January 2019

Vertec Software Release 6.3

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

Table of contents

1 Introduction

These are the Release Notes for Vertec version 6.3. Please consider the following points:

1.1 Vertec first set-up

For a first set-up, use the set-up program VertecSetup-6.3.0.exe, available for downloading. Detailed information on the set-up procedure is available in the Online Knowledge Base under www.vertec.com/kb/neuinstallation.

1.2 Before upgrading to 6.3

Check reports

From Vertec Version 6.3 onwards, the old reporting systems **Internal Reports** and **Reports** (Rb-Reports) are discontinued. Since they can no longer be run, all reports of this type are removed from the report templates.

- ➔ **These reports will not be available after the upgrade to Vertec 6.3!**

Check your reports BEFORE upgrading to Vertec 6.3 and transfer those you want to keep to the new system.

Detailed information is available in chapter 2.2.

Remove manually installed Vertec Service services

Notif, the service used for upgrading data, is now part of the Cloud Server service (see 12.1). Therefore, the Vertec Service is discontinued.

Vertec Service (**VertecServer.exe**) has been removed from the update.

Normally, no action is necessary. If, however, Vertec Service was registered under a different name, it must be removed manually. This must be done BEFORE the update to Vertec 6.3 since VertecServer.exe, needed for the removal, will no longer be available.

More information is available in chapter 12.1.

1.3 Vertec Upgrade

To upgrade an existing Vertec system, use the set-up program as well. 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 in the article **Testinstallationen** in the Online Knowledge Base under www.vertec.com/kb/testinstallation.

Backwards compatibility

Backwards compatibility issues are discussed in detail in the section **Backwards compatibility** for the following features:

Article	Page
2.2 Discontinued features in Vertec 6.3	7
6.1 Keyword folder rights	25
8.3 Vertec lists in Python	36
9 Config Sets	38

Article	Page
12.1 Notif part of the CloudServer Service	52
12.3 XML Server with fixed number format	53

1.4 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.

1.5 Firebird version 3.0.4

Vertec 6.3 comes with the latest Firebird Version 3.0.4. 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/kb/firebird.

2 Versions

2.1 Highlights of version 6.3

Version 6.3 again comes with several highlights. The most interesting ones are listed here:

Article	Page
3.1 SharePoint Online DMS interface	10
3.2 Extended Office reports output to Excel	11
7.1 Choice field in search dialogues for SQL folders	27
8.2 Python Console	34
12.1 Notif part of the CloudServer Service	52

2.2 Discontinued features in Vertec 6.3

Internal reports and Rb reports are discontinued

From Vertec Version 6.3 onwards, the old reporting systems **Internal Reports** and **Reports** (Rb-Reports) are discontinued. Since they can no longer be run, all reports of this type are removed from the report templates. After an upgrade to Vertec 6.3 they are no longer available.

This concerns all reports in **Settings > Report templates**, which are of type `Report` or `Internal Report`:

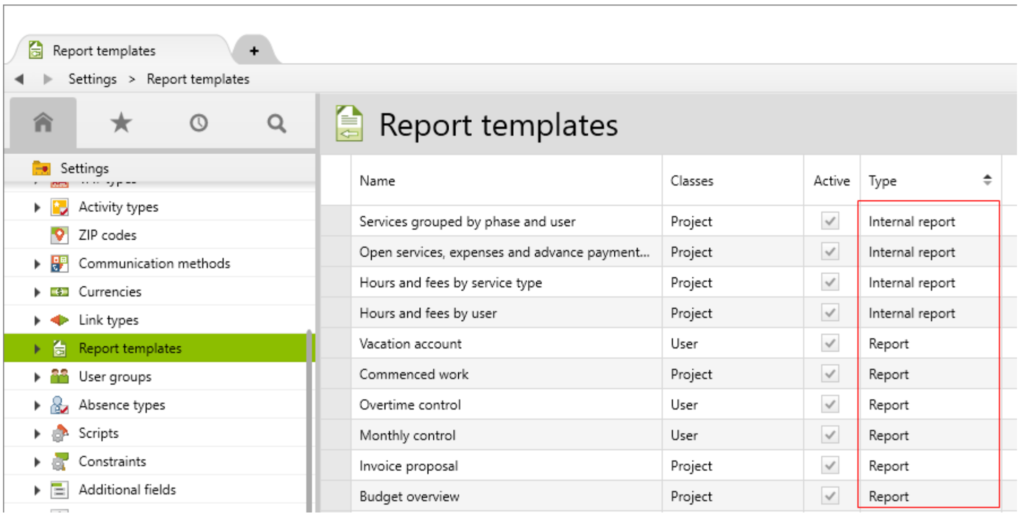


Fig. 1: Legacy reports in the report templates

Vertec provides the following reports as extended Office reports:

Report	Registered on	Template in Vertec reports directory
Commenced work	Project/mandate	CommencedWork.docx CommencedWork.xlsx
User Analysis	Collaborator	UserAnalysis.docx
Budget overview	Project/mandate	BudgetOverview.docx
Invoice proposal	Project/mandate	InvoiceProposal.docx

Report	Registered on	Template in Vertec reports directory
Vacation account	Collaborator	VacationReport.docx
Monthly control services	Collaborator	MonthlyControl.docx
Monthly overview with target times	Collaborator	MonthlyOverview.docx
Project evaluation	Project/mandate	ProjectEvaluation.docx
Invoice with services list	Invoice	InvoiceWithList.docx
Recapitulation of bookings	Project/mandate	RecapitulationOfBookings.docx
Expenses to reimburse by type	Collaborator	ExpensesToReimburse.docx
Overtime control	Collaborator	OvertimeReport.docx
Advance payments	Project/mandate	AdvancePayments.docx
Time balance	Collaborator	TimeSheet.docx

If you wish to compare old and new reports, create a test installation before upgrading to Vertec 6.3 (see chapter 1.3). Please note that the old reports will no longer be available after the upgrade to 6.3.

The extended Office reports provided by Vertec are automatically activated in version 6.3 and all corresponding old reports are removed.

Additional information related to extended Office reports can be found in the Online Knowledge Base under www.vertec.com/kb/erweiterte-office-berichte.

If you use these reports "as is", without customisation, you will be able to continue to work with the same functionalities. Otherwise adapt the reports to your requirements.

If you have question, please contact your Vertec partner.

Collaborator years removed

In the Allocated time system modified in December 2008 with Vertec 5.4, the previously used `BearbeiterJahr` (collaborator year) was replaced by `BearbeiterVorgaben`.

For this purpose, an OCL Operator `getJahre` was introduced which could be instantiated with BearbeiterJahr.

With version 6.3, the class BearbeiterJahr and the corresponding OCL Operator are totally removed.

Backwards compatibility

Customers who worked with Vertec before version 5.4 and are using customer specific reports, report Settings, Expression-Folders, etc. based on BearbeiterJahre or the OCL Operator getJahre must now definitively make the necessary changes.

An overview of the necessary changes with several before/after comparisons can be found in the article **Umstellung des Sollzeitsystems ab Vertec Version 5.4** in the Online Knowledge Base under www.vertec.com/kb/umstellungsollzeitsystem.

More information All OCL operators related to allocated times and vacations are found in the Online Knowledge Base under www.vertec.com/kb/ocl#bearbeiteroperatoren.

VB Script Editor removed

VB Script Editor is no longer available in version 6.3. The corresponding menu entry in System Settings has been removed.

VB Scripts can still be registered as scripts and executed there via the Execute button for testing.

A new Python script editor is available as script editor. It is described in chapter 8.1.

3 Document management and Reports

3.1 SharePoint Online DMS interface

Line: Standard, Expert | Module: Services & CRM | Version: 6.3

The SharePoint Online DMS Extension permits integration with SharePoint Online.

- Vertec-generated and extended Office reports can store their output directly in SharePoint. The result is displayed in Office Online and can directly be processed further.
- Documents and mails from the Office-, Explorer- und Outlook add-ins can directly be stored in SharePoint.
- Documents from Vertec can be opened and processed in Office Online.

The following processes are not possible with the present version:

- Storing Office-generated reports in SharePoint
- Using SharePoint templates

The following steps are necessary to implement this interface (explained here for Office 365):

Register the Vertec application in Office 365

This process must be performed once for each Company. The exact procedure is described in the article SharePoint Online DMS Schnittstelle in the Online Knowledge Base under www.vertec.com/kb/dmssharepoint.

A key is provided which must be entered into the Vertec system Settings under **DMS SharePoint Application Secret**, as well as the application ID, which must be entered into the Vertec system Settings under **DMS SharePoint Application ID** (see the section Installing an interface in Vertec below).

Create or specify a SharePoint site

For a Company, this process must be performed once. The exact procedure is described in the article SharePoint Online DMS Schnittstelle in the Online Knowledge Base under www.vertec.com/kb/dmssharepoint.

A path (URL) is provided which must be entered into the Vertec system Settings under **DMS SharePoint Document Library** (see the section Installing the interface in Vertec below).

Installing the interface in Vertec

To install the interface, open in the folder Settings > Extensions the **SharePoint Online DMS Extension** and click on `Install`.

The interface sets up the following system settings CRM / Activities:

DMS SharePoint Application ID: the Vertec registration key in SharePoint Online, see section Register the Vertec application in Office 365 above.

PropertyName: DmsSharePointApplicationId. StringProperty.

DMS SharePoint Application Secret: the application ID in SharePoint Online, see section Register the Vertec application in Office 365 above.

PropertyName: DmsSharePointApplicationSecret. StringProperty.

DMS SharePoint Library URL: Path to the provided SharePoint site, see section Create or specify a SharePoint site above.

PropertyName: DmsSharePointDocumentLibraryUrl. StringProperty.

After changing the values of the system Settings, it is necessary to restart Vertec or to click on the `Reload` button in the SharePoint Online DMS Extension.

Document path

The base path – System settings General > Document path – is specified as `SharePoint:` followed by the directory in which the Vertec documents should be stored in SharePoint, e.g. `SharePoint:\mydir`.

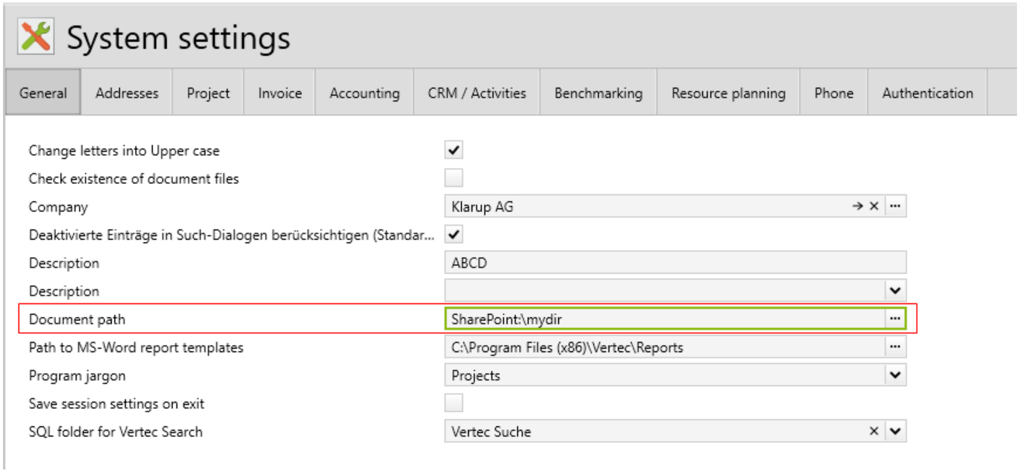


Fig. 2: The document path points to SharePoint

The documents created in Vertec or stored in Vertec through Office add-ins are then directly stored in SharePoint Online.

The Vertec path system has been extended so that the `SharePoint:` prefix (instead of e.g. `C:`) can be used to specify a sharepoint path. It is also possible to save only certain documents in Sharepoint without changing the general document path to Sharepoint.

3.2 Extended Office reports output to Excel

Lines: Standard, Expert | Module: Services & CRM | Version 6.3

Extended Office reports can now be output in Excel format.

The necessary Python-Code is the same as for Word reports, the two formats only differ by their templates.

Excel templates are structured like for Word templates. The following examples use the template CommencedWork.xlsx. It is automatically provided with the present Vertec version. (Report Commenced Work).

Bands

To set a band, mark the corresponding cells and insert the band name in the field on the top left.

As for Word templates, there is a main band which includes everything and is related to the main object, therefore also containing the main frame.

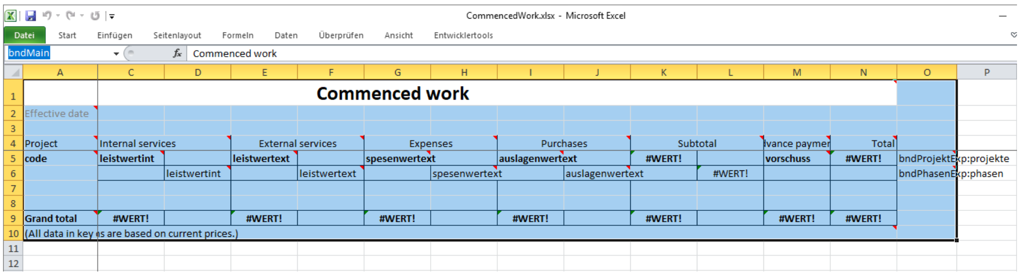


Fig. 3: The main band is specified

Other bands and sub-bands are specified in the same way. Select the cells and input the name on top left:

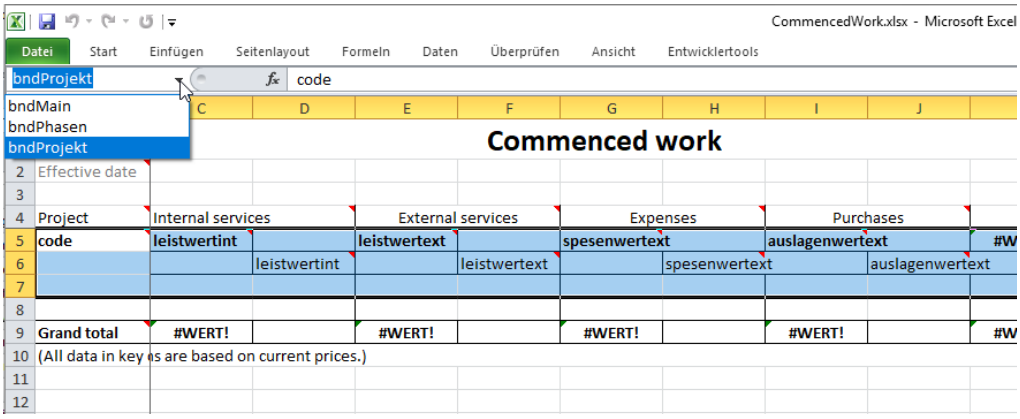


Fig. 4: Various bands of an extended Excel report

Important When setting the bands, be careful not to mark all the cells (from the right edge to infinity), but only the cells which are really needed. Otherwise the build will take forever as Vertec must iterate through all cells to find out where a value is stored.

Band expressions

There are two types of band expressions: Cond and Exp bands.

– The text begins with the name of the band to which the expression relates, followed by Exp or Cond, depending on the type of the expression (see below).
– The reference to the code is specified by `:name`.

