

verantwortungsvollen Projektmanager und Mandatsverantwortlichen. Mit der Option Budget & Teilprojekte optimieren und verfeinern Sie Ihre Datenstruktur und schaffen dadurch höchste Transparenz im Mandats- und Projektmanagement. Statt sämtliche erbrachte Leistungen einem einzelnen Grossprojekt zuzuordnen, lassen sich Teilmandaten der Teilprojekte erstellen, die dem Hauptprojekt oder weiteren Teilprojekten untergeordnet sind. Somit können Sie präzise nachverfolgen, wo die einzelnen Aufwände generiert sind. Mit Hilfe der **Budget-Funktion** haben Sie Ihre Projekte und Mandate besser unter Kontrolle. Besonders bei grösseren Projekten schafft die Budgetierung von Teilmandaten, bzw. Teilprojekten wertvolle Transparenz. Verschiedene Analysen wie Kostenvorgang-Vergleiche sowie Projekt-Budget-Kontrollen sind perfekte Instrumente für den verantwortungsvollen Projektmanager und Mandatsverantwortlichen. Ein wirklich gutes Gefühl, wenn man die Kosten jederzeit sicher im Griff hat. **Vertec-Budget & Teilprojekte** unterstützen nicht nur das Projektmanagement bei der täglichen Arbeit, sondern liefert der Unternehmensleitung wertvolle Informationen über den aktuellen Auftragsbestand. Dies schafft Sicherheit, wenn es darum geht, fundierte Entscheidungen zu treffen. Insbesondere

# Vertec - Software Cours configurer

# REPertoire

<b>1. QU'ENTEND-ON PAR "CONFIGURER VERTEC"?</b>	<b>4</b>
1.1 Buts du cours	4
<b>2. COURTE INTRODUCTION DANS OCL</b>	<b>5</b>
2.1 Le modèle de classes de Vertec	7
2.2 Les opérateurs OCL importants	9
2.3 Exercices	9
<b>3. RÉGLAGES DE LISTES</b>	<b>11</b>
3.1 Editer l'affichage des listes	12
3.2 Paramétrage supplémentaire du tableau	16
3.3 Exercices	20
<b>4. DOSSIERS</b>	<b>21</b>
4.1 Dossiers manuels	21
4.2 Dossier à expression	26
4.3 Dossier de recherche	26
4.4 Dossier SQL	29
4.5 Exercices	30
<b>5. TYPES DE LIEN (LINK-TYPEN)</b>	<b>31</b>
5.1 Wrapper-Link-Type	31
5.2 Types de liens génériques	32
5.3 Exercices	34
<b>6. CHAMPS ET CLASSES COMPLÉMENTAIRES</b>	<b>35</b>
6.1 Champs complémentaires	35
6.2 Classes complémentaires	37
<b>7. CONSTRAINTS</b>	<b>41</b>
7.1 Exercices	44
<b>8. D'AUTRES POSSIBILITÉS DE CUSTOMIZING</b>	<b>45</b>
8.1 Générateur de numéros	45
8.2 Reports	45

8.3	Scripts	50
8.4	Interface COM	50
<b>9.</b>	<b>RÉFÉRENCES: CONNAISSANCE DE BASE OCL</b>	<b>51</b>
9.1	Members	51
9.2	Listes de classes	52
9.3	Opérateurs	52
9.4	Constantes	56
9.5	Les classes les plus importantes	57
9.6	Aperçu des classes d'adresses (Adressklassen)	61
9.7	Aperçu des classes de dossiers	63
9.8	Exemples de modèles de classes	64
<b>10.</b>	<b>SOLUTIONS</b>	<b>66</b>

## 1. Qu'entend-on par "Configurer Vertec"?

La société Vertec a des clients actifs dans les branches les plus diverses et ayant des exigences les plus variées à l'encontre d'un logiciel de gestion des prestations. Chaque client a des particularités et des attentes individuelles qu'il souhaiterait pouvoir intégrer.

Une entreprise a, par exemple, regroupé ses collaborateurs dans des départements, une autre répartit ses projets par branche. Une autre entreprise encore aimerait volontiers enregistrer des informations supplémentaires au niveau des phases d'un projet et un cabinet d'avocat souhaiterait représenter les relations entre les administrateurs et les différentes sociétés.

Pour nous, la société Vertec, il est en revanche important de développer un seul produit. Cela simplifie la maintenabilité et le support et nous permet ainsi garantir que tous les clients disposent en tous temps d'un accès facilité aux mises à jour les plus récentes et que les modifications et les caractéristiques intégrées puissent atteindre tous les clients.

Afin que nos clients issus des branches les plus diverses puissent appliquer leurs exigences spécifiques, nous avons intégré divers instruments avec lesquels il est possible de configurer Vertec selon les spécificités du client.

En voici les plus importants :

Structures par dossiers	Dans Vertec, les objets peuvent être catégorisés à l'aide de la structure par dossiers.  En outre, des dossiers de recherche et d'évaluation peuvent être créés afin de rassembler les informations les plus diverses et les rendre plus accessibles.
Types link	Au moyen des types lien, différents objets peuvent être reliés entre eux dans Vertec.
Paramétrage des listes	Toutes les informations qui caractérisent un objet Vertec peuvent être affichées dans des listes multicolonnées permettant un aperçu de toutes les données.
Conditions de validité	Les règles présentes dans dans Vertec peuvent être définies et vérifiées automatiquement grâce aux conditions de validité.
Champs et classes supplémentaires	Existe-t-il des informations supplémentaires, associées à un objet, que vous souhaiteriez enregistrer ? Ou y a-t-il dans vertec des objets qui n'existent pas ? Grâce aux champs ou aux classes supplémentaires, vous pouvez sans problème représenter le déroulement de vos processus.

En plus des éléments mentionnés ci-dessus, il existe une multitude d'autres possibilités de configurer et d'automatiser Vertec afin que vous disposiez d'un produit taillé pour vous. Les instruments de cette personnalisation vous sont présentés dans ce cours.

### 1.1 Buts du cours

- Vous connaissez les options techniques déterminantes pour le paramétrage de Vertec
- Vous êtes en mesure d'identifier quel instrument est le plus adapté à une exigence définie
- Vous apprenez à élaborer vous-même des expressions OCL.

## 2. Courte introduction dans OCL

Les possibilités de configuration (customizing) sont marquées par le langage d'interrogation **OCL** (Object Constraint Language). Ce langage permet d'accéder aux données Vertec sans pour autant connaître les détails techniques du logiciel. Les requêtes (formulation d'interrogations) sont appelées des **Expressions OCL**.

Afin de faciliter la saisie des expressions OCL dans Vertec, un **Editeur d'expression OCL** est à disposition. A l'aide de celui-ci, vous pouvez rechercher et assembler les expressions désirées à partir du modèle de données de Vertec.

Si vous souhaitez par exemple afficher tous les projets dans un dossier. Cliquez sur la touche droite de la souris en pointant Dossiers publics et sélectionnez **Nouveau > Dossier à expression**.

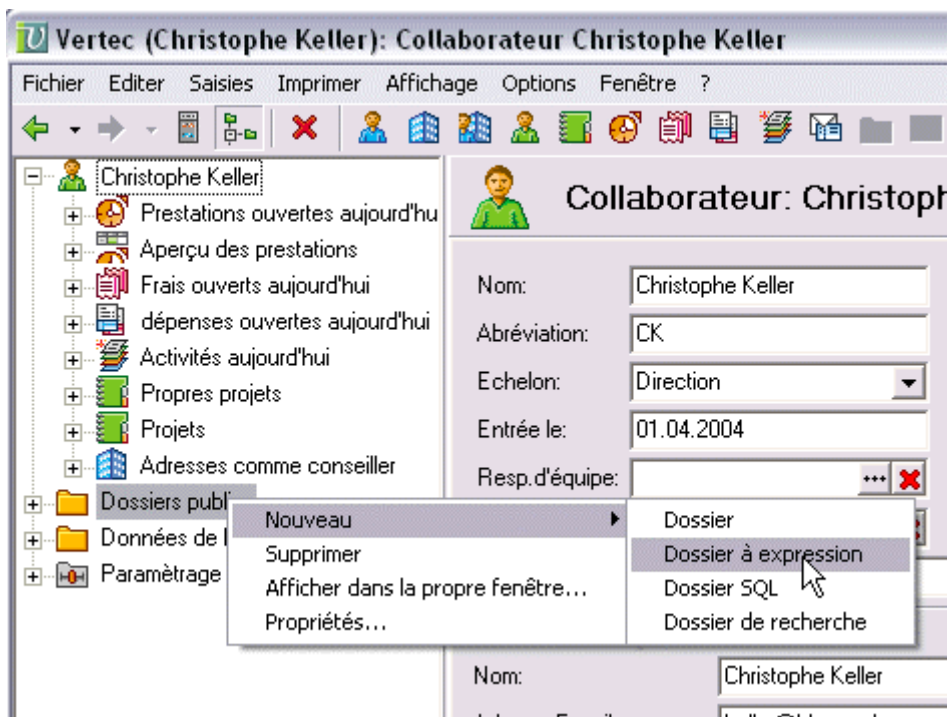


Figure 1 La création d'un dossier à expression

La fenêtre des propriétés du nouveau dossier apparaît. Nommez le dossier **Tous les projets**.

A côté du champ **Expression** cliquez sur le bouton avec les 3 points, l'éditeur d'expressions OCL s'ouvre:

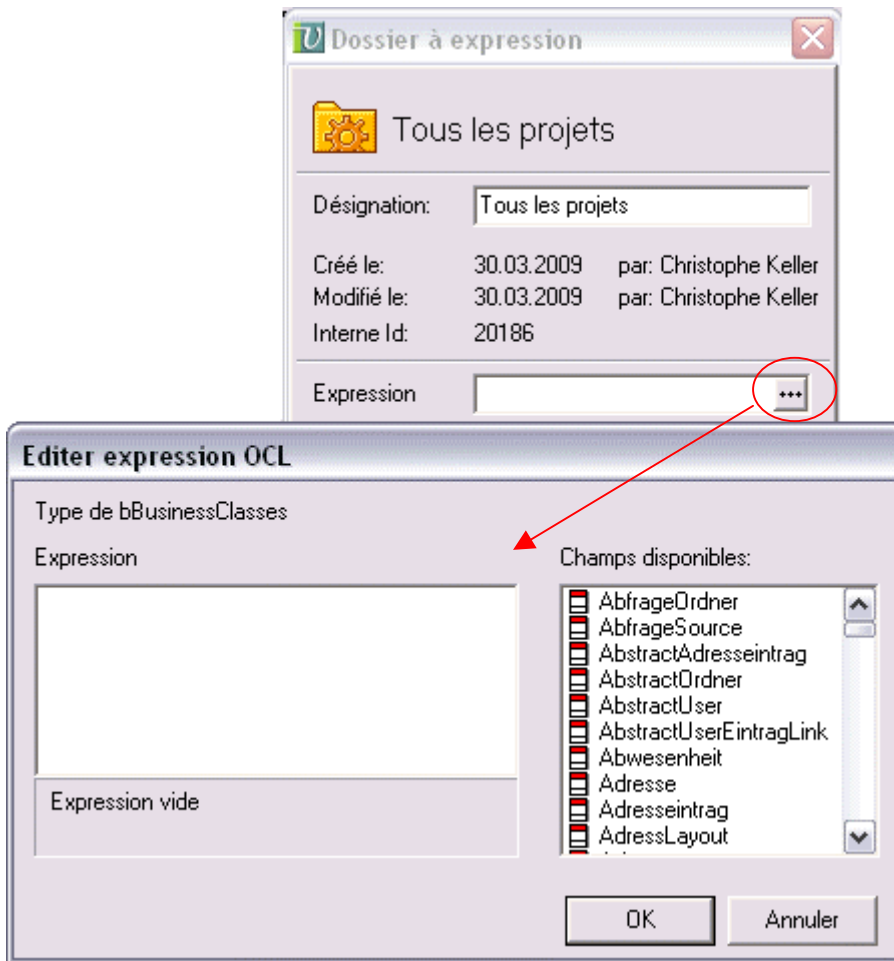


Figure 2 L'éditeur d'expressions OCL

Ecrivez `projekt` dans le champ et l'éditeur d'expression vous indique que l'expression est OK et qu'il s'agit d'une liste de projets : `Collection(Projekt)`.

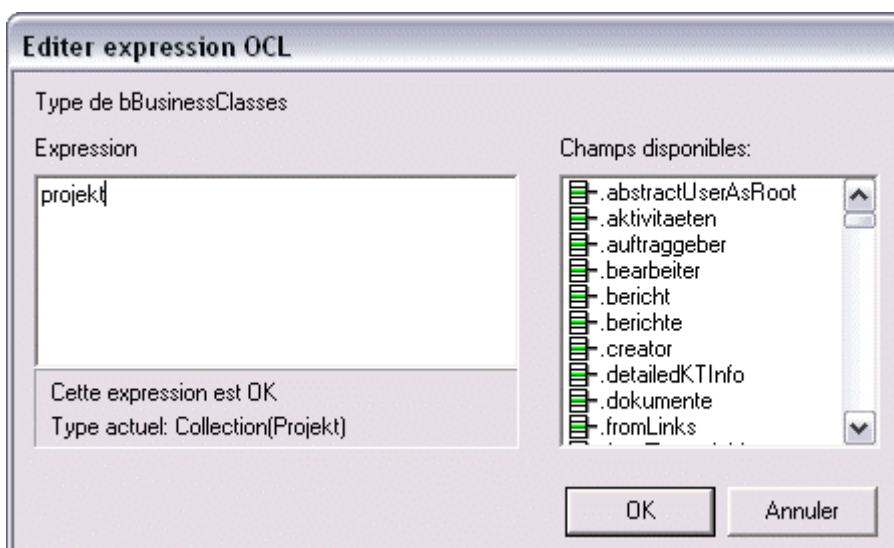


Figure 3 Cette expression est correcte

Une expression interprète toujours *tous d'un type*. C'est pourquoi, dans le cas de l'expression `pro-`

jekt, même si l'on écrit l'unité, on sous-entend *tous du type* projet, c'est-à-dire une liste de projets.

