.

  The signature of the function is `calculate_main_frame(context)`; the returned value is a frame instance.

  The implementation should check the type of the frame created, in agreement with the report definition.

  Example:
  The report is registered to a list of projects; the main frame however corresponds to a list of collaborators. The main-frame class is called "collaborator".

```
main_frame=Bearbeiter

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

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

- `calculate_main_frame` is a String. It is evaluated as an OCL expression and must return a list. The expression is evaluated using the original list of the report and can use the OCL variables in context.

  Same example as above, with expressions:

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

4.5 Context object

A context object is passed to every computation within the report generation. This object is globally available and has the following properties:

Variables

The following variables are defined in every report:

currentobject	The actual object of the computation. `context.currentobject`
currentdate	Today’s date, without the time part. `context.currentdate`

optarg

The optional address argument in the case of reports. To work with it, it is possible to access this object in the following way:

```
class Projekt(Frame):
    fields = [
        OclTextField("code"),
        OclTextField("beschrieb"),
        OclTextField("creationdatetime"),
        TextField('adresstext', 'calcadresse'),
    ]

     def calcadresse (context):
        return context.optarg.adresstext

main_frame = Projekt
```

rootlist

The list on which the report was executed. Usually, this is the `eintraege` list of the container on which the report is executed, or a list containing the object on which the report was executed.

container

The container on which the report was executed.

var<frame>

For each superior frame a variable is defined with its actual object.

var<frame>List

For each superior frame a variable is defined with the list of the objects of the frame.

Other variables can be defined at will and attributed to the context object.

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

Such variables can later be output on the report using a context expression (see chapter 4.11).

Method evalocl(expression)

In addition, the context object possesses the method `evalocl(<expression>` which evaluates an OCL expression on the current object (`currentobject`).

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

4.6 Before report logic

The method `before_report(context)` allows executing Python code before the execution of the report (e.g. to display dialogues and set parameters of the context object):

```
def before_report(context):

  """Ask the user to enter the date."""
  initValues = {}
  initValues["Referencedate"] = vtcapp.currentdate()
  dlgDefinition="""
    <Dialog Title="{Translate 'Choose date'}" Width="400">
      <Group Orientation="Vertical">
        <DatePicker Name=" Referencedate" Label=" Referencedate" />
  </Group>
  <Dialog.Buttons>
   <Button Text="OK" IsAccept="True" Command="{Binding OkCommand}" />
   <Button Text="Cancel" IsCancel="True" Command="{Binding CancelCommand}" />
    </Dialog.Buttons>
 </Dialog>
"""

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

context.referencedate = values["Referencedate"]
```

4.7 Structure of the Word templates

Unlike existing Word reports which used .dotx files as templates, extended Word reports use normal Word Documents (.docx) as templates.

4.8 Expressions

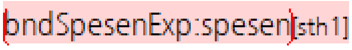
Since computations now occur in the code rather than in the report, the Word report no longer contains long OCL Expressions, only references to the corresponding code.

As previously, control happens through a text marked by a comment within a band. The references however are no longer contained in the comment but in the text which is marked. The comment is empty or holds the keyword `translation` to activate the translation for this field (see below).

4.9 Bands

As previously, everything that is printed is located within a band. There exists therefore at least a main band holding everything. Included sublists are also located in a band, as before.

As before, there are Exp-, Cond- and Locale-bands. The marked text performs the control (see above). The reference to the code is indicated as `:name.` The band expression is, for instance, written as `bndExpensesExp:expenses`; this text is marked and includes an empty comment:



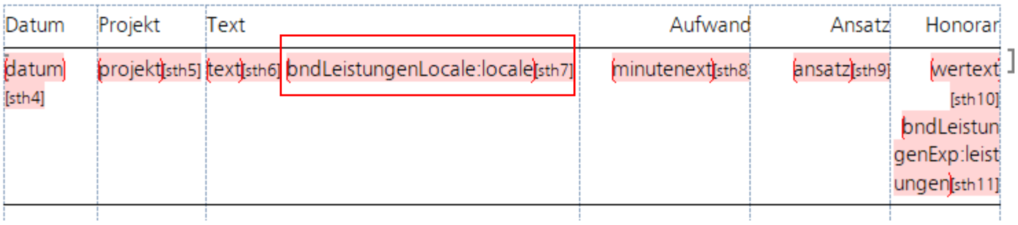
bndSpesenExp:spesen[sth1]

More information on bands in Word reports in the Online Knowledge Base under https://www.vertec.com/de/kb/wordreports.

4.10 Locale

Numerical values are formatted according to the regional settings. With locale expressions, numerical values can also be formatted according to other country settings

Locale expressions can be expressed in a way similar to band and conditional expressions in the report template. The reference to the code is given with `:name`. The band expression then becomes for instance `bndServicesLocale:locale`; the text is marked and completed with an empty comment:



Datum	Projekt	Text		Aufwand	Ansatz	Honorar
datum[sth4]	projekt[sth5]	text[sth6]	bndLeistungenLocale:locale[sth7]	minutenext[sth8]	ansatz[sth9]	wertext[sth10] bndLeistungenExp:leistungen[sth11]

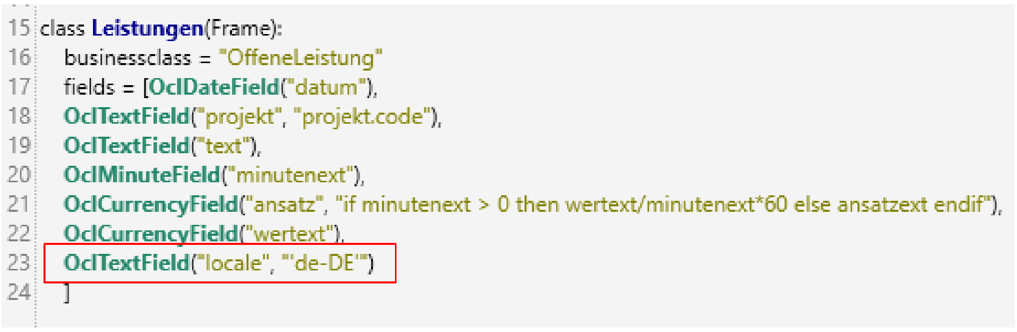
Figure 23: Locale band

The country code can also be indicated directly, e.g.:

`bndLeistungenLocale:de-DE`

All numerical values inside this band are then formatted according to the specified country settings.

To this effect, an `OclTextField` with the specified name is inserted in the corresponding frame (in this example `locale`):



```
class Leistungen(Frame):
    businessclass = "OffeneLeistung"
    fields = [OclDateField("datum"),
    OclTextField("projekt", "projekt.code"),
    OclTextField("text"),
    OclMinuteField("minutenext"),
    OclCurrencyField("ansatz", "if minutenext > 0 then wertext/minutenext*60 else ansatzext endif"),
    OclCurrencyField("wertext"),
    OclTextField("locale", "'de-DE'")
    ]
```

Figure 24: Field to specify the locale expression

The result of the specified OCL expression must be a country code. It can be directly given as in the above example (`'de-DE'`) or be for instance selected from a field OclTextField("locale", "projekt.zusatzfeldasstring('locale')").

More information Additional information on locale expressions can be found in the Online Knowledge Base under https://www.vertec.com/de/kb/wordreports#locale.

4.11 Context expressions

With so-called context expressions, variables can be directly output on the report context without specially having to create a field on a frame definition.

A context expression is designated with the prefix `context:` and references a context variable of the report (see 4.6). Example:

```
context:referencedate
```

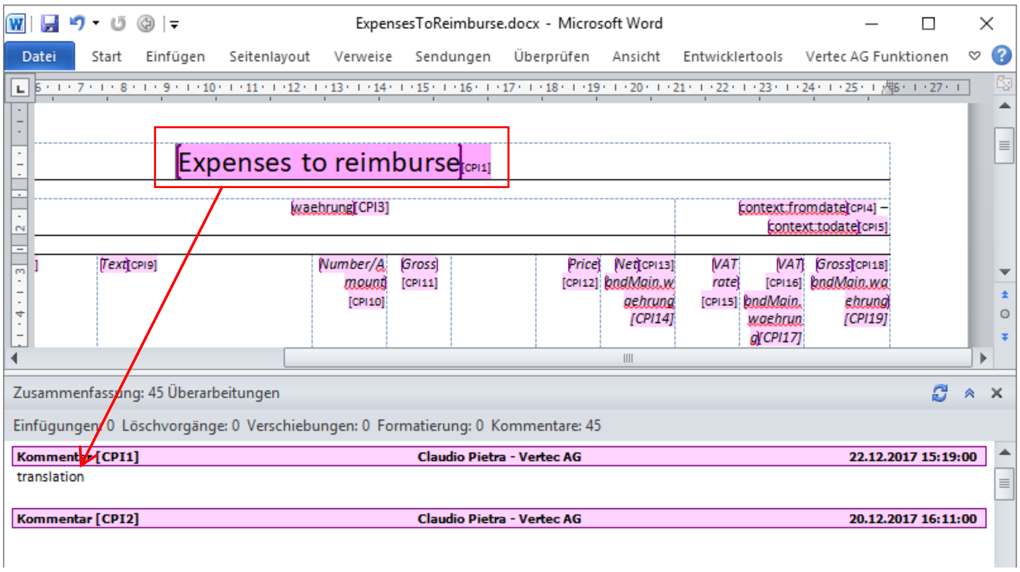
The data type of the field corresponds to the value of the context variable. If the context variable is an object reference or a list, the context expression returns a string value (string representation).

Use case: reports with computations often include a band which is iterated, while the rest of the report consists of static content such as titles, etc. Reports of this type can be implemented by defining the iterated band as the main band (and thus as main frame). If it is nevertheless desired to display a report parameter in the title, the reference date of the report for instance, it is not necessary to create a new reference frame with a calculated frame field: the context expression can directly be specified.

4.12 Translations

Separate Word templates for the various languages no longer exist. The Vertec translation system is used instead.

To translate a field, the keyword `translation` is indicated as comment. The translation mechanism looks up the term in the translation file and, if available, inserts the translated value.



Expenses to reimburse[CPI1]

waehrung[CPI3]

context.fromdate[CPI4] – context.todate[CPI5]

Text[CPI9] Number/Amount[CPI10] Gross[CPI11] Price[CPI12] Net[CPI13] bndMain.waehrung[CPI14] VAT rate[CPI15] VAT[CPI16] bndMain.waehrung[CPI17] Gross[CPI18] bndMain.waehrung[CPI19]

Zusammenfassung: 45 Überarbeitungen

Einfügungen: 0 Löschvorgänge: 0 Verschiebungen: 0 Formatierung: 0 Kommentare: 45

Kommentar [CPI1] Claudio Pietra - Vertec AG 22.12.2017 15:19:00

translation

Kommentar [CPI2] Claudio Pietra - Vertec AG 20.12.2017 16:11:00

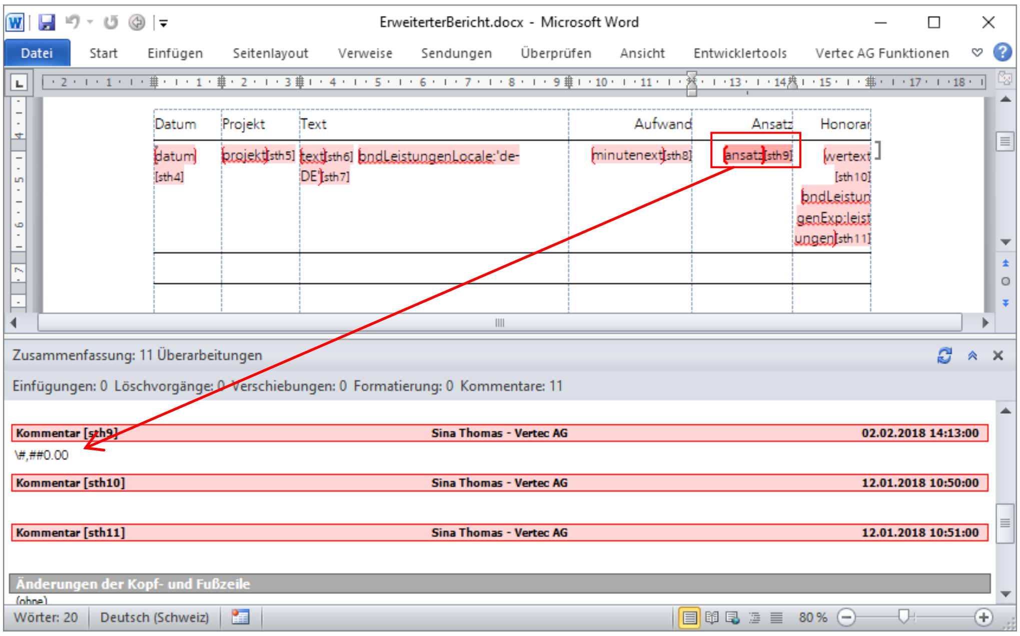
Figure 25: The expression is translated

This implies that expressions which are exclusively used in reports (e.g. the report title) must be inserted into the translation file. How this is done is explained in the Online Knowledge Base under https://www.vertec.com/de/kb/sprachen#userdict.

4.13 Formatting fields

In extended Office reports also, date and numerical values can be formatted according to Visual Basic formatting rules.

The format to be used is inserted in the comment part of the field to be formatted:



Datum	Projekt	Text	Aufwand	Ansatz	Honorar
datum [sth4]	projekt [sth5]	text [sth6] bndLeistungenLocale:'de-DE' [sth7]	minutenext [sth8]	ansatz [sth9]	wertext [sth10] bndLeistungenExp:leistungen [sth11]

Zusammenfassung: 11 Überarbeitungen

Einfügungen: 0 Löschvorgänge: 0 Verschiebungen: 0 Formatierung: 0 Kommentare: 11

Kommentar [sth9] Sina Thomas - Vertec AG 02.02.2018 14:13:00

#,##0.00

Kommentar [sth10] Sina Thomas - Vertec AG 12.01.2018 10:50:00

Kommentar [sth11] Sina Thomas - Vertec AG 12.01.2018 10:51:00

Änderungen der Kopf- und Fußzeile

(ohne)

Figure 26: The format is indicated in the comment field

The formatting only works with numerical values (currency, integer). The local regional settings are used for this purpose. To use other regional settings, a locale band can be specified (see section 4.10).

More information Additional information on the formatting of numbers and dates is available in the Online Knowledge Base under https://www.vertec.com/de/kb/vbformatierung.

4.14 Pictures

Pictures from the data base (ImageData) are declared in Python Code as `ImageField` or `OclImageField` (see 4.3). In the report document, they may be inserted as normal data fields (commented text).

Specifying a path to an image file (`ImagePath`) is not yet implemented and will be available in a future release.

4.15 Expressions in headers and footers

Context variables (see 4.11) can also be used in headers and footers.

Since comments are not available in headers and footers, variables are specified here in double curly braces, e.g.:

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

5 Config Sets

Vertec installations can to a very large extent be adapted to the requirements of the customers. The main part of the possible configurations occurs in the data-base of the customer.

Configurations, grouped by theme, can now be stored in a file in a data-base independent format and used for another installation. Such a configuration definition is called a Config Set.

5.1 Config Set dialogue

Config Sets are managed with the Config Set dialogue. This dialogue is started with menu `Settings > Config Sets`.

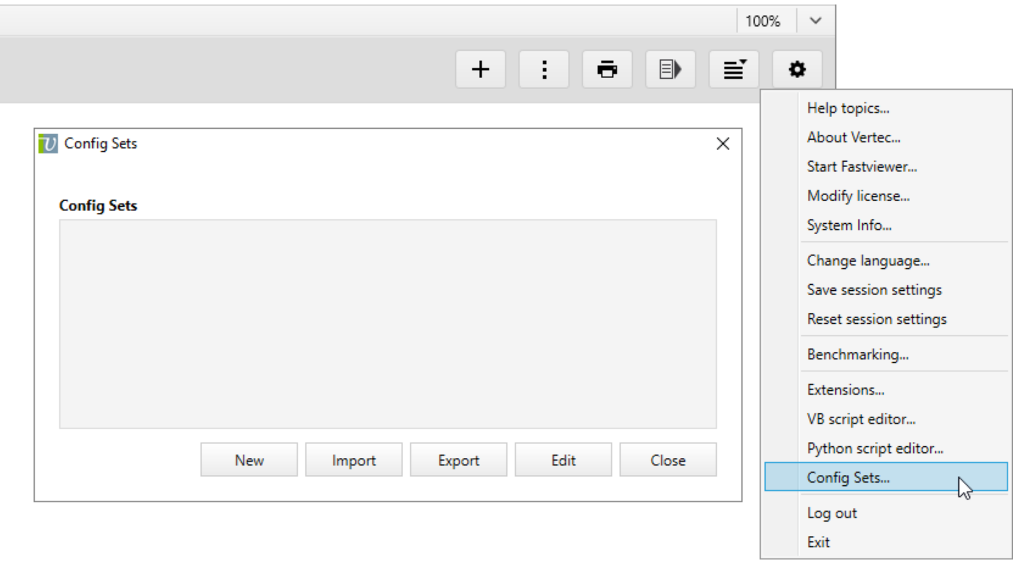


Figure 27: The Config Set dialog

5.2 Importing Config Sets

A Config Set is imported with the Import action in the Config Set dialogue. A click on the `Import` button displays an explorer in which the Config Set XML-file can be selected.

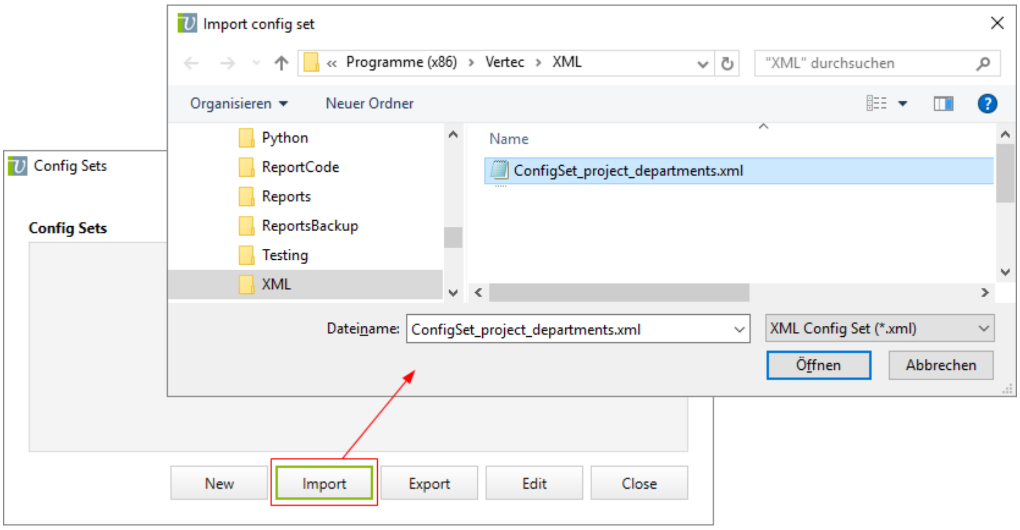


Figure 28: A Config Set is imported

Plausibility check

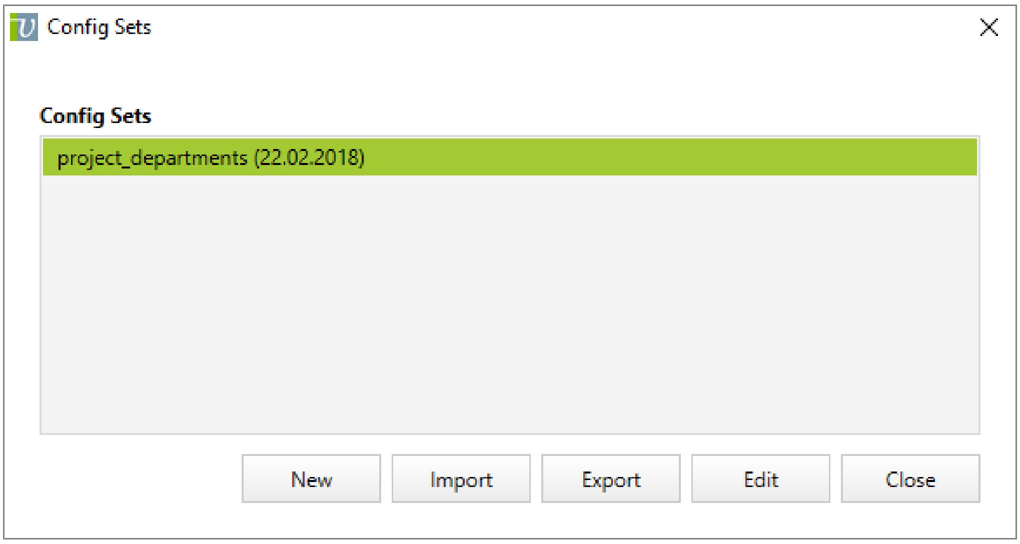
When a Config Set is imported, its plausibility must first be checked. The following requirements must be met before the set is used:

- Preconditions explicitly expressed in the Config Set must be met
- All external references listed exist
- All internal aliases are unique
- Members with references point to defined aliases
- Additional classes in the Config Set do not yet exist in the system
- Additional fields in the Config Set do not yet exist in the system

If errors occur, an error message appears and the Config Set is not applied.

If no errors occur, the Config Set is directly applied and the corresponding objects are created in Vertec.

After a Config Set has been applied, it appears in the list in the Config Set dialogue.



project_departments (22.02.2018)

Figure 29: The imported Config Set appears in the list

Applying a Config Set via Python

There exists a new Python method `vtcapp.importconfigset()` which accepts a Config Set as string (XML text), e.g.:

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

The Config Set imported in this way also figures in the list of the Config Set dialog.

5.3 Creating Config Sets

Vertec provides the Config Set Builder to create Config Sets. It allows specifying the configuration interactively and to export the Config Set as an XML data file.

To create a new Config Set, click on the `New` button in the Config Set dialogue. The Config Set Builder opens:

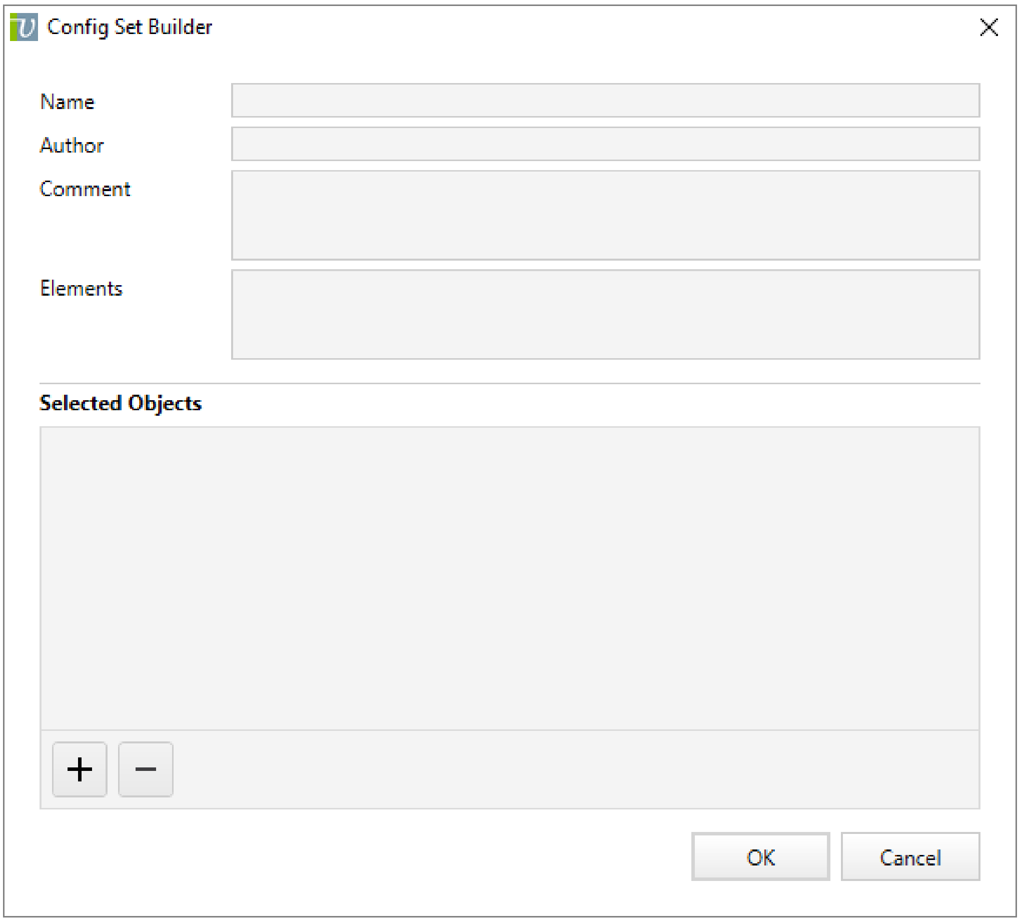


Figure 30: The Config Set Builder

Name	Name of the Config Set. Since Config Sets can reference each other (see Elements), the name must be unique.
Author	Author of the Config Set.
Comment	Comment text to precisely describe the Config Set.

Elements

Requirements, references and settings are specified here. The text must be a valid XML expression (see 5.4).

The following elements are supported:

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

The elements can be indicated directly, without superior node.

Instead of:

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

is indicated:

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

Adding/removing objects

In Selected objects, the Config Set Builder displays a list of the objects which are part of the Config Set.

To add a specific object to the list, select it in the active Vertec window, click on the + button in the Config Set Builder or simply pull it into the field with drag & drop.

To remove an object from the list, mark it and click on the - button.

In the case of strictly hierarchical objects (e.g. folders) which possess lower level objects, the lower-level objects are automatically included.

Exporting Config Sets

A click on the `Export` button lets the Config Set Builder create an XML file which can be stored.

This Config Set can then be imported in a target installation (see chapter 5.2).

If, during the export, an error message of the following type appears:

```
No program attributed to execute the action for this file…
```

You must in Windows attribute a standard app to the file type .xml, for instance the Editor.

5.4 Config Set XML

The Config Set definition consists of the following elements (some of which are optional):

Attributes

author	Author of the Config Set. The value of the field `Author` in the Config Set Builder (5.3) is automatically inserted.
name	Name of the Config Set. The value of the field `Name` in the Config Set Builder (5.3) is automatically inserted.
ref-name	Name of a referenced Config Set.
class	Class name of the object.
entryid	Entry Id of the object in Vertec (see 5.5)
alias	Alias of the element within the Config Set
ref-alias	Alias of a referenced Config Set
expression	Expression attribute of OCL references

Elements

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

<comment />

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

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

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

Requirements

A collection of pre-requirements which must be satisfied to allow the Config Set to be accepted.

The requirements are entered in the field `Elements` of the Config Set Builder (see 5.3).

The following types of requirements exist:

- `configset-requirement`: designates an underlying Config Set which must be installed already.

```
<configset-requirement>Project_Departments</configset-requirement>
```

- `ocl-requirement`: any OCL expression which must return true. This allows checking preconditions required to import the Config Set, e.g. the fact that a certain additional class does not yet exist (see 5.6):

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

References
A collection of references used within the Config Set. Each reference has a unique name within the Config Set.

References can be of the following types:

- `entryid-reference`: references an object on the base of its class and its Entry Id. This is the way to identify objects already existing in the target system (see chapter 5.5).

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

  When the config set is created, entryid-references are automatically inserted via the Config Set Builder.

- `configset-reference`: references a named object from another Config Set. This object must already exist and should therefore be entered in the requirements.

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

  The configset-references are entered in the field `Elements` of the Config Set Builder (see 5.3).
- `ocl-reference`: references an object through an OCL expression. The OCL expression is evaluated globally. This is the way to identify an object existing in the target system and operate on it in the Config Set.

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

  The ocl-references are entered in the field `Elements` of the Config Set Builder (see 5.3).

Settings
Settings can be specified as well. They are entered in the field `Elements` in the Config Set Builder (see 5.3).

- `System settings`: Used to set a system setting. As name, use its **PropertyName** which can be found with the description of the setting in the Online Knowledge Base under www.vertec.com/kb.

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

Objects

The Config Set mainly contains a list of objects which must be created by the Config Set. The objects contain values of properties and links.

```
<objects>
    <object class="Ordner" alias="Ordner_ProjectsbyDepartments0">
      <member name="Bezeichnung">Projekte by Departments</member>
      ...
    </object>
</objects>
```

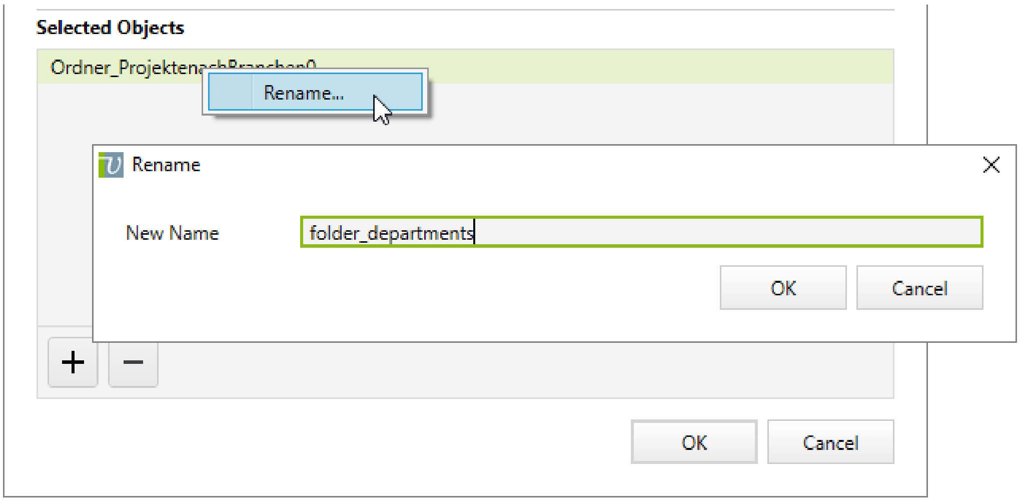
In the case of links, either an object generated in the same Config Set or a reference defined at the beginning with its alias is specified (see next section).