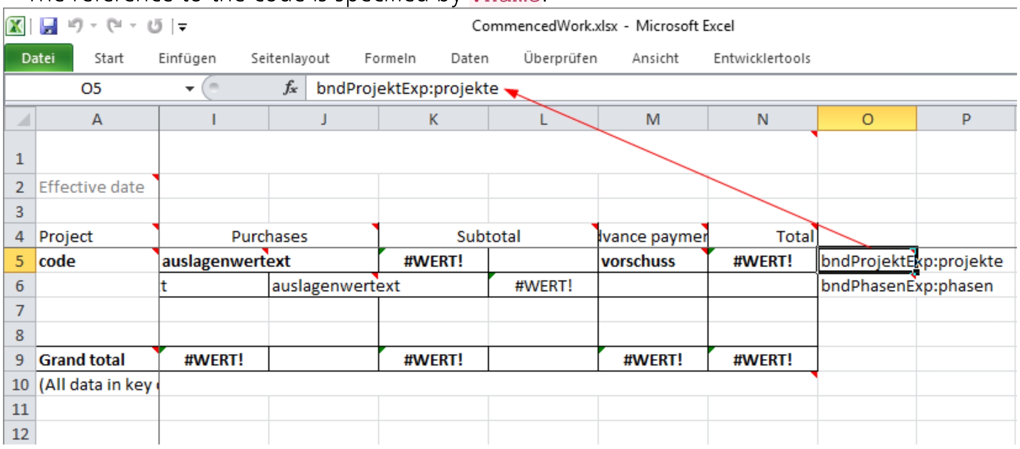
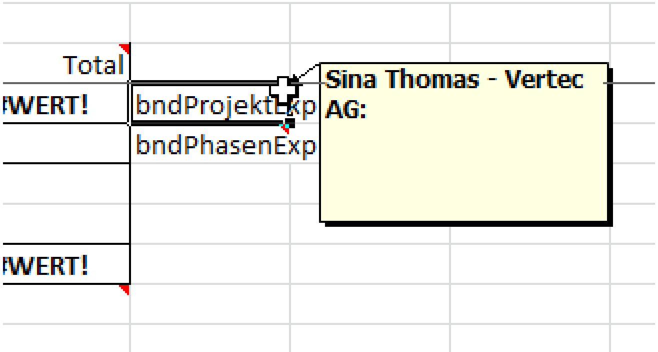


Fig. 5: A band expression

As band expression, one writes for instance `bndProjektExp:projekte` marks the text and adds an empty comment.

Comment Excel automatically inserts the author when the comment is created. It is important that this value be kept and not erased. «Empty comment» therefore does not mean a completely empty comment, but one which only contains the author.



Fields

A field is a text within a band marked by a comment. It will be replaced by data from Vertec when the report is executed. The marked text contains the reference to the fields defined in the Python code. The comment itself is empty (see above) or contains the keyword `translation` to activate the translation for this field.

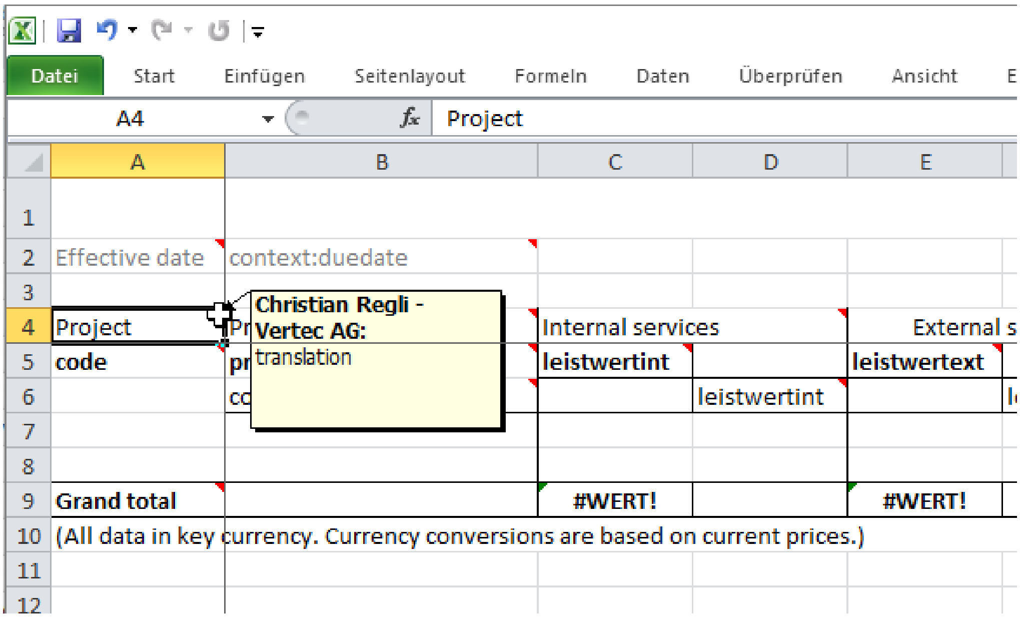


Fig. 6: An item is translated

Comment Each cell can hold one item only. It is therefore not possible to set both a field and a band expression into a cell. In the sample report in Fig. 3, this was solved by inserting the band expression in a cell at the end of the line and making the band one column wider.

Context variables

Context variables are supported as well. Instead of a Python field, the cell contains the text context:, followed by the corresponding variable:

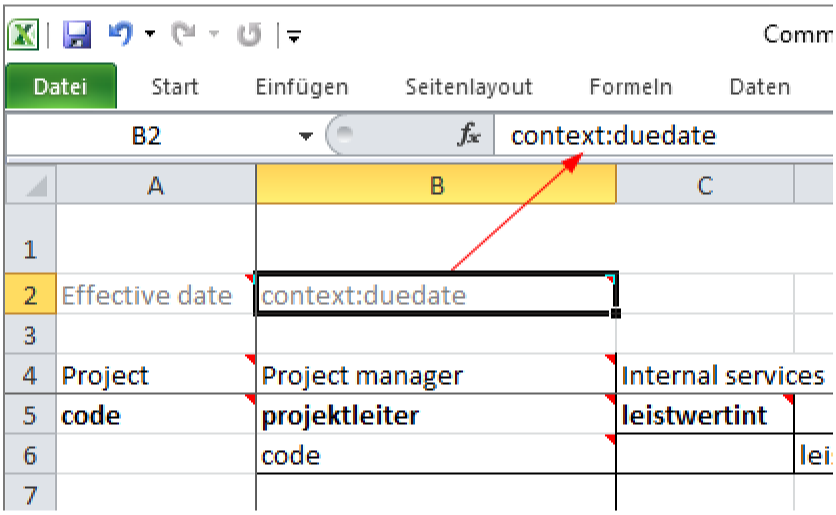


Fig. 7: A context variable is referenced

Data types in Excel

Contrarily to Word, there are different data types in Excel. The automatic type recognition in Excel however does not always make sense. For this reason, it is possible in the Excel template to specify the desired formatting of the cell. When data is inserted or lines are copied down, the formatting is copied as well.

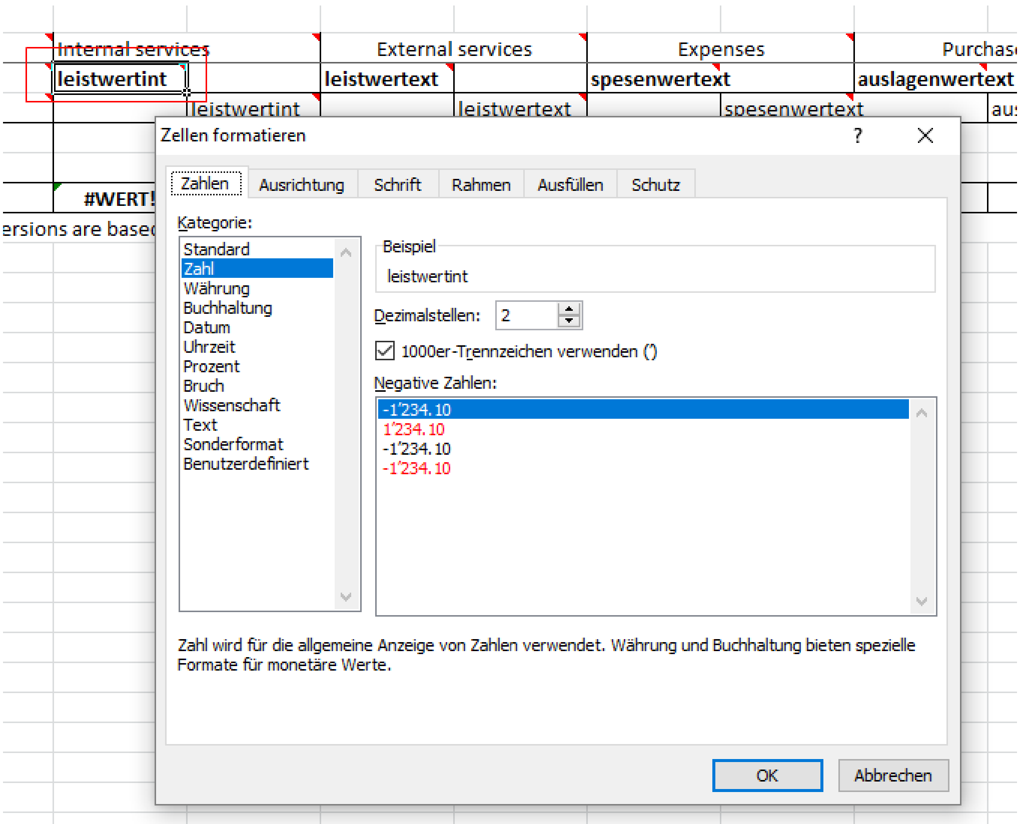


Fig. 8: A cell is formatted as number.

Sums

Performing a sum in Excel is done with the usual Excel formula =Summe(<Band>).

Sums in lines are specified by cell references:

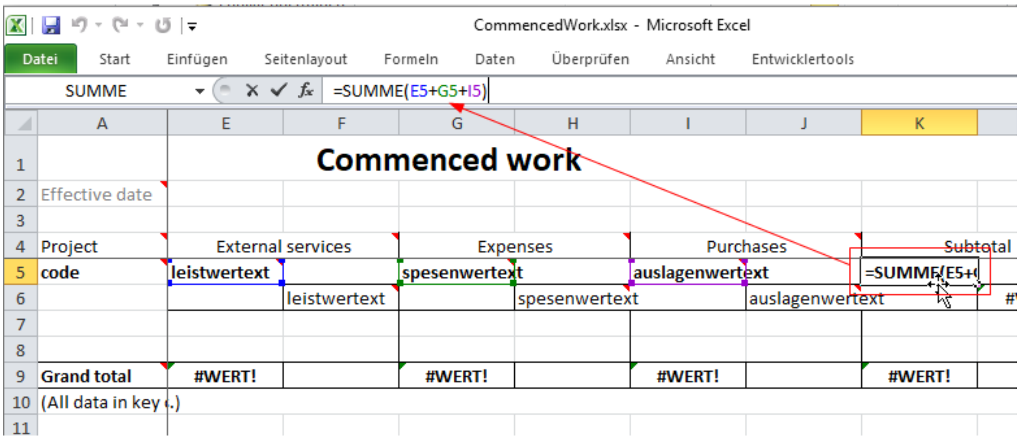


Fig. 9: Intermediary sum referring to values in the line

When summing up columns, the following rules apply:

– All Excel operators are accepted. The format must however be =Operator(<Band>).
– The formula must be outside the band referenced by the sum, but within the band including it (= one level lower).
– The column holding the formula is summed up.
– Subtotals are possible, but only within the same data record.
– Summing of all subtotals in the same column is not supported.

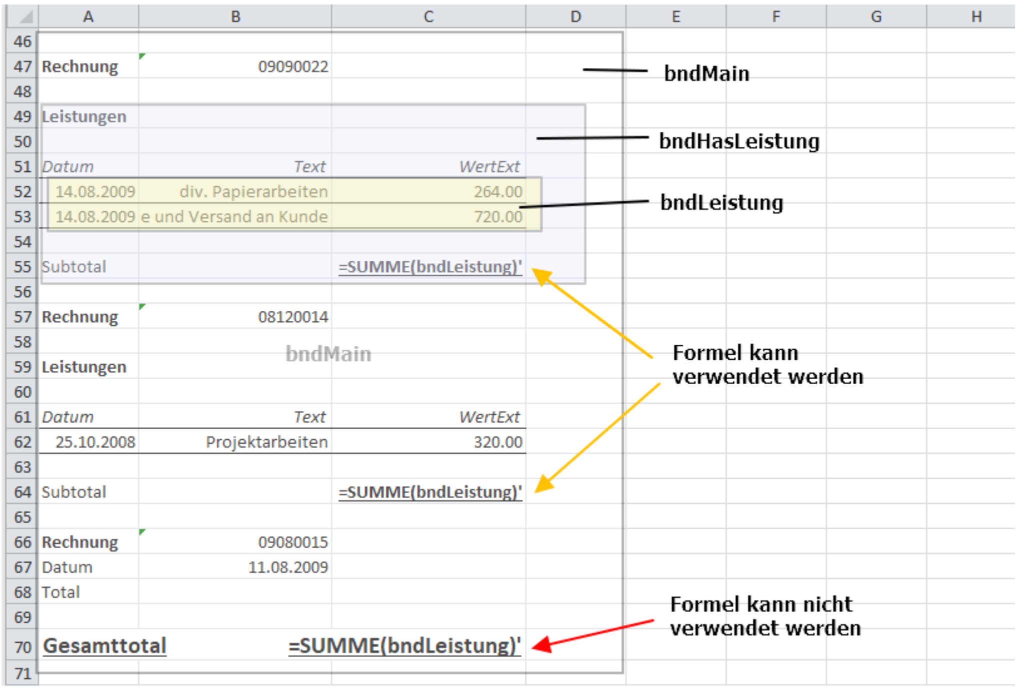


Fig. 10: Summing is only possible within the same data record.

Due to the formatting of the data types (see section Data types in Excel), it may occur that the following message appears when cells are referenced: "A value used in the formula is of the wrong data type."

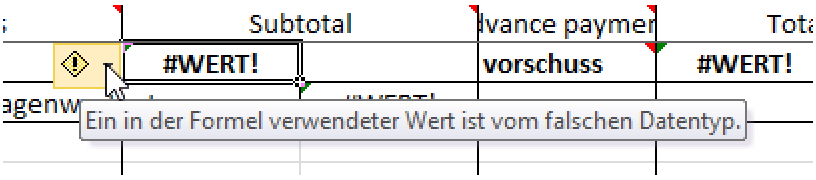


Fig. 11: Reference to a "wrong" data type

The reason for this is that the fields contain for instance `leistwertint`, `leistwertext` etc. which of course cannot be summed. When the report is generated, the sum works because the cells now contain the numerical values.