<b>Astuce</b>	Les expressions OCL sont écrites en minuscules.
---------------	---

C'est fait. Cliquez sur OK et refermez la fenêtre des propriétés. A présent, si vous sélectionnez le dossier dans l'arborescence, la liste des projets apparaît :

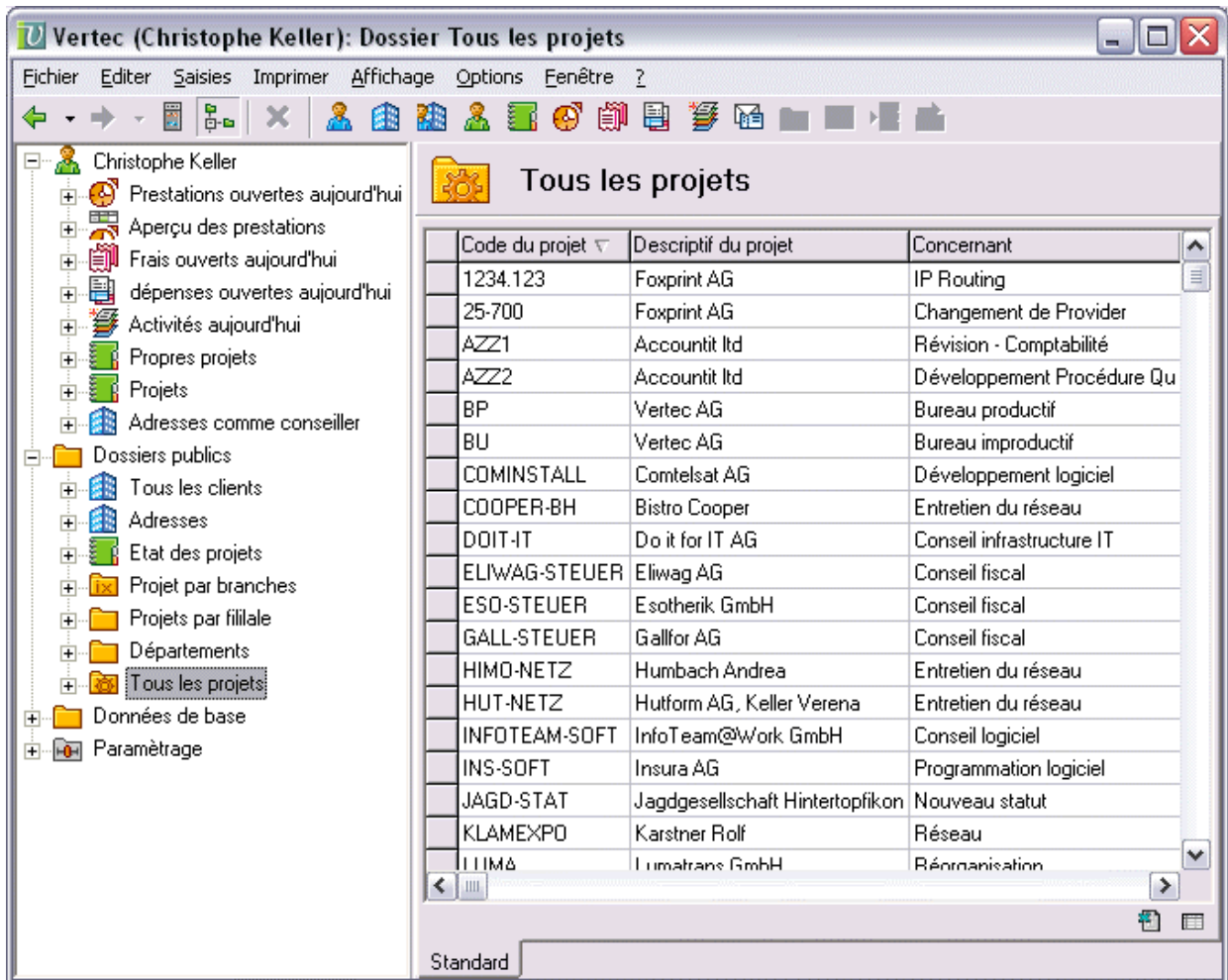


Figure 4 La dossier nouvellement créé affiche tous les projets

L'expression OCL de cet exemple est libellée **projeckt**. C'est la forme la plus simple d'une expression OCL. Les expressions OCL peuvent être simple ou hautement complexes; on peut en effet faire des sommes, comparer, extraire, trier et faire bien d'autres choses avec OCL.

Vous avez à présent un aperçu des propriétés et des fonctions les plus importantes de OCL. Celles-ci seront plusieurs fois utilisées durant le déroulement du cours. Vous pouvez donc lentement apprivoiser ce langage de requête.

## 2.1 Le modèle de classes de Vertec

Pour cette petite introduction, nous partons d'un extrait du modèle de classes de Vertec. Un modèle de classe décrit les propriétés des différents types de données (**classes**), les liens entre les données (**links**) et les champs existants (**Members**).

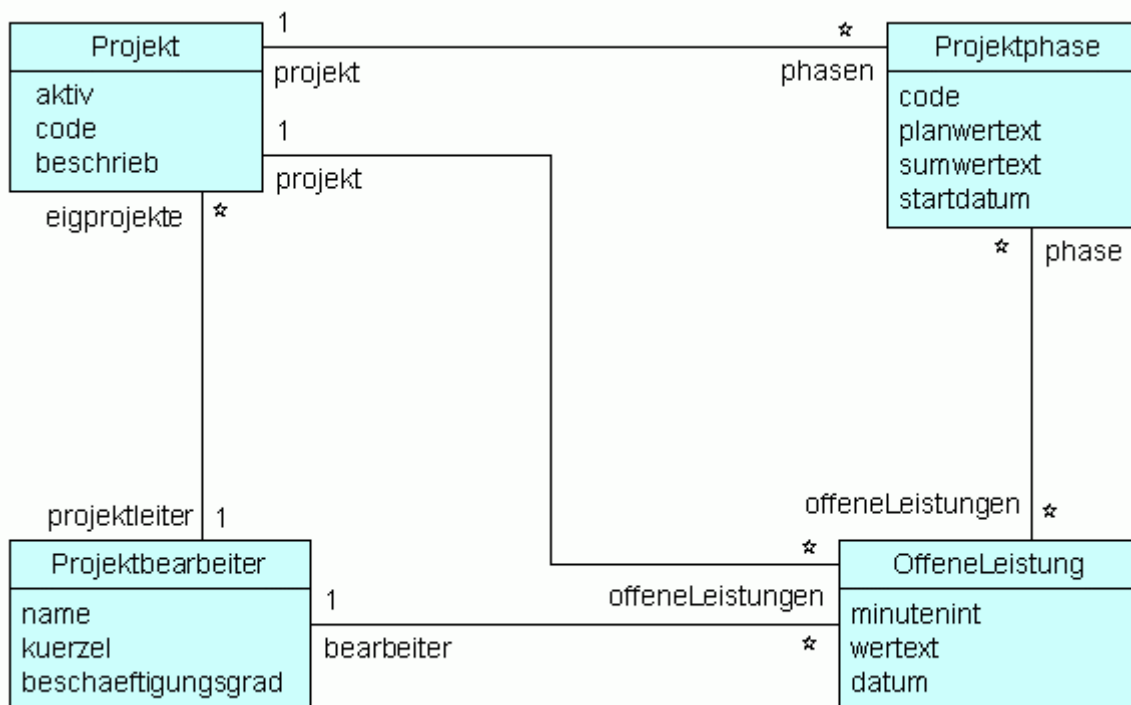


Figure 5 Extrait du modèle de classes Vertec

Ce graphique (diagramme des classes) se lit comme suit : un projet/Projekt possède les champs (members) **aktiv**, **code**, **description** (entre beaucoup d'autres). Un projet peut avoir des liens et liaisons avec plusieurs **phases** (celles-ci sont du type `Projektphase`), tandis que les phases ont des liens vers plusieurs **prestations ouvertes**. Une prestation ouverte (nom de classe `OffeneLeistung`) a des liens avec **Projekt** (une prestation ne peut être attribuée qu'à un seul projet) et avec **Projektbearbeiter** (une prestation ne peut être attribuée qu'à un seul collaborateur). Les liens correspondants s'appellent **projekt** et **bearbeiter**.

Dans OCL

```
Projekt
```

Signifie le type de données **Projekt** (projet, la classe **Projekt**), tandis que

```
projekt
```

la liste de tous les projets, qui existent dans le système. Dans OCL, c'est la première lettre qui décide s'il s'agit d'un type (majuscule) ou d'un membre, lien ou d'une liste (minuscule). Que les autres caractères soient des majuscules ou minuscules ne joue aucun rôle.

La navigation dans les liens se fait très simplement dans OCL. Ainsi l'expression OCL

```
projekt.projektleiter
```

donne comme résultat une liste de tous les chefs de projets. (techniquement, on navigue d'une liste de projets vers les chefs de projets, c'est pour cela que le résultat est une liste).

Les expressions OCL ne sont pas seulement exécutées sur un système Vertec, mais également sur les différents objets (ceci est par exemple utilisé pour les colonnes des listes, voir aussi chapitre 2). L'expression

```
projektleiter
```

sur un seul projet donne le **Chef de projet** (un seul objet, pas de liste d'objets).

## 2.2 Les opérateurs OCL importants

La recherche de données spécifiques dans OCL se fait avec l'opérateur **->select('critère')**. Ainsi l'expression suivante cherche tous les projets actifs à l'intérieur des projets. (**aktiv** est le membre **aktiv** sur le type de données **Projekt**):

```
projekt->select(aktiv)
```

Le tri de la liste se fait à l'aide de l'opérateur **->orderBy('Expression')**. L'expression suivante donne une liste des projets actifs, triés selon le code de projet:

```
projekt->select(aktiv)->orderBy(code)
```

L'opérateur **->select** donne toujours une liste. Vous pouvez compter le nombre d'objets dans la liste à l'aide de l'opérateur **->size**. L'expression suivante indique le nombre de projets actifs :

```
projekt->select(aktiv)->size
```

Maintenant que nous connaissons l'opérateur **->size**, l'expression pour tous les chefs de projets pourrait aussi être formulée comme suit:

```
projektbearbeiter->select(eigprojekte->size>0)
```

D'autres opérateurs importants existent sur des listes : **->first**, **->last**, **->sum**. L'expression suivante donne le code du premier projet de l'alphabet (trié selon code de projet):

```
projekt->select(aktiv)->first.code
```

**->sum** est utilisé pour effectuer la somme de valeurs numériques sur des listes. Ainsi l'expression

```
phasen.planwertext->sum
```

exécutée sur un projet donne la somme des budgets (Honoraires) de toutes les phases du projet et ainsi le budget du projet. On peut aussi calculer le budget de toutes les phases actives du projet:

```
phasen->select(aktiv).planwertext->sum
```

La manipulation avec des valeurs de dates est plus simple, si l'on peut mettre quelque chose en relation avec la date du jour. L'expression

```
phasen->select(startdatum>=now)->size
```

indique le nombre de phases d'un projet, dont la date de démarrage est dans le futur (**now** donne la date système). La façon d'utiliser une date fixe, à la place de **now**, est mentionnée dans l'exemple suivant :

```
phasen->select(startdatum>encodeDate(2006,12,31))->size
```

D'autres opérateurs sont décrits de manière détaillée dans le chapitre 9.

## 2.3 Exercices

*Exercice 1*      *Créez un dossier à expression affichant tous les projets actifs*

*Exercice 2*      *Créez un dossier à expression affichant toutes les phases de tous les projets actifs. Référez-vous pour cela au modèle de la Figure 5 à la page **Fehler! Textmarke nicht definiert.***

*Exercice 3*      *Créez un dossier à expression avec tous les collaborateurs des projets, qui sont désignés comme chef de projet dans au moins un projet. Référez-vous pour cela au modèle de la Figure 5 à la page 8 (Astuce :utiliser `->size`)*

### 3. Réglages de listes

Un des réglages très importants dans Vertec est la possibilité d'adapter les listes des objets.

Voici quelques exemples de listes:

Cette liste est un dossier appelé Link-Container. Celui-ci affiche des données qui sont en relation avec d'autres données, dans l'exemple les prestations d'un collaborateur d'une journée précise. Si l'on modifie les réglages d'un Link-Container, tous les mêmes Link-Container sont adaptés. (si l'on enlève de l'exemple **prestations ouvertes aujourd'hui** la colonne **prestation**), elle ne serait plus visible chez tous les collaborateurs). La configuration de Link-Container via des types de liens est décrit dans le chapitre 5.

Projet	Phase	Date	Collaborateur	Prestation	Texte	Heures	Taux	Honoraires
AZZ1	CONS	28.04.2008	CK	ADM	Lettre	1:00	350.00	350.00
1-TRASTA	1 - PLANNING	28.04.2008	CK	CONCEPT	nouvelle version	4:00	450.00	1'800.00
*		28.04.2008	CK					

Figure 6 Exemple d'un Link-Container

L'autre exemple affiche un dossier. Un dossier représente des données qui possèdent plus ou moins les mêmes attributs - dans l'exemple tous les projets actifs. Si le réglage de liste d'un tel dossier est changé, il n'est valable que pour ce dossier. Vous pouvez par exemple compléter un dossier comportant une liste de projets avec des colonnes d'analyses (p.ex. horaires ouverts par projet) tandis qu'un autre dossier ne contient que les informations les plus importantes comme le code de projet etc. La manière de configurer le dossier est décrite dans le chapitre 4.

Code du projet	Descriptif du projet	Concernant	Type du projet	Chef de projet
1-TRASTA	Trasta AG	Installation IT	Projet comm.	PT
12	Trasta AG	Divers questions	projet interne	CK
123	Hostettler Informatique	Divers	projet interne	Admin
1234.123	Foxprint AG	IP Routing	Projet comm.	PT
25-700	Foxprint AG	Changement de Provider	Projet comm.	CK
AZZ1	Accountit ltd	Révision - Comptabilité	Projet comm.	PT
AZZ2	Accountit ltd	Développement Procédure Qu	Projet comm.	WS
BP	Vertec AG	Bureau productif	projet interne	WS
BU	Vertec AG	Bureau improductif	projet interne	WS
COMINSTALL	Comtelsat AG	Développement logiciel	Projet comm.	CK
COOPER-BH	Bistro Cooper	Entretien du réseau	Projet comm.	PT
DDIT-IT	Do it for IT AG	Conseil infrastructure IT	Projet comm.	WS
ELIWAG-STEUER	Eliwag AG	Conseil fiscal	Projet comm.	WS

Figure 7 Liste de tous les projets actifs

Comme mentionné, les colonnes de cette liste peuvent être adaptées individuellement. Chaque liste dans Vertec possède 2 icônes en dessous de l'affichage de la liste:



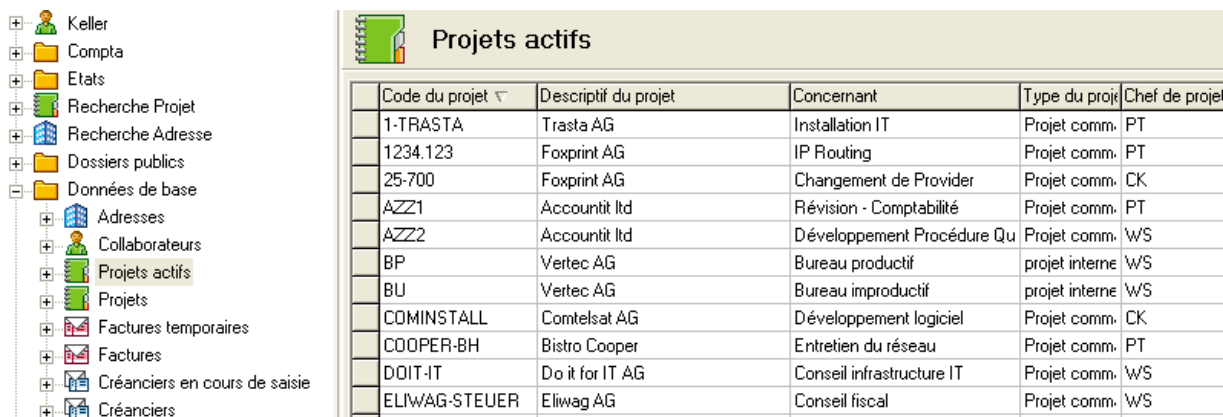
Figure 8 Les icônes de configuration de la liste

1. Ce bouton permet de transférer le contenu de la liste directement dans Excel (transférer liste dans Excel).
2. Ce bouton permet d'éditer la liste (Réglage du tableau).