```
      <member name="Parentordner">
        <reference-object
           alias=" Ordner_ProjectsbyDepartments0_Parentordner" />
      </member>
```

Object aliases

Objects within the Config Set can optionally be provided with an alias and thus be explicitly referenced within the Config Set or from another Config Set. Within a Config Set, an alias must be unique.

When an object is added in the Config Set Builder, it automatically receives an alias. If you wish to change it, right- click on the object in the list > `Rename` and input the desired alias.



Selected Objects

Ordner_ProjektenachBranchen0

New Name: folder_departments

Figure 31: Rename an object

Existing objects

In certain situations, rather than creating a new object the properties of an existing object are used.

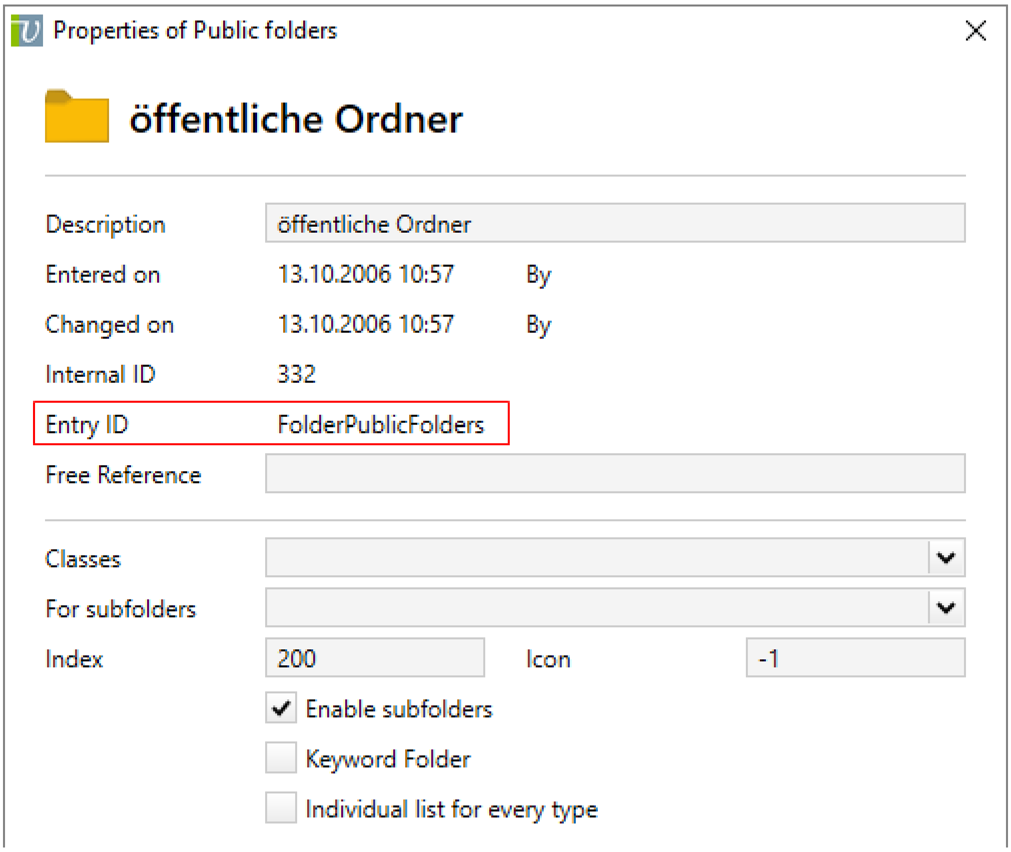
- If the object has an Entry Id, the system looks for the existing object in the corresponding class and updates it. See chapter 5.5 regarding the Entry Id.
- Class settings are treated in a special way: the system looks for the existing object with the name of the class. This special procedure is not applicable to class settings of additional classes and an error is generated if one attempts to change class settings for an existing additional class (see 5.6)

– Setting properties to their default value (e.g. string to empty string) is not supported for existing objects.

5.5 Using the Entry Id

The Entry Id of Vertec objects occupy a central position in the use of Config Sets. It is used to determine if the object already exists in the target data-base or not.

The Entry Id is found in the `Properties` dialogue for the corresponding objects (right mouse button > Properties):



Properties of Public folders

öffentliche Ordner

Description	öffentliche Ordner		
Entered on	13.10.2006 10:57	By	
Changed on	13.10.2006 10:57	By	
Internal ID	332		
Entry ID	FolderPublicFolders		
Free Reference			
Classes			
For subfolders			
Index	200	Icon	-1

☑ Enable subfolders

□ Keyword Folder

□ Individual list for every type

Figure 32: Entry Id of a public folder

Objects passed without Entry Id reference (`entryid="XX"`) are created in the target system.

When a Config Set is created (see 5.3), it is checked at the time the object is inserted that the parent folder possesses an Entry Id. In this case, it is assumed that the parent folder already exists in the target system and that the object can be generated in it.

If an object referenced by its Entry Id is not found in the target system, the import raises an error (see 5.2).

If the parent folder of an object does not yet have an Entry Id, it is assumed that the entire folder does not exist in the target system. Consequently the parent folder with all its subfolders (see Hierarchy in 5.4) is exported or newly created in the target system.

It is therefore important that all data structures (folders, link types) provided by Vertec possess an Entry Id in Vertec. This is now assured in Vertec version 6.2 (see chapter 11.1).

Here are two scenarios illustrating how to deal with the Entry Id.

Scenario: inputting a new configuration

In this scenario the customer receives a Config Set to input for a specific configuration.

It is admitted here that the data structures provided in a standard way by Vertec (master data folder, public folders, link types, etc.) exist and can be referenced.

If any such structure is missing in Vertec when a Config Set is imported an error message appears.

In such a scenario, the customer must not take care of the Entry Id.

Scenario: test environment – live system

Another situation occurs when configurations are elaborated on a test system and to be later transferred to the live system. In this case the Entry Id plays a central role.

The problem here is to determine if the new objects should be inserted into an existing structure in the target system or if this structure should be newly created as well.

If for example a new root folder must be generated in the target system, it must not possess an Entry Id in the source system. If it is however later necessary to access this root folder on the target system, possibly to generate new subfolders, it must possess an Entry Id. In such a case, the Entry Id must be set manually in the source and in the target system.

This can be done with Python by specifying `argobject.entryid="XX"` on the corresponding object.

5.6 Special cases

Additional classes

If an additional class must be configured within a Config Set, it must be checked that this class is not yet used within the target system. The Config Set cannot check this at the time of the import; it must already be taken into account when the configuration is created (especially when other configurations, such as scripts, also provided with the Config Set make a reference to this additional class).

To prevent that an additional class which already exists is imported “by accident”, an appropriate OCL requirement can be stated (see 5.4).

Additional fields

If a Config Set creates additional fields, one must ensure that the target data-base does not already have additional fields of same name within the same class. This is not checked at the time of the import: the additional fields will otherwise exist twice.

As is the case with additional classes (see above), this must be taken into account when the Config Set is created.

To prevent that an additional field that already exists is imported by accident, an appropriate OCL requirement can be stated (see 5.4).

5.7 Removing /deleting Config Sets

Config Sets can be removed or deleted. Click with the right mouse button on the corresponding Config Set in the Config Set dialogue (see 5.1):

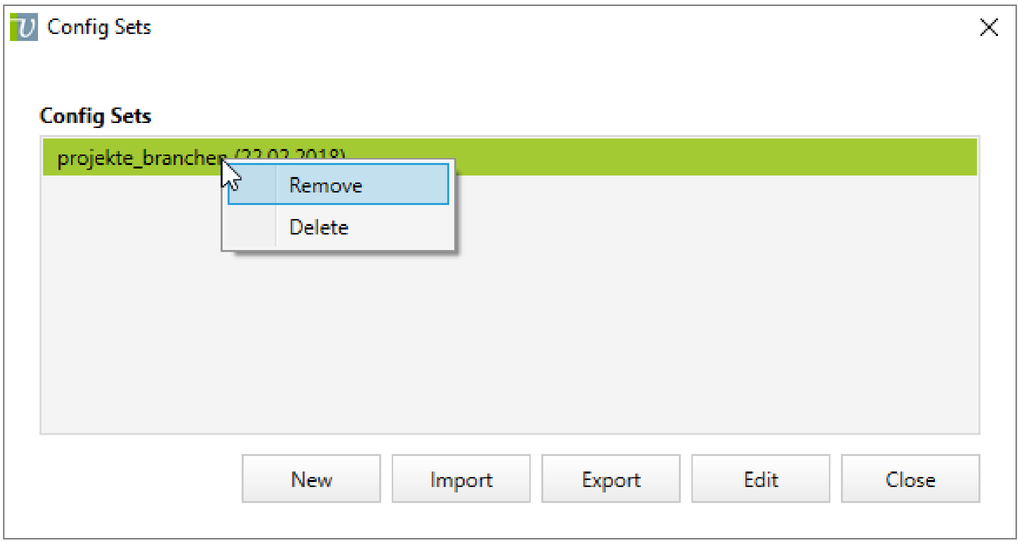


Figure 33: Remove or delete a Config Set

- `Remove`: The Config Set is removed from the list but the objects remain in the system. Removing a Config Set from the list primarily protects it against being deleted.
  The Config Set cannot be imported a second time since the objects would then exist twice. There is no way to add a Config Set to the list without generating the objects.
- `Delete`: The Config Set is deleted. **Careful! All the objects belonging to the set will be deleted in Vertec**. This procedure is used to remove from Vertec all objects generated by the Config Set. If further changes have been made with these objects, the changes are lost as well. To protect a Config Set from being deleted it can be removed from the list. (Remove).

6 Customisation/Parameterisation

6.1 Tags on user entries

Line: Expert| Module: Services & CRM | Version 6.1.0.10

In Vertec, objects (e.g. projects, services) are often characterized through additional Boolean fields. The resulting OCL queries of the type

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

are not efficient because the additional fields must be loaded for every element in the list.

To represent such a requirement in an efficient way, the possibility now exists to add tags to user entries which can be used as filters.

For this reason there is a `Tags` field for every user entry (user entries are all objects in Vertec which can be stored in folders, such as addresses, projects, collaborators, services, etc.) which can be filled via Python with desired values (tags).

Setting a tag
Tags are set with the Python method `addtag(tagname)`, e.g.

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

Spaces in tag names are allowed. If a tag already exists, it is not added a second time.

The maximal total length of all tags set on a user entry is 1000 characters. If this length is exceeded, an error message appears.

Querying tags
Tags can be queried via Python or OCL. In both cases, the query is:

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

Python	OCL
argobject.hastag('mytag')	hasTag('mytag')

The result is `true` if the tag is set, `false` if it doesn't exist.

In this way, lists of the type

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

can be filtered rapidly in an efficient way.

Removing tags
To remove tags use the Python method `removetag(tagname)`, e.g.

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

Nothing happens if a non-existing tag is deleted, no error is raised.

6.2 Key values on user entries

Line: Expert| Module: Services & CRM | Version 6.1.0.10

To efficiently filter long lists on different criteria, the possibility has been added to store customer-specific values on user entries which can be queried via OCL.

For this reason, there is a new field `Keys` on user entries (user entries are all objects in Vertec which can be stored in folders, such as addresses, projects, collaborators, services, etc.) in which (via Python) key-value value pairs can be stored. The following data types are supported:

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

Setting a key value
Key values are set with the Python method `setkeyvlaue(key,value)`, e.g.

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

All the above listed data types are accepted as value. The conversion into the corresponding key-value type is as follows:

Data type	Input	Example
String	String	argobject.setkeyvalue("Mykey", "My text here")
Boolean	"True", "False"	argobject.setkeyvalue("Finished", "True")
Date	ISO Datestring ("2017-02-03")	argobject.setkeyvalue('date', vtcapp.currentdate())
DateTime	ISO Datetime ("2017-02-03T08:30")	argobject.setkeyvalue('date', vtcapp.currentdatetime())
Integer	Integer value	argobject.setkeyvalue("Quantity", 5)
Currency	Floating point number	argobject.setkeyvalue("Amount", 10573.65) setkeyvalue() produces numerical types with the smallest possible number of decimal positions, at most 4. Numbers are rounded off to the 4th decimal position.
None	Empty string	argobject.setkeyvalue("Mykey", None) or argobject.setkeyvalue("Mykey", "")

The maximal total length of all key-value value pairs set on a user entry is 1000 characters. If this length is exceeded, an error message appears.

Querying key values
Python methods as well as OCL operators are available to query key values. Queries are type dependant.

Python

Method	Result
getkeystring(key)	Returns exactly what is stored in the key value
getkeybool(key)	Returns False in the case of an empty string, "False" (case insensitive) or a numerical 0 value, otherwise True
getkeycurr(key)	Returns a float or 0 as float
getkeyint(key)	Returns an integer or 0 as integer
getkeydate(key)	Returns date and time or None

`getkeyint(key)` and `getkeycurr(key)` return numerical values of the other type (integer vs. float) if there is no data loss.

OCL

Operator	Result
keystring(key)	everything -> string or ""
keybool(key)	"", "False", "0", "0.0" -> False, or True
keydate(key)	iso-date (with or without time) -> datetime, otherwise nil
keyint(key)	integer string -> integer, 0
keycurr(key)	currency string -> currency, otherwise 0.00

Empty or invalid values lead to a default value of the corresponding type. The default values are:

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

Deleting a key value

There is no method to delete key values. The removal of such value pairs is done by setting the corresponding key to None or an empty string:

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

or

```
setkeyvalue('mykey', empty string)
```

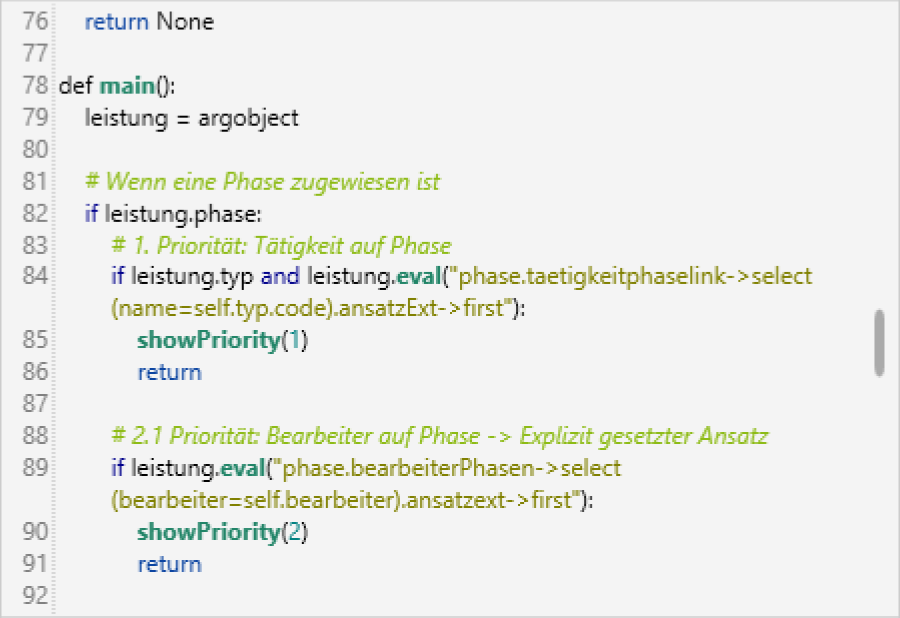
Notice on currency key values

In the Vertec versions 6.1.0.10 to 6.1.0.12, currency key values were erroneously stored in a region-specific way. Currency values stored in this time interval appear as strings when queried, or lead to an error when queried as floats. If this is the case, the values must be set again.

6.3 Highlighting the syntax in code editors

Line: Standard, Expert | Module: Services & CRM | Version 6.2

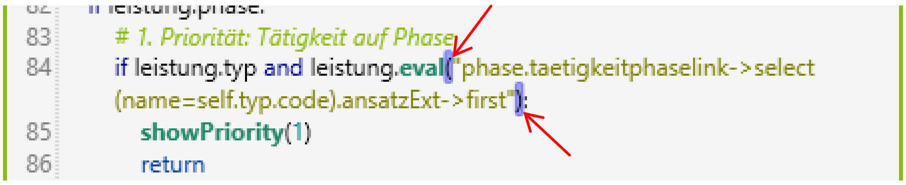
The code editors for OCL, XML, Python and VBScript now feature syntax highlighting in the Desktop App and Cloud App. This makes the code easier to read and clearer.