Registration

Extended Excel reports are registered exactly like the corresponding Word reports. Since they share the code basis, the registration can be performed simultaneously for both:

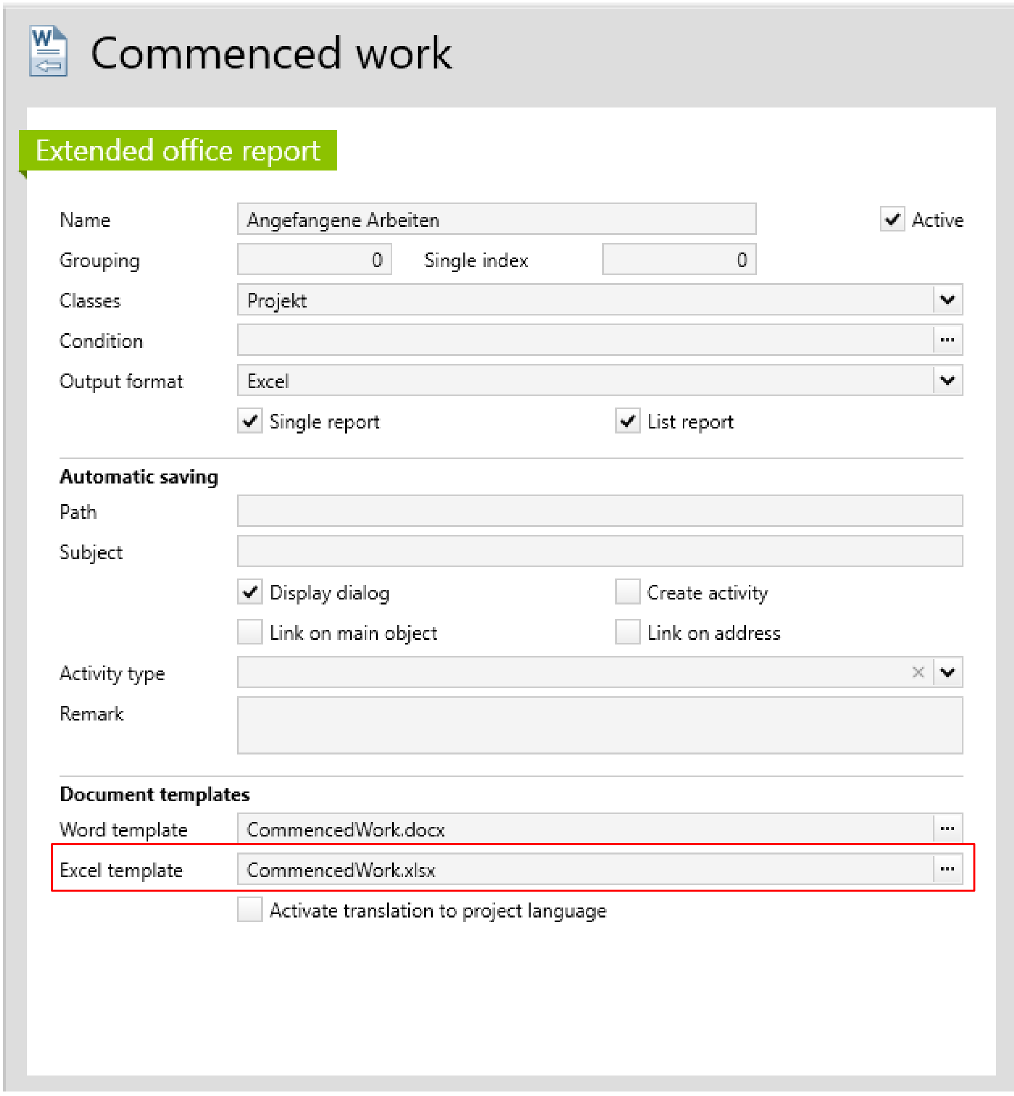


Fig. 12: An Excel template is registered

If the output should be made with Excel, **Output format** must be set to `Excel`:

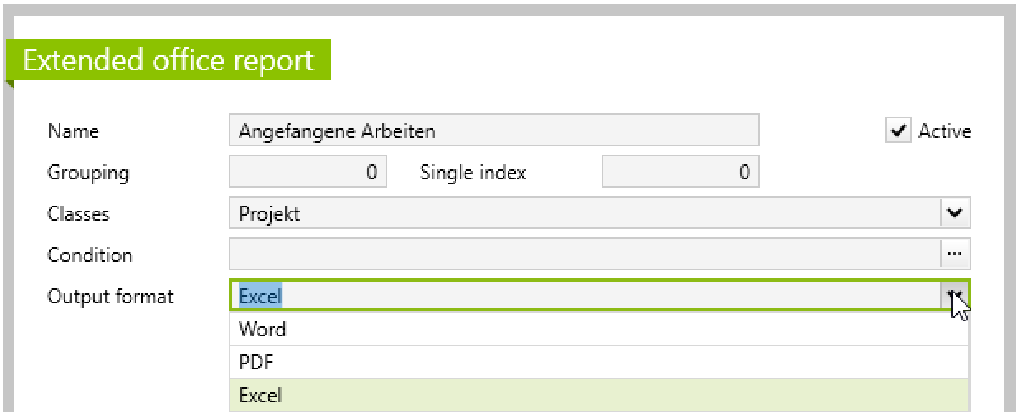


Fig. 13: Output format is set to Excel

If you wish to choose the output format ad hoc at the time you execute the report, click the check box `Display dialog`. A dialog will appear before the execution of the report and allow you to choose the output format:

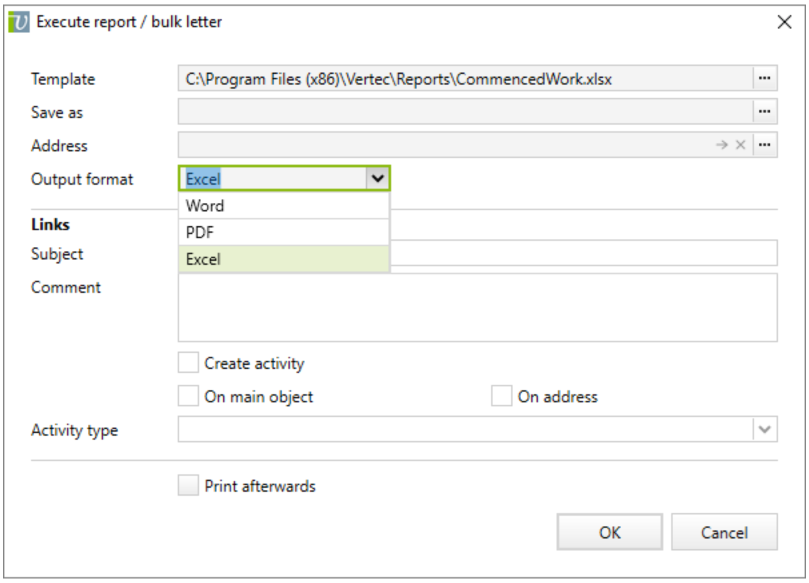


Fig. 14: The output format is selected at the time of execution

– **Word** and **PDF** always use the Word template
– **Excel** always uses the Excel template
If there is no template for the specified output format – if for instance Excel is chosen and there is only a Word template – an error message occurs.

Additional Information More information on extended Office reports is available in the Online Knowledge Base under www.vertec.com/kb/erweiterte-office-berichte.

3.3 Consideration of project language configurable

Line: Standard, Expert | Module: Services & CRM | Version 6.3.

There is a new check mark `Activate translation to project language` for the report registration of extended Office reports:

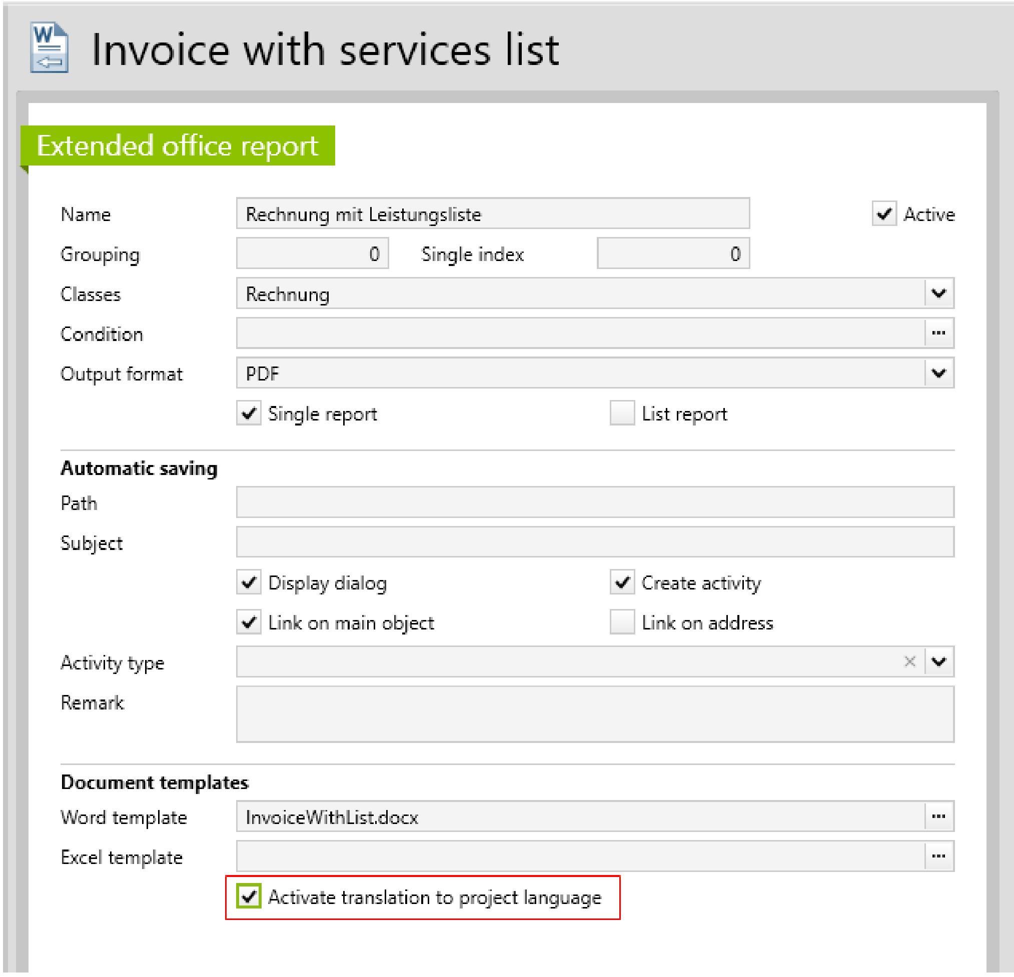


Fig. 15: Activate translation into project language

If translation is activated, this has the following consequences:

– Reports executed on objects which have a defined link to a project (e.g. invoices or project evaluations) will be translated into the project language (respectively all labels marked with `translation`).
– Reports executed on several objects (list reports) continue to be translated into the existing Vertec interface language.

For Vertec-provided extended Office reports (see 2.2) this option is activated by default only for the Invoice with services list. For all others, this option must be activated manually if required.

3.4 Using context variables in OCL

Lines: Standard, Expert | Module: Services & CRM | Version 6.2.0.7

From now on, context variables in extended Office reports can directly be used in OCL. Computations of the following type are now considerably easier to perform:

```
class Collaborator(Frame):
    fields = [
        MinuteField("balance", "calc_balance"),
    ]

    def calc_balance(context):
        return context.currentobject.evalocl("self-
>getFerienSaldo(%s)" % vtcapp.ocldate(context.todate))
```

With the possibility of using the context variable in OCL this code reduces to:

```
class Collaborator(Frame):
    fields = [
        OclMinuteField("balance", "self->getFerienSaldo(todate)"),
    ]
```

The following rules apply:

- Naming: names of the variables set automatically are the same as on the context object: `varXY` and `varXYList`. Names of the variables set manually on the context object may collide with OCL keywords (self, date, now). In that case an error message appears.
- Reporting OCL variables are set on the private OCL evaluator of the executed report. This implies that the variables can only be used in OCL expressions of OCL fields or in method `context.evalocl()`. All other OCL calls, especially the `evalocl()` methods of business objects, use the global OCL evaluator which has no knowledge of local report variables. Instead of:

```
    templist = spesenList.evalocl("self->select(getBearbeiter=
                        varCollaborator)-
>ordermulti('datum;boldid')")
```

  one uses:

```
    templist = context.evalocl("spesenList->select(getBearbeiter=
                        varCollaborator)-
>ordermulti('datum;boldid')")
```

- Only OCL-compatible values are supported (Vertec objects and lists, simple values). With incompatible values (e.g. Python dictionaries, Python lists, etc.) an error appears.

3.5 Access on other fields in field computations

Lines: Standard, Expert | Module: Services & CRM | Version 6.2.0.8

The function `context.get_fieldvalue(fieldname)` in extended Office reports allows to access other fields of the same frame. The computations must therefore be performed once only and the value can directly be used in further calculations. Not only is this clearer, it is also more efficient.

Example

```
class Project(Frame):
    businessclass = "Projekt"
    fields =  [
        OclMinuteField("bdgaufwand", "if (phasen->size>0) then
             phasen.bdgvalue('planminutenint', duedate, -1)->sum else
             planminutenint endif"),
        OclMinuteField("effaufwand", "leistsums->select(Project=
             varProject)->collect(minutenintoffen+
             minutenintverrechnet)->sum"),
        MinuteField("restaufwand", "calculate_restaufwand"),

    def calculate_restaufwand(context):
        return context.get_fieldvalue("bdgaufwand")-
             context.get_fieldvalue("effaufwand")
```

3.6 Context expression as conditional band expression

Lines: Standard, Expert | Module: Services & CRM | Version 6.2.0.4

It is now in extended Office reports possible to specify the conditional expression of a band by means of a context expression. The syntax is the following:

`bndXXXCond:context:name`.

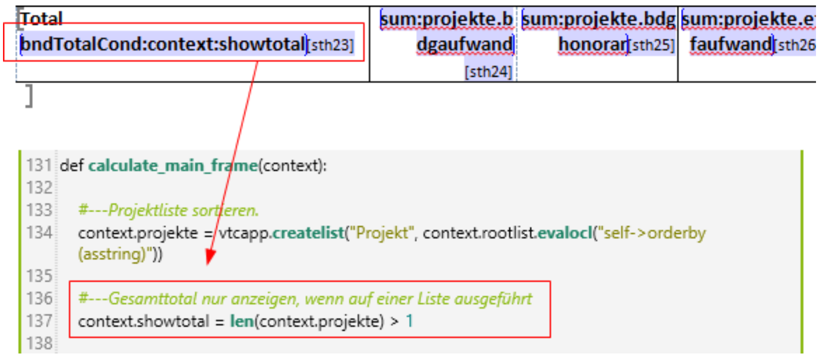


Fig. 16: A context expression to specify a conditional expression.

Further information
More information on conditional bands in extended Office reports can be found in the Online Knowledge Base under www.vertec.com/kb/erweiterte-office-berichte#band-expressions.

3.7 Translations of header and footers

Lines: Standard, Expert | Module: Services & CRM | Version 6.2.0.4

In headers and footers of extended Office reports, terms enclosed in double quotes can be translated with `Translation`. As an example, we want to translate "Page" X "of" XX:

```
{{Translation:Page}} X {{Translation:of}} XX
```

4 Phone App

4.1 Document images for expenses

Lines: Standard, Expert | Module: Services & CRM | Version 6.2.0.7

When an expense is input with Phone App, a picture of the receipt can be stored at the bottom.

By clicking on the field Document image, a picture can be taken or selected from the image gallery.