### 3.1 Editer l'affichage des listes

Nous avons comme tâche d'ajouter une colonne qui affiche des données du chef de projet dans une liste de projets.

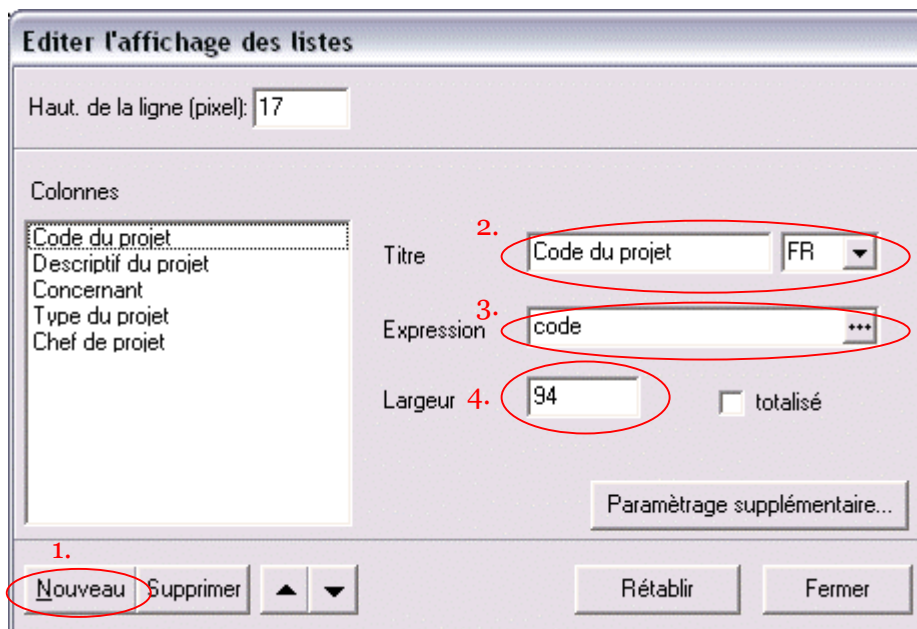
Si l'on fait afficher les projets actifs dans Vertec, cela peut se présenter comme ceci :



Code du projet	Descriptif du projet	Concernant	Type du projet	Chef de projet
1-TRASTA	Trasta AG	Installation IT	Projet comm.	PT
1234.123	Foxprint AG	IP Routing	Projet comm.	PT
25-700	Foxprint AG	Changement de Provider	Projet comm.	CK
AZZ1	Accountit ltd	Révision - Comptabilité	Projet comm.	PT
AZZ2	Accountit ltd	Développement Procédure Qu	Projet comm.	WS
BP	Vertec AG	Bureau productif	projet interne	WS
BU	Vertec AG	Bureau improductif	projet interne	WS
COMINSTALL	Comtelsat AG	Développement logiciel	Projet comm.	CK
COOPER-BH	Bistro Cooper	Entretien du réseau	Projet comm.	PT
DOIT-IT	Do it for IT AG	Conseil infrastructure IT	Projet comm.	WS
ELIWAG-STEUER	Eliwag AG	Conseil fiscal	Projet comm.	WS

Figure 9 Liste standard des projets actifs

Après un clic sur le bouton **Réglage du tableau** la fenêtre suivante apparaît:



Editer l'affichage des listes

Haut. de la ligne (pixel): 17

Colonnes

- Code du projet
- Descriptif du projet
- Concernant
- Type du projet
- Chef de projet

Titre 2. Code du projet FR

Expression 3. code

Largeur 4. 94  totalisé

Paramétrage supplémentaire...

1. Nouveau Supprimer ▲ ▼ Rétablir Fermer

Figure 10 Dialogue configuration de la liste

1. Le bouton **Nouveau** permet d'ajouter une colonne
2. La désignation dans le champ **Titre** est plus tard le titre de la colonne. Les titres des colonnes peuvent être saisis dans les différentes langues (DE, EN, FR, IT). Dans le menu déroulant situé à côté du champ *Titre* on peut choisir à quelle langue correspond le libellé du titre.
3. Vous pouvez choisir ici l'expression OCL souhaitée à l'aide de l'éditeur ou la taper
4. Permet de définir la largeur de la colonne.

Créez une nouvelle colonne et activez là dans la partie gauche. Par un clic sur le bouton avec les 3 points à la fin du champ **Expression** l'éditeur OCL s'ouvre (Expression-Editor):

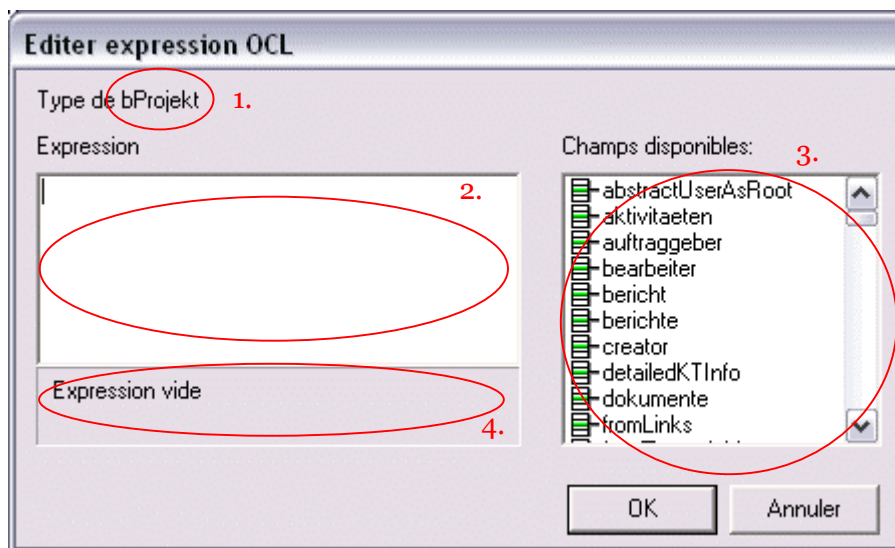


Figure 11 L'éditeur d'expressions OCL

L'éditeur d'expressions OCL se compose des champs suivants

1. Le type de base est indiqué. L'expression se réfère à ce type, dans l'exemple **Projekt**
2. Ce champ permet d'éditer l'expression OCL, soit en la tapant directement soit par un double-clic sur l'expression dans la liste déroulante dans la partie de droite.
3. La liste **Verfügbare Felder/champs disponibles** indique les Links, Members et opérations qui sont disponibles sur le type actuel.
4. Ce champ affiche l'information si l'expression est valable ou pas.

Dans la partie de droite, les champs disponibles sont affichés comme suit :

vert	Tous les liens vers d'autres objets
turquoise	Members
jaune	opérations

Nous trouvons alors le **chef de projet** (projektleiter) dans le domaine **vert**, tandis que le champ **actif** (aktiv) doit se trouver dans la partie **turquoise**:

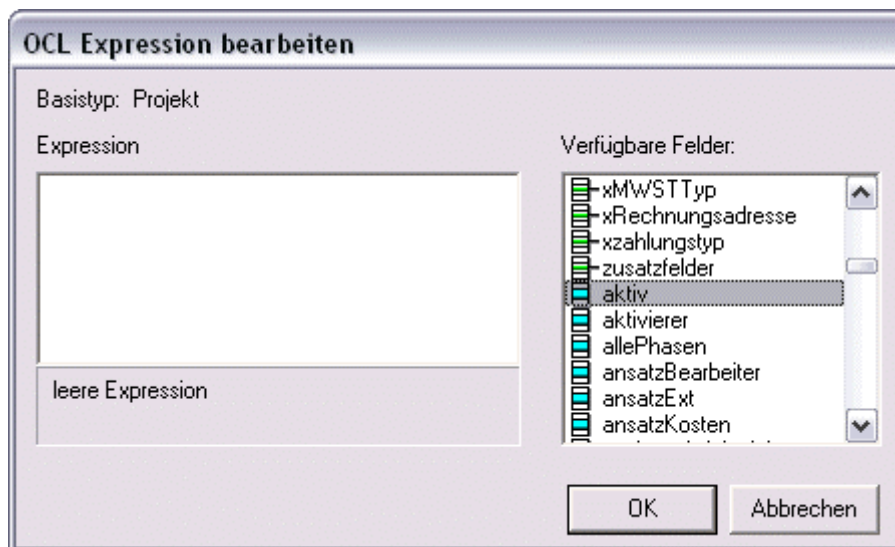


Figure 12 La champ **aktiv** se trouve parmi les Members

Si l'on sélectionne le **chef de projet(projektleiter)** avec un double-clic ou l'on tape l'expression **proj-  
jektleiter** directement dans le champ, l'expression se présente comme suit:

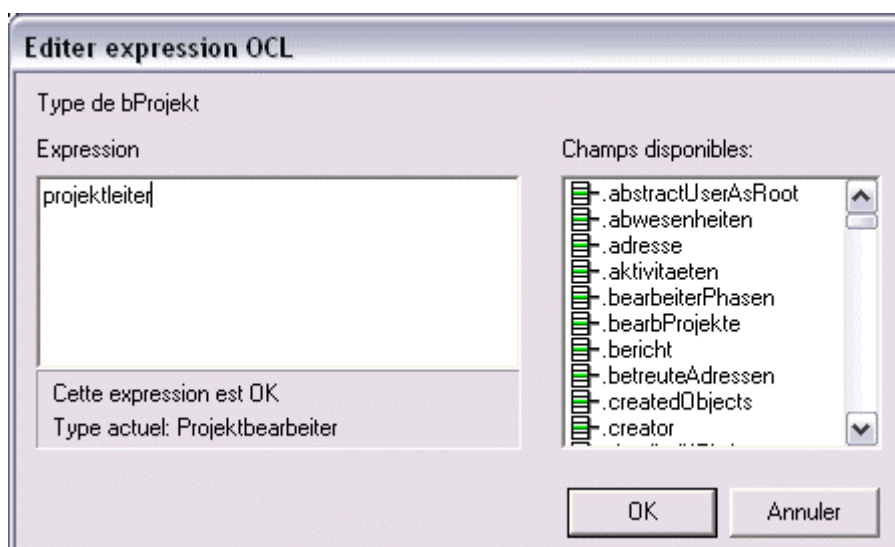


Figure 13 Une expression OCL est construite

L'éditeur annonce que l'expression est OK et que le type actuel est Projektbearbeiter (Collaborateur). La partie de droite affiche maintenant les champs du Collaborateur, entre autres, les **presta-  
tions ouvertes (offene Leistungen)**. Si l'on double-clic sur **offeneLeistungen**, l'expression se présente  
comme suit:

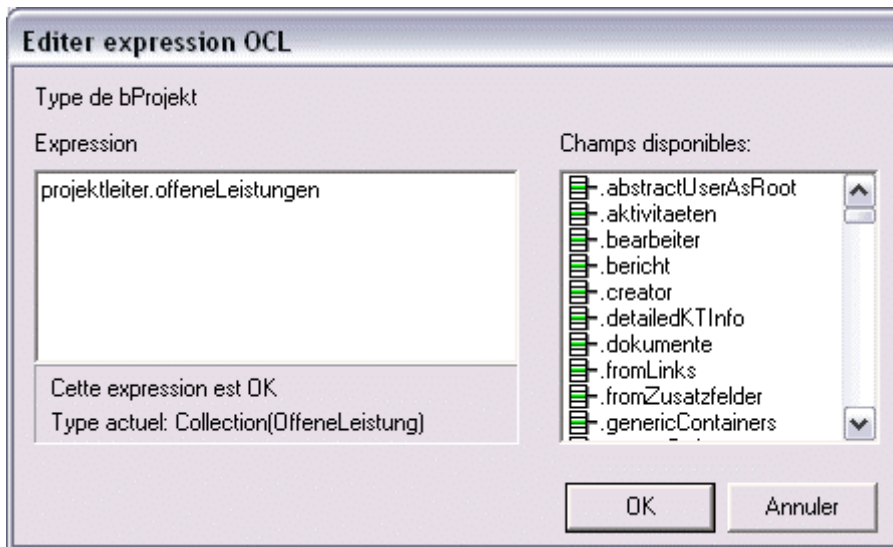


Figure 14 Prestations ouvertes du chef de projet

L'éditeur annonce maintenant que le type actuel est une **Collection** (liste) des prestations ouvertes, et affiche à droite le champ du type OffeneLeistung/prestation ouverte. Nous choisissons dans la liste **wertext** (à trouver dans les Membres turquois) ou nous tappons le texte **.wertext** directement à la suite de l'expression existante. L'expression s'appelle alors :

```
projektleiter.offeneLeistungen.wertext
```

L'éditeur affiche alors maintenant:

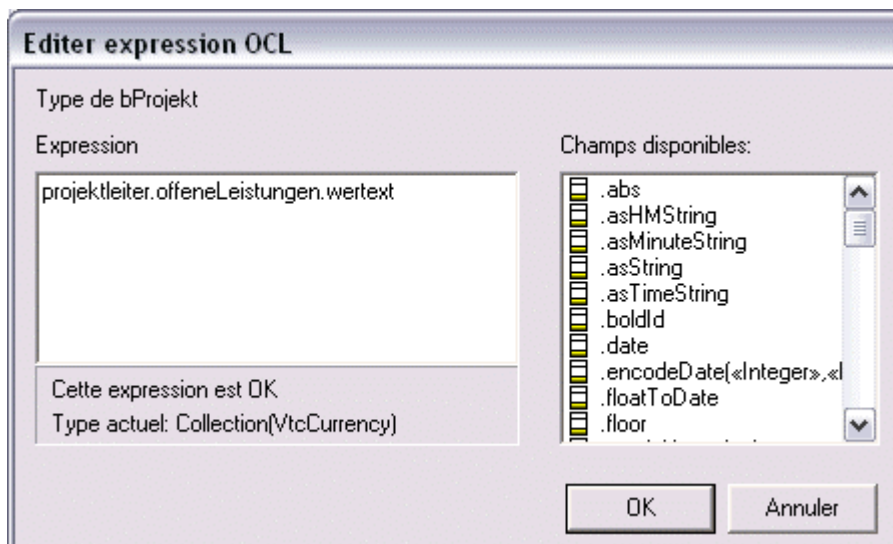


Figure 15 Expression avec une valeur numérique

Le type actuel est une liste de valeurs numériques (Currency) et la partie de droite n'affiche plus que des éléments jaunes (opérateurs).