```
    return None

def main():
    leistung = argobject

    # Wenn eine Phase zugewiesen ist
    if leistung.phase:
        # 1. Priorität: Tätigkeit auf Phase
        if leistung.typ and leistung.eval("phase.taetigkeitphaselink->select
        (name=self.typ.code).ansatzExt->first"):
            showPriority(1)
            return

        # 2.1 Priorität: Bearbeiter auf Phase -> Explizit gesetzter Ansatz
        if leistung.eval("phase.bearbeiterPhasen->select
        (bearbeiter=self.bearbeiter).ansatzext->first"):
            showPriority(2)
            return

```

Figure 34: Python script with syntax highlighting

Except with OCL, lines are now numbered and brackets show beginning and end when clicked:



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

Figure 35: Brackets show beginning and end

6.4 Change of control names on invoices

Line: Expert| Module: Services & CRM | Version 6.2

In the XML code of the Invoice two controls had an incorrect name and have therefore been renamed:

- TotalAfterCosts: In table TotalAfterCostsRow, there is a TextBlock named `TotalAfterCosts`. In fact it holds the total after a discount and is therefore now called `TotalAfterDiscount`.
- FixedPrice: The field `FixedPrice` is now called `FlatExpenses`.

Backwards compatibility

The change is backwards compatible. Even at customers where these elements have been overwritten, the display works as before. The interface is not affected by this change.

In this case, however, it is no longer possible to attribute the overwritten element to the corresponding element within the inserted XML.

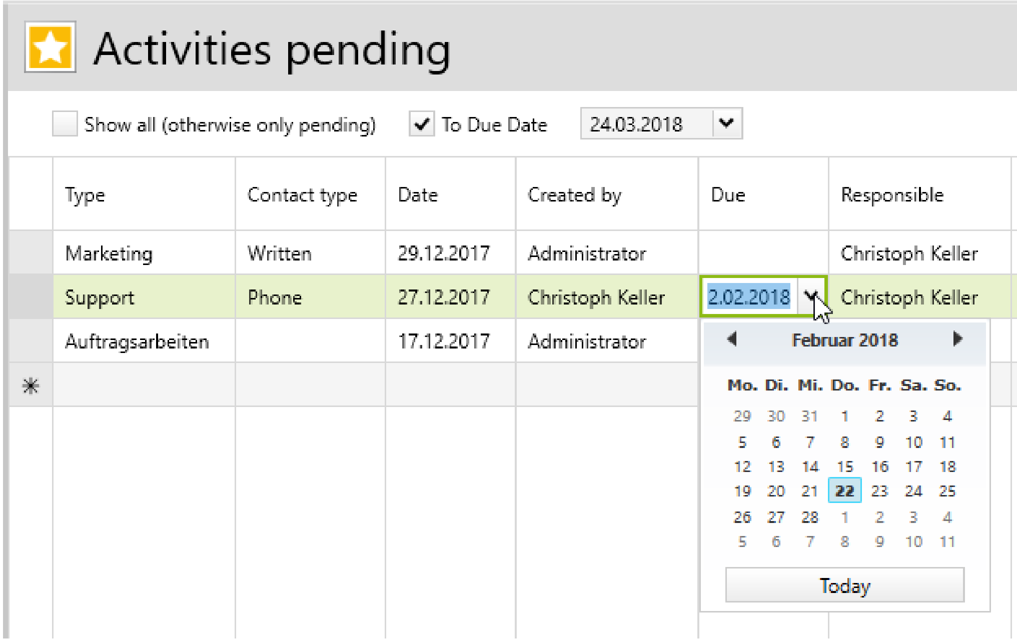
6.5 DatePicker in lists

Line: Expert| Module: Services & CRM | Version 6.2

A DatePicker can now be used in lists. By customizing the Control XML of columns, the user can introduce a DatePicker.

Example: the `Control XML` for the due-date in an activity list:

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



Activities pending

□ Show all (otherwise only pending) ☑ To Due Date 24.03.2018

Type	Contact type	Date	Created by	Due	Responsible
Marketing	Written	29.12.2017	Administrator		Christoph Keller
Support	Phone	27.12.2017	Christoph Keller	2.02.2018	Christoph Keller
Auftragsarbeiten		17.12.2017	Administrator		

Februar 2018

Mo.	Di.	Mi.	Do.	Fr.	Sa.	So.
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	1	2	3	4
5	6	7	8	9	10	11

Figure 36: Activity list with DatePicker

6.6 Support of dates in custom dialogues

Line: Expert| Module: Services & CRM | Version 6.1.0.14

In the past, specifying a date value (`datetime`) as initial value for a date picker has resulted in a serialization error. Date values could therefore only be used with a workaround.

This problem has been solved. As of now, date values can be passed normally:

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

Backwards compatibility

In Vertec versions up to 6.1.0.14, date values were handled differently. Scripts of these versions using date values must be adapted correspondingly:

– Date values could then be passed in the following way:

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

initValues = {}
initValues["FromDate"] = from.isoformat() + "T00:00:00"
initValues["ToDate"] = to.isoformat() + "T00:00:00"
```

Values must now be passed in the following way:

```
initValues["FromDate"] = date.today() #or:
initValues["ToDate"] = date(2017, 1, 1)
```

- The returned value could be transformed into a date as follows:

```
#Example. 01.11.16
from = datetime.strptime(values['FromDate'], '%Y-%m-%dT%H:%M:%S')
#Example. 31.01.17
to = datetime.strptime(values['ToDate'], '%Y-%m-%dT%H:%M:%S')
```

Starting with 6.1.0.14 this will produce an error message:

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

Solution: The returned vale is already of type date and can directly be attributed:

```
from = values['FromDate']
to = values['ToDate']
```

6.7 Custom dialogues: DatePicker with null value

Line: Expert| Module: Services & CRM | Version 6.2.

DatePickers in custom dialogues up to now returned a special date value (1.1.0001), when empty. Now they return `None`.

Backwards compatibility
Scripts which logically have queried this special value in the following way:

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

must be adapted to now query the date with `None`:

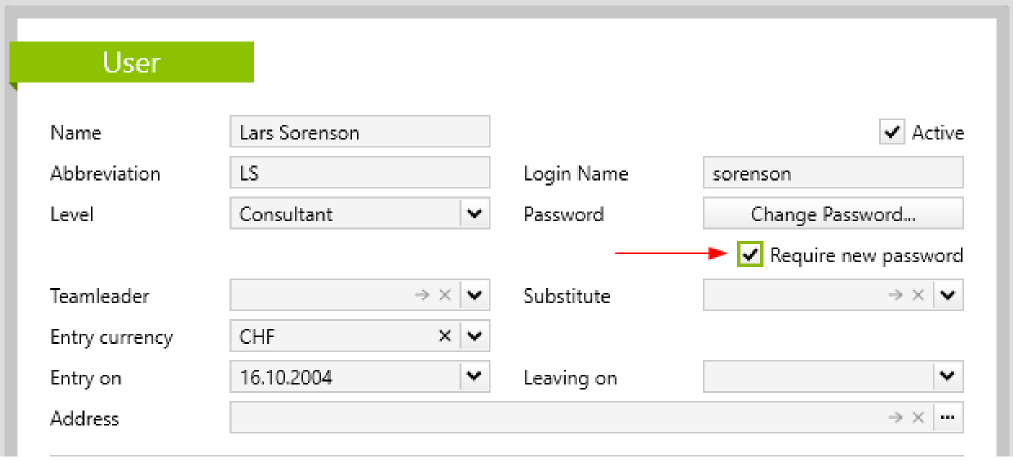
```
if values['Date']:
```

6.8 Requesting a new password

Line: Standard, Expert | Module: Services & CRM | Version 6.1.0.11

The Administrator now has the possibility to force the user to change her/his password at the next login. This option exists only if authentication is not done with LDAP.

For this purpose, there is a new checkbox on the collaborator which is only shown to the Administrator:

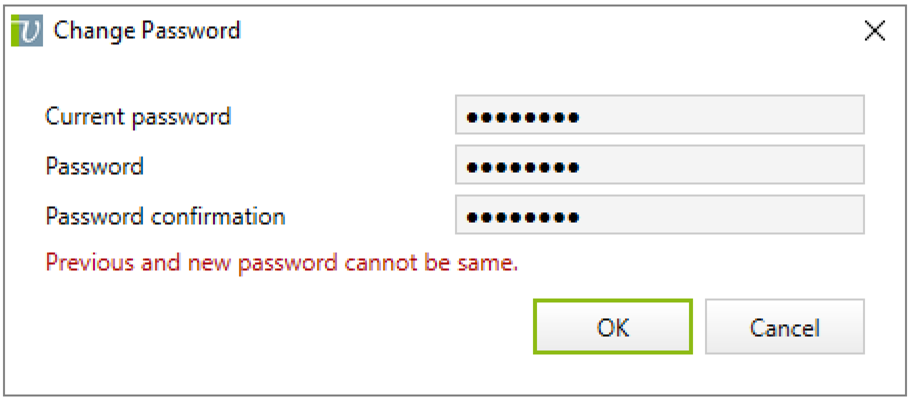


User

Name	Lars Sorenson		☑ Active
Abbreviation	LS	Login Name	sorenson
Level	Consultant	Password	Change Password...
			☑ Require new password
Teamleader		Substitute	
Entry currency	CHF		
Entry on	16.10.2004	Leaving on	
Address			

Figure 37: Setting the checkbox to request a new password

The next time the corresponding collaborator logs into Vertec, the Change password dialogue appears and the collaborator must then input a new password conforming to the existing password policy. It cannot be the same as the old password or the following message appears:



Change Password

Current password ••••••••

Password ••••••••

Password confirmation ••••••••

Previous and new password cannot be same.

Figure 38: A new different password is requested

Password Policy Further information on passwords can be found in the Online Knowledge Base under www.vertec.com/de/kb/verteclogin.

6.9 New OCL operators

Line: Expert| Module: Services & CRM | Version 6.2

The following OCL Operators are now available:

Function	On	Description
datetostrgerman	Datum	Returns the date in German date format as string. In certain places date values in German date format are needed, independently of the regional settings (see for example the from /to arguments in `groupLeistungen` operators in https://www.vertec.com/de/kb/leistungssummen#syntax). `argobject.eval('creationdatetime.datetostrgerman')`
getPresenceEntries(date)	Bearbeiter	Produces a list of presence-time entries on a given date for the employee. `bearb.getPresenceEntries(date)`

6.10 New Python methods

Line: Expert| Module: Services & CRM

The following Python methods have been introduced:

For module "vtcapp"

Function	Version	Description
datetostr(date): string	6.1.0.14	Changes a date into a string and returns it according to country settings. E.g. in Australian English: **>>> mydate = vtcapp.strtodate('13/06/17')** **>>> print mydate** **2017-06-13** **>>> string = vtcapp.datetostr(mydate)** **>>> print string** **13/06/2017**
datetostrgerman(date): string	6.2.	Returns the date in German date format as string. In certain cases it is necessary to get date values in German format, independently of the regional settings (see e.g. the from /to arguments in `groupLeistungen` operators in https://www.vertec.com/de/kb/leistungssummen#syntax). The datetostrgerman(date) method transforms a date to a string and returns it in the German date format. **>>> vtcapp.datetostrgerman(argobject.eval('creationdatetime'))** **13.10.2016**
currentdate(): date	6.1.0.10	Returns today's date. **>>> vtcapp.currentdate()** **2017-11-24**

Function	Version	Description
currentdatetime(): datetime	6.1.0.10	Returns today’s date and time. **>>> vtcapp.currentdatetime()** **2017-11-24 15:01:12.880000**
executeserverwordreport(rootObj, optarg, templatePath, [outputFormat, saveAs, showDialog, showApp])	6.1.0.11	This method is used to produce Vertec generated Word reports without report object. **vtcapp.executeserverwordreport(argobject, None, r"C:\Program Files (x86)\Vertec\Reports\English\Letter.dotx", "", "C:\\Dokumente\\Vertec\\Test", True, True)**
createlist(classname, [list]): list	6.1.0.14	Creates a Vertec list with information on types for the construction of a Vertec list directly on the basis of another (Vertec or Python) list **mylist = vtcapp.createlist('Projekt')** **mylist.append(argobject)** oder **liste = vtcapp.createlist("Projekt", vtcapp.evalocl("projekt ->select(code.sqllike('A%'))"))** **liste2 = liste.evalocl("self->select(aktiv)")**
importconfigset(xmltext: string)	6.2	Imports a Config Set in Vertec and uses it (see chapter 5.2). The Config Set must be a valid XML text transmitted as string.

On individual objects

Function	Version	Description
evalocl(ocl:string)	6.1.0.10	Evaluates an OCL expression on the object. Same function as eval; was introduced to have the same name as the global method. **projectcode = argobject.evalocl("code")**
Aktivitaet.setpfad(pfad:string)	6.1.0.14	Sets the document path on the activity. **argobject.setpfad('C:\\Documents\\text.txt')** Requires writing rights on the activity.
Waehrung.getkursto(currency: object, date: date): currency	6.1.0.14	Returns the currency exchange value at the specified date. **chf.getkursto(eur, vtcapp.currentdate())**
Leistung.updatesatz()	6.1.0.14	Hourly rate is recomputed (on the basis of the rate system). **argobject.updatesatz()** Requires `Project Administrator` or `Super` rights.

On Vertec lists

Function	Version	Description
extend()	6.2	With the extend() method, Vertec lists in Python can attach a Python list (e.g. resulting from an OCL evaluation). Previously, only the append() method offered the possibility to attach individual objects. In this context, it is checked whether the Vertec list is a derived list (e.g. Container.eintraege). In that case, the access is denied. **cominstall = vtcapp.evalocl("projekt->select(code='COMINSTALL')->first")** **comssup = vtcapp.evalocl("projekt->select(code='COM-SSUP')->first")** **cominstall.offeneleistungen.extend(comssup.offeneleistungen)**
idstring()	6.2	A frequent request in Python scripts is to specify with `bold_id in (...)` an SQL query based on a list of Vertec objects. For this purpose, the IDs of a list of objects must be transformed into a string of objects separated by commas. – `List.idstring()` returns the IDs of the Vertec objects as a string of values separated by commas. – If the list includes non-persistent objects (object ID is not storable) the method returns an error message. **vtcapp.evalocl('Projekt->allinstances').idstring()**

Changes have been made to the following methods:

Function	Version	Change
vtcapp.evaltovariable()	6.1.0.14	"self" may no longer be used a variable name since it is a key value. A corresponding error message appears.
vtcapp.executereport(rootObj, optarg, berichtObj, [saveAs, showDialog, doPrint, showApp])	6.1.0.14	The parameter `showDialog` now also allows controlling the display of messages such as "overwrite existing file" and "create new path" by means of this setting. If the setting is not specified, the setting from the report object (`berichtObj`) is used.

Detailed information on the various functions is found in the article Python Interfaces in the Online Knowledge Base under www.vertec.com/de/kb/pythoninterfaces.

6.11 Python method vtcapp.sendfile()

Line: Expert| Module: Services & CRM | Version 6.2

This new Python method allows sending a file or a string (as file) to a client. Syntax:

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

Parameter	
file: string	A path to a file, or a string, can be specified here. For this reason, this also works on cloud clients without file access system.
filename: string	Name of the file if stored.
showsavedialog : boolean	Optional. The transmitted file shall, in accordance with the request and the possibilities of the client, be stored or immediately be executed (shell-execute). – Web App: download only – Desktop and Cloud App: dialogue and execution are possible. In Desktop App and Cloud App, the showsavedialog argument, if set, indicates that a save dialogue is displayed. By default, the desktop is displayed as storage destination. If `showsavedialog=false`, the file is stored in the Temp folder of the client, an existing file stored there will be overwritten. This makes sense with `openfile=true`, the file is then stored in Temp and immediately opened. Default is `false`.
openfile: boolean	Optional. If `true`, the file is opened after being stored. Default is `true`.

With the Web App client, the file always appears as download in the web browser. The arguments showsavedialog and openfile have no meaning.

Note The combination `showsavedialog=false` and `openfile=false` doesn't make sense as the file is then stored in the Temp folder and nothing more happens.

The method also works in restrict scripting mode.

Examples

Example of a simple project export:

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

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

Example showing how an image existing locally can be sent:

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

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

Files should be explicitly closed after use. Otherwise they will be closed by the garbage collector or, in case of an error, only when the next exception occurs. It is recommended to use `open` with `with` as in the above example.