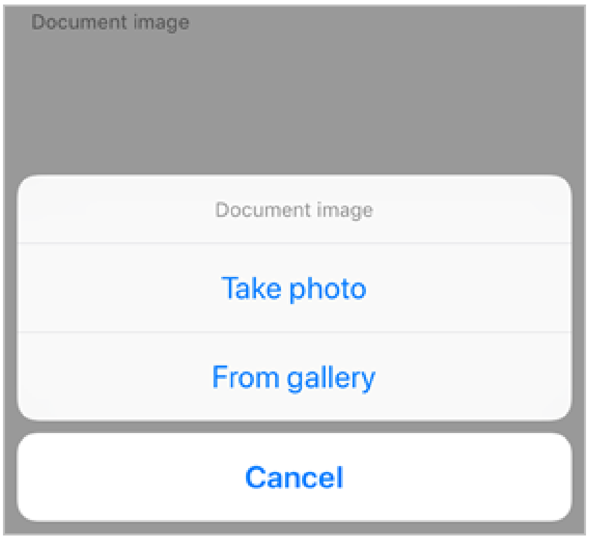


Fig. 17: A picture of the receipt is added

The picture of the receipt then appears with the expense:

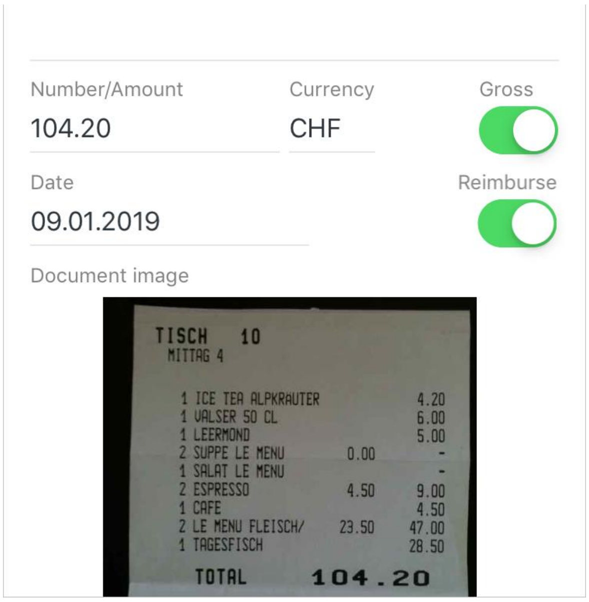


Fig. 18: The picture of the receipt appears with the expense

4.2 Inputting a service with from, to and break

Lines: Standard, Expert | Module: Services & CRM | Version 6.2.0.8

In the Phone App, it is now also possible to input services by specifying the time **from** - **to** with a possible **break**:

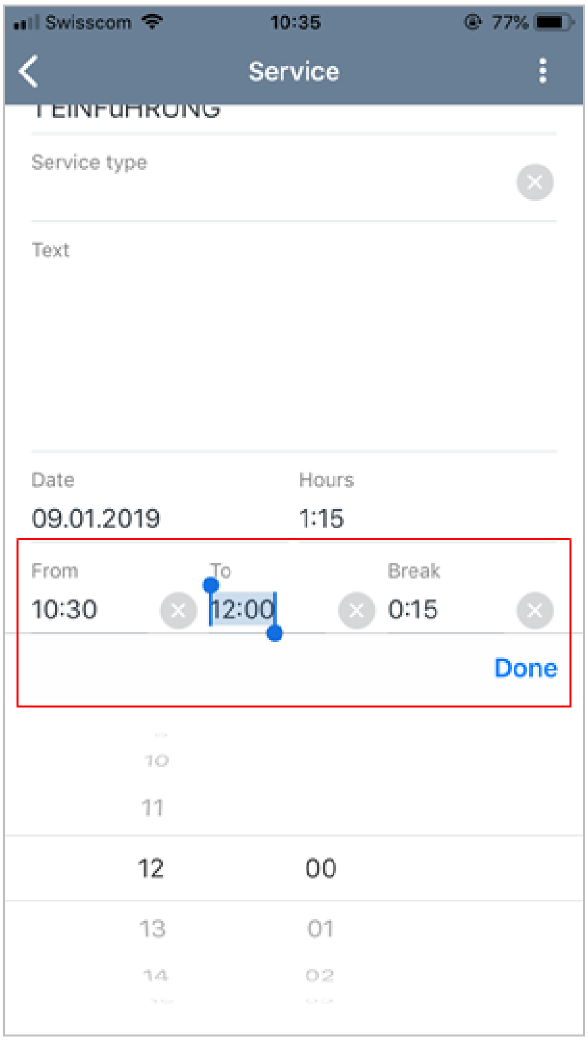


Fig. 19: Specifying the time when inputting a service

Vertec automatically computes the corresponding hours.

If a value is entered into the field **Hours**, the values of **From**, **To** and **Break** are reset to null.

4.3 Third-party products / Data protection statement in the Phone App

Line: Standard, Expert | Module: Services & CRM | Version 6.3.

To display the Vertec data protection statement and the list of third-party products in Phone App there are now links in **Settings**:

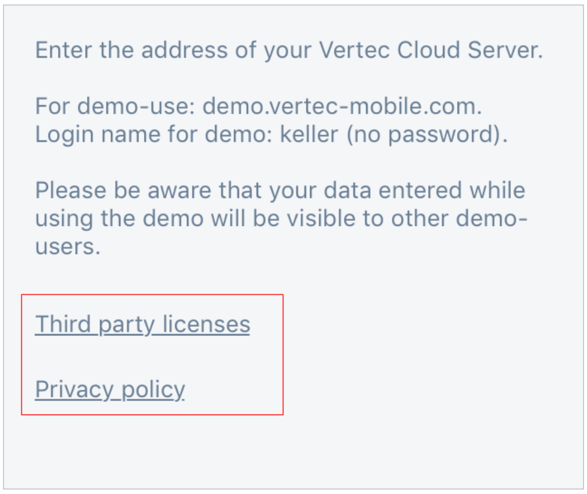


Fig. 20 Links to third-party products and to the data protection statement in the Vertec Phone App

4.4 Support of mobile device management

Line: Standard, Expert | Module: Services & CRM | Version 6.2.0.11

Phone App now supports centralised mobile device management. The path to the server (ServerUrl) can be stored in AppConfig and must no longer be manually entered by the user.

The implementation is as follows:

- When Phone App is started, it is checked whether an AppConfig setting for the ServerUrl is available.
- In that case, ServerUrl is set to this value.

This has the following consequences:

- If ServerUrl has been set via AppConfig, this URL is used whenever Phone App is restarted. The set-up dialogue is omitted and login occurs directly.
- If ServerUrl is modified via AppConfig, the new value becomes active only when Phone App is restarted.
- If ServerUrl is changed manually in Phone App, this value remains active as long as the app is running. As soon as the app is restarted, the value in AppConfig is automatically used again.

Additional information Additional information on this topic can be found in the Online Knowledge Base under www.vertec.com/kb/vertecphoneapp.

5 Invoicing

5.1 Support of company address in Payment type

Line: Expert | Module: Services & CRM | Version: 6.3

Two new fields have been added to Payment type: CompanyAddress and BIC.

This allows for the management of several companies with a single Vertec Installation.

CompanyAddress

There is now a field `Company` in the Payment type.

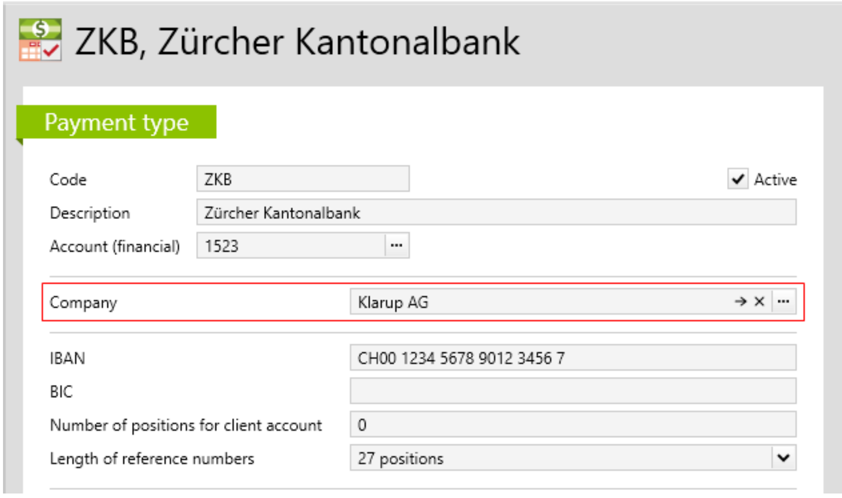


Fig. 21: Company in Payment type

By default, it holds the company defined in `System settings General > Company`. In that case, the company appears in black.

By clicking on the button with three dots, a different company can be attributed to this payment type. In that case, the company appears in green.

BIC

The field BIC (Bank Identifier Code) is a text field.

Backwards compatibility

Invoices and the QR-Code now use CompanyAddress as address of the recipient. Since, by default, CompanyAddress takes the company address from the system Settings, there is no difference unless a company is explicitly specified in the payment type.

The BIC field is purely informative and is not used in any standard report.

6 Collaborators

6.1 Keyword folder rights

Line: Standard, Expert | Module: Services & CRM | Version 6.3.

The existing right **Folder/Links administrator** has been changed to **Keyword folder administrator** and modified as follows:

- Users with the Keyword folder administrator right can create, delete and modify subfolders of keyword folders.
- In addition, users can modify folders only when they are located in their own personal folder.
- These two points also refer to modifications of the list Settings.
- All users can now add new entries to folders.

The existing user group Folder/Links administrator has been renamed Keyword folder administrators.

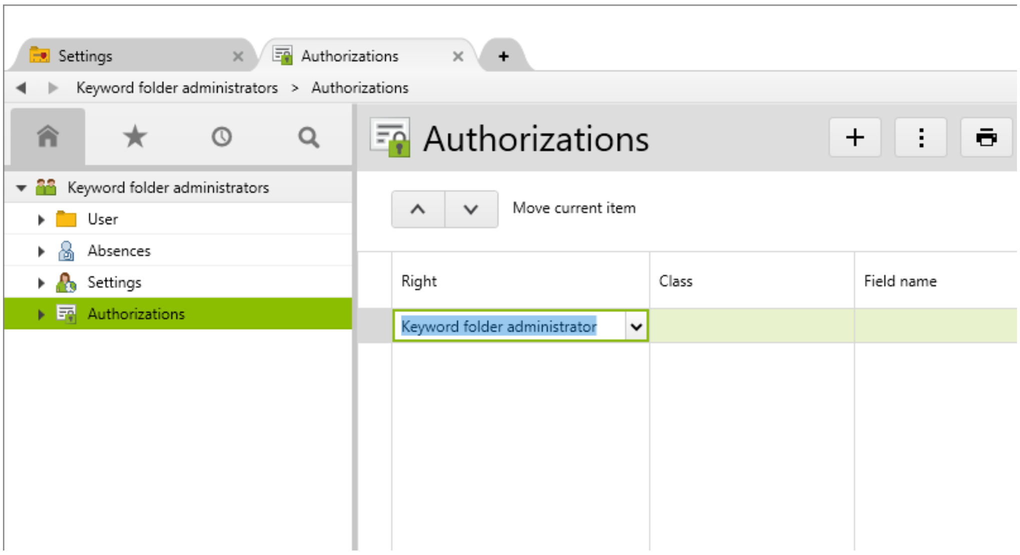


Fig. 22: Folder/Links administrators are now called Keyword folder administrators

The **Folder/Links user** right and the corresponding rights attributions have been abolished without replacement.

Backwards compatibility

Creation, modification and deletion of subfolders as well as modification of list Settings can now also be performed by former Folder/Link administrators – now called Keyword folder administrators – but only for keyword folders or their personal folders.

Creating folder structures and adapting list Settings are configurations which can now only be set by users with administrator rights. If necessary, additional rights must be granted in Vertec.

6.2 Input of absences now possible with DatePicker

Line: Standard, Expert | Module: Services & CRM | Version 6.3

When inputting absences, the date can now be selected with the DatePicker.

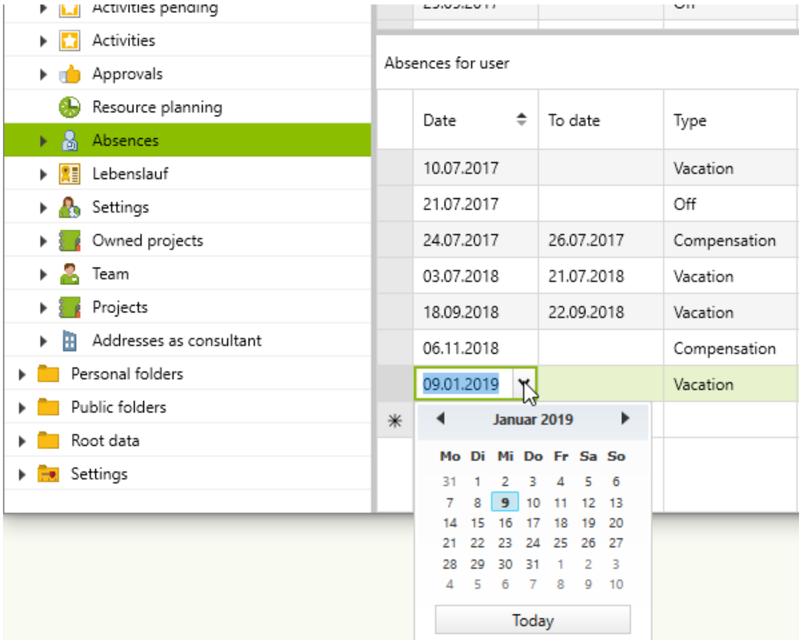


Fig. 23: The date of the absence can be set with the date picker.

As before, a manual input into the field is also possible.

7 Customizing

7.1 Choice field in search dialogues for SQL folders

Lines: Standard, Expert | Module: Services & CRM | Version: 6.3

There is a new field type **Choice** in the dialogue for SQL queries:

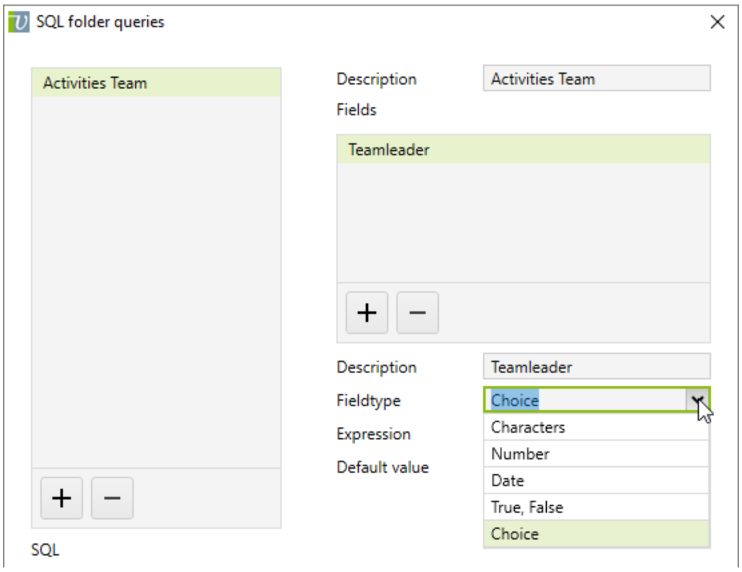


Fig. 24: Field type Choice

As soon as this type is set, a field for the expression is inserted below.

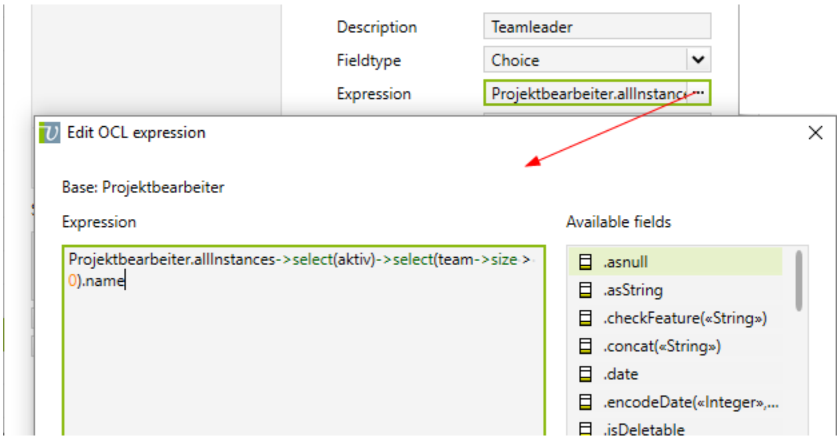


Fig. 25: The expression for the selection

The expression must result in a list of strings.