Choisissez maintenant **->sum** et vous obtenez l'image suivante:

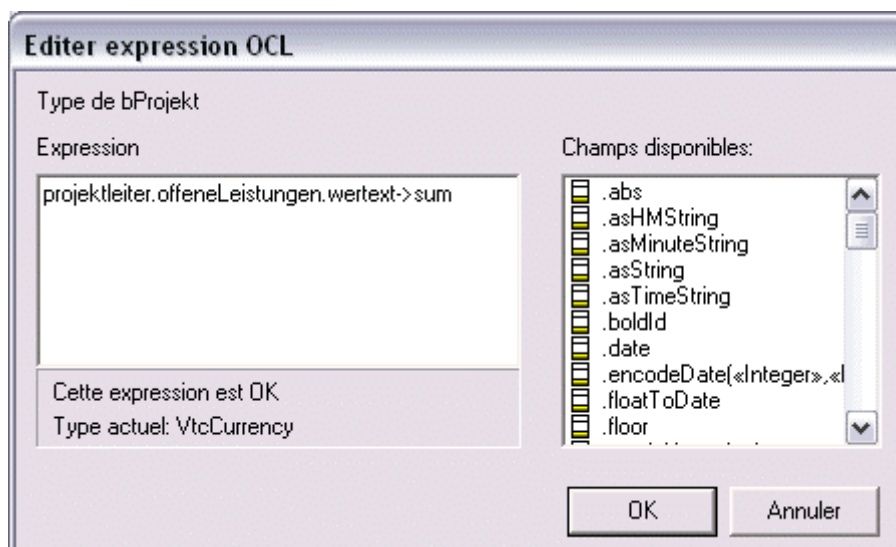


Figure 16 Expression avec un opérateur de sommes

Le type actuel est une valeur numérique. Si vous cliquez sur **OK**, l'expression créée est transférée dans la configuration de la liste.

Qu'est-ce que fournit cette expression OCL maintenant? Elle affiche par projet la somme de toutes les prestations ouvertes du chef de projet. Ceci n'est peut-être pas très utile, mais montre ce qui est possible. (une expression similaire vous attend dans l'Exercice 5).

L'éditeur se présente maintenant comme suit:

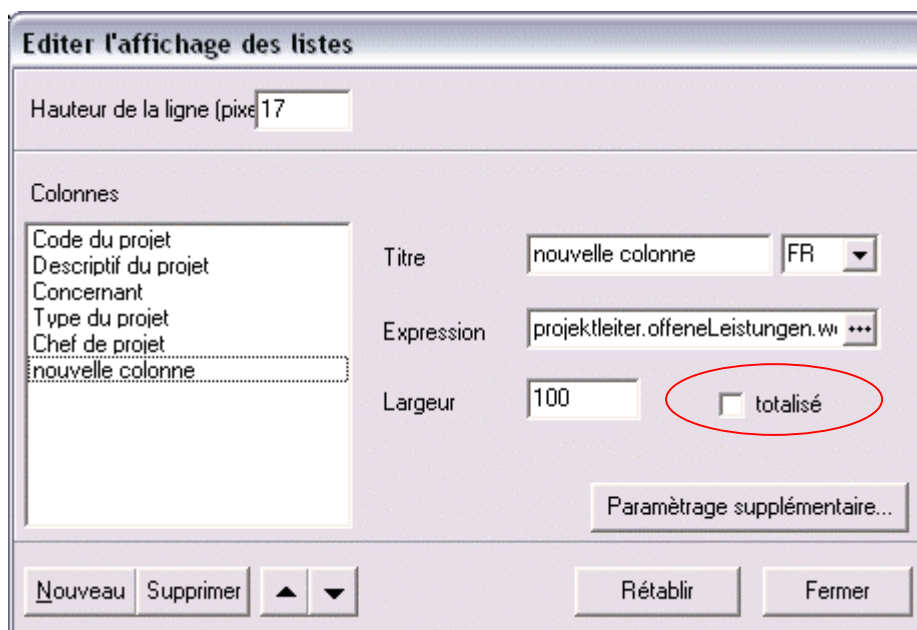


Figure 17 Nouvelle colonne dans l'éditeur

Si vous cliquez sur **totalisé**, la colonne correspondante est totalisée.

Si vous cliquez sur **paramètre supplémentaire...**, les possibilités de paramétrage de cette colonne sont affichées:

### 3.2 Paramétrage supplémentaire du tableau

Figure 18 Paramétrages supplémentaires

Les termes **Nom élément/contrôle** et **Nom du renderer** seront expliqués plus bas dans le présent chapitre.

Alignement

0: aligné à gauche  
1: aligné à droite  
2: centré

Tri

Vous définissez si la liste doit être triée selon cette colonne (ceci est la valeur par défaut lors du premier affichage de la liste). Même en cas d'indication d'un tri, la liste peut être triée par un double-clic sur le titre de chaque colonne.

Arrêt du tabulateur

Indique s'il y a un arrêt dans cette colonne si l'on navigue à l'aide de la touche de tabulation. Très important dans la liste de saisie des prestations.

Seulement lire (read only)

Indique si la colonne peut être éditée ou pas. En cas de sélection de 'seulement lire', les données de cette colonne ne peuvent pas être modifiées.

Couleur de fond.

Vous pouvez indiquer la manière de formater cette colonne. Un choix de couleurs de fond est parfois utile pour faire ressortir certaines informations.

Type

Le type doit être introduite, p. ex. "Arial" (voir dans **Fonts** sous **Police** ou p. ex. dans Word)

- Couleur de texte** La couleur de la police peut être sélectionnée dans la palette des couleurs. La couleur standard se réfère aux paramètres de Windows. Si cette valeur est modifiée, la couleur choisie sera utilisée dans tous les cas.
- Taille de texte** La taille standard est 8. Vous pouvez modifier la taille selon vos désirs. Tenez compte du fait que des tailles plus grandes ne sont en partie plus lisibles.
- Style de texte**
- 0 : Bold (gras)
  - 1 : italique
  - 2 : souligné
  - 3 : barré
- Ces valeurs peuvent également être combinées, p. ex.  
0,1 : bold-italique

Les champs supérieurs se réfèrent au texte dans les listes. La partie inférieure "Titre" se réfère à l'en-tête de la colonne.

Si vous choisissez pour cette colonne l'alignement **1**, la liste des projets se présente comme suit :

Projets actifs						
Code du projet ▾	Descriptif du projet	Concernant	Type du projet	Chef de projet	Prestations ouv.	
1-TRASTA	Trasta AG	Installation IT	Projet comm.	PT		43'876.75
1234.123	Foxprint AG	IP Routing	Projet comm.	PT		43'876.75
25-700	Foxprint AG	Changement de Provider	Projet comm.	CK		128'998.20
AZZ1	Accountit Ltd	Révision - Comptabilité	Projet comm.	PT		43'876.75
AZZ2	Accountit Ltd	Développement Procédure Qu	Projet comm.	WS		9'450.00
BP	Vertec AG	Bureau productif	projet interne	WS		9'450.00
BU	Vertec AG	Bureau improductif	projet interne	WS		9'450.00
COMINSTALL	Comtelsat AG	Développement logiciel	Projet comm.	CK		128'998.20
COOPER-BH	Bistro Cooper	Entretien du réseau	Projet comm.	PT		43'876.75
DOIT-IT	Do it for IT AG	Conseil infrastructure IT	Projet comm.	WS		9'450.00
ELIWAG-STEUER	Eliwag AG	Conseil fiscal	Projet comm.	WS		9'450.00
ESD-STEUER	Esotherik GmbH	Conseil fiscal	Projet comm.	Keller		0.00
GALL-STEUER	Gallfor AG	Conseil fiscal	Projet comm.	CK		128'998.20
HIMO-NETZ	Humbach Andrea	Entretien du réseau	Projet comm.	Keller		0.00
HUT-NETZ	Hutform AG, Keller Verena	Entretien du réseau	Projet comm.	WS		9'450.00
INFOTEAM-SOFT	InfoTeam@Work GmbH	Conseil logiciel	Projet comm.	Keller		0.00
INS-SOFT	Insura AG	Programmation logiciel	Projet comm.	Keller		0.00
JAGD-STAT	Jagdgesellschaft Hintertopfikon	Nouveau statut	Projet comm.	Keller		0.00
KLAMEXPO	Karstner Rolf	Réseau	Projet comm.	CK		128'998.20
LUMA	Lumatrans GmbH	Réorganisation	Projet comm.	PT		43'876.75
MR-NFT7	Meuer Blatter Rechtsanwälte	Conseil en réseau	Projet comm.	PT		43'876.75

Standard

Figure 19 Liste avec une nouvelle colonne, alignée à droite

La largeur de la colonne peut être réglée à l'aide de la souris ou dans les paramètres supplémentaires. (attention, tous les utilisateurs peuvent modifier la largeur de colonne, mais seulement la modification par l'administrateur la fige).

### Nom élément/contrôle

**Paramétrage supplémentaire du tableau**

Nom élément/contrôle  Control Expression

Nom du renderer

Alignement  Représentation

Tri  pas trié  croissant  décroissant

Arrêt du tabulateur  seulement lire

Figure 20 Indiquer un élément de contrôle

Le champ **nom élément/contrôle** permet de saisir un élément de contrôle, dans l'exemple l'élément avec le nom **cmbPhase** pour la colonne **Phase** de la liste de prestations. Le résultat est que cet élément de contrôle s'affiche dans la liste :

Projet	Phase	Date	Collaborateur	Prestation	Texte	Heures	Taux	Honoraires
COMINSTALI	1 ANA	28.04.2008	Keller			0:00	180.00	0.00
	1 ANA			Analyse				
	2 IMPLEMENTATION			Implémentation				
	3 FINAL			Phase finale				

Figure 21 L'élément de contrôle dans la liste

Puisque les listes de Vertec sont déjà pré-configurées, le contenu de ce champ n'est que rarement modifié. Une liste de tous les éléments de contrôle est contenue dans le manuel resp. dans la documentation online sur le site (<http://www.vertec.ch/support>) sous **Customizing > Rendre-r/Steuer-elemente**

### Control Expression

Il est parfois nécessaire de donner dans ce champ une expression spéciale pour un élément de commande. La façon de faire est indiquée pour chaque cas dans la description des éléments de commande dans la documentation Online (voir plus haut).

## Nom du renderer

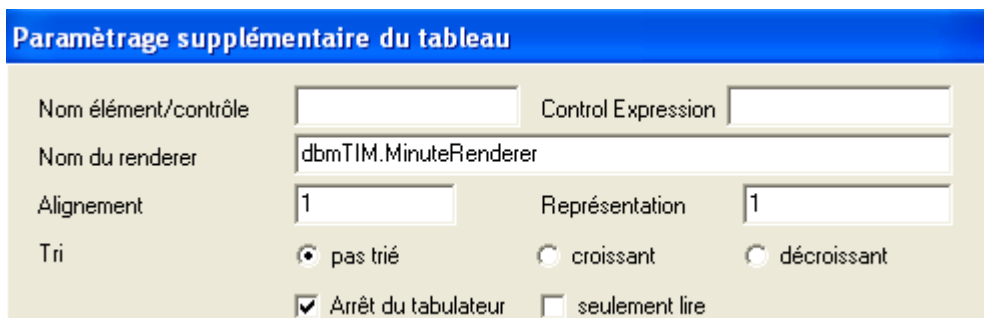


Figure 22 Indiquer un nom d'un renderer

Le champ **Nom du renderer** permet d'indiquer un renderer. Un renderer représente des données Vertec internes de manière différente à la l'écran. Ainsi les temps saisis d'une prestation (**minute-nint**) sont enregistrés dans Vertec en minutes. Le renderer **dbmTIM.MinuteRenderer** assure que ces valeurs soient affichées chez l'utilisateur en fonction des réglages individuels (Heures:Minutes, heures.décimales etc.) . Voici une liste des renderer usuels :

dbmTIM.MinuteRenderer	Affiche une valeur Integer comme valeur de temps, correspondant aux paramètres système.
dbmTIM.TwoDigitCurrencyRenderer	Affiche un chiffre comme chiffre en devises, avec séparateur de milliers et 2 positions après la virgule.
dbmTIM.ProjektEintragWertRenderer	Renderer pour le Member <b>wertext</b> de prestations, frais et dépenses.

Une liste complète des renderer disponibles se trouve dans le manuel ou dans la documentation en ligne. (<http://www.vertec.ch/support>) sous **Customizing > Renderer/Steuer-elemente**.

### 3.3 Exercices

- Exercice 4* Affichez sur une liste de collaborateurs (p.ex. **Collaborateurs** sous **Données de base**) le taux d'occupation. Comment pouvez-vous voir l'occupation totale dans votre entreprise ?
- Exercice 5* Créez une liste de projets avec une somme des prestations ouvertes par projet.
- Exercice 6* Créez une liste **propres projets** d'un collaborateur (cette liste indique les projets dans lesquels le collaborateur est chef de projet) une nouvelle colonne qui indique le budget restant du projet (conseil: budget restant (ou réserve de travail) est la différence entre le budget et les prestations déjà saisies).
- Exercice 7* Sur une liste de collaborateurs, créez une colonne qui indique le nombre de projets actifs dans lesquels le collaborateur est chef de projet. Référez-vous pour cela au modèle présenté dans la Figure 5 à la page 8.
- Exercice 8* Sur une liste de projets, créez une colonne qui affiche par projet les honoraires ouverts du chef de projet. Référez-vous pour cela au modèle présenté dans la Figure 5 à la page 8. Conseil: l'expression OCL **self** référence le point de départ (**self.offeneLeistungen...**).

## 4. Dossiers

Des dossiers dans Vertec représentent des pots ou listes d'objets qui ont des choses en commun. Les dossiers sont comparables avec des répertoires de données ou dossiers dans Outlook avec une différence importante: Vertec peut afficher le même objet dans différents dossiers. Le dossier ne contient qu'une référence vers l'objet (il n'y a donc pas de copie de l'objet).

Vertec connaît 4 types de dossiers, qui sont décrit ci-après.

### 4.1 Dossiers manuels

L'utilisation d'un dossier manuel (**dossier**) s'impose si Vertec n'est pas en mesure de calculer le contenu du dossier de manière automatique. Ce genre de dossier contient typiquement des **rubriques**, **catégories** etc.. Un objet fait partie de ce dossier, s'il a été ajouté manuellement.

Un nouveau dossier peut être créé à l'aide d'un clic droit sur un dossier existant :

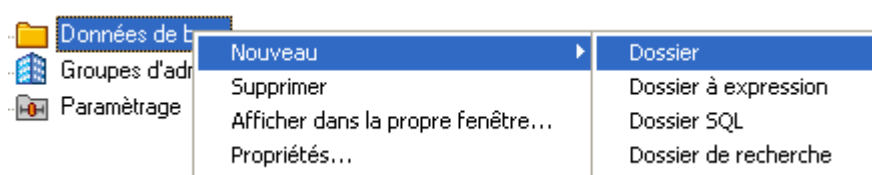


Figure 23 Créer un nouveau dossier

Par un clic droit sur **Dossier**, un nouveau dossier manuel est créé. Le masque suivant apparaît :



Figure 24 Les propriétés du dossier

**Désignation:** le dossier apparaîtra sous ce nom.

**Classes:** vous pouvez indiquer les types de données (classes) qui peuvent être transférés dans ce dossier. Dans la plupart des cas, cela ne fait pas de sens de mélanger les types de données dans un dossier. Dans l'exemple, nous aimerions créer un nouveau département, nous choisissons donc

**Projektbearbeiter/collaborateur** et cherchons le choix de classes les plus usuelles à l'aide du bouton avec les 3 points.

**Sous-dossiers:** Si l'on crée des structures de dossiers, on peut indiquer sur le dossier supérieur, quelles classes peuvent contenir les sous-dossiers. Lors de la création d'un tel sous-dossier, les **classes** sont automatiquement sélectionnées.