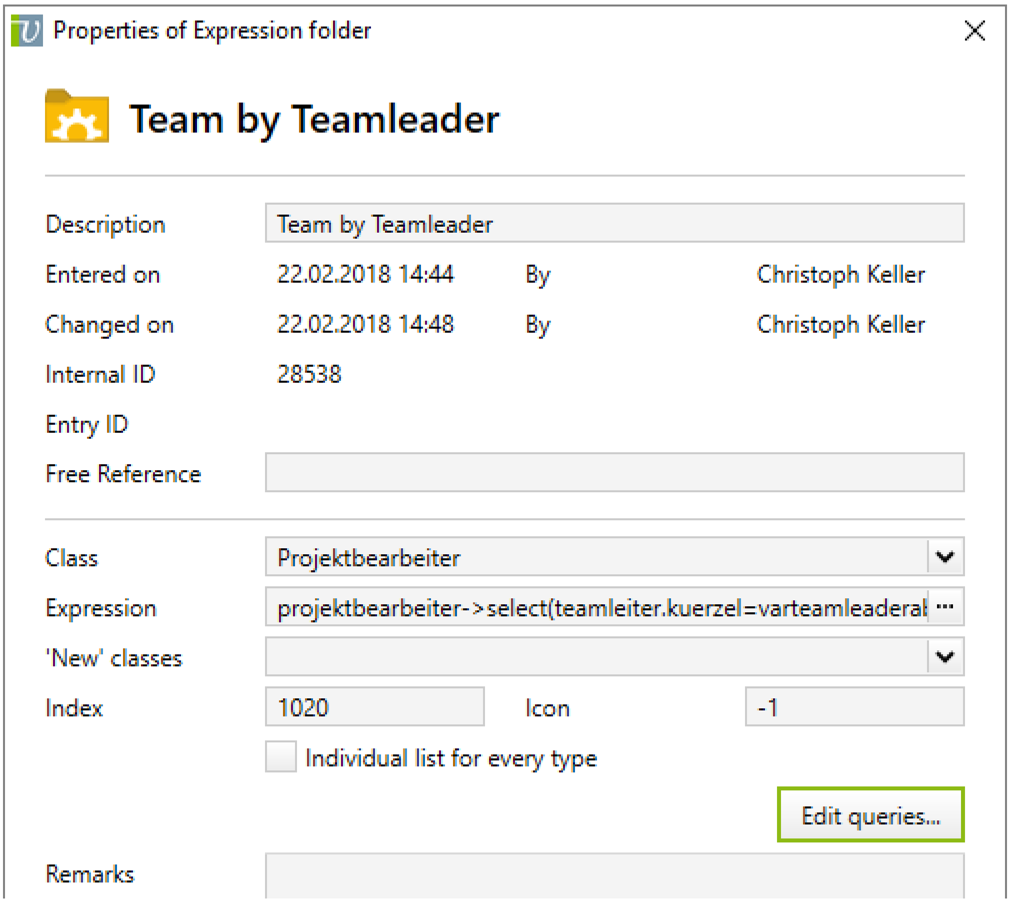
6.12 Search dialogues for expression folders

Line: Expert| Module: Services & CRM | Version 6.2

In the present version, the handling of expression folders has been adapted to the handling of SQL folders. Thus, queries can now also be defined for expression folders and input parameters used in the folder and column expressions.

As before, an expression can be input without defining a query. In this case, the expression folder performs as usual und existing expression folders perform as before.

If however one wishes to pass search parameters to the expression to be used by the user for querying, queries can now be defined for this purpose.



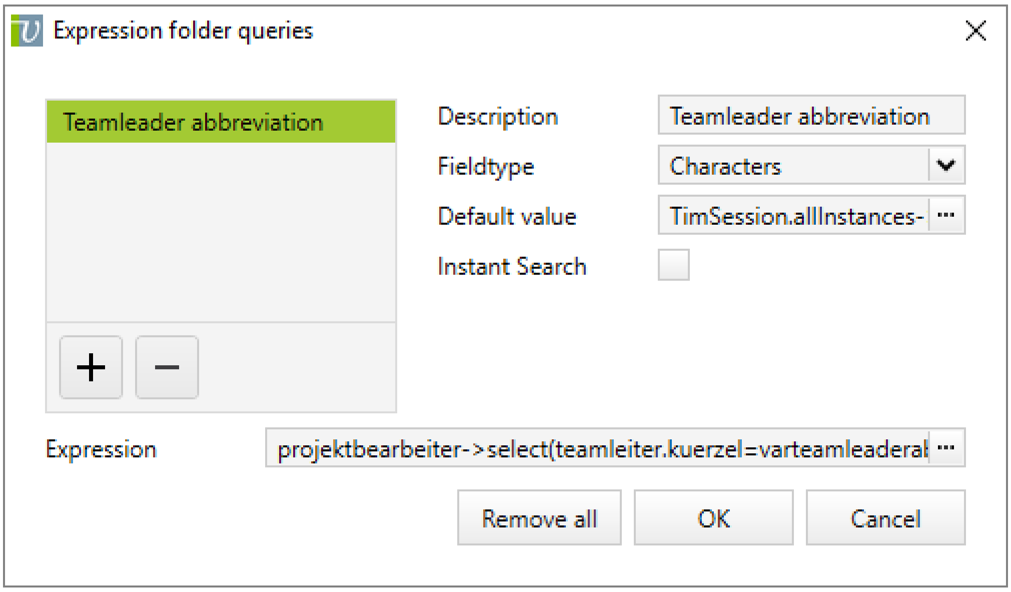
Properties of Expression folder

Team by Teamleader

Description	Team by Teamleader		
Entered on	22.02.2018 14:44	By	Christoph Keller
Changed on	22.02.2018 14:48	By	Christoph Keller
Internal ID	28538		
Entry ID			
Free Reference			
Class	Projektbearbeiter		
Expression	projektbearbeiter->select(teamleiter.kuerzel=varteamleaderal		
'New' classes			
Index	1020	Icon	-1
	□ Individual list for every type		
Remarks			

Figure 39: Property dialogue of an expression folder

The button `Edit queries…` opens the dialogue for inputting the query.



Expression folder queries

Teamleader abbreviation

Description: Teamleader abbreviation

Fieldtype: Characters

Default value: TimSession.allInstances-

Instant Search: □

Expression: projektbearbeiter->select(teamleiter.kuerzel=varteamleaderat

Figure 40: The query dialogue

As with SQL folders, the field designations are used as variable names prefixed with `var`. In the example above, the field produces a variable `varteamleaderabbreviation,` which can be used in the OCL expression as well as in the column expressions of the parameters of the list:

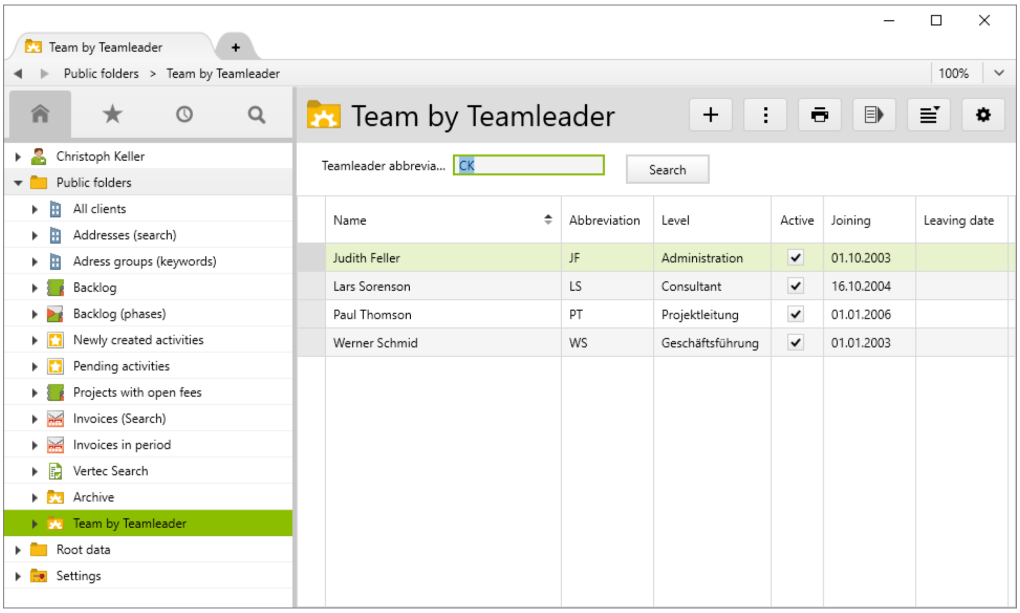
```
projektbearbeiter->select(
teamleiter.kuerzel=varteamleaderabbreviation)
```

The OCL expression editor knows these variable names and declares the expression valid.

As with SQL folders, several query fields can be defined. They are created with `+` in the query dialogue.

Individual query fields can be removed with the `–` button.

With a click on `OK` in the query dialogue, the query is created and the corresponding field is displayed on the interface:



Teamleader abbrevia... CK

Search

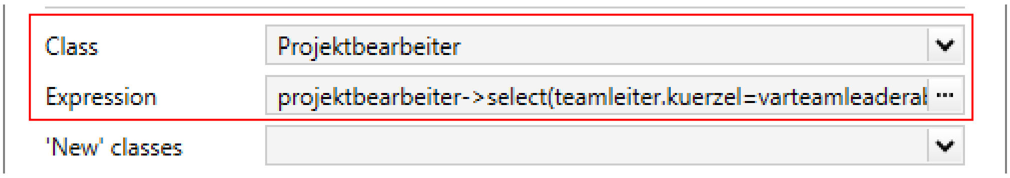
Name	Abbreviation	Level	Active	Joining	Leaving date
Judith Feller	JF	Administration	☑	01.10.2003	
Lars Sorenson	LS	Consultant	☑	16.10.2004	
Paul Thomson	PT	Projektleitung	☑	01.01.2006	
Werner Schmid	WS	Geschäftsführung	☑	01.01.2003	

Figure 41: Expression folder with query

Instant search The folder search can also be executed immediately by checking `Instant search` in the query dialogue. See the separate description in chapter 6.13.

The complete query can be erased with a click on the button `Remove all` in the query dialogue. The expression folder then behaves like a “normal” expression folder.

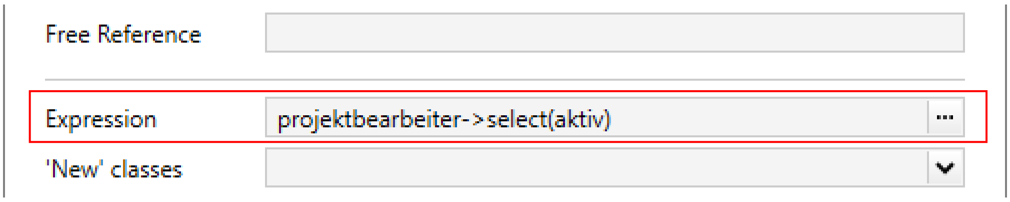
Expression folders with query are displayed in the property dialogue with the name of the class:



Class	Projektbearbeiter
Expression	projektbearbeiter->select(teamleiter.kuerzel=varteamleaderal
'New' classes	

Figure 42: An expression folder with query shows the class name

For expression folders without query this field does not appear, not being necessary since the class results from the expression.



Free Reference	
Expression	projektbearbeiter->select(aktiv)
'New' classes	

Figure 43: Expression folder without query

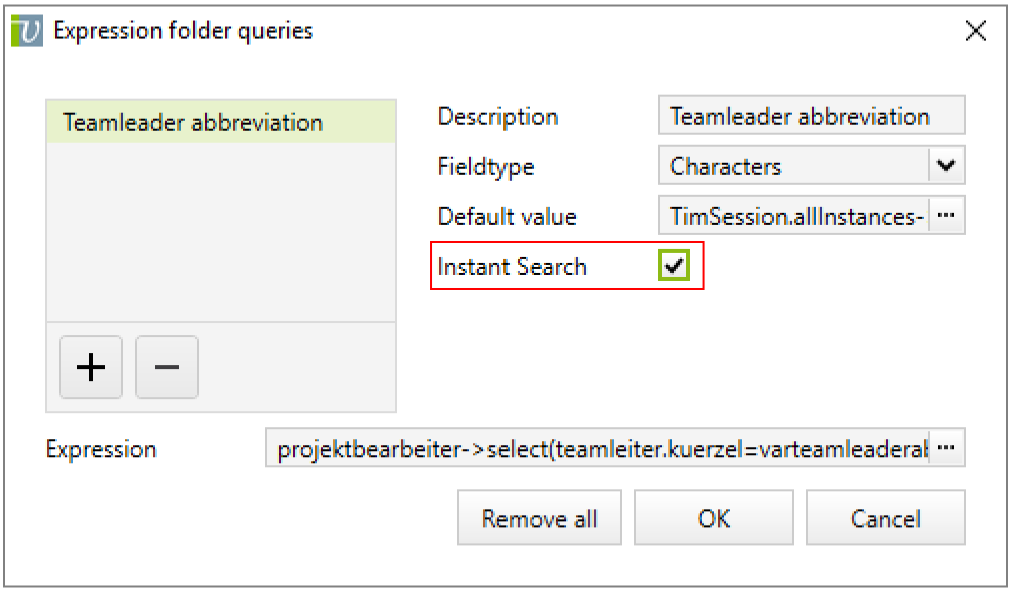
6.13 Execute a folder search immediately

Line: Standard & Expert | Module: Services & CRM | Version 6.2

SQL or expression folders with queries (see 6.12) can now be directly executed with standard search parameters, without having to click on `Search`.

This is useful when the search parameters already hold meaningful values (e.g. the logged-in user) und are ready to be executed.

For this purpose there is a new checkbox Instant search:



Description: Teamleader abbreviation

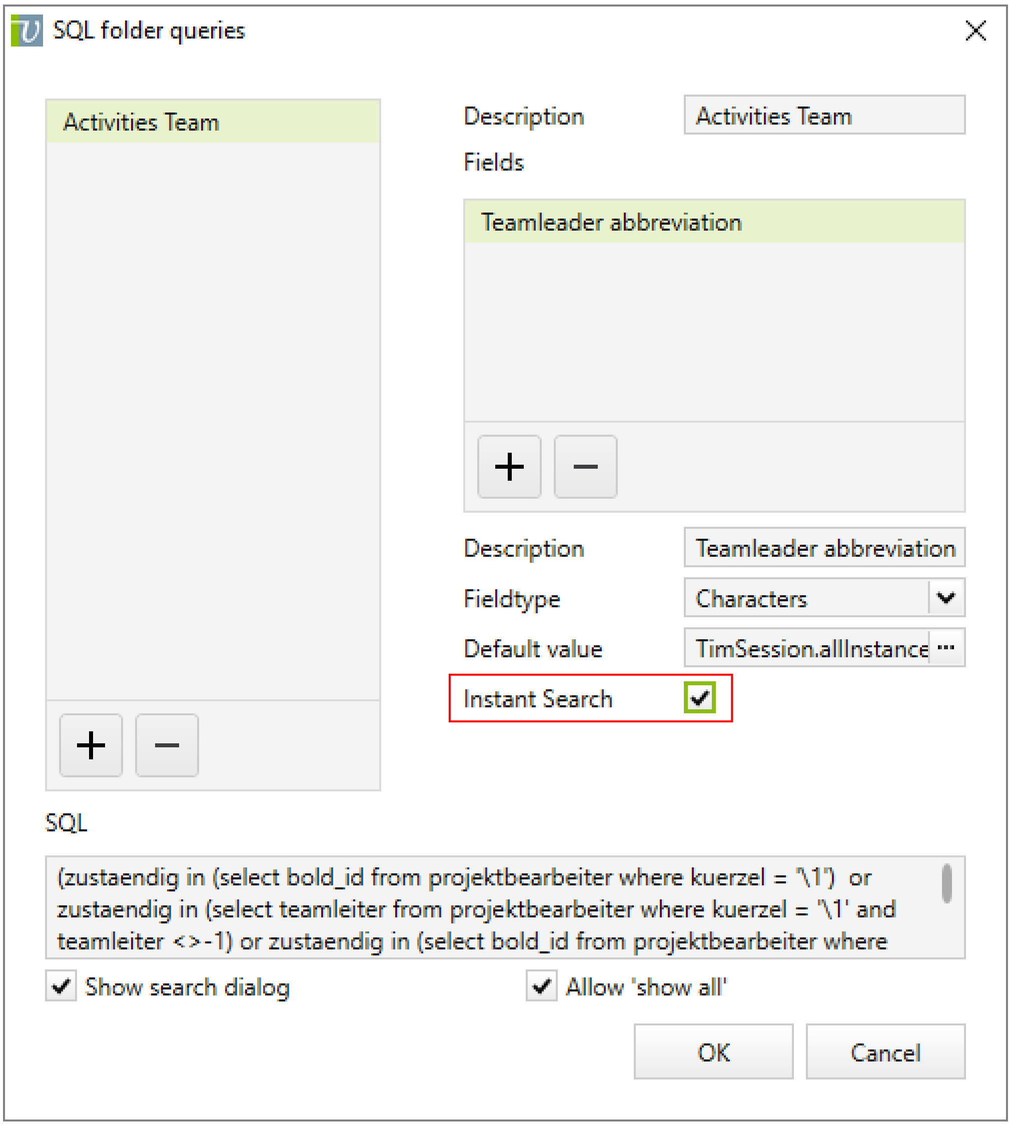
Fieldtype: Characters

Default value: TimSession.allInstances-

Instant Search ☑

Expression: projektbearbeiter->select(teamleiter.kuerzel=varteamleaderal

Figure 44: Expression folder query with instant search



SQL folder queries

Activities Team

Description: Activities Team

Fields

Teamleader abbreviation

Description: Teamleader abbreviation

Fieldtype: Characters

Default value: TimSession.allInstance

Instant Search ☑

SQL

```
(zustaendig in (select bold_id from projektbearbeiter where kuerzel = '\1')  or
zustaendig in (select teamleiter from projektbearbeiter where kuerzel = '\1' and
teamleiter <>-1) or zustaendig in (select bold_id from projektbearbeiter where
```

☑ Show search dialog ☑ Allow 'show all'

Figure 45: SQL folder with instant search

Thus, the search is immediately executed and the results displayed in the list.