A selection field is then displayed on the interface, from which the desired value can be selected.

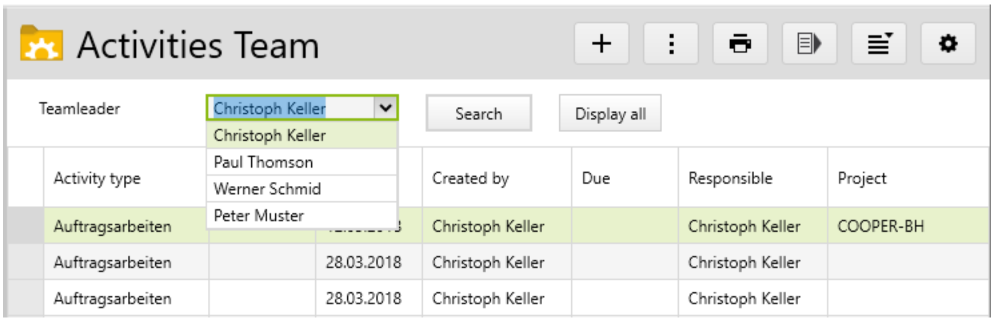


Fig. 26: The desired value can be selected from the selection field

Normally the entry specified in Default value is displayed if such a value is defined.

If a default value is defined, it is displayed by default.

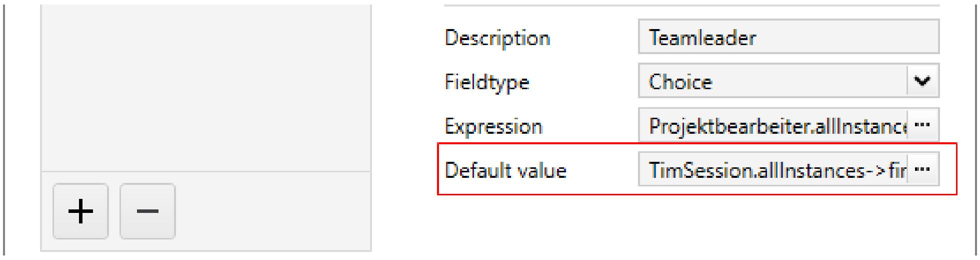


Fig. 27: A default value is defined.

If the specified default value is not present in the selection list or if there is no default value, the first entry in the selection list is used.

7.2 The display of inactive entries in SQL folders can be configured

Lines: Standard, Expert | Module: Services & CRM | Version: 6.3

In SQL folders for addresses and projects, the option **Only active** is available when a dialogue occurs.

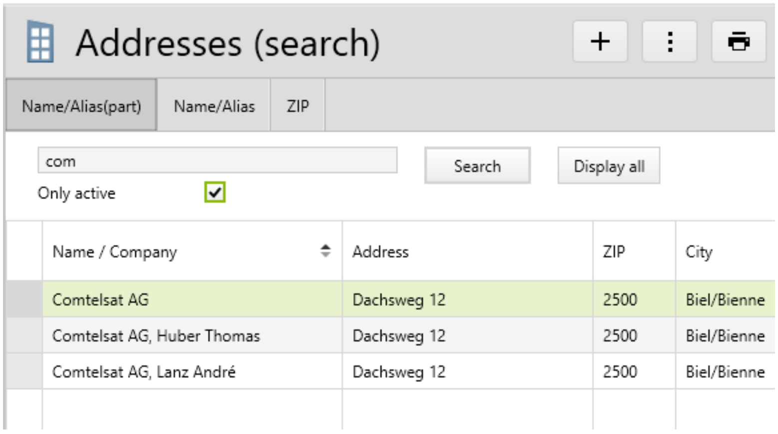


Fig. 28: Only active addresses are searched

Whether this field is set by default or not can be specified in System settings General> **Consider deactivated entries in search dialogs (default value)**.

This option can now be completely hidden. There is an option for this in the definition dialogue of the queries: **Hide 'only active'**:

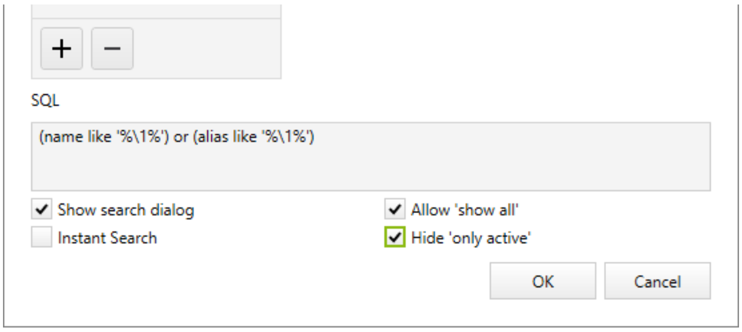


Fig. 29: The option 'only active' is hidden

This has the following consequences:

– The checkbox only active no longer appears on the interface.
– When this checkbox is hidden, the search always includes all entries, active or not. If this option is hidden, the system parameter Consider deactivated entries in search dialogs (default value) is ignored.

Backwards compatibility

By default, the new option is not set. As a result, there is no change of behaviour for existing search configurations.

7.3 OCL null operator

Line: Standard, Expert | Module: Services & CRM | Version 6.2.0.4

The OCL null operator generates null values for any attribute type.

This is useful in the case of numerical values to display an empty field instead of 0, without having to transform the expression into a string. Thus, expressions of the type:

```
if usespesen then spesenint.asstring else '' endif
```

can be replaced by the following expression:

```
if usespesen then spesenint else spesenint.asnull endif
```

Since OCL is a static and strongly typed language, even null values must have a type, so that the validity of the expression can be checked. For this reason, the operators .asnull for instances and .null for types were introduced:

– <type>.null creates a null value of an OCL type. Using the null operator on a type, the above expression becomes:

```
if usespesen then spesenint else VtcCurrency.null endif.
```

– <value>.asnull creates, on an existing value (member reference) a null value of the same type without having to know the name of the type:

```
if usespesen then spesenint else spesenint.asnull endif.
```

7.4 The initial display with ReferenceBox can be configured

Line: Expert | Module: Services & CRM | Version 6.2.0.3

With **ProjectReferenceBox**, the control element to select a project by SQL query, the display shows by default the projects for which services have been registered during the last month.

This can lead to performance problems for systems with a large number of collaborators and projects.

The property **ShowInitialContent** has been introduced for this reason. It allows to turn on or off the search for an initial result list.

Its value is `True` by default; the existing behaviour remains unchanged.

If it is set to `False`, the list of favourites is no longer displayed during the search dialogue.

Example

If you wish for instance to include in the list Settings a project search without display of the favourites (instead of the control element **edtProjekt**) the code in control element XML is as follows:

```
<ProjectReferenceBox ShowInitialContent="False" />
```

7.5 Insert keywords with Entry Id

Line: Expert | Module: Services & CRM | Version 6.2.0.9

Keywords can be inserted in pages with the element **KeywordGroup**. It displays a group with the designation of the keyword folder and, depending on the type, a choice between check boxes or a combo box.

To identify the keyword folder, it was previously necessary to specify the internal Id of the folder with the attribute `FolderId`.

It is now also possible to use the attribute `FolderEntryId` which holds the Entry Id of the folder.

```
<KeywordGroup FolderEntryId="branches" PlaceAfter="InvoicePeriod" />
```

The Entry Id of the folder can be found in the folder properties:

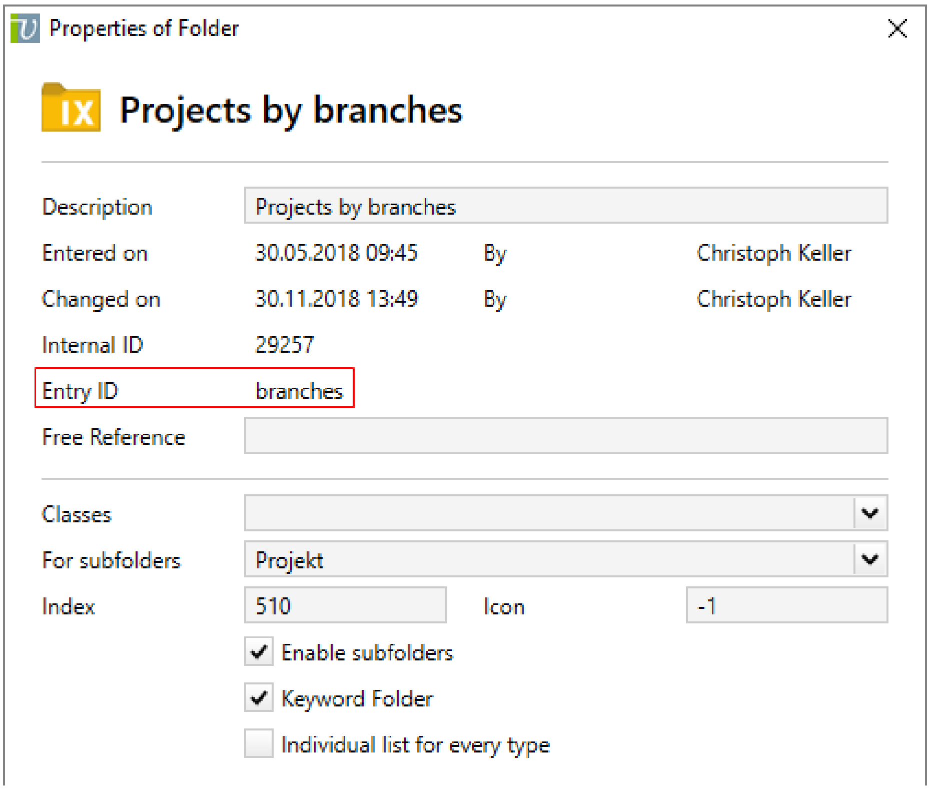


Fig. 30: The FolderEntryId of the keyword folder is specified in Entry ID

Additional information Additional information about page customizing is available in the Online Knowledge Base under www.vertec.com/kb/seiten.

7.6 Extension of syntax highlighting in the XML Editor

Line: Expert | Module: Services & CRM | Version 6.3.

Syntax highlighting of XML Code now includes matching tag.

Begin and end of tags are now highlighted when the cursor is placed on them:

Fig. 31: Begin and end of tag <Group> are displayed

As a result, the code is easier to read and to understand.

7.7 TextArea Info Appearance

Line: Expert | Module: Services & CRM | Version 6.3

TextArea has been extended to display text with several lines, but which cannot be edited.

For this reason, Appearance can now have the value `="Info"`.

Example

```
<TextArea Label="Text Info" Lines="2" Appearance="Info" Text="This is
a text with 2 lines. It is displayed to look like an info text. It
cannot be edited and only serves to display information." />
```

On the interface, this will look as follows:

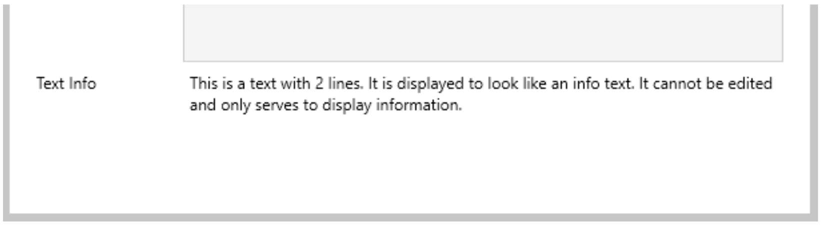


Fig. 32: A text with Appearance="Info" is displayed

Additional information Additional information to customising the Vertec user interface is available in the Online Knowledge Base under www.vertec.com/kb/grundlagen.

---

8 Scripting

8.1 New Script Editor

Line: Standard, Expert | Module: Services & CRM | Version 6.3

A new Python script editor has been implemented for version 6.3 which can also be used in the Cloud App and Web App.

The new script editor is displayed as a separate page in a new tab.

With a click on **Settings > Script Editor** it appears as follows:

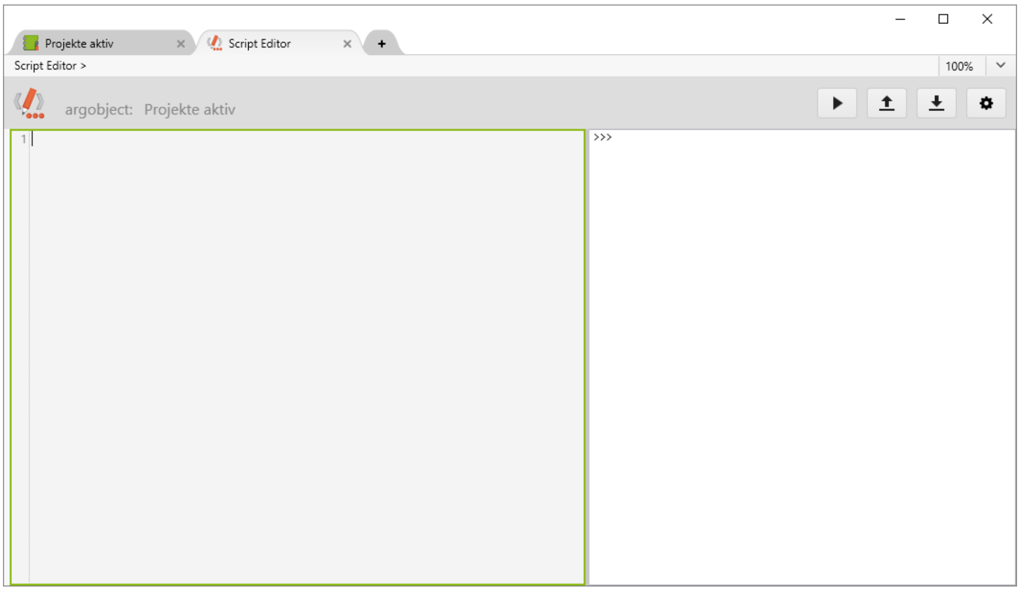


Fig. 33: the new script editor opens with a new tab

On the left part of the window, the code is displayed and/or modified. The code must be valid Python code.

The right side of the window is the output window.

In addition, the Python console exists as a stand-alone program. It only displays the output window. The Python console is fully described in chapter 3.3.

The following 4 buttons are available:

	Execute: Execute the code displayed on the left side of the window.
	Open file: Open a script of the file system in the script editor.
	Save file: Store the code displayed in the left part of the window to the corresponding data file. In the case of new code, a dialogue appears to specify the storage.
	The Vertec Settings menu.

The Script Editor uses as `argobject` (actual object) the last selected Vertec object in another tab. The present argobject is displayed at the top of the Script Editor window.

The Script Editor can also be started with the button **Script Editor** in any script entry. In that case, the text of the script displays the contents of the corresponding script entry.