**Index:** l'index permet de déterminer la position dans l'arborescence. Vertec tri les dossiers selon l'index croissant.

**Icône:** toutes les icônes dans Vertec possèdent un numéro qui est documenté dans la knowledge-base. Si vous n'avez pas accès à cette base, vous pouvez copier le numéro depuis un dossier existant. Si vous voulez utiliser une autre icône que la standard (-1), vous pouvez indiquer l'index de cette icône (ce réglage n'est activé qu'après redémarrage de Vertec).

**Permettre sous-dossiers:** si vous voulez créer une structure de données qui doit avoir plusieurs niveaux et que le dossier doit contenir des sous-dossiers, vous devez activer permettre sous-dossiers.

**Dossier à rubriques:** Permet de sélectionner simplement des enregistrements. Voir aussi le chapitre. **Dossiers comme dossiers à rubriques** voir plus loin dans le texte.

**Propre liste pour chaque type:** Si le dossier contient des enregistrements de différents types, (classes), on peut afficher une propre liste pour chaque type. Celles-ci apparaissent alors sur des onglets séparés. Les listes peuvent ainsi être affichées séparément.

Dès que vous avez terminé les réglages, le nouveau dossier apparaît dans la structure. Les enregistrements autorisés peuvent maintenant être ajoutés, soit par drag&drop ou à l'aide de l'onglet **Informations supplémentaires** de l'enregistrement en question.

### Inclusion / Exclusion

Suivant le cas, on voudrait assurer que l'enregistrement ne figure que dans l'un des sous-dossiers (un collaborateur ne fait partie que d'un département). Vous avez 2 possibilités pour déterminer cela:

Soit par une **Condition (Constraints)** (pour Conditions/Constraints voir chapitre 7) qui défini quand un enregistrement est valable. L'expression d'une telle condition serait par exemple:

```
ordner->select (parentordner.bezeichnung='Abteilungen')->size=1
```

ou par un **dossier IX**. Un dossier normal devient un dossier IX, en configurant les propriétés à l'aide du masque accessible par **l'onglet Inclusion/Exclusion** (peut être sélectionné par le menu contextuel de la souris):

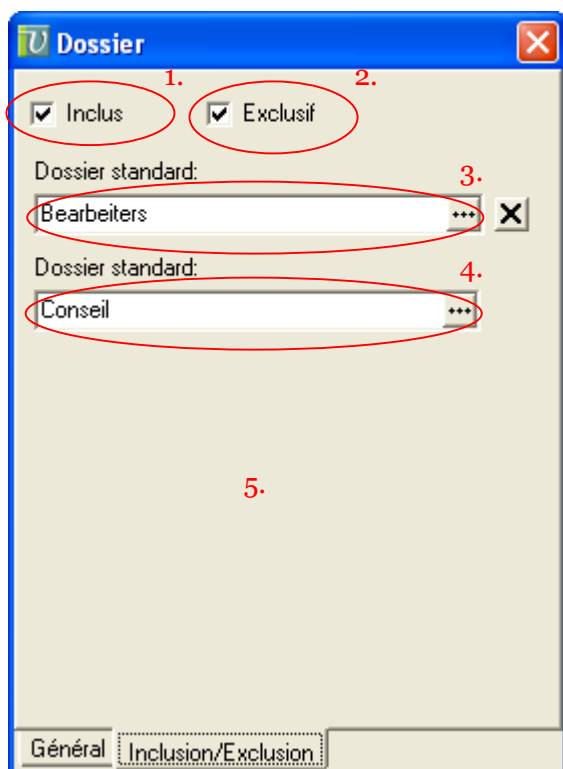


Figure 25 Onglet Inclusion/Exclusion des propriétés du dossier

1. Inclusive: signifie que tous les enregistrements du dossier de base doivent se trouver au moins une fois dans le dossier IX ou un sous-dossier.
2. Exclusive: signifie qu'un enregistrement ne puisse se trouver qu'une fois dans le dossier IX ou l'un des sous-dossiers.

Si l'on choisit Inclusive et Exclusive, le dossier IX fera attention à ce que tous les enregistrements du dossier de base n'apparaissent qu'une fois dans le dossier IX ou un sous-dossier.

3. le dossier de base indique la quantité de base sur laquelle agit l'option Inclusive. Si Inclusive n'est pas sélectionné, le dossier de base n'a pas de signification.
4. Si Inclusive est sélectionné, le dossier par défaut contient tous les enregistrements du dossier de base qui ne sont pas soit dans le dossier IX soit un sous-dossier.

Un dossier IX avec beaucoup d'enregistrements et le choix **Inclusive** peut exiger beaucoup de temps jusqu'à ce qu'il a constaté la consistance des données (le dossier doit déterminer pour chaque enregistrement du dossier de base, s'il se trouve dans un sous-dossier). Ce **dossier IX Inclusive** n'est alors pas indiqué pour des grandes quantités de données (par exemple des phases de projets). Dans ce cas, il est conseillé de préparer un dossier **Exclusive** (les enregistrements sont déplacés par Drag&Drop d'un sous-dossier vers l'autre). Par une condition, on surveille que chaque enregistrement se trouve dans un sous-dossier.

### Dossiers comme dossiers à rubriques

Pour la création de catégories des différents enregistrements dans Vertec, on peut déterminer les dossiers manuels comme **dossiers à rubriques**. Cette catégorie est attribuée dans les propriétés (clic droit sur le dossier > **Propriétés**).

Si un dossier est marqué comme dossier à rubriques, tous les sous-dossiers de ce dossier apparais-

sent comme rubriques sur les enregistrements de cette classe qui conviennent à ce dossier. Ce procédé ne fonctionne qu'à l'aide d'un niveau inférieur. Si les sous-dossiers ont de nouveau des sous-dossiers, ceux-ci ne sont pas pris en considération pour les rubriques.

Comme exemple, les projets devraient être attribués à des filiales. Le dossier **projets par filiales** est qualifié comme dossier à rubriques:

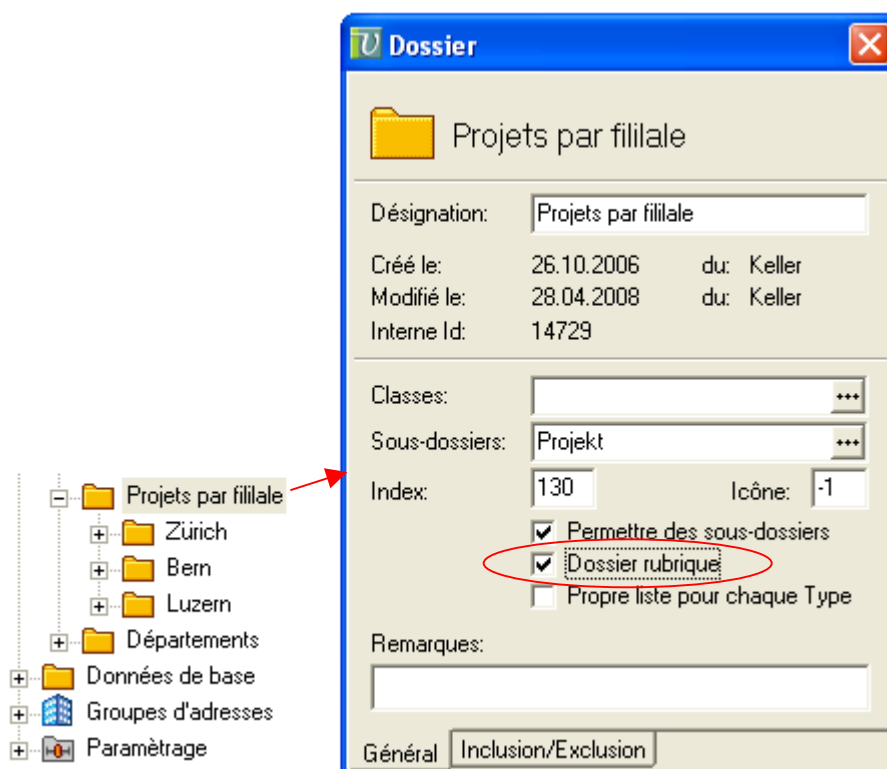


Figure 26 Projets selon filiales comme dossier à rubriques

Les sous-dossiers de ce dossier apparaissent ensuite sur les projets comme rubriques. Les projets peuvent ensuite être attribués facilement par un cliquer de souris:

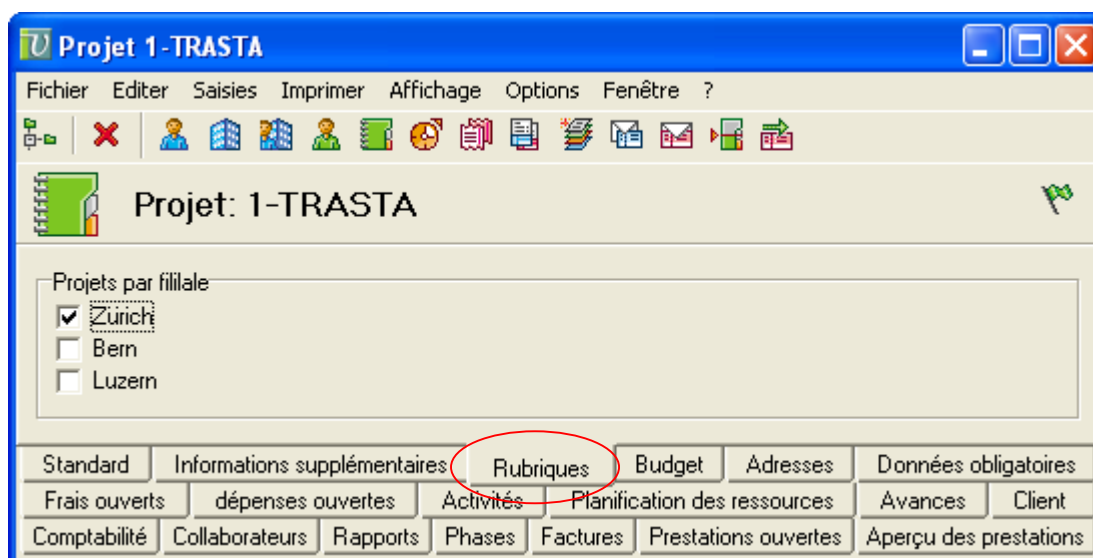


Figure 27 Rubriques sur un projet

Ainsi vous pouvez facilement créer des rubriques pour différentes sortes d'enregistrements comme (adresses, projets, collaborateurs etc.).

### Comportement des rubriques dans des dossiers IX

Des dossiers à rubriques qui contiennent en même temps des conditions Inclusive/Exclusive (voir ci-dessus), la sélection des rubriques varie comme suit:

1. en cas de condition **Exclusive** (Enregistrement ne doit apparaître que dans un sous-dossier) un champ combiné apparaît avec les dossiers à choix. Le bouton d'effacement X permet de supprimer le choix.

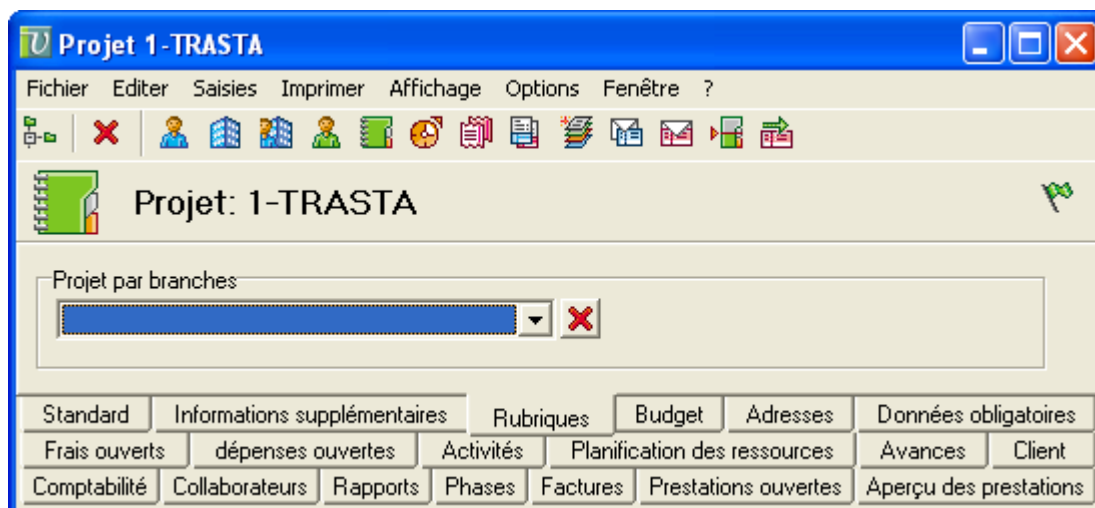


Figure 28 Choix de rubriques en cas de condition Exclusive

2. en cas de condition **Inclusive ET Exclusive** (l'enregistrement doit être attribué exactement à un sous-dossier) un champ combiné de sélection apparaît. Le bouton d'effacement n'est pas disponible, car l'enregistrement doit être attribué à une rubrique. De nouveaux enregistrements ou des enregistrements pas encore attribués sont reliés sur le dossier par défaut (correspond au dossier par défaut de la condition Inclusive).

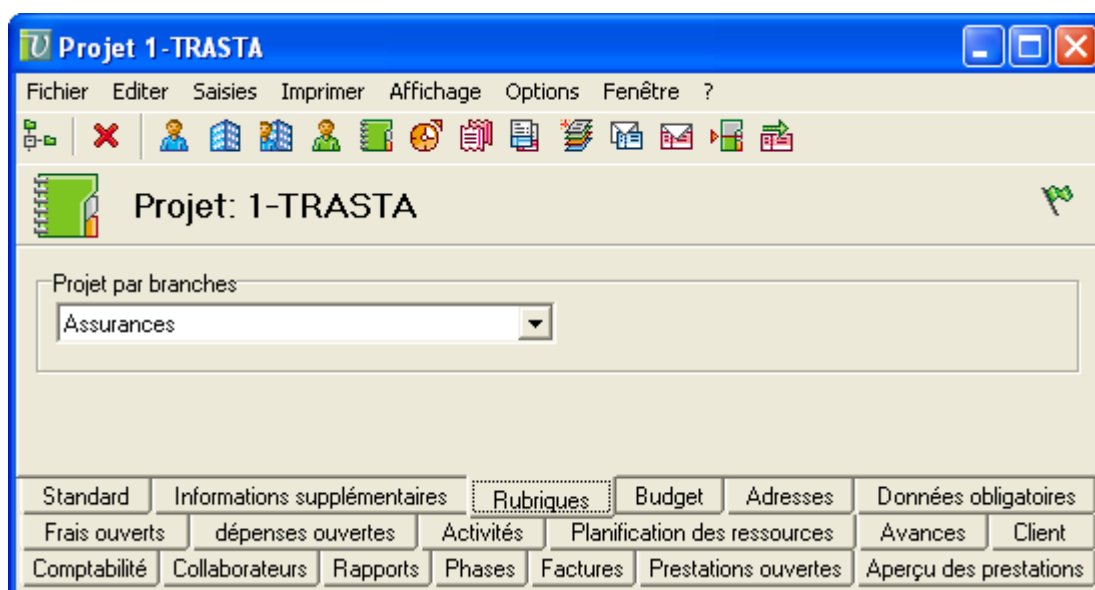


Figure 29 Choix de rubriques en cas de conditions Inclusive / Exclusive

3. En cas de condition **Inclusive** (le dossier doit se trouver dans au moins un des sous-dossiers) des cases à cocher apparaissent comme pour les dossiers à rubriques normaux. La différence réside dans le fait qu'une case doit toujours être activée. Si la dernière case est désactivée, la case par défaut est automatiquement activée (correspond au dossier par défaut de la condition Inclusive).