Further searches can as usual be started by changing the search parameters and clicking on the search button.

6.14 Authorisation for SQL queries

Line: Standard & Expert | Module: Services & CRM | Version 6.2

SQL queries now require administrator rights or a special SQL right called SQL Query introduced with the present version. If the user does not have such a right, a SQL query call will lead to an error message.

SQL queries exist in the following areas:

Scripts (Python und VBScript)
In scripts, this concerns the use of the `GetWithSQL` and `GetMemberWithSQL` script methods. If these methods are called, the calling user must possess administrator rights or the SQL Query right.

With scripts, there also exists the variant of allocating extended rights. They give the user executing the query administrator rights for the range between `beginSystemContext` and `EndSystemContext`.

Extended rights More information on Extended rights in scripts can be found in the Online Knowledge Base under www.vertec.com/de/kb/erweiterte-berechtigungen.

ActiveX/COM

In calls via COM (Word, Excel) this concerns the methods `GetWithSQL` and `Member-WithSql`.

The user executing the query must possess administrator rights or the SQL Query right.

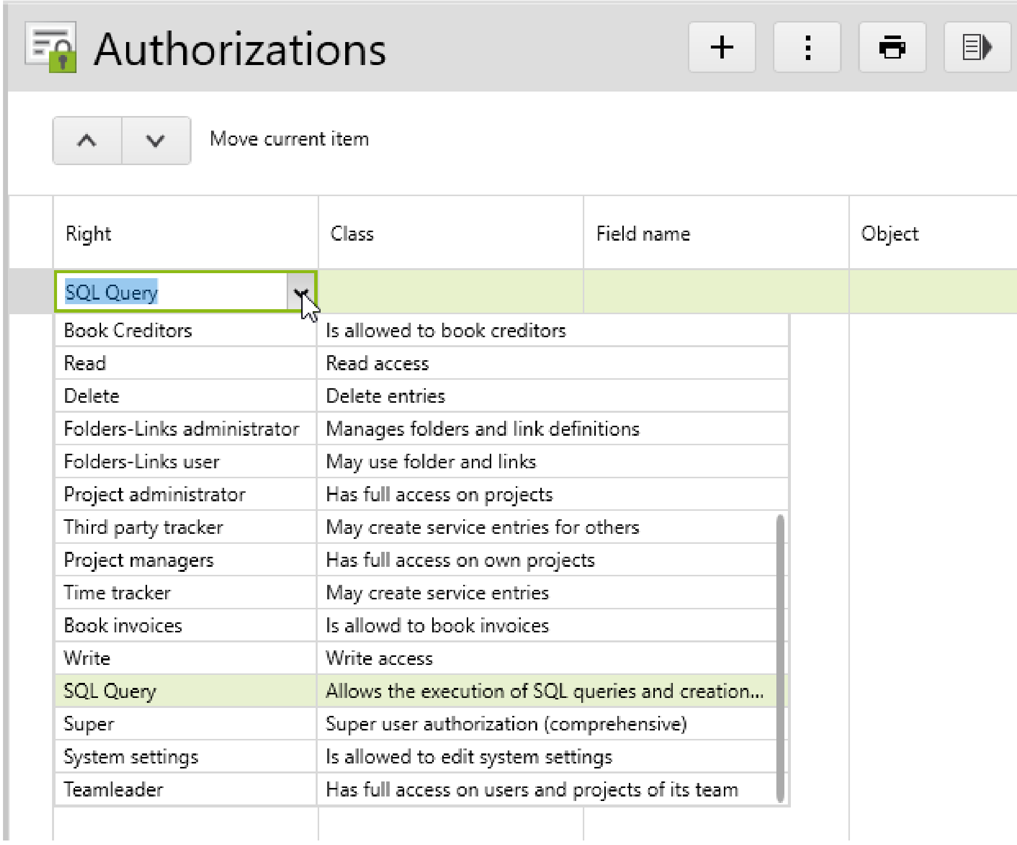
XML

With XML, the corresponding method is `sqlwhere`.

The user executing the query must possess administrator rights or the SQL Query right.

SQL Query right

There is a new right called SQL Query:



Authorizations

Move current item

Right	Class	Field name	Object
SQL Query			

Book Creditors	Is allowed to book creditors
Read	Read access
Delete	Delete entries
Folders-Links administrator	Manages folders and link definitions
Folders-Links user	May use folder and links
Project administrator	Has full access on projects
Third party tracker	May create service entries for others
Project managers	Has full access on own projects
Time tracker	May create service entries
Book invoices	Is allowd to book invoices
Write	Write access
SQL Query	Allows the execution of SQL queries and creation...
Super	Super user authorization (comprehensive)
System settings	Is allowed to edit system settings
Teamleader	Has full access on users and projects of its team

Figure 46: SQL Query right

We recommend granting the SQL Query right very restrictively. The right gives users a wide SQL access to the data.

OCL hasright The OCL hasright right name is called `sqlquery`. See in this respect the section on user rights in the OCL article under www.vertec.com/de/kb/ocl#user-rechte.

Backwards compatibility

Even existing data-bases now require administrator rights or a special SQL right for SQL queries. It is therefore possible that error messages appear after an update of scripts, Excel reports or XML calls.

In the case of scripts, we recommend to adapt them in such a way that the call occurs within `beginSystemContext`/`endSystemContext` and to grant the script extended rights.

For calls via COM and XML, the users must be granted the SQL Query right. We recommend creating a special user group for this purpose.

6.15 Authorisation check for personal folders modified

Line: Expert| Module: Services & CRM | Version: 6.2

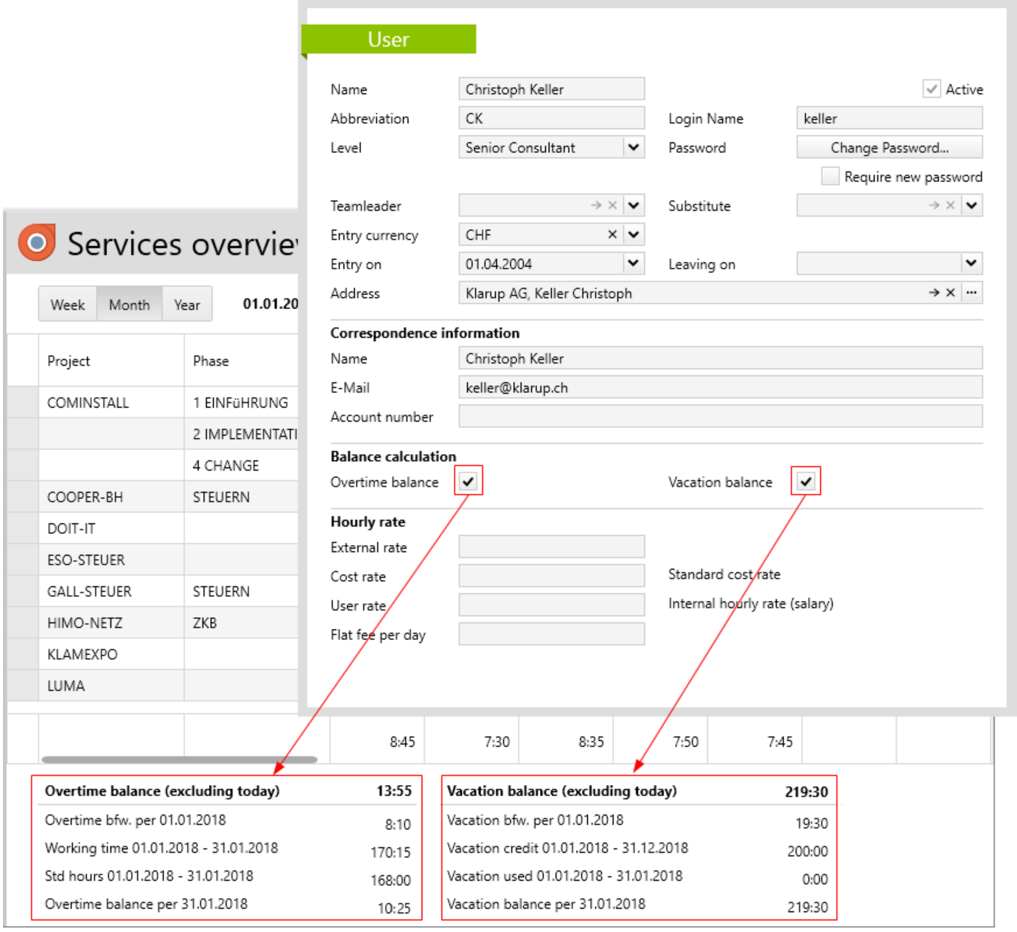
The authorisation check for personal folders has been modified. Should you work with personal folders (in that case you have a `Personal folders` root folder), you may notice a change, for instance that you cannot execute certain computations. In this case, you'll have to modify the authorisations accordingly.

7 Services/CRM

7.1 Representation of account balances in Services Overview

Line: Expert| Module: Services & CRM | Version: 6.2

The display of overtime and vacations balances in Services overview now conforms to the options chosen for computing user balances.



User

Name	Christoph Keller		☑ Active
Abbreviation	CK	Login Name	keller
Level	Senior Consultant	Password	Change Password...
			□ Require new password
Teamleader		Substitute	
Entry currency	CHF		
Entry on	01.04.2004	Leaving on	
Address	Klarup AG, Keller Christoph		

Correspondence information

Name	Christoph Keller
E-Mail	keller@klarup.ch
Account number	

Balance calculation

Overtime balance ☑ Vacation balance ☑

Hourly rate

External rate		
Cost rate		Standard cost rate
User rate		Internal hourly rate (salary)
Flat fee per day		

Services overvie

Week Month Year 01.01.20

Project	Phase
COMINSTALL	1 EINFüHRUNG
	2 IMPLEMENTATI
	4 CHANGE
COOPER-BH	STEUERN
DOIT-IT	
ESO-STEUER	
GALL-STEUER	STEUERN
HIMO-NETZ	ZKB
KLAMEXPO	
LUMA	

8:45 7:30 8:35 7:50 7:45

Overtime balance (excluding today)	13:55
Overtime bfw. per 01.01.2018	8:10
Working time 01.01.2018 - 31.01.2018	170:15
Std hours 01.01.2018 - 31.01.2018	168:00
Overtime balance per 31.01.2018	10:25

Vacation balance (excluding today)	219:30
Vacation bfw. per 01.01.2018	19:30
Vacation credit 01.01.2018 - 31.12.2018	200:00
Vacation used 01.01.2018 - 31.01.2018	0:00
Vacation balance per 31.01.2018	219:30

Figure 47: Controlling the display with the options for computing balances

If, for a collaborator, one or both options are deactivated, the corresponding values on Services overview are automatically hidden.

7.2 Activities folder for offers

Line: Expert| Module: Budget & Subproject | Version 6.2

There is a new link type for activities on offers. It is called Offer – Activities and is available in a standard way.

8 Invoicing

8.1 Support of pay-in slips with QR code

Line: Expert| Module: Services & CRM | Version: 6.2

There is a new standard for pay-in slips in Switzerland. The existing OCR-B ESR code is replaced by a QR code with the relevant invoice data.

QR-invoices More information on the subject in

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

For this purpose, there is now a derived member on the invoice called `qrcode` which generates the QR code as binary graphic (PNG). This member can be inserted as an image in the Word report.

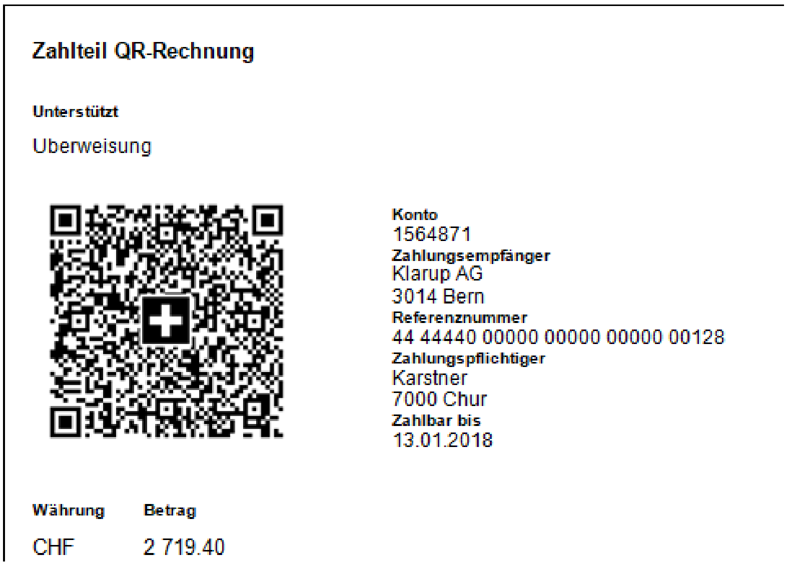
– The content of the QR code is a text encoded in ANSI Latin-1.
– The individual data fields are separated by CR/LF.
– The meaning of the fields is given by their position (order).

Supplied invoice reports

The standard invoice templates supplied by Vertec with pay-in slips are modified in the following way:

– ESR and QR codes are both output on a separate page.
– The QR code is output when an IBAN number is indicated on the payment type (see 8.2).
– As before, a participant number is required for the output of the ESR.

If, for any reason, an error occurs when the QR code is computed, an error message is output in place of the QR code. The corresponding attribute is called `qrcodeerror`.



Zahlteil QR-Rechnung

Unterstützt
Uberweisung

Konto
1564871
Zahlungsempfänger
Klarup AG
3014 Bern
Referenznummer
44 44440 00000 00000 00000 00128
Zahlungspflichtiger
Karstner
7000 Chur
Zahlbar bis
13.01.2018

Währung	**Betrag**
CHF	2 719.40

Figure 48: Pay-in slip with QR code

8.2 Support of IBAN number on payment type

Line: Expert| Module: Services & CRM | Version: 6.2

To support pay-in slips with QR-Code (see 8.1) a new field IBAN has been introduced on the payment type.