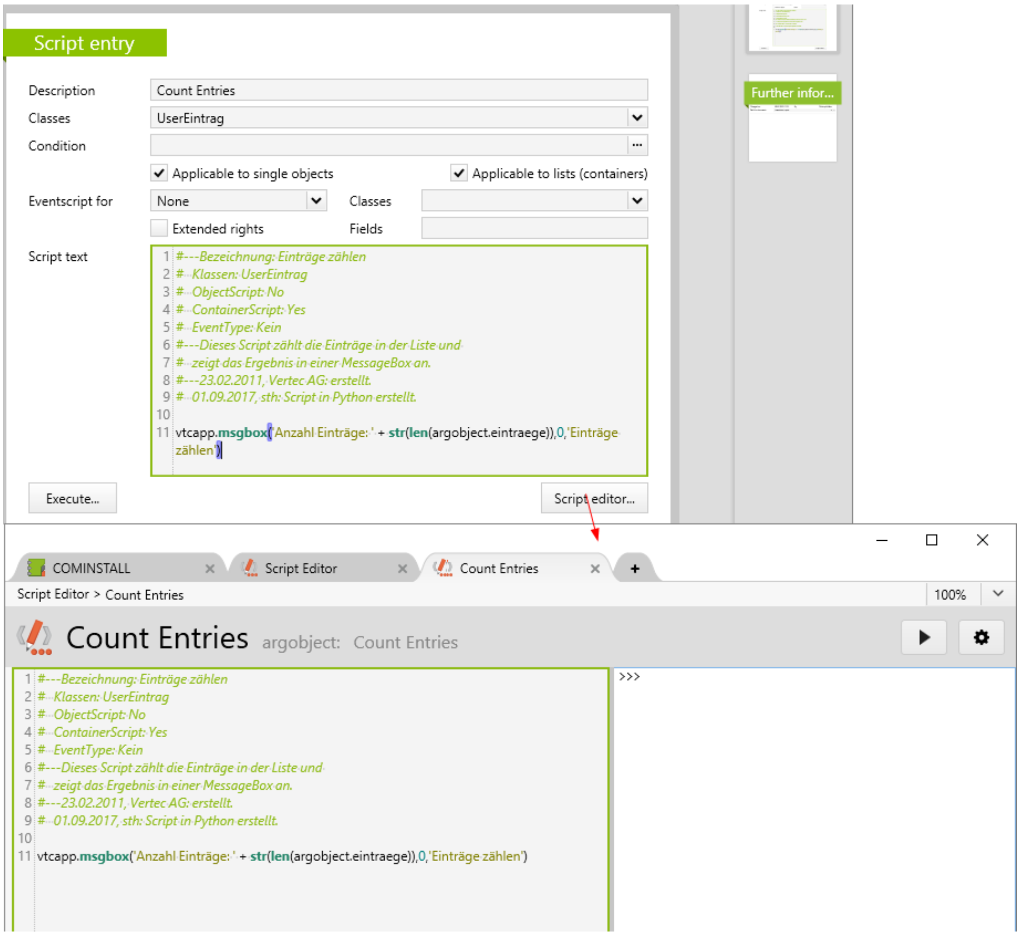


Fig. 34: A registered script is opened in the Script Editor.

In that case, the Script Editor is directly linked to the script entry. If the code is modified here, the text of the script is modified in the script entry as well.

Several Script Editors An unlimited number of script editor windows can be opened simultaneously. They operate independently, each of them however displays as argobject the last selected Vertec object.

The new Script Editor replaces the old existing Python Script Editor.

The old also existing VB Script Editor has been discontinued with this version. See the corresponding note in chapter 2.2.

8.2 Python Console

Line: Standard, Expert | Module: Services & CRM | Version 6.3

An important feature of the Script Editor (see 8.1) is the possibility to execute a Python instruction on the present object and to see its result – the console functionality.

To make this easier, a Python Console that can be rapidly and easily started has been created. The console can be displayed and hidden via the menu **Settings** or with the shortcut `F3`:

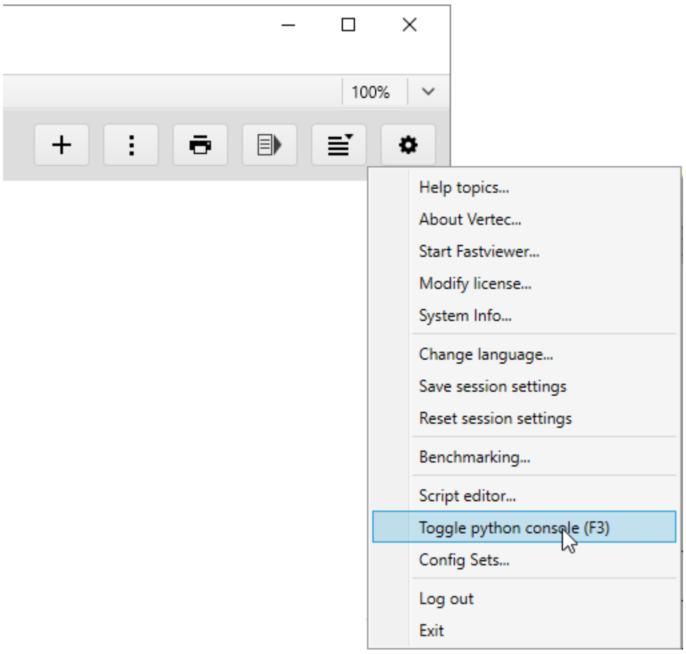


Fig. 35: Settings > Toggle python console (F3)

The console is displayed at the lower edge of the Vertec window:

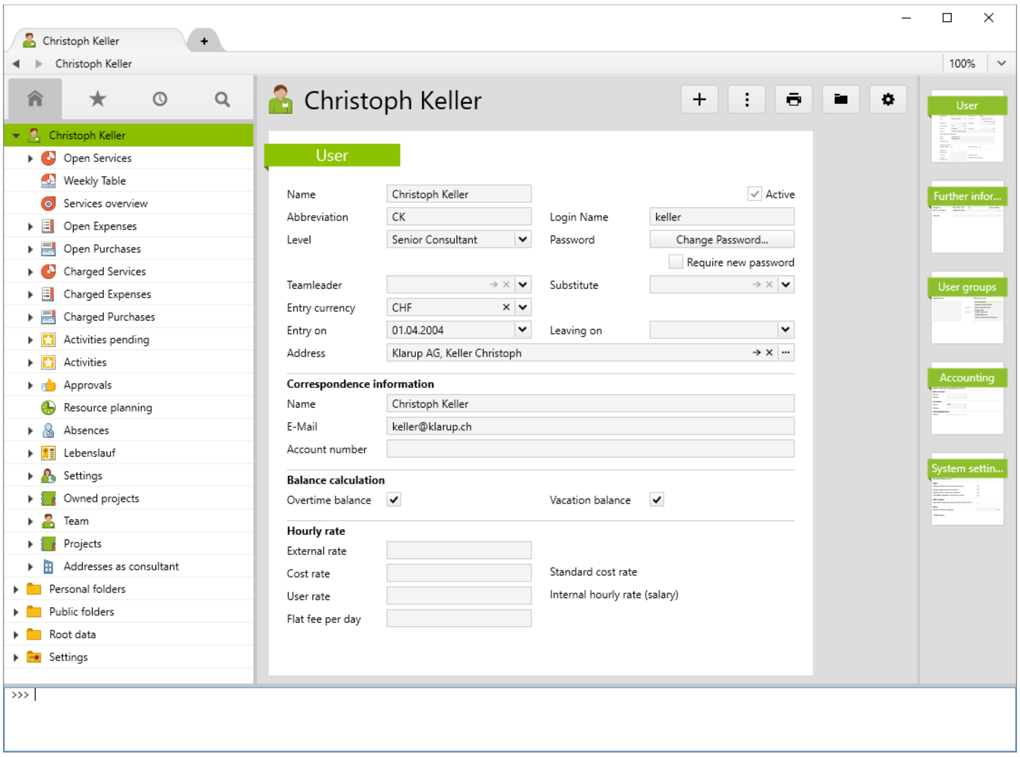


Fig. 36: The Python Console is displayed

Python commands can be executed here as usual. The height of the console can be modified with the mouse.

A right click in the Python console opens a context menu with the option to delete the console content.

8.3 Vertec lists in Python

Line: Expert | Module: Services & CRM | Version 6.2.0.7

All Python methods generating lists now directly generate a Vertec list and no longer a Python list. This is the case for the methods

– evalocl()
– getwithsql()
– getmemberwithsql().

The resulting lists are directly Vertec list and must no longer be generated using `vtcapp.createlist()`.

Before:

```
list = vtcapp.createlist("Projekt", vtcapp.evalocl("projekt->select(code.sqllike('A%'))"))
```

New:

```
list = vtcapp.evalocl("Projekt->select(code.sqllike('A%'))")
```

Methods in Vertec lists A list of the methods in Vertec lists is given in the article Vertec Python Interfaces in the Online Knowledge Base under www.vertec.com/kb/pythoninterfaces#methoden-auf-vertec-listen.

Backwards compatibility

Vertec and Python lists differ by the following aspects:

The + operator cannot be used in Vertec lists. Code of the type

```
projektIds = argobject.eval("eintraege").idstring()
services = vtcapp.getwithsql("OffeneLeistung", "projekt IN (%s)" % projektIds, "") \
             + vtcapp.getwithsql("VerrechneteLeistung", "projekt IN (%s)" % projektIds, "")
```

will not work. An error message will appear: `TypeError: can only concatenate list (not "BoldListAdapter") to list`.

Instead, `extend()` or `+=` must be used:

```
services = vtcapp.getwithsql("OffeneLeistung", "projekt IN (%s)" % projektIds, "")
services.extend(vtcapp.getwithsql("VerrechneteLeistung", "projekt IN (%s)" % projektIds, ""))
```

or:

```
services = vtcapp.getwithsql("OffeneLeistung", "projekt IN (%s)" % projektIds, "")
services += vtcapp.getwithsql("VerrechneteLeistung", "projekt IN (%s)" % projektIds, "")
```

To ensure compatibility, Python lists can be generated from Vertec lists with `list(vertecliste)`.

Please note that Vertec lists are "live". Thus, if actions are performed on the computation logic of the lists, they may be changed, as shown in the following examples:

With `remove()` and `delete()` the list is shortened by one item after every iteration. At the end, only half of the list is erased. To delete all objects from a list, it is better to transform it to a Python list:

```
for l in list(argobject.offeneleistungen):
    l.delete()
```

The following example deals with a list of open services:

```
services = phase.evalocl("offeneleistungen")
for l in services:
    l.makeverrechnet()
```

Each service invoiced is automatically removed from the list. The effect is the same as above: only half the services in the list are invoiced. Instead, transform the list into a Python list:

```
for l in list(phase.evalocl("offeneleistungen")):
    l.makeverrechnet()
```

'count', 'pop', 'reverse' and 'sort' only exist on Python lists. If you use these methods, you must rewrite them or recreate a Python list from the Vertec list with `list(vertecliste)`.

8.4 New Python modules in Restricted Scripting mode

Line: Expert | Modules: Services & CRM | Version 6.2.0.8/6.2.0.11

The following Python modules are now available for use in Restricted Scripting mode:

Starting with version 6.2.0.8

– requests

Starting with version 6.2.0.11

– string
– re
– json
– urlparse
– email
– smtplib
– base64

Additional information on the subject **Restricted Scripting** is available in the Online Knowledge Base under www.vertec.com/kb/restrict-scripting.

9 Config Sets

Config Sets were introduced in Vertec 6.2. Some changes have been made and are described here.

Backwards compatibility

The changes are not backwards compatible: the present state is mandatory. Config Sets previously established may still function but must be tested with the present version and possibly be set up again.

9.1 Folders no longer export subfolders automatically

Lines: Standard, Expert | Module: Services & CRM | Version 6.2.0.9

Previously, subfolders of a folder were implicitly included in the Config Set. For this reason, it was impossible to partition folder hierarchies into different Config Sets.

Subfolders must now be included in a Config Set explicitly. Otherwise they will not be included.

9.2 Configset-reference removed

Line: Standard, Expert | Module: Services & CRM | Version 6.2.0.9

The element `configset-reference` has been removed. Other Config Sets can therefore no longer be referenced.

The elements `entryid-reference` and `ocl-reference` remain.

Additional information Additional information about references can be found in the Online Knowledge Base under www.vertec.com/kb/config-set-xml#references.

9.3 Configset-requirements removed

Lines: Standard, Expert | Module: Services & CRM | Version 6.2.0.9

The element `configset-requirements` has been removed as well.

The element `ocl-requirements` can be used to ensure that certain necessary objects are present in Vertec.

Additional information Additional information about Requirements can be found in the Online Knowledge Base under www.vertec.com/kb/config-set-xml#requirements.

9.4 Include list settings in a Config Set

Lines: Standard, Expert | Module: Services & CRM | Version 6.2.0.9

If a folder is included in a Config Set, the corresponding list settings are included as well and applied to the target system.

It is now possible to include in the Config Set only the list settings.

When a folder is added to the Config Set Builder, it is possible to click on it with the right mouse button. In the context menu appears the menu item `Include list settings instead of folder`.

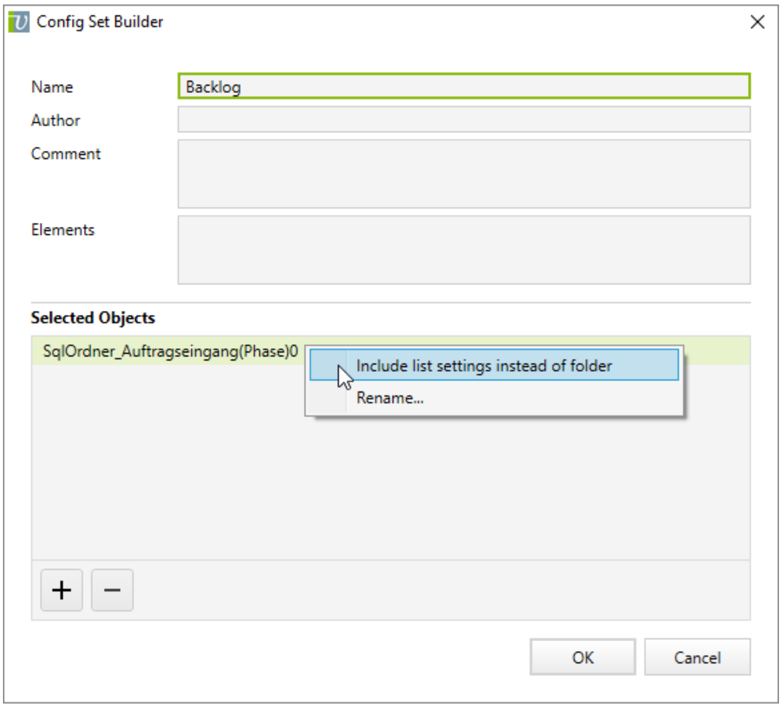


Fig. 37: Include list settings to the Config Set.

Only the list settings are included if one clicks on this option. The line then looks as follows:

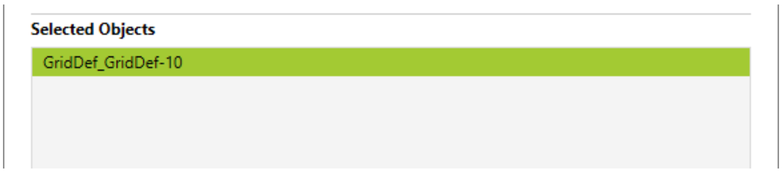


Fig. 38: The list settings have been included.

This means that ONLY the list settings, not the folder, are exported. In that case the folder must already be present in the target system via Entry Id.

9.5 File size for the import of Config Sets increased

Lines: Standard, Expert | Module: Services & CRM | Version 6.2.0.7

The maximum size of Config Set files to be imported in Vertec has been increased from 1 MB to 50 MB.