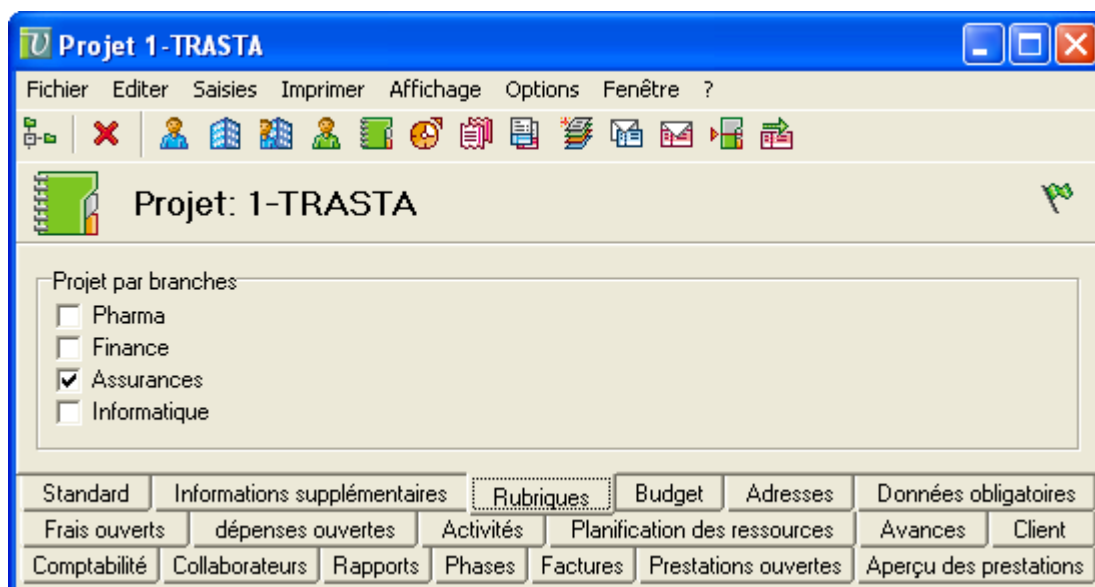


Figure 30 Choix de rubriques en cas de condition Inclusive

## 4.2 Dossier à expression

Un dossier à expression détermine son contenu à l'aide d'une expression OCL. L'expression OCL se réfère toujours au système Vertec, le type de base dans l'éditeur d'expression s'appelle **Business-Classes**. Des dossiers à expression peuvent donc travailler avec des expressions OCL comme

```
projekt->select(aktiv)
```

(pour une introduction courte dans OCL – voir chapitre 1).

## 4.3 Dossier de recherche

Un dossier de recherche combine les listes d'autres dossiers (aussi d'autres dossiers de recherche) selon la théorie des ensembles, avec des opérateurs (**ajouter**, **exclure** et **Intersection**). Comme source de recherche, on ne peut pas seulement indiquer un seul dossier, mais aussi n'importe quelle **expression OCL**.

### Ajouter, exclure, Intersection

L'exemple ci-après montre un dossier de recherche, qui montre tous les collaborateurs du département IT Consulting qui ont des heures supplémentaires.

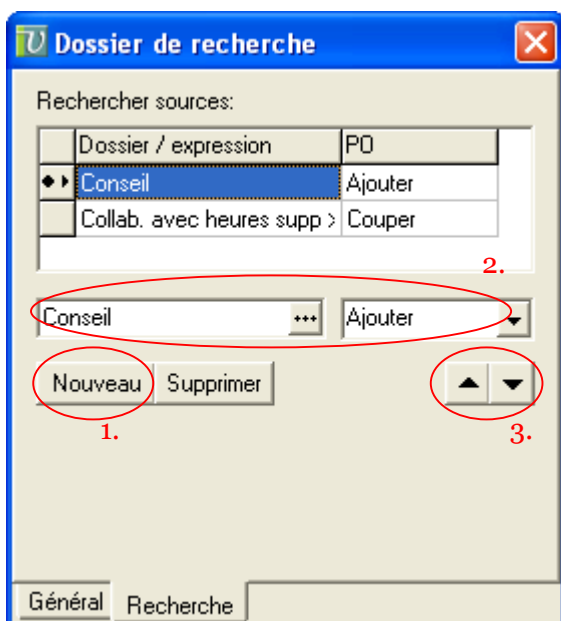


Figure 31 Les propriétés d'un dossier de recherche

1. A l'aide du bouton **Nouveau** une nouvelle ligne est créée dans la liste des sources de recherche.
2. La ligne marquée dans la liste des sources de recherche peut être éditée. A l'aide des 3 points, une fenêtre de dialogue s'ouvre qui permet de sélectionner les dossiers souhaités. **L'opération (PO)** est sélectionnée à l'aide du champ combiné à droite.

Ajouter	Tous les enregistrements du dossier indiqué sont ajoutés à la recherche.
Intersection	Les enregistrements de ce dossier sont comparés/coups avec les enregistrements des lignes précédentes. C'est à dire, seulement les enregistrements qui apparaissent dans tous les dossiers sont pris en considération.
Exclure	Des enregistrements qui se trouvent dans ce dossier sont exclus du résultat. Le résultat ne montre donc que des enregistrements qui ne se trouvent pas dans ce dossier.

La ligne sélectionnée de la liste source de recherche marquée est déplacée à l'aide du bouton fléché vers le haut ou le bas.

Les différentes recherches sont toujours traitées séquentiellement. Un dossier de recherche peut aussi se trouver dans un autre dossier de recherche. Ainsi vous pouvez procéder à des recherches à travers plusieurs dossiers.

Puisque vous ne pouvez pas placer des parenthèses, vous devriez utiliser des dossiers de recherche pour des résultats intermédiaires lors d'opérations logiques complexes (avant tout si vous combinez **intersection** et **ajouter**).

### Combiné avec une expression OCL

1. Si une expression OCL doit être utilisée, vous sélectionnez **Expression** comme opérateur. L'éditeur OCL apparaît en cliquant sur le bouton avec les 3 points.

2. L'introduction de l'expression se réfère toujours sur le type du résultat intermédiaire actuel. Si un dossier de projets a été sélectionné comme source, l'expression se réfère ensuite sur le type Projekt.

La détermination du type de résultat se fait sur la base des types du contenu des dossiers. Elle se fait comme suit en fonction des différents opérateurs :

Ajouter	Classe de base commune des deux types de dossiers
Intersection	Typeinfo du premier dossier
Exclure	Typeinfo du premier dossier
Expression	Type de résultat de l'expression

Des résultats intermédiaires via des expressions peuvent avoir des types de résultats différents (cela n'est pas obligatoirement une liste de UserEintraege). A la fin, le résultat devrait être une liste ou un seul UserEintrag.

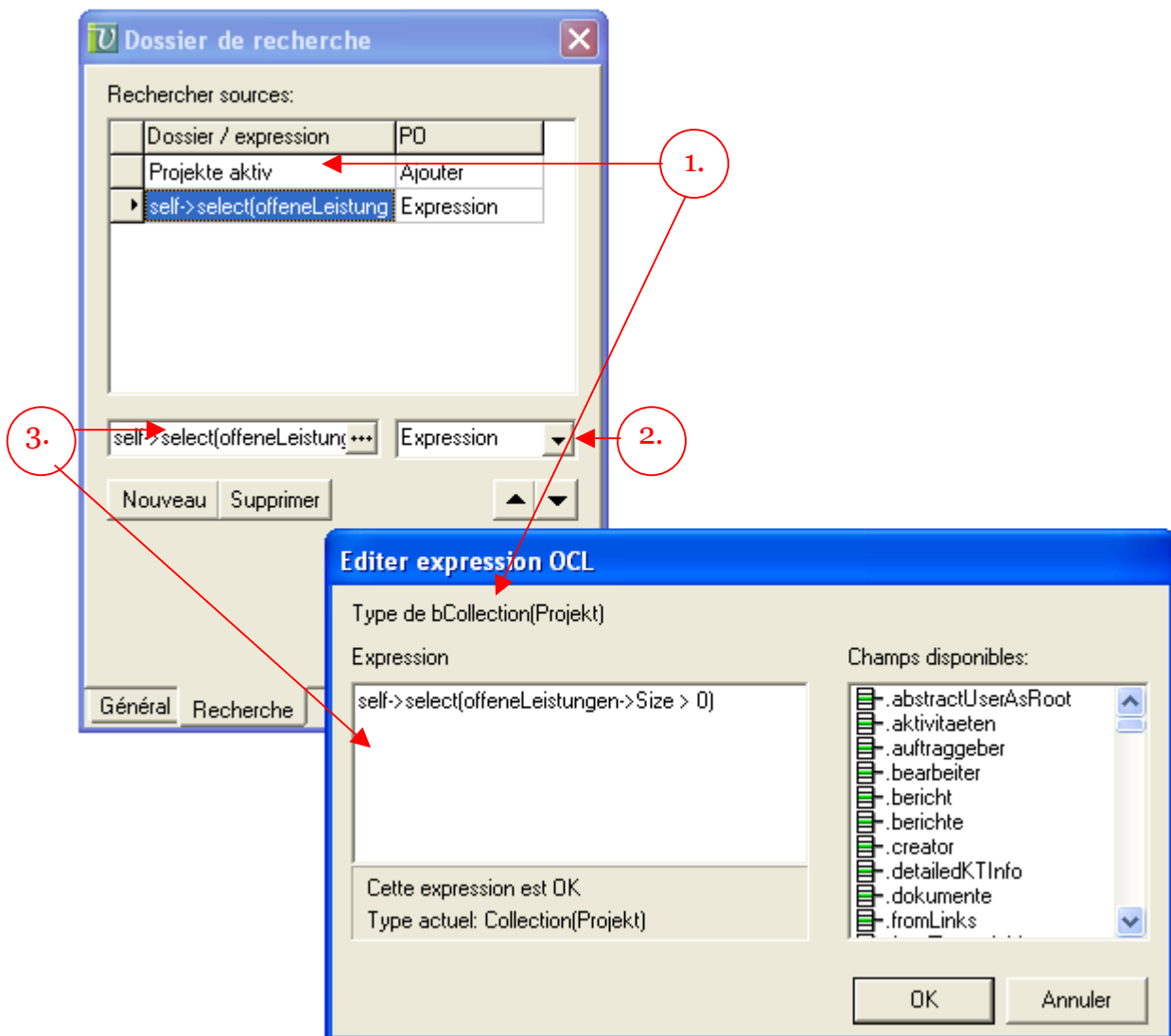


Figure 32 Dossier de recherche avec expression OCL

#### 4.4 Dossier SQL

L'utilisation du dossier SQL-Ordner est conseillée lorsque des objets définis doivent être sélectionnés sur la base d'une condition simple dans une importante quantité de données. Contrairement au dossier à expression normal, la sélection des données est effectuée par le serveur de la base de données, ce qui est plus rapide dans la plupart des cas. L'inconvénient des dossiers SQL réside dans le fait qu'ils ne réagissent pas automatiquement aux changements des données et qu'ainsi la flexibilité des extractions est limitée face à celle des dossiers à expression normale (avec OCL).

Un exemple de dossier SQL est celui du dossier des **Adresses** dans les données de base de Vertec. Si vous sélectionnez ce dossier, une fenêtre de recherche apparaît dans laquelle vous pouvez indiquer des critères définis. Seuls les enregistrements répondant à ces critères apparaîtront alors dans le dossier.

Le dialogue des propriétés d'un dossier SQL permet un traitement interactif de la requête SQL.

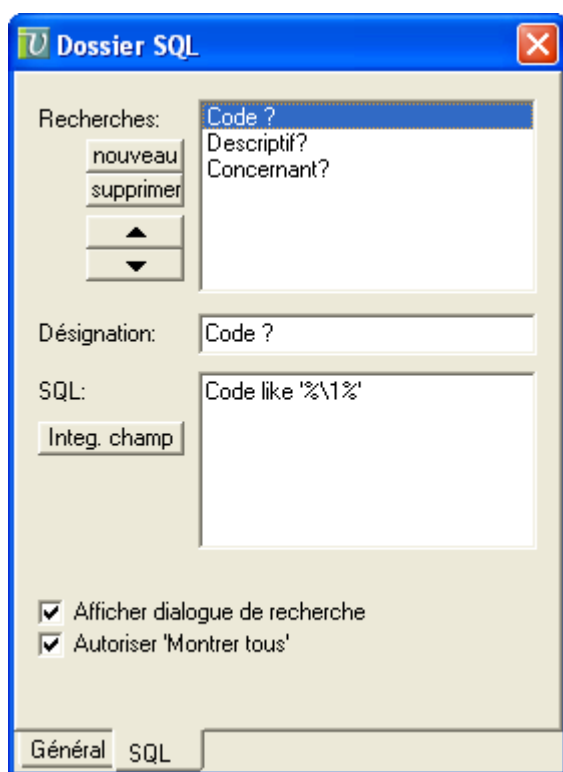


Figure 33 Le dialogue du dossier SQL

Une requête d'un dossier SQL est composée d'une **désignation** et d'une **condition SQL**. La condition SQL doit être une expression SQL valable qui fournit une valeur booléenne. Pour l'analyse, la condition SQL est utilisée comme condition WHERE dans une commande SQL SELECT.

Comme aide lors de construction de la condition, vous trouverez le champ **Intég. champ**. Il affiche un dialogue de sélection des noms de champs disponibles des tables de la base de données. La majorité des champs de la table correspondent aux noms d'attributs utilisés dans OCL. Des différences existent en cas d'attributs calculés et de liaisons (Verknüpfung).

Plusieurs requêtes peuvent être formulées pour un dossier SQL. Si le dossier est configuré de telle manière à ce que le dialogue de recherche est affiché, les différentes requêtes sont disponibles dans un dialogue de sélection. Si aucun dialogue de recherche n'est affiché, le dossier utilise toujours la première requête pour calculer son contenu.

A ce titre, il est possible de masquer l'option *afficher tous* dans la fenêtre de dialogue. Pour cela, désactivez la case *Autoriser 'Montrer tous'* dans les propriétés du dossier:

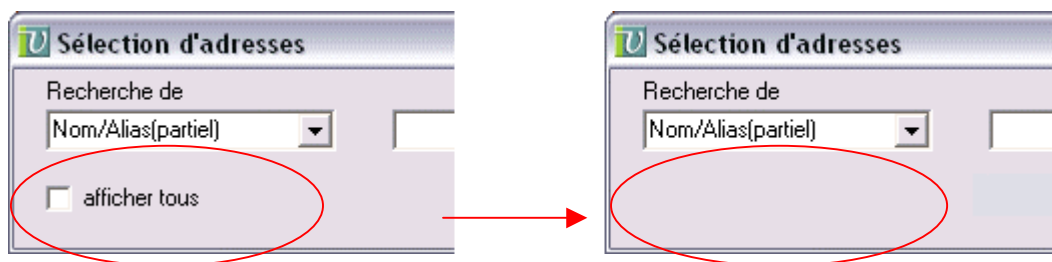


Figure 34 Afficher tous visible ou masqué

ATTENTION: des dossiers SQL avec un dialogue de recherche ne doivent pas être utilisés comme source de recherche pour un dossier de recherche. Ceci ne s'applique pas aux dossiers SQL sans dialogue de recherche. En fait, le dossier SQL contiendrait toujours le résultat de la dernière requête.