In addition, there is a new setting `Standard payment type` in System settings > Invoice.

Project types and projects inherit this value which can however be overridden on both levels.

Backwards compatibility
During an update, a new standard payment type is created and the existing system settings for the ESR details are taken over for it. This payment type is set in the system settings as standard payment type.

The following settings are taken over and the following system settings are removed:

- BESR-Bank
- BESR-Bank customer identification number
- Type of blue inpayment slip
- Blue inpayment slip reference number length
- Blue inpayment slip Participant number
- Use blue inpayment slip or blue bank inpayment slip

Hence, from now on, ESR information from the payment type is used, no longer from system settings. This however has no effect on operations.

Existing payment types remain unchanged and are used where they set (project type, project).

If you wish to set an existing payment type as standard, specify it as standard in System settings > Invoice.

More information The system setting `BESR-Bank` previously was a reference to a Vertec address. It is now a text field on the standard payment type. This is without consequence as the text alone was used on the report. The bank address entry is no longer used in the ESR procedure.

9 Reporting

9.1 New implementation of Vertec-generated reports

Line: Standard, Expert | Module: Services & CRM | Version 6.2

The introduction of extended Office reports (see chapter 4) has led to a totally new implementation of Vertec-generated Word reports.

This should not be noticeable to the user, but some differences will occur:

Locale bands
Previously, when using a locale band, all numerical and date values inside of the band formatted according to Visual Basic (VBA) formatting rules (e.g. `\#,##0.00`) were formatted according to the country setting indicated in the locale expression.

This system was extended in the Vertec-generated Word-Reports in such a way that ALL numerical and date values within the band are now automatically formatted according to the indicated locale expression. It is therefore no longer necessary to provide extra VBA formatting indications for numerical values which should be formatted according to other country representations (but existing VBA formattings are accepted).

Differences in Office- and Vertec-generated reports
Differences in representations of Office- and Vertec-generated reports usually have to do with the handling of empty paragraphs.

Depending on how the bookmark of a band is defined, the scope of a band is interpreted slightly differently in Office- and Vertec-generated reports.

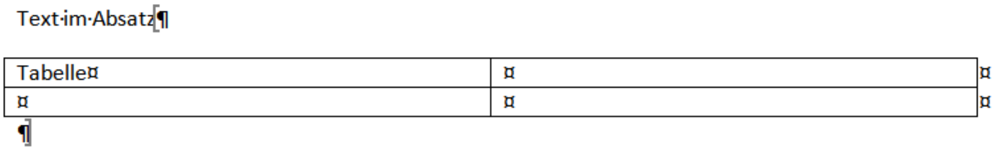
There are two general types of differences:

Bands crossing document hierarchy levels
Contrarily to the VBA-based report generator of Office-generated reports, the server-based reporting system for Vertec-generated reports only supports band definitions for which beginning and end lie on the same level. This implies:

Band beginning	Band end	Vertec generated reports
Text (normal paragraph)	Text (normal paragraph)	OK
Text (normal paragraph)	Text (within a table)	not OK
Text (normal paragraph)	End of table line	not OK
Beginning of table line	End of table line	OK
Beginning of table line	Text in table line, not at the end	not OK

In order for these conditions to be met as easily as possible, the beginning and end of a band are interpreted as generously as possible.

If e.g. a band begins at the end of a paragraph with text, but only includes the paragraph symbol, this is in principle not valid. To nevertheless allow the report to execute properly, the beginning of the band is advanced to the beginning of the next element (paragraph or table), in cases where the end of the band lies at the end of a table line or of a paragraph.



Text·im·Absatz¶

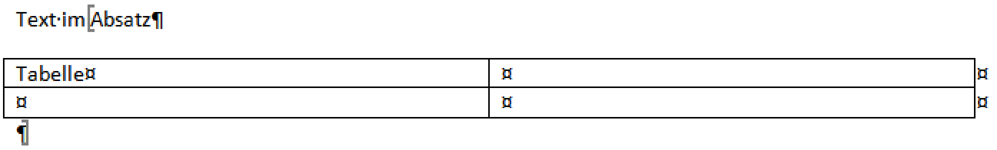
Tabelle¤	¤	¤
¤	¤	¤

¶

This automatic adaptation however results in one less paragraph being represented, which is of course visible in the output.

Such bands should, if possible, not be defined. Band definitions should always correspond to complete paragraphs or their beginning and end should lie within a text.

If the beginning of a band lies in the middle of the text of the paragraph, as shown in the figure below, an error message appears: Invalid band definition.



Text·im·Absatz¶

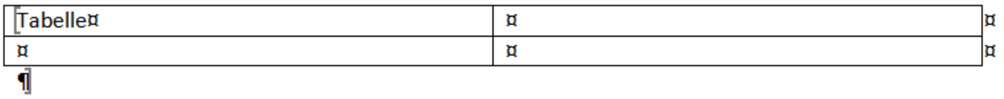
Tabelle¤	¤	¤
¤	¤	¤

¶

Indistinguishable variants

There are cases of band definitions which cannot be distinguished in the server-based code, even though they can be visually distinguished in Word.

This is for instance the case when a text mark occurs at the end of an empty paragraph following a table:



Tabelle¤	¤	¤
¤	¤	¤

¶

In this case, the bookmark's end lies within the empty paragraph. If however the band ends at the end of the table and does not include the empty paragraph, the bookmark's end nevertheless lies in the following paragraph.

Bookmark ends in an otherwise empty paragraph are always interpreted as lying at the end of the preceding table.

If an empty paragraph following a table must explicitly be included in the band, this can be achieved by inserting any text (e.g. a space) within the paragraph. The paragraph will then be considered as part of the band.

Backwards compatibility

If you are already using Vertec generated reports or wish to have Office-generated reports to be newly generated by Vertec, check these reports for the points listed above and adapt them if necessary.

9.2 Office templates changed to new Office formats

Line: Standard, Expert | Module: Services & CRM | Version 6.2

Vertec no longer provides Office report templates with formats older than Office 2005 (xlt, dot):

- Excel report templates in `.xlt` format are no longer provided.
- All Word templates have been changed from `.dot` to `.dotx`.

- The macro-file VertecReport.dot is replaced by VertecReport.dotm. If VertecReport.dotm is not present in the system, reports cannot be executed.

Backwards compatibility

The `.dot`- and `.xlt` files of existing installations are not removed.

If old and new versions of the same files are present in the same folder, the newer ones have priority if the report is registered in Vertec without ending (standard).

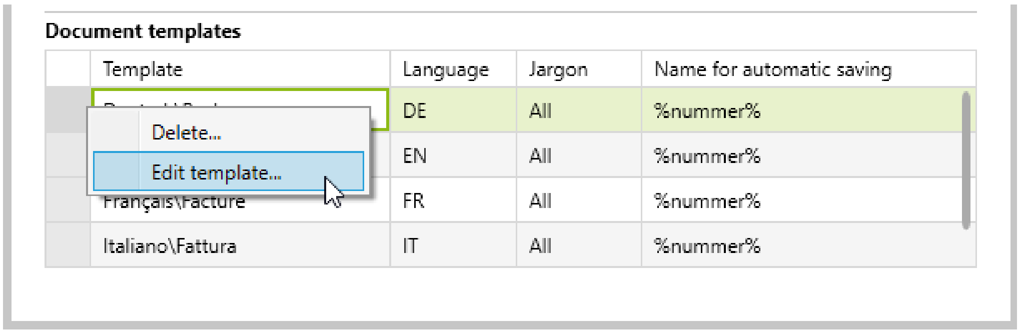
This implies: if your Vertec reports are registered with an ending, the old versions are used. If you wish to use the new Vertec templates, remove the endings from the registrations.

9.3 Create/modify templates in cloud clients

Line: Standard, Expert | Module: Services & CRM | Version 6.1.0.7

To manage document templates with cloud clients a new upload/download mechanism is available. It works as follows:

Click with the right mouse button next to the template you wish to work on and select `Edit template...`:

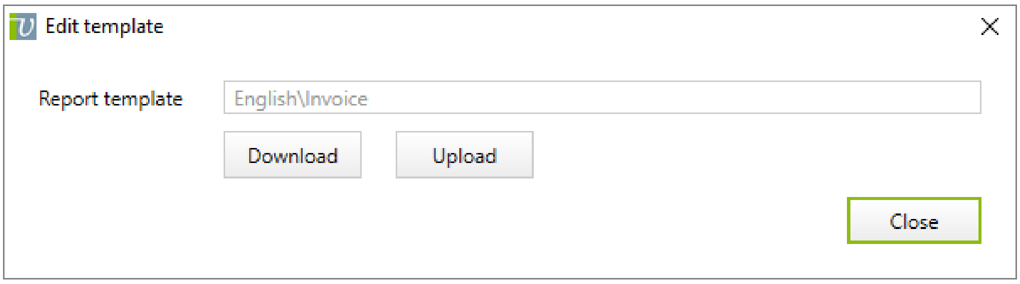


Document templates

Template	Language	Jargon	Name for automatic saving
	DE	All	%nummer%
	EN	All	%nummer%
Français\Facture	FR	All	%nummer%
Italiano\Fattura	IT	All	%nummer%

Figure 49: Select Edit template

Or double-click on the corresponding line. The upload/download dialogue appears:



Edit template

Report template: English\Invoice

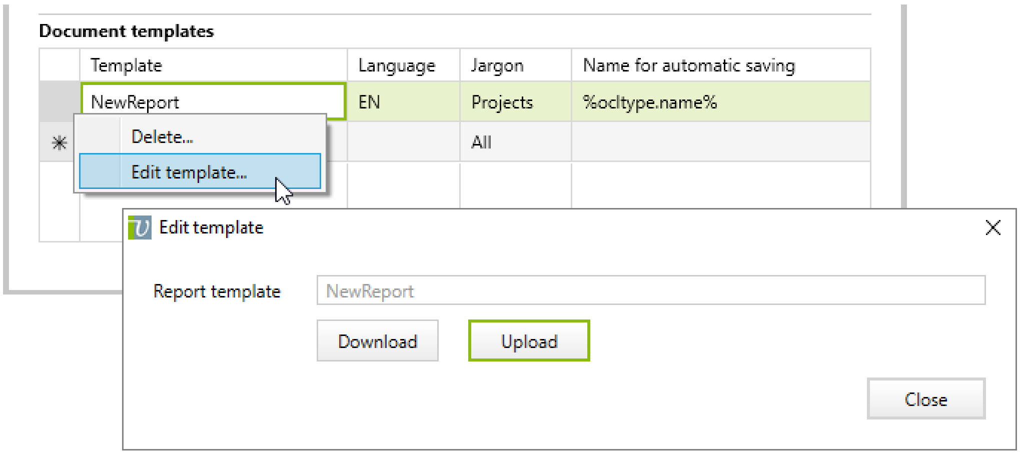
Figure 50: Upload/download dialogue for report templates

To adapt an existing report

- **Download**: To work on an existing report, click on `Download`. The report is downloaded and you are asked where to store it locally (in Web App the report is stored in the Temp directory and immediately opened for modification).
- **Upload**: To upload the report to the server after modification, click on `Upload`. In the Explorer, select the report you have modified. It will automatically be stored at the path location indicated for Word report templates.

Record a new report

To record a new report, input the desired name of the report in the Template field and click on `Edit template…` > `Upload`.



Document templates

	Template	Language	Jargon	Name for automatic saving
	NewReport	EN	Projects	%ocltype.name%
*			All	

Edit template

Report template: NewReport

Figure 51: Creating a new report

Choose the desired name and the report will be stored with this name at the location of the path specified for Word report templates.

Additional information: the size for uploads is limited to 1 Mb. This restriction is checked on the client and on the server side.

Download of report templates at execution time (Cloud App)

Starting with version 6.1.0.7, client-side Office reports are possible with Cloud App even when there is no access to the file system on the server. When an Office report is produced, the corresponding template is downloaded to the local Temp directory for this purpose and the report is generated with this template. Excel reports as well as non-Vertec generated reports newly work in this way.

- The templates must be stored at the path defined for templates. Files stored elsewhere cannot be downloaded.
- If a template already exists in the Temp directory of the client it will be overwritten. If this is not possible, a number is added to the file name.

9.4 Excel Export: Selection fields are not exported

Line: Standard, Expert | Module: Services & CRM | Version 6.1.0.14

Selection fields in lists were not correctly represented in Excel. The class name appeared instead of the value, as for instance:

`Vertec.ServerControls.AdditionalFields.AdditionalFieldIntValue`

This problem has been corrected.

9.5 Locale-bands in Word report now also accepts OCL expressions

Line: Standard, Expert | Module: Services & CRM | Version 6.2.

With so-called locale-bands, it is possible in Word reports to specify the regional settings to be used within the band for the formatting of numbers and dates.

Up to now a country code had to be specified here. It is now possible to specify an OCL expression which returns a string with a country code. The following variants will work:

– A locale-name as an expression: `de-CH`
– An OCL expression which produces a locale-name as string: `'de-CH'`

Additional Information Additional Information on locale-bands is found in the Online Knowledge Base under https://www.vertec.com/de/kb/wordreports#locale.

---

10 Interfaces

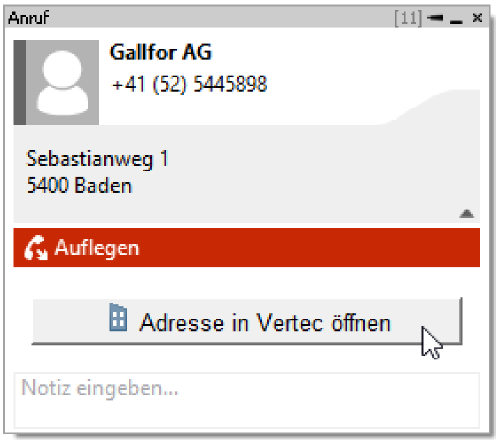
10.1 ProCall Addin

Line: Standard, Expert | Module: Services & CRM | Version 6.1.0.11

Vertec's ProCall Addin provides telephony integration in Vertec through Estos ProCall (https://www.estos.de/). The addin works with ProCall versions 5.1.104.41061 or later.

The telephony journal

During incoming or outgoing calls, ProCall displays a telephony window (popup). If the Vertec ProCall Addin is installed, a button `Adresse in Vertec öffnen` appears. This makes it for instance possible during an incoming call to directly open in Vertec the corresponding address with customer history and to have all necessary data available on a click.