9.6 Better error messages with Config Sets

Lines: Standard, Expert | Module: Services & CRM | Version 6.2.0.9

For Config Sets, more detailed information on what is missing is now provided when a reference points to a non-existent object. This during the import and when the Config Set is created. Example:

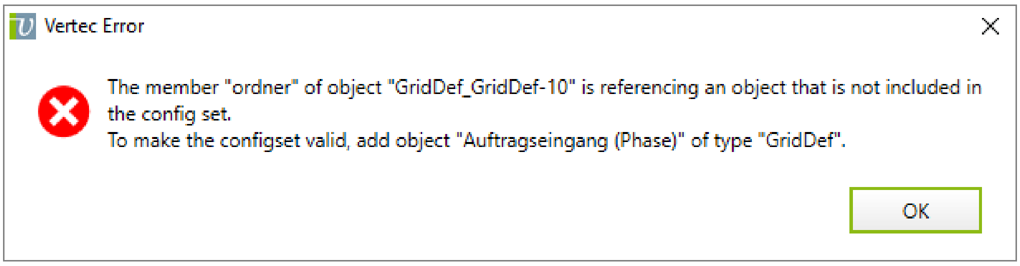


Fig. 39: Detailed error message when a Config Set is created

9.7 Storage dialog when creating Config Sets

Lines: Standard, Expert | Module: Services & CRM | Version 6.2.0.7

A storage dialog now appears when exporting Config Set, which allows to specify where the Config Set will be stored (up to now the file was automatically stored in the Temp folder and immediately opened).

- Web App: With Web App, the dialog asks whether one wishes to open or store (in Downloads).
- Cloud App and Desktop App suggest the storage path of the last storage.

10 Interfaces

10.1 Extensions as Vertec objects

Lines: Standard, Expert | Module: Services & CRM | Version: 6.3

The handling of extensions has been completely changed in version 6.3. Extensions are now independent Vertec objects (of type `UserEintrag`). In the folder **Settings** there is a folder **Extensions**:

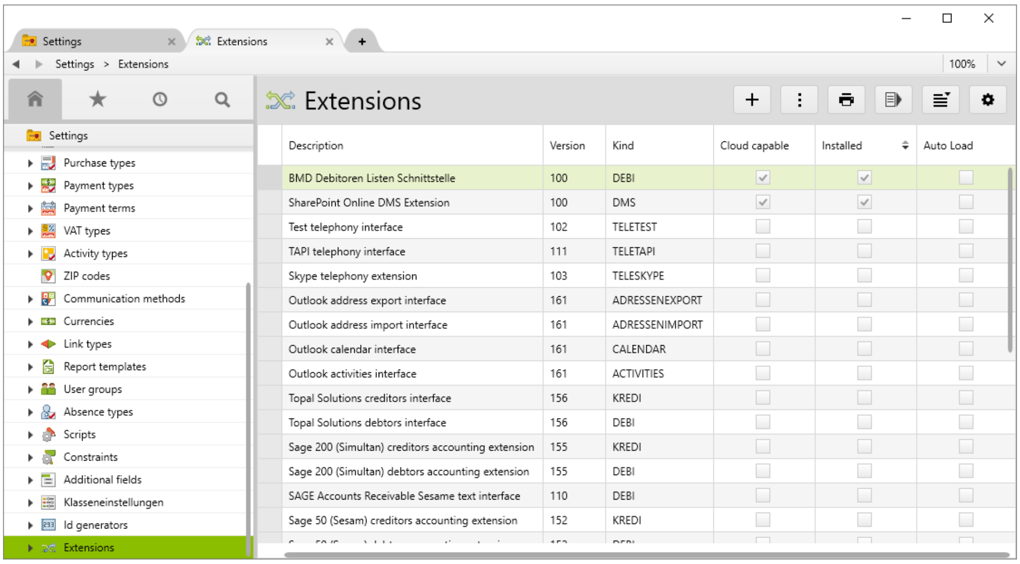


Fig. 40: List of extensions in Settings

There is an extension object for all available extensions. Whether the extension is installed or not is determined by the "Installed" property.

A double-click on an extension opens the corresponding screen:

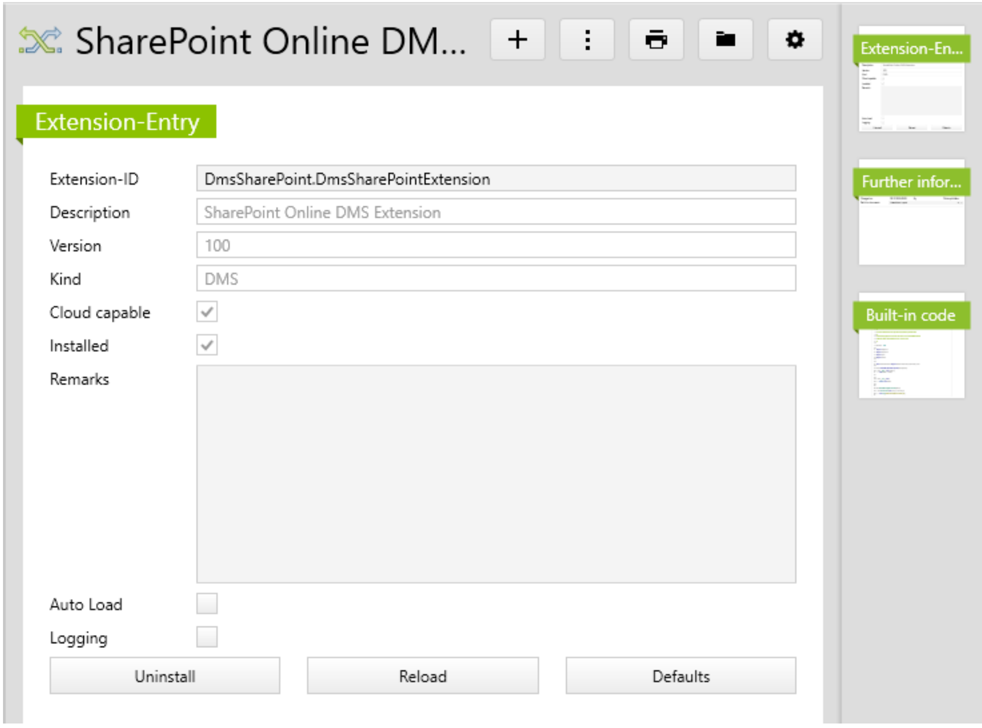


Fig. 41: Window of an extension

Install the interface

To install the interface, click on the button `Install`.

The text on the button thereafter changes to `Uninstall`. Click on it to uninstall the interface.

Auto Load

Auto Load means that the extension is loaded when Vertec is started.

Usually, an extension is only loaded and initialised when it is accessed for the first time. This is sufficient for interfaces to accounting systems. Interfaces to telephony systems, however, which must handle incoming calls, must be loaded when the application is started, independently of any access. The auto load parameter is needed for this reason.

Logging

The Logging checkbox starts the log output for the corresponding extension. This setting is only valid as the application is running. The log output appears on the Python console (see 8.2) or in the output window of the script editor (see 8.1), if they are open, and in the Vertec logfile.

Default values

Resets all Settings of the extension to their initial value at the time of the installation of the extension.

Reload

Restarts the extension or reloads the code after a code change, without having to restart the application for this reason.

Provided code

Two new cloud-enabled extensions are provided in Vertec version 6.3:

– SharePoint Online DMS Extension (see 10.3)
– BMD Debitoren List Extension (see 10.4)

The code for these two extensions is available in the Vertec application and no longer in the Extensions Folder. It can directly be viewed in the extension on page **Built-in code**:

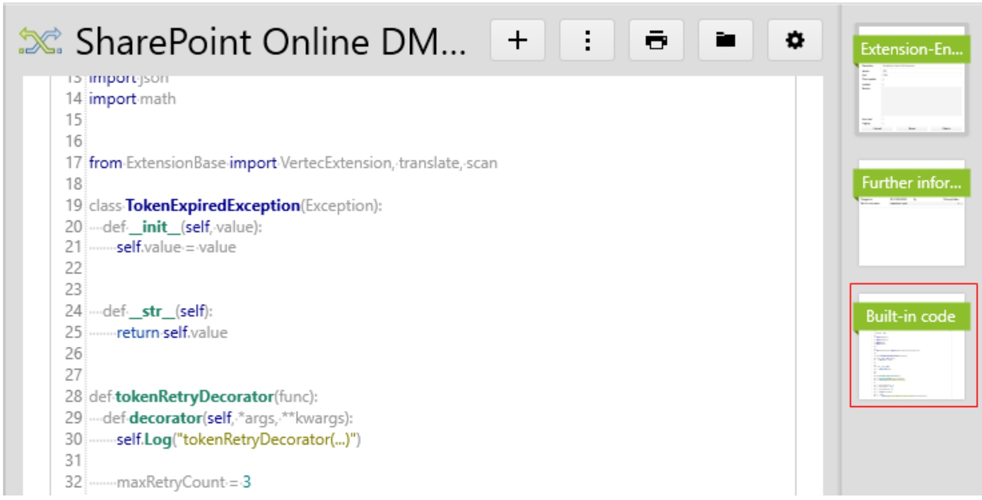


Fig. 42: Built-in code of an extension

10.2 Customizing extensions

Lines: Standard, Expert | Module: Services & CRM | Version: 6.3

Create an extension script

To create a new extension or modify an existing one, create a new script entry in the folder Settings > Scripts and enter the code of the interface as a Python script.

The `Description` of the script entry is the name of the module. Together with the class name of the extension, it forms the Extension-ID. Specify it when registering the interface in Extensions (see 10.1). Module names must be unique: you cannot "overwrite" an existing module by creating a new module with the same name, you do it by creating a new module with a new name and installing it in place of the existing one.

Registering an extension

Create a new extension in the folder Settings > Extensions and complete it in the following way:

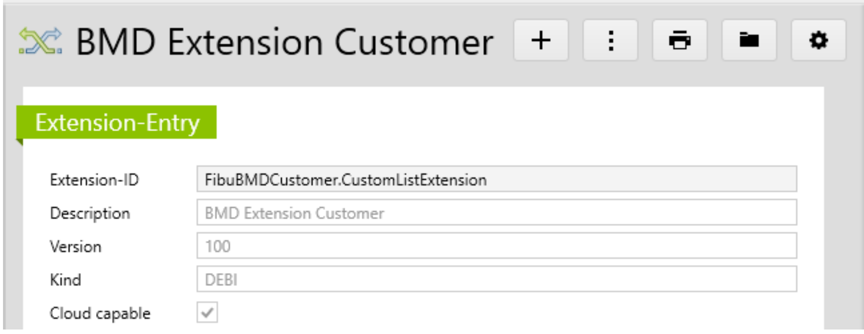
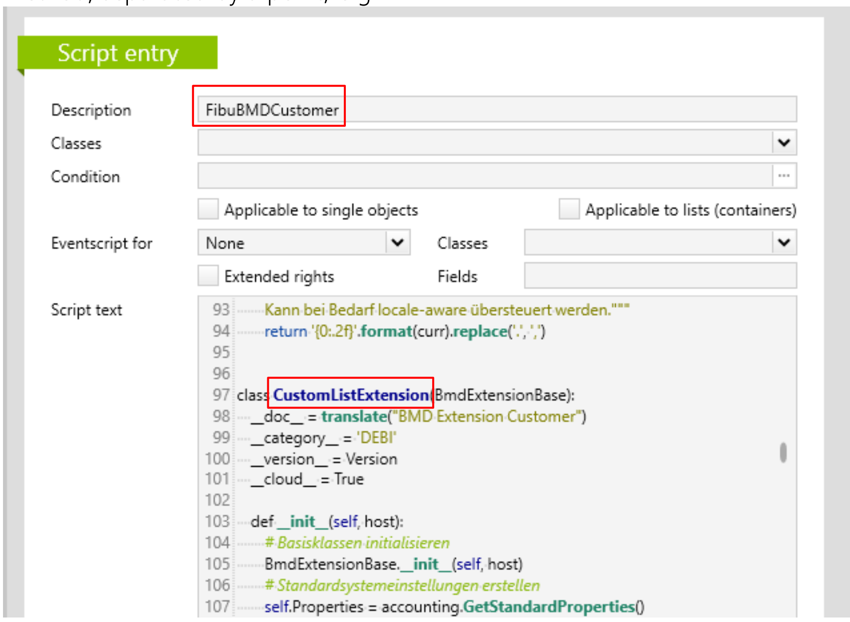


Fig. 43: Registering an Extension

- **Extension-ID: made up of the module name** (name of the script entry) and the called method, separated by a point, e.g.



The following attributes are controlled by properties of the class in the code:

- **Description**: `__doc__`
- **Version**: `__version__`
- **Category**: `__category__` (See the list of categories in Settings > Extensions)
- **Cloud capable**: `__cloud__` (True or False)

10.3 Booking with lists

Lines: Standard, Expert | Module: Services & CRM | Version: 6.3

To support file-based interfaces to accounting software systems in cloud scenarios, Vertec 6.3 now offers the possibility to create lists of booking entries.

In this way, when bookings or cancellations are made, a list of invoices or debtors is transmitted to the interface instead of single objects.

- If booking/cancelling occurs with the right mouse button, the full selected list is transmitted to the interface.
- With a click on the booking/cancelling button in a single window, a list with a single object is transmitted to the interface.

On the DEBI extensions (see 10.1 about extensions), two new interface methods are provided for this:

– `PostInvoiceBookings(list of invoices)` books a list of invoices
– `CancelInvoiceBookings(list of invoices)` cancels a list of invoices

A check is performed to verify if booking is possible. If an invoice in the list does not meet the requirements and can therefore not be booked, the booking will not occur and the invoice is removed from the list. If no invoices can be booked, an error message appears and the booking file is not produced.

At the end of the operation a booking file is returned as a download to the client.

In case of the Desktop App, the value of the Settings parameter Accounting > Buchhaltungsdateien Speicherpfad "AccountingFilePath" is taken into account: if a local path has been specified, the file is stored there and a message appears at the end.

The BMD accounts receivable list interface (see 10.4) is the first implementation of list-base booking.

10.4 BMD Debitoren List Extension

Line: Standard, Expert | Module: Services & CRM | Version 6.3

The BMD Debitoren List Extension is an interface generating CSV files.

It allows booking and cancelling debtor invoices into BMD.

The interface is cloud-enabled and implements a list-based booking process (see 10.3). Since a file is generated for every booking process, bookings can occur for complete lists and not only for single objects.

Installing the interface

To install the interface, open the BMD Debitoren Listen Schnittstelle in the folder Settings > Extensions and click on `Install`.

The interface creates the following System Settings Accounting:

Buchhaltungsdateien Speicherpfad: optional. A data path may be specified here where the generated accounting files should be stored. This is only used with Desktop App (see section Booking/cancelling invoices below).

PropertyName: AccountingFilePath. StringProperty.

Nächste Belegnummer: default value: 0.

If a value > 0 is stored here, it is the counter for record numbers. Each booking (or cancellation) is then booked with this counter as record number. When the booking occurs, this record number becomes the record number in the invoice, but not in the case of a cancellation (standard behaviour of Vertec interfaces).

If anything else is stored here (0 or a negative number), the invoice number is always used as record number. An invoice and its cancellation will then have the same record number.

PropertyName: AccountingNextDocumentnumber. IntegerProperty.

Booking/cancelling invoices

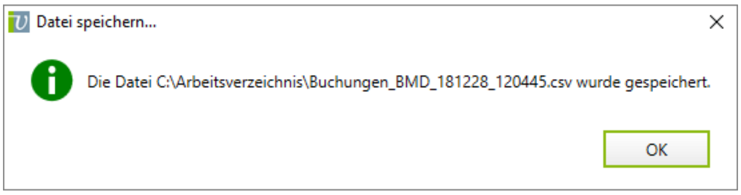
Booking and cancelling of invoices occurs either for a single invoice on page Accounting, or for a list of invoices by right-click > `Book` resp. `Cancel`.

All data records in a file must belong to the same client. This is checked when the booking occurs, otherwise an error message appears.

A file is created for every booking and cancellation process.