#### 4.5 Exercices

- Exercice 9* Créez un dossier dans lequel se trouvent tous les clients (rubrique : qu'est-ce qu'un client?). Voyez à ce titre le modèle de la Figure 55 à la page 65.
- Exercice 10* Créez une structure de dossiers pour des rubriques d'adresses (Branches: banques, assurances, pharma, ...). Attribuez des adresses à ces branches.
- Exercice 11* Créez une liste de tous les clients selon Exercice 9, qui appartiennent à la branche **banques** (voir Exercice 10) (conseil: dossier de recherche).
- Exercice 12* Créez un dossier, dans lequel sont présentés tous les projets qui contiennent des prestations non facturées (prestations ouvertes).

## 5. Types de lien (Link-Typen)

Les types de liens déterminent l’affichage des liens entre les enregistrements (links) dans Vertec.

Dans le cas où ce lien est déjà créé dans le modèle de base de Vertec (p.ex. **prestation ouverte - projet**) ou peut être dérivé du modèle (p.ex. d’une **prestation ouverte -chef de projet** à travers le projet), cette liaison est configurée à l’aide d’un **Wrapper-Link-Typ**.

Si le lien n’existe pas dans le modèle Vertec (p.ex. le lien entre l’adresse et le projet avec nom **Consultants externes** (adresse) et **Collaboration au projet** (projet), ce nouveau lien doit être créé à l’aide d’un **type de lien générique**.

Tous les types de liens sont présentés dans le dossier **Paramétrage**:

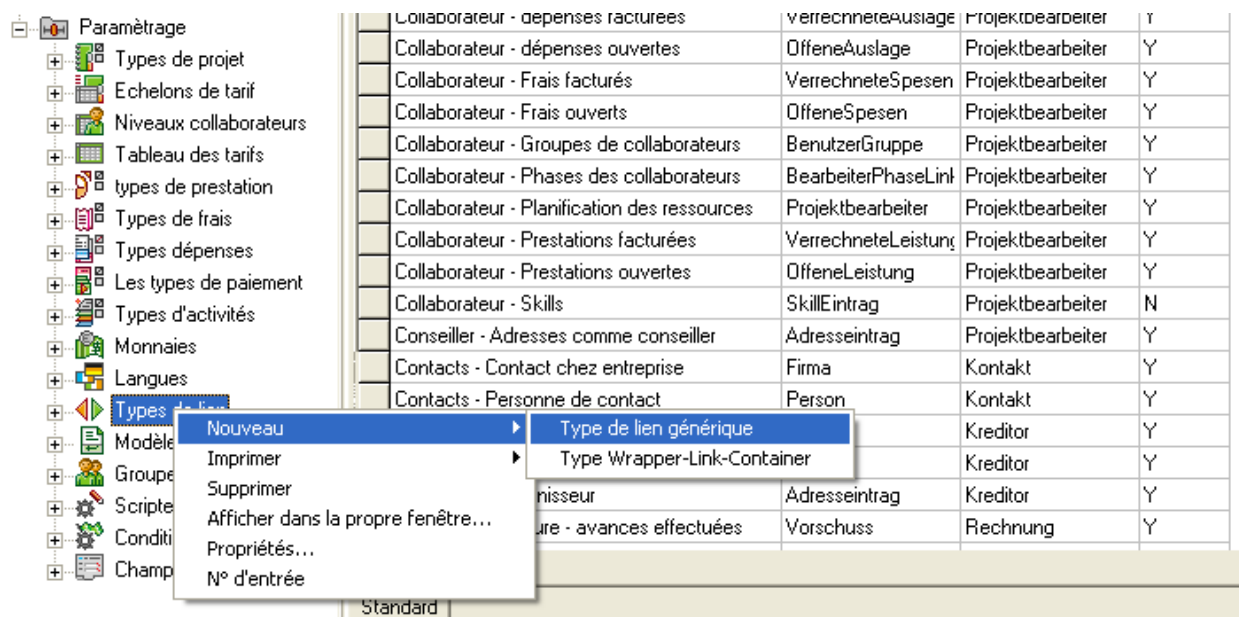


Figure 35 Les types de liens

ATTENTION: lors de la saisie d’un nouveau type ou de la modification d’un type de lien existant, il est possible que celui-ci ne fonctionne qu’après un redémarrage de Vertec.

### 5.1 Wrapper-Link-Type

Un Wrapper-Link-Type permet d’afficher des liens qui existent dans le modèle de base Vertec (p.ex. **projet – chef de projet**) ou qui peuvent être dérivés depuis le modèle. Un Wrapper-Link-Type remplit deux tâches:

1. Les Wrapper-Link-Types peuvent être modifiés. Par exemple, si vous ne voulez pas travailler avec des activités dans Vertec, vous pouvez désactiver les Wrapper-Link-Types en question et les dossiers correspondant ne s’affichent plus.
2. Vous pouvez créer de nouveaux Wrapper-Link-Types pour afficher des données d’utilisateurs au bon endroit. On peut par exemple créer un Wrapper-Link-Type qui permet au chef de projet d’afficher par un seul clic toutes les activités de ses projets actifs (nous allons créer ce type de lien dans l’Exercice 14.

Voici un exemple d’un Wrapper-Link-Type:

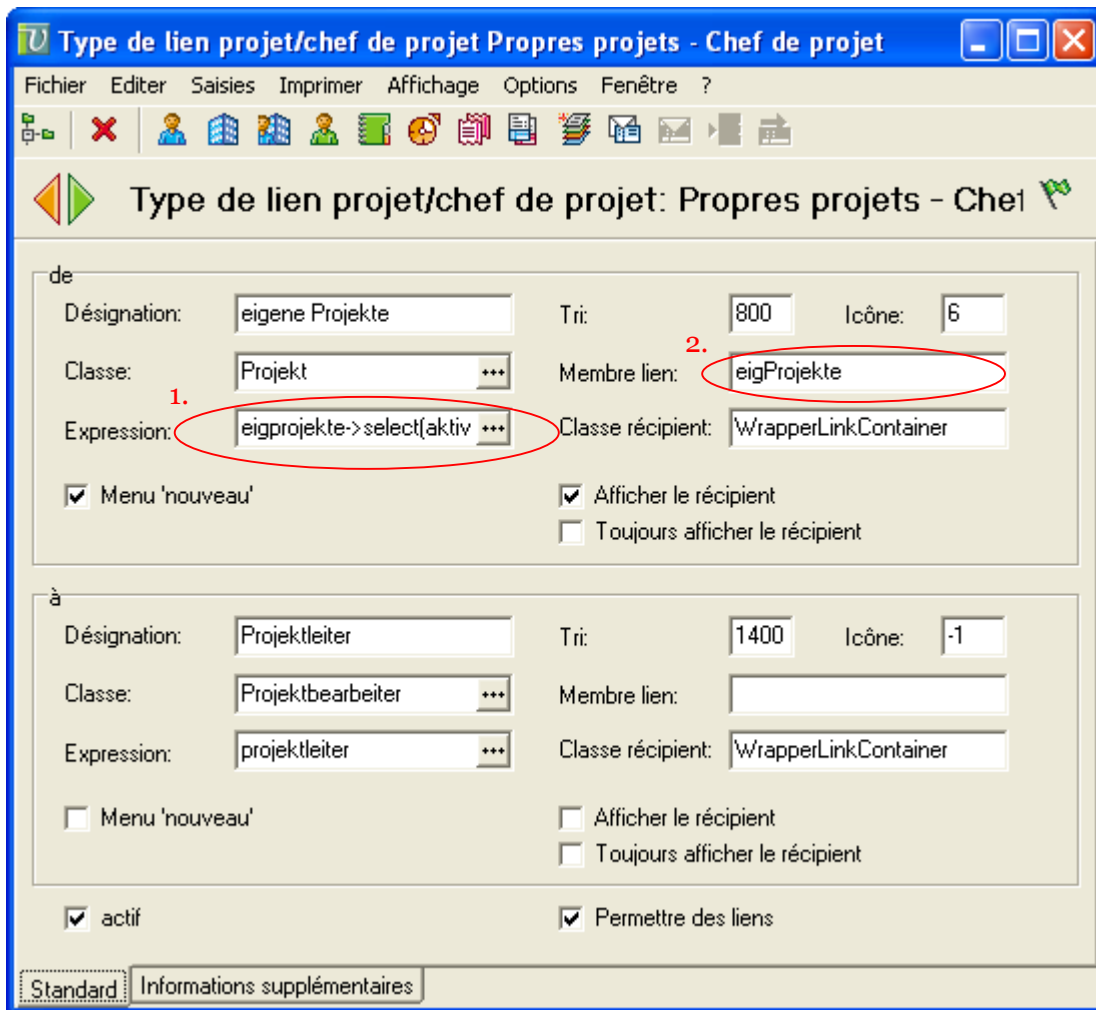


Figure 36 Un Wrapper-Link-Type

Voici l'explication des nouveaux éléments d'affichage non nécessaires dans 5.2. Tous les autres éléments à l'écran sont identiques aux type Generic-Link.

1. Expression: indique l'expression, comme on arrive de ce côté en partant depuis le côté opposé. Dans l'exemple, on navigue depuis le collaborateur du projet (Projektbearbeiter) (nom de classe dans le champ **a**) vers le projet avec l'expression `eigprojekte->select(aktiv)`. Comme vous voyez, on peut aussi utiliser des expressions très complexes.
2. Linkmember: Si une expression complexe (pas seulement un lien simple, qui existe dans le modèle de base Vertec) est utilisée dans le champ **Expression**, le champ Linkmember/membre lien doit contenir le nom du lien direct (Member) -> dans l'exemple `eigprojekte`. Ceci signifie que ce membre est fixé si l'utilisateur crée un nouveau lien.

## 5.2 Types de liens génériques

Des types de liens génériques permettent de définir de nouveaux liens dans Vertec. Ceux-ci n'existent pas dans le modèle de base de Vertec et ne peuvent pas être générés automatiquement. Un nouveau type de lien générique est créé à l'aide de la touche droite de la souris sur le dossier **Types de lien**. La fenêtre suivante s'ouvre (voici un type de lien générique déjà rempli qui définit le lien entre une personne et une entreprise (**Membre du conseil – mandats comme CA**)):

Figure 37 Un type de lien générique

Le comportement des deux côtés doit toujours être défini dans un type de lien (**Rôles**), c'est pourquoi ce masque se compose des deux parties **De** et **A**, qui définissent les deux côtés du type de lien.

1. Désignation: Le terme indiqué ici sera affiché à l'écran pour ce rôle (Container/dossier dans l'arbre ou onglet dans le masque individuel)
2. Nom de classe: Nom du type d'enregistrement (**Classe**) de ce rôle. Peut être sélectionné à l'aide du bouton à 3 points ou saisi directement dans le champ (Attention: première lettre en majuscule)
3. Tri et icône: Indiquez le tri et une icône alternative au dossier. Analogue aux dossiers (voir chapitre 4)
4. Menu nouveau : Si cette option est sélectionnée, l'option de créer un nouvel enregistrement et de le lier apparaît dans le menu **Nouveau**. Si l'on sélectionnait cette option pour le rôle du conseil d'administration dans l'exemple, le menu Nouveau de chaque entreprise ferait apparaître le point Conseil d'administration. En cliquant sur ce point du menu, on saisirait une personne et la lierait directement comme conseil d'administration avec l'entreprise.
5. Afficher container : Si cette option est sélectionnée, le container est affiché (comme arbre et comme onglet dans le masque individuel). Il peut être judicieux de ne pas montrer les deux côtés de certains liens, ainsi on ne peut pas sélectionner cette option dans ce rôle. De créer des types de lien où l'on sélectionne 'ne pas afficher le container' pour les deux rôles, ne fait

pas de sens.

6. Afficher le container toujours: si cette option est sélectionnée, le container est affiché tout le temps - que des liens existent ou pas. Un exemple pour un tel rôle est la liste des prestations ouvertes sur le collaborateur. Le container des factures sur le projet n'est par contre affiché que si des factures existent. L'option n'est pas cochée dans ce rôle.
7. actif: le type de lien est actif et est ainsi affiché
8. permettre liens : ce type de lien permet de créer des liens entre des enregistrements.

Les champs **Expression** et **Linkmember** n'ont pas de signification pour des types de liens génériques. Ces champs sont expliqués dans le chapitre 5.1 à propos des types Wrapper-Link.

<b>Astuce</b>	<ul style="list-style-type: none"> <li>• <b>WrapperLinkTyp</b>: représente une liaison entre des données que Vertec connaît déjà.</li> <li>• <b>GenericLinkTyp</b>: créer une relation entre des données que Vertec ne connaît pas encore.</li> </ul>
---------------	---

### 5.3 Exercices

- Exercice 13* Vous voudriez que la condition avancée soit toujours affiché sur le projet indépendamment du fait que des avances existent ou pas. Comment procédez-vous ?
- Exercice 14* Vous aimeriez réaliser que le chef de projet ait toujours un accès direct aux factures impayées de ses projets. (conseil: voir exemple dans Figure 55 à la page 65).
- Exercice 15* Créer un nouveau type de lien qui permet d'attribuer des conseillers externes (Adresses) aux projets (Conseil: la classe pour adresses s'appelle **Adresseintrag**). Saisissez aussi quelques liens entre les adresses et les projets et surveillez ce qui se passe avec les enregistrements liés dans l'arbre et dans le masque individuel.

## 6. Champs et Classes complémentaires

### 6.1 Champs complémentaires

La liste des champs disponibles dans Vertec est indiquée dans le modèle de base et ne peut pas être modifié. Il peut par contre arriver que vous souhaitez saisir des informations supplémentaires dans vos données. (p ex. **Numéros d'investissements** sur **Projekt**). Pour cela existent les champs complémentaires (seulement dans la ligne **Expert**).