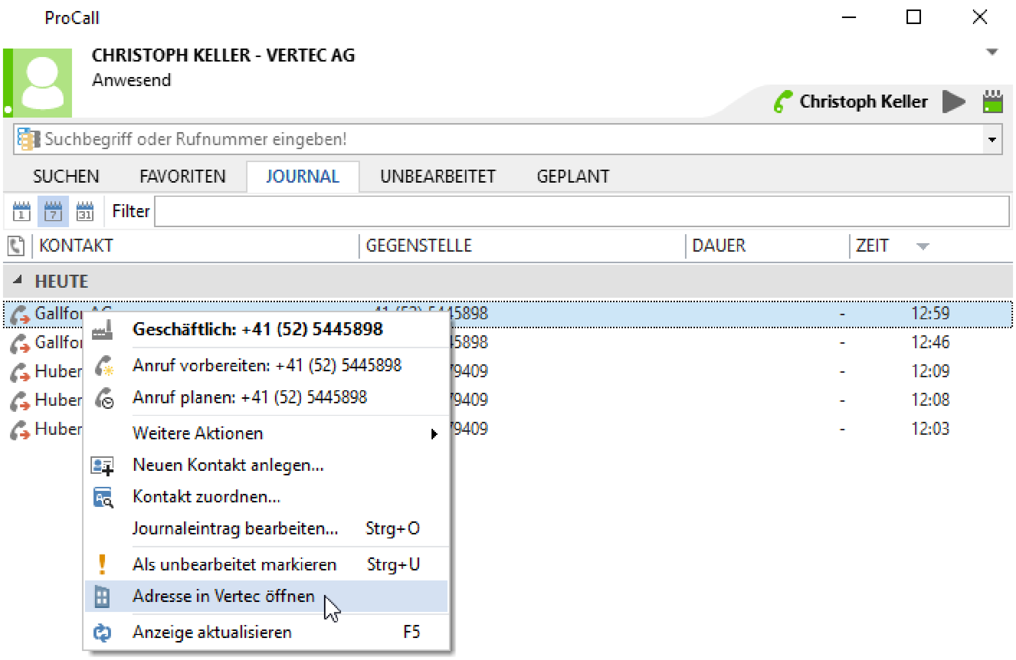
Anruf [11]

Gallfor AG
+41 (52) 5445898

Sebastianweg 1
5400 Baden

Figure 52: The ProCall telephony window

In the journal, as with the contacts, this function is available with a right-click on the mouse:



ProCall
CHRISTOPH KELLER - VERTEC AG
Anwesend
Christoph Keller
Suchbegriff oder Rufnummer eingeben!
SUCHEN
FAVORITEN
JOURNAL
UNBEARBEITET
GEPLANT
Filter
KONTAKT
GEGENSTELLE
DAUER
ZEIT
HEUTE
Geschäftlich: +41 (52) 5445898
Anruf vorbereiten: +41 (52) 5445898
Anruf planen: +41 (52) 5445898
Weitere Aktionen
Neuen Kontakt anlegen...
Kontakt zuordnen...
Journaleintrag bearbeiten... Strg+O
Als unbearbeitet markieren Strg+U
Adresse in Vertec öffnen
Anzeige aktualisieren F5

KONTAKT	GEGENSTELLE	DAUER	ZEIT
Gallfor AG	+41 (52) 5445898	-	12:59
Gallfor	5898	-	12:46
Huber	9409	-	12:09
Huber	9409	-	12:08
Huber	9409	-	12:03

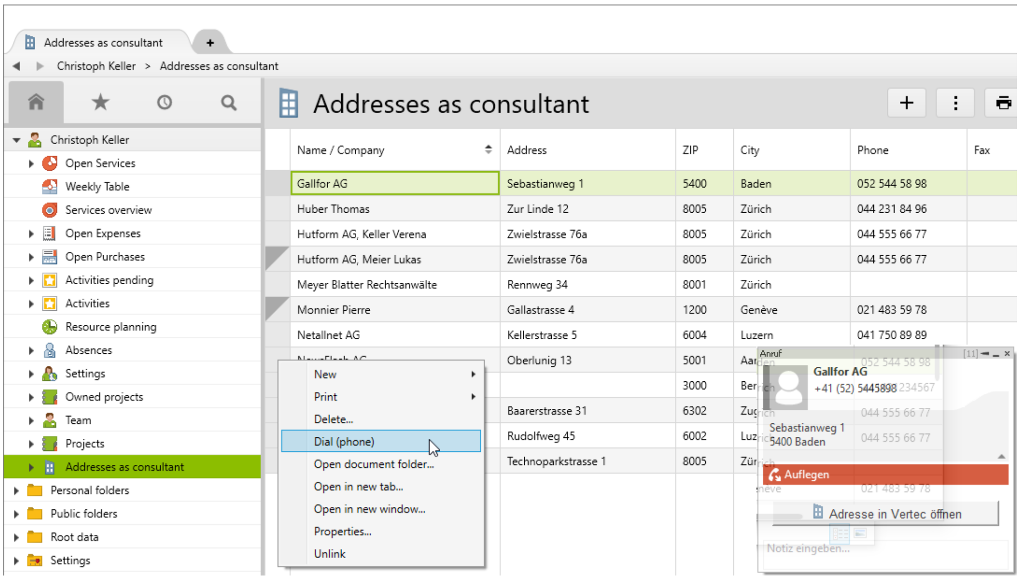
Figure 53: The ProCall telephony journal

Access to addresses in Vertec occurs via COM.

Several addresses can be selected and opened in Vertec. For each selected address, a new tab is opened in Vertec.

Outgoing calls

If ProCall Addin is installed, it can be directly selected from Vertec. The numbers are transmitted via URI to the ProCall client.



Name / Company	Address	ZIP	City	Phone	Fax
Gallfor AG	Sebastianweg 1	5400	Baden	052 544 58 98	
Huber Thomas	Zur Linde 12	8005	Zürich	044 231 84 96	
Hutform AG, Keller Verena	Zwielstrasse 76a	8005	Zürich	044 555 66 77	
Hutform AG, Meier Lukas	Zwielstrasse 76a	8005	Zürich	044 555 66 77	
Meyer Blatter Rechtsanwälte	Rennweg 34	8001	Zürich		
Monnier Pierre	Gallastrasse 4	1200	Genève	021 483 59 78	
Netallnet AG	Kellerstrasse 5	6004	Luzern	041 750 89 89	
NewsFlash AG	Oberlunig 13	5001	Aar		
		3000	Ber		
	Baarerstrasse 31	6302	Zug		
	Rudolfweg 45	6002	Luz		
	Technoparkstrasse 1	8005	Zür		

Figure 54: Direct selection out of Vertec

Incoming calls

In the case of incoming calls, ProCall searches all Communication Means for the corresponding number and gets the data related to the contact such as name, town and address. If several addresses in Vertec have the same number, the one that was input first is selected (lowest ID).

Installation

To install the Vertec ProCall Addin, ProCall must be terminated.

The addin `VertecProCall.dll` lies in the subfolder Addins in the Vertec Installation directory.

To install the addin, execute with local administrator rights the file `InstallProCallAddin.bat` found in subfolder Addins of the Vertec installation.

Removal

To remove the addin, execute with local administrator rights the file `UninstallProCallAddin.bat` found in subfolder Addins of the Vertec installation.

This is only necessary if the addin should no longer be used (no buttons or menus in the corresponding applications). The removal of the registration or a new registration is not necessary for an update (replacement of the DLL file). To overwrite the DLL, one must simply verify that ProCall is not running.

10.2 Abacus: Amounts must only have two decimal places

Line: Standard, Expert | Module: Services & CRM | Version 6.1.0.12

All Abacus interfaces only accept amounts with 2 decimals places at most. This has been implemented in the present version.

10.3 Abacus: Use country code from Vertec addresses

Line: Standard, Expert | Module: Services & CRM | Version 6.1.0.12

Abacus Fibu extensions have an internal mapping table to determine which country code should be used for entries.

This table allows full-text country names in Vertec addresses to be transformed to the country codes used by Abacus.

This mapping table is stored in the code of the extension and only includes values for Switzerland, Germany, Italy and France. If a country is missing in the table, an empty country code is used and Abacus signals an error.

Newly, the country in the address (`standardland` in the address) is used as country code for the interface if the country does not exist in the mapping table.

Backwards compatibility
Before, all values not present in the mapping table were converted to "CH". If values such as "Canary Islands" etc. appear, an error now occurs. It must be ensured that the field holds a valid country code or is empty (in this case "CH" is used).

10.4 Licensing for SelectLine COM interfaces

Line: Standard, Expert | Module: Services & CRM | Version 6.2

The COM interface of the SelectLine software, for which we offer extension interfaces, is subject to licensing for SelectLine Order Management (Auftrag) starting with version 15.1 and for Accounting and Salaries starting with version 16.1.

Starting with SelectLine version 16.1, it is possible to unlock the COM interface directly by using an auxiliary tool of the interface provider without having to buy a license. This is now provided by Vertec.

Since every new SelectLine main version requires a new COM unlocking code, this code must be entered by the customer into Vertec.

In System settings > Accounting there appears a new field `Fibu Unlocking code` when a SelectLine extension has been installed. Input here the unlocking code which you will receive from your SelectLine partner.

11 Technical aspects and database

11.1 Introduction of a free reference field

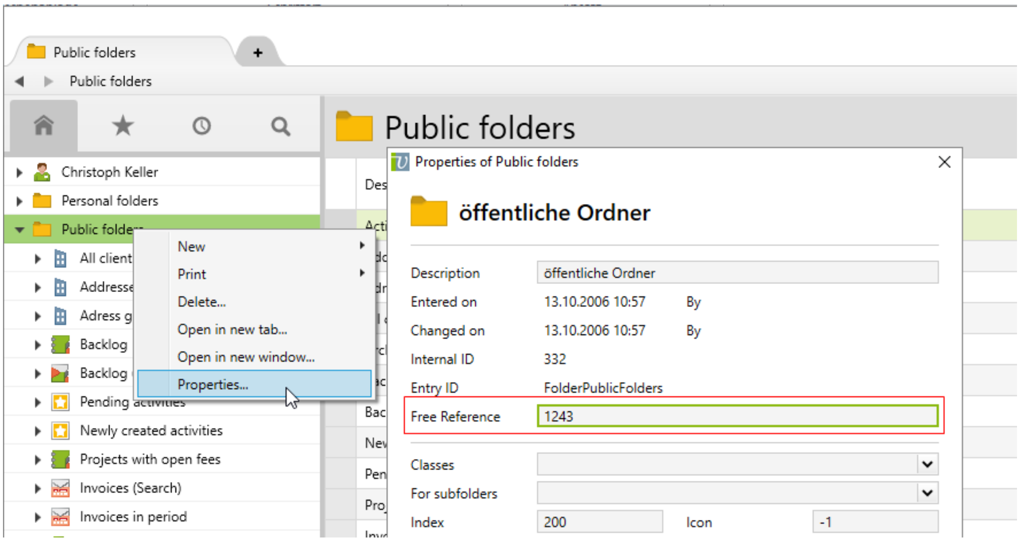
Line: Standard, Expert | Module: Services & CRM | Version 6.2.

On every Vertec object there is now a field called Free reference. It can be freely used.

The reason for this is that field Entry Id can no longer be freely used, since it is now used for the Config Sets (see chapter 5).

Entry Id was sometimes used to store references in external systems during data import (data migrations). The field Free reference can now be used for that purpose.

On the interface, the field appears in the Properties dialogue (right mouse button > Properties):



öffentliche Ordner

Description: öffentliche Ordner

Entered on: 13.10.2006 10:57 By

Changed on: 13.10.2006 10:57 By

Internal ID: 332

Entry ID: FolderPublicFolders

Free Reference: 1243

Classes:

For subfolders:

Index: 200 Icon: -1

Figure 55: Field for Free reference in the Properties dialogue

The value can also be set in the list or in the code. The corresponding OCL attribute is called `freereference`.

Update to version 6.2

When Vertec version 6.2 is started for the first time (see 1.3) an Entry Id is added to all structural data initially created by Vertec.

11.2 Determine the minimal version for DB converts

Line: Standard, Expert | Module: Services & CRM | Version 6.2.

When updating older Vertec versions to the latest version, it may be necessary first to pass through an intermediary version. The corresponding list is found in Update von älteren Vertec Versionen of the article Update von Vertec in the Online Knowledge Base under https://www.vertec.com/de/kb/updateinstallation#olderversions.

Newly, Vertec determines this automatically. If a convert from a version that is too old is attempted, an error message of the following type appears:

```
A conversion of the data-base version 5.7.0 to the program version
6.2.0.1 is not directly possible. Start with a Vertec version older
than 6.2.0 to execute the first step of the convert.
```

11.3 Parameters of the command line

Line: Standard, Expert | Module: Services & CRM | Version: 6.2

Various parameters have been introduced or adapted.

Parameter	Version	Description
/script	6.1.0.11	In addition to registered scripts, Python scripts (ending .py) are also supported on the file system. In this case, the data path must be passed as parameter.
/setpassword	6.1.0.11	Allows setting the password of a user in Vertec Desktop.
/batch_nologin	6.1.0.12	Starts Vertec in batch mode without login dialogue. In this case only the procedures PRECEEDING the login are executed and Vertec is then terminated. It is not possible to bypass the login in this way. To execute Vertec in batch mode without login the possibility to remain logged-in has been extended to the batch mode (see 11.9).

Additional information Detailed information on the various parameters can be found in the article Kommandozeilen-Parameter in the Online Knowledge Base under www.vertec.com/de/kb/parameter.

11.4 Parameter /noevents restricted to administrators

Line: Standard, Expert | Module: Services & CRM | Version 6.2.

Starting Vertec with the `/noevents` parameter is only meant for debugging and data migrations and can lead to inconsistent data during productive operation. For this reason, this parameter is now only active for users with administrator rights.

Since it must be checked if the logged-in user is administrator, `/noevents` can only be evaluated after the login. This means that event scripts of the type Login can be executed as before.

11.5 Problems with VBA in Excel report macros solved

Line: Standard, Expert | Module: Services & CRM | Version 6.1.0.12

The following problems with Excel reports have been solved:

- COM errors from Excel when querying large amounts of data, also when ProcessMessages is called.
- Vertec Desktop is blocked when the report is finished.

Changes

- Vertec is no longer blocked during production of the Excel report. This should not cause problems since the Excel report only uses the object passed to it and does not independently access `vertec.argobject` or `vertec.currentobject`.
- `vertec.ProcessMessages` calls are no longer necessary, yet cause no harm.
- If Excel crashed during report production, or is terminated, the following error message can appear:

```
The specified conversion is invalid.
```

11.6 Vertec.Updater self-autoupdate

Line: Standard, Expert | Module: Services & CRM | Version 6.1.0.14

Cloud App automatically downloads Vertec.Updater during an update. This had to be done manually in the past.

Vertec.Updater is the tool for an automatic update of Cloud App, controlled by the setting `AutoUpdate` in the configuration file Vertec.ini.

11.7 Activate cloud server services by configuration

Line: Standard, Expert | Module: Services & CRM | Version 6.2

The various services of a cloud-server instance can be individually started in the Vertec.ini file. A cloud-server instance on the internet can for example only provide Phone-App service to minimise the risk of attacks.

The following parameters are available in the section [CloudServer]:

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

Default value for all three parameters is true. To turn off a specific service, the corresponding parameter must be set to false.

11.8 Web App: Caching of static files on the client side

Line: Standard, Expert | Module: Services & CRM | Version 6.2

Static files (Images, style sheets, etc.) are now cached on the client. This means that they are no longer transmitted with every loading, but only during the first access or when the cloud server is updated.

This has a positive influence on the performance of the Web App.

11.9 "Remember me" in batch mode

Line: Standard, Expert | Module: Services & CRM | Version 6.2

Vertec Desktop can be started in batch mode (without interface). Previously, username and password were in that case transmitted as parameters. This is not elegant as the credentials might then appear in some scheduled script.

Starting with version 6.2, authentication occurs via Remember me in batch mode also. Desktop App now generates a new AuthToken for the next login which therefore does not expire.

The procedure is as follows: the user logs into Vertec on the server once only and sets the option `Remember me`. If Vertec is then started in batch mode, the user is logged in automatically.

Additional information Additional information on automating processes in Vertec can be found in the Online Knowledge Base under www.vertec.com/kb/task.

11.10 Brute-force prevention mechanism

Line: Standard, Expert | Module: Services & CRM | Version 6.2

To make it more difficult to perform a large number of unsuccessful login attempts, as is usually the case with brute force authentication attempts; a brute-force prevention mechanism has now been added.

The authentication server counts the number of unsuccessful attempts per login name. If the number of unsuccessful attempts for a specific login name reaches a threshold, no more authentication requests are processed for this login name during a specific time interval.

The number of possible attempts and the time interval can be specified in the Vertec.ini file.

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

If one of these values is set to 0 the brute force prevention mechanism is deactivated.

If a successful login occurs before reaching the threshold, the counter for this login name is reset.