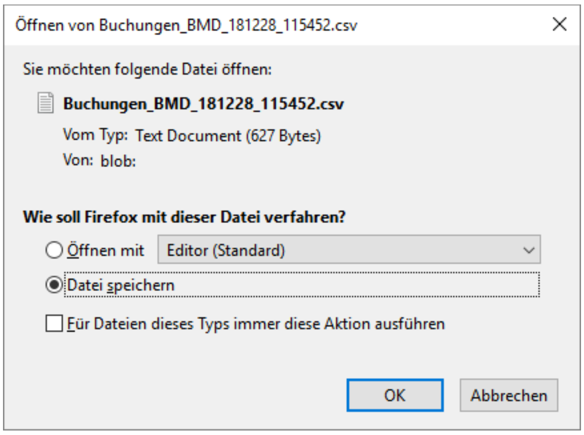
The generated booking files are called `Buchungen_codeacc_YYMMDD_HHMMSS.csv`.

For **Desktop App** clients, if a storage path is entered in the System Settings Accounting, the file is directly stored at this address:



For Cloud App, a dialogue appears at the time of the booking to specify where the file should be stored:

For Web App, the file is directly available for downloading:



10.5 Support of Geschäftsbereiche in the AbaConnect Webservice Extension

Lines: Standard, Expert | Module: Services & CRM | Version 6.2.0.3

This interface now supports the concept of Geschäftsbereiche. In Abacus, Geschäftsbereiche only relate to transaction data (bookings), not master data (personal accounts).

In Vertec, Geschäftsbereiche are implemented as a property of the personal account. This means that the personal account knows into which Geschäftsbereich booking must occur and always books into the same Geschäftsbereich. Bookings covering multiple Geschäftsbereiche are not supported at this point.

The Geschäftsbereich must be specified after the code of the client, separated by hyphen. If therefore a personal account should be booked into Geschäftsbereich `10` of client `7777`, the entry should read `7777-10`.

10.6 Abacus: Booking without "Customer" license

Lines: Standard, Expert | Module: Services & CRM | Version 6.2.0.4

Booking of accounts receivable invoices now functions without the Abacus license for the interface "Customer". This means that, from now on, clients with a "Document" license can also use the interface.

This is valid for the AbaConnect Webservice as well as for the XML Extension.

10.7 Sage50 cancellation implemented

Lines: Standard, Expert | Module: Services & CRM | Version 6.3

Invoices and vendors can now also be cancelled using the Sage50 interface. There are two new System settings Accounting in Vertec for this purpose:

Stornierungsmethode: The reversal method is stored here. The default value is `Beleg löschen`. If you want to use the new reversal method instead, set this setting to `Beleg stornieren`.

PropertyName: Stornierungsmethode. AuswahlProperty.

Durchgangskonto Storno: If you have selected `Beleg stornieren` under Stornierungsmethode, you must enter a Sage50 Transit Account for the reversal here.

PropertyName: Durchgangskonto. StringProperty.

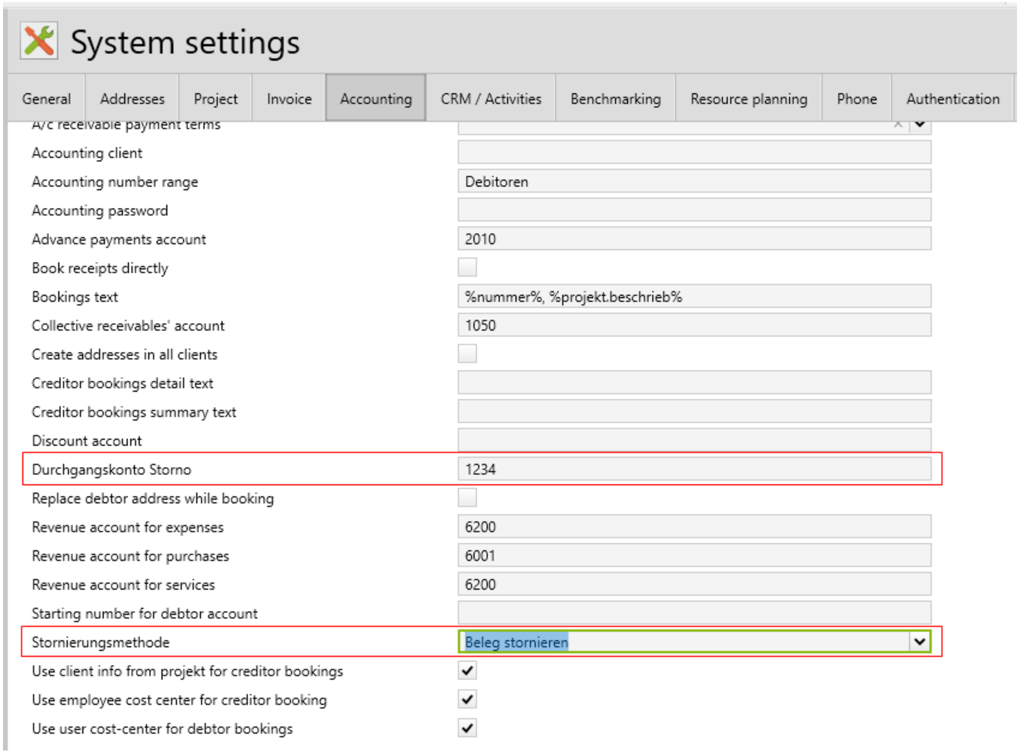


Abbildung 44: These two fields must be filled in to be able to cancel with the Sage50 interface

If you click on `Cancel` in Vertec on the invoice or the vendor under Accounting, the following procedure will be performed in Sage50:

- The reversal makes a new document on the original open item (OP) to the transit account and gives it a new document number. The open item therefore has 2 documents after the reversal. The document number of the reversal document cannot be processed by Vertec. The document number on invoice or vendor therefore remains that of the first document.
- The reversal then creates another open item (the reversal item). Then 2 documents are created: One document clears the original document (debit/credit swapped), the other document clears the transition account document of the original open item.
- The cancellation open item receives the number of the original open item with a capital 'S' appended.

In other words, the reversal is carried out by creating another open item (reversal OP), which is used to balance the postings via a transit account.

The invoice or vendor then appears in Vertec as "not booked" again.

Note It is no longer possible to post with the same OP number. If you want to book the invoice or the vendor again, you must change the invoice number or the vendor number in Vertec.

This transaction can also be carried out across fiscal year boundaries.

10.8 Selectline: use the COM access code only when indicated

Lines: Standard, Expert | Module: Services & CRM | Version 6.2.0.4

The COM access code (Freischaltcode) required via API by newer Selectline versions is now transmitted to the Selectline API only if it has been entered.

If this code has not been entered, the corresponding method in the API is not called. This allows to use newer as well as older Selectline versions, which do not use this API method.

This is valid for Selectline accounting and order interfaces.

This works as follows:

- For Selectline version 16 and later: If a correct COM access code is stored in Vertec, the actions are executed. If no valid COM code is stored, an error message occurs.
- For Selectline versions prior to 16: If no COM access code is stored in Vertec, the actions are executed. If a COM access code is stored, an error occurs since Selectline does not yet know this field.

10.9 The DATEV Extension has been redesigned

Line: Standard, Expert | Module: Services & CRM | Version 6.3

The following changes have been brought to the DATEV Extension:

Supported accounting files

In addition to the existing format Version 3.0, the interface now supports all the subsequent compatible versions.

Using Belegfeld2 is optional

There is now an accounting parameter named **Verwendung Belegfeld2**.

It can be specified here if Belegfeld2 must be described with a payment proposition or with the due date.

11 Performance

11.1 Firebird page and buffer sizes increased

Line: Standard, Expert | Module: Services & CRM | Version 6.3

The performance of the Firebird database server can be improved if more memory is allocated to the database.

The following values are changed during database conversion to version 6.3:

– Page size: increased from 8192 bytes to a maximum value of 16384 bytes.
– Buffer size: increased from 2048 bytes to 10'000 bytes.

This additional memory available for Vertec can have a positive effect on the performance.

For existing Firebird databases, these new values are automatically set during the upgrade.

Memory From version 6.3 onwards, Vertec needs more memory for the database server. Sufficient memory must therefore be available.

11.2 Consideration of activities optimised for project selection in the Outlook Addin

Lines: Standard, Expert | Module: Services & CRM | Version 6.2.0.8

The logic to determine the suggested projects in the Outlook Addin up to now also included activities linked to the found address by the general activity link (`eintraege`) (attributed to the activity through "Further addresses" or "Objects from folder").

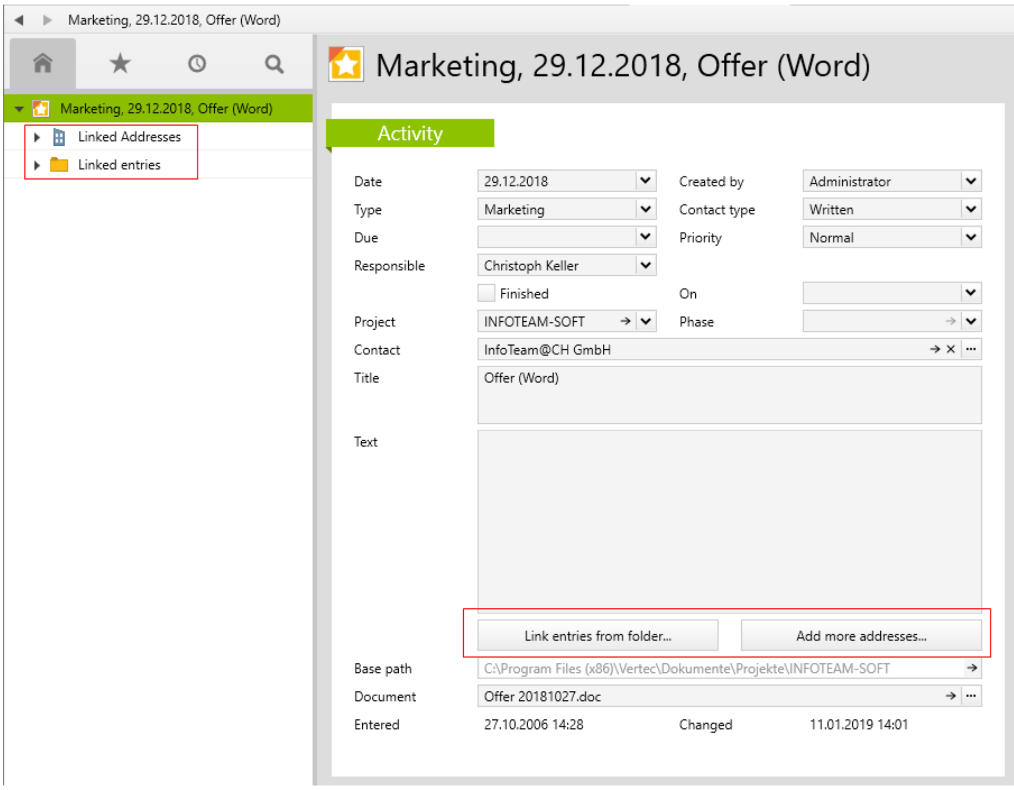


Fig. 45: Linked addresses and entries are no longer considered in the Outlook Addin

This query required a lot of time to execute without bringing much benefit. It has therefore been discarded.

11.3 Store the project persistently on all phases

Line: Standard, Expert | Module: Services & CRM | Version 6.3

The data model of hierarchical phases (subphases) has been designed in such a way that each subphase points to its parent phase.

This means that a subphase has no direct link to the project to which the phase hierarchy belongs.

It is therefore difficult for instance to specify an SQL query on all phases of a single or several projects.

On Projects there has up to now been a derived (computed during execution) attribute named `allePhasen`. This attribute is now persistent and holds the list of all the phases and subphases at all levels.

On Phases, this attribute is called `owningProjekt` and holds the corresponding project.

12 Technical aspects

12.1 Notif part of the CloudServer Service

Line: Standard, Expert | Module: Services & CRM | Version 6.3

The Notif Server is now integrated in the CloudServer Service.

No action is necessary for normal operation.

Indicating `Port=XXXX` in the **[Notif]** section of the configuration file Vertec.ini, as before activates or deactivates the Notif Server.

In the **[CloudServer]** section, the Notif server can be activated and deactivated. `Notif Server = false` in the **[CloudServer]** section overrides the activation described above.

```
[CloudServer]
Server Address=SERVER1
Server Port=8081
Management Port=8082
Notif Server = false

[Notif]
Host= SERVER1
Port=8091
PollInterval=30000
```

The line `Notif Server = xxx` is optional.

If this line is not present, it is equivalent to `Notif Server = true`.

If Vertec.Desktop.exe is started with `/DB`, Notif is explicitly turned off since this involves a different database.

Vertec Service (VertecServer.exe) discontinued

The Vertec Service is therefore no longer used and has been removed from the upgrade.

Backwards compatibility

No action is normally necessary. If, however, Vertec Service has been registered under a different name, it must be removed manually. The Vertec Update only removes the standard registration.

This deinstallation must occur before the update since VertecServer.exe, necessary for an `/uninstall`, is no longer available later.

At least it must be ensured that such a service does not run parallel to the CloudServer with integrated Notif. This would cause port problems, or the CloudServer could not be started if the other service is still running.

Additional information about installation/deinstallation of Vertec Services is available in the Online Knowledge Base under www.vertec.com/kb/serviceinstanzen.

12.2 Improved upgrade behaviour of Vertec.Desktop

Line: Expert | Module: Services & CRM | Version 6.3

Up to now a problem arose when a single user unpredictably started Vertec.Desktop with a newer version: all other clients then announced an error stating that the database version was newer than the version of the application.

Version upgrades with the Desktop app are now accepted only if the app has been started on the database server. On the clients, an error message appears.

12.3 XML Server with fixed number format

Line: Expert | Module: Services & CRM | Version 6.2.0.8

Up to now, number values have been presented and interpreted in region-specific formats by the XML server.

The XML server no longer uses region-specific number formats. It only accepts the point as decimal separator and no longer accepts thousands separators.

Backwards compatibility

Regional number formats are no longer accepted. As a result, existing scripts must be checked and adapted if necessary.

12.4 Set key values and tags via the XML server

Line: Expert | Module: Services & CRM | Version 6.3

Key values and tags can now be set and removed with the XML server.

For this purpose, the following new elements are available in the XML update element:

– `<keyvalue>` with sub-elements `<key>` and `<value>`
– `<tag>` with sub-elements `<add>` and `<remove>`

```
<Body>
  <Update>
    <Projektbearbeiter>
      <objref>4019</objref>
      <kuerzel>ABC</kuerzel>
      <keyvalue>
        <key>ExternalId</key>
        <value>"123465"</value>
      </keyvalue>
      <tag>
        <add>selected</add>
      </tag>
    </Projektbearbeiter>
    ...
  </Update>
</Body>
```

The query of existing key values or tags can be performed with OCL as usual. No new elements exist here:

```
<Body>
  <Query>
    <Selection>
      <ocl>ProjectCollaborator->select(hasTag('selected'))</ocl>
    </Selection>
    <ResultDef>
      <member>kuerzel</member>
    </ResultDef>
  </Query>
</Body>
```

12.5 Set key values and tags via COM

Line: Expert | Module: Services & CRM | Version 6.3

Key values and tags can now be set and removed with COM.

For this purpose, the following methods have been added to **IVtcObject**, in analogy to Python:

SetKeyValue(Key as String, Value as Variant)

```
Set Vertec = CreateObject("Vertec.App")

Set argobject = Vertec.argobject
argobject.SetKeyValue "Date", Date
argobject.SetKeyValue "Boolean", True
```

AddTag(Tagname as String)

```
argobject.addTag ("mytag")
```

RemoveTag(Tagname as String)

```
argobject.removeTag ("mytag")
```

HasTag(Tagname as String): boolean

```
If argobject.hasTag("mytag") Then
```