Ces champs complémentaires sont définis dans l'arborescence sous **Paramétrage**:

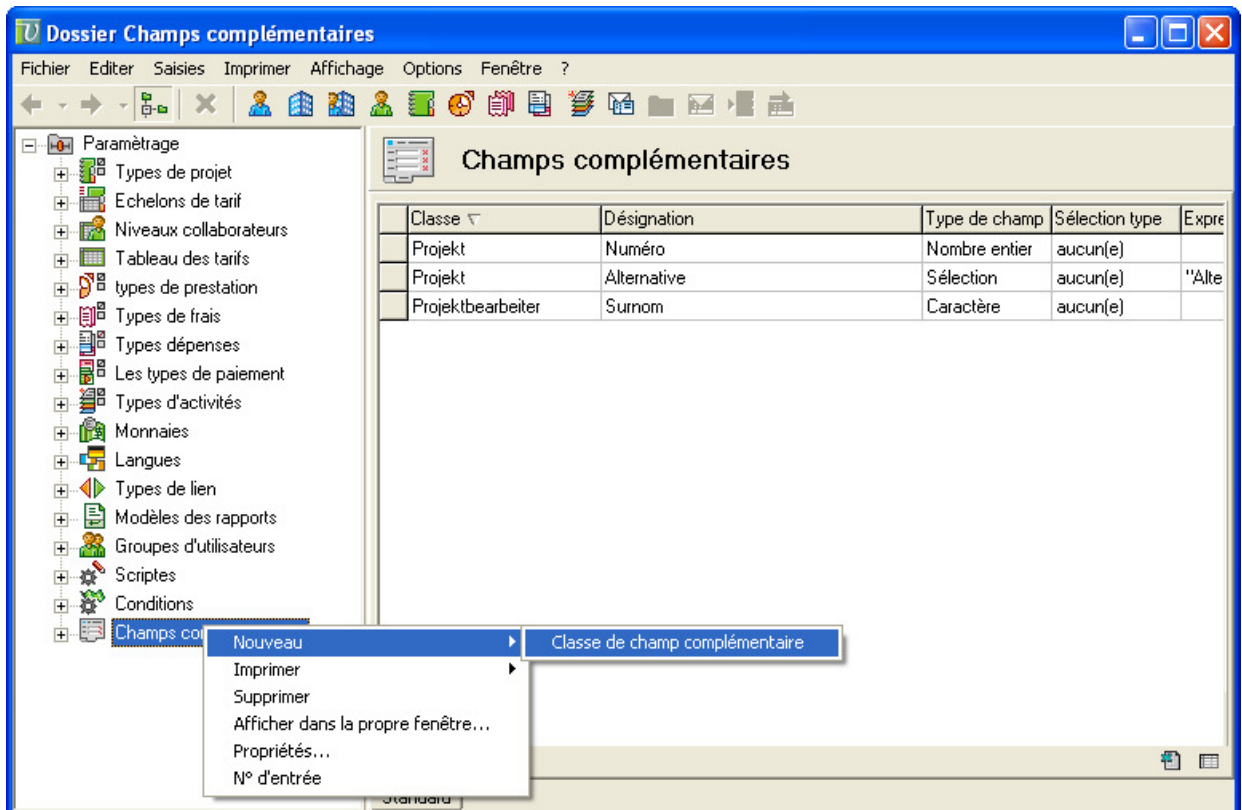


Figure 38 Définir des champs complémentaires

Le masque de définition des champs complémentaires s'ouvre :

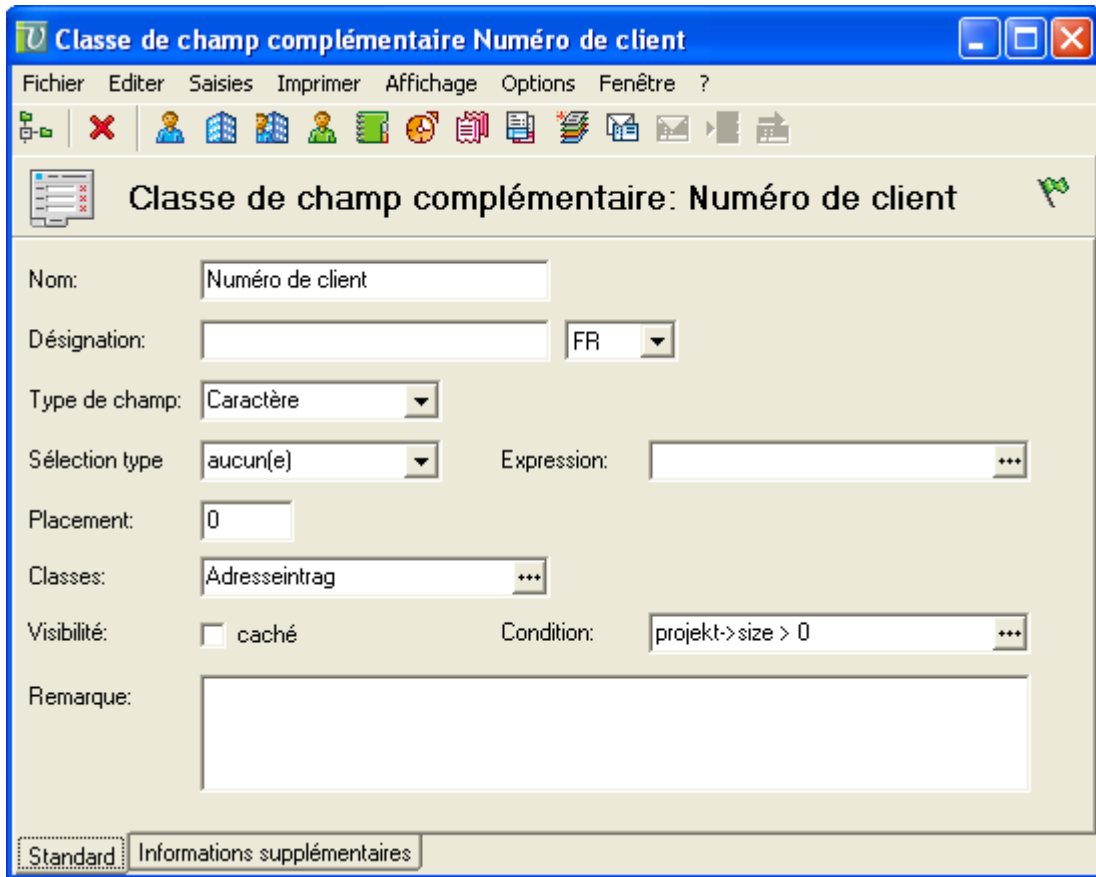


Figure 39 Éditer un champ complémentaire

Le champ complémentaire est défini à l'aide des champs **Type de champ**, **Sélection type** et **Expression**. **Placement** règle la disposition sur l'onglet **Informations complémentaires** tandis que **Classes** indique sur quelles classes le champ doit apparaître. Expression est toujours une expression OCL qui règle la sélection (n'est pris en considération que si un type de sélection est défini à l'exception du type de champ **Sélection**). L'expression OCL se base sur l'objet actuel. Par exemple :

```
phasen.code
```

sur un champ d'un projet donnerait la liste de toutes les phases de ce projet. Si l'on voulait connaître tous les codes des phases qui existent dans le système, l'expression devrait être :

```
Phasen->allInstances.code
```

Voici une liste de type des champs possibles :

TYPE CHAMP	EXPRESSION OCL	REMARQUE
signe	zusatzfeld('Nom')	Un champ d'édition simple.
Vrai, Faux	zusatzfeldbool('Nom')	Affiche une case de sélection.
Nombre entier	zusatzfeldint('Nom')	Nombre entier comme 0,1,3 etc.
Minutes	zusatzfeldint('Nom')	Présentation en minutes comme pour des prestations. Afin d'obtenir une présentation habituelle de 15 minutes (p.ex. 0:15), il faut utiliser le renderer <code>dbmTim.minuteRenderer</code> (Renderer voir 3.2).

Chiffre à virgule fixe	<code>zusatzfeldcurr('Nom')</code>	Chiffre avec 2 positions après la virgule, p.ex. 1'015.60.
Date	<code>zusatzfelddate('Nom')</code>	Valeur de date. Si vous aimiez afficher un dialogue de date, vous devriez définir un <b>type de sélection</b> . Vous pouvez choisir un type quelconque de sélection à l'exception de Via Extension.
Texte	<code>zusatzfeldblob('Nom')</code>	Affiche un grand champ de texte (Mémo).
Image		Image. L'image peut être attribuée via Drag and Drop.
Objekt	<code>zusatzfeldobj('Nom')</code>	Un objet du système Vertec. En liaison avec un objet, les types de sélection <b>Sélection d'objet</b> , <b>Sélection d'adresses</b> , <b>Dossier Vertec</b> et <b>Sélection de dossier</b> sont utiles.
Sélection	<code>zusatzfeldint('Nom')</code>	Représente une Combobox. Les contenus de la Combobox sont introduits par <b>Expression</b> précisément dans le format <b>"Machines", "Consommations", "Commerce"</b> . En interne, ces champs sont enregistrés comme Integer, ainsi <code>Machines=0</code> , etc. Si l'on souhaite voir le texte complet à l'écran, on peut l'afficher à l'aide de l'expression <code>zusatzfeldasString('Name')</code> .
tous	<code>zusatzfeldAsString('Nom')</code>	Tous les types de champs mentionnés peuvent être lus à l'aide de cette expression. Le résultat est toujours une valeur String.

A l'aide d'une condition d'affichage, vous pouvez déterminer si un champ complémentaire est affiché sur un enregistrement individuel ou pas. La formulation de cette condition se fait à l'aide d'une expression OCL et doit donner une valeur booléenne Vrai ou Faux. Si le résultat est Vrai, le champ complémentaire est affiché. Si aucune condition n'est indiquée, le champ complémentaire est affiché.

Si l'option **Visibilité: caché** est mis sur **caché**, le champ complémentaire n'est pas affiché, indépendamment du fait qu'une condition vrai ou faux ou aucune condition n'est mentionnée.

Le nouveau champ complémentaire s'affiche dans l'affichage individuel de l'enregistrement en question sous l'onglet **Infos supplémentaires**.

## 6.2 Classes complémentaires

Une autre possibilité d'enregistrer des informations complémentaires est assurée par les **Classes complémentaires**. Vertec permet de définir 20 classes complémentaires que vous pouvez définir à votre guise. Elles servent à représenter vos propres informations ou à les réunir.

Des classes complémentaires sont des enregistrements normaux dans Vertec et peuvent être traitées ainsi. Elles peuvent être classées dans des dossiers et filtrées par des dossiers à expression et vous pouvez définir des types de liens.

Vous pouvez définir des champs complémentaires sur les classes complémentaires. (voir chapitre 6.1). Ceux-ci sont affichés sur les classes complémentaires et peuvent être remplis avec des valeurs et affichés dans les listes.

Les classes complémentaires 0 – 19 se trouvent dans **Paramètres système > Classes complémentaires**.

res:

Nom	Valeur
Zusatzklasse 00 Name	Curriculum vitae
Zusatzklasse 01 Name	
Zusatzklasse 02 Name	
Zusatzklasse 03 Name	
Zusatzklasse 04 Name	
Zusatzklasse 05 Name	
Zusatzklasse 06 Name	
Zusatzklasse 07 Name	

**Option:** Zusatzklasse 00 Name     
**Valeur actuelle:** Curriculum vitae     
**Nouvelle valeur:** Curriculum vitae     

Benchmarking | Planification de ressources | **Classes Compl.** | TVA | Moyen de comr

Figure 40 Définir le nom de classes complémentaires

Afin qu'une classe complémentaire s'affiche au bon endroit, elle doit être reliée par un GenericLink avec l'enregistrement en question. (voir chapitre 5.2). Un onglet s'affiche ensuite avec le nom de la classe complémentaire dès qu'un enregistrement correspondant existe.

### Exemple

Vous aimeriez pouvoir saisir des curriculum vitae de toutes les personnes de votre gestion d'adresses.

1. Donnez à votre classe complémentaire le nom **Curriculum vitae** (voir Figure 40).
2. créez 3 champs complémentaires : **Formation**, **Durée** et **Entreprise**. Pour ces 3 champs, vous sélectionnez le type de données **Zusatzklasse0** :

Classe de champ complémentaire Formation

Fichier Editer Saisies Imprimer Affichage Options Fenêtre ?

Classe de champ complémentaire: Formation

Nom: Formation

Désignation: FR

Type de champ: Caractère

Sélection type: aucun(e) Expression: ...

Placement: 0

Classes: ZusatzKlasse0

Visibilité:  caché Condition: ...

3. Créez un type de lien générique. Mettez les valeurs suivantes:

- De: Désignation **Curriculum vitae**, Nom de classe **ZusatzKlasse0**. Cochez la case **Menu Nouveau**, afin que vous puissiez générer des enregistrements correspondants plus tard.
- A: Désignation **Personne**, Nom de classe **Personne**.

Type de lien générique: Curriculum vitae - Personne

de

Désignation: Curriculum vitae Tri: 0 Icône: -1

Classe: ZusatzKlasse0 Membre lien: ...

Expression: ... Classe récipent: GenericLinkContainer

Menu 'nouveau'  Afficher le récipent

Toujours afficher le récipent

à

Désignation: Personne Tri: 0 Icône: -1

Classe: Person Membre lien: ...

Expression: ... Classe récipent: GenericLinkContainer

Menu 'nouveau'  Afficher le récipent

Toujours afficher le récipent

actif  Permettre des liens

A partir de maintenant, vous pouvez créer un enregistrement **Curriculum Vitae** chez chaque personne à l'aide du menu **Nouveau** et remplir les champs correspondants.



Figure 41 Créez un nouvel enregistrement Curriculum Vitae

Les masques des personnes qui possèdent des enregistrements de Curriculum Vitae affichent l'onglet **Curriculum Vitae** avec les informations correspondantes.

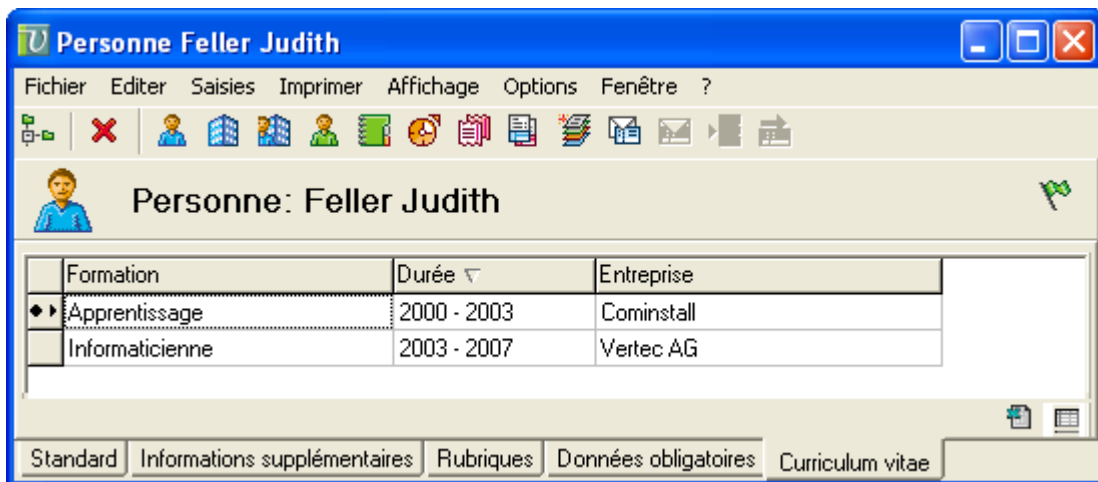


Figure 42 Affichez les enregistrements dans une liste

L'accès à un champ de la liste se fait à l'aide d'une expression normale d'un champ complémentaire, par exemple `zusatzfeldAsString('Formation')`.

## Exercices

- Exercice 16** Créez un champ complémentaire sur les projets qui indique le taux de couverture (marge contributive) projeté du projet. Représentez ce champ dans une liste de projets et totalisez la colonne.
- Exercice 17** Vous aimeriez attribuer chaque avance à une phase de projet. Créez un champ complémentaire qui vous permet d'attribuer une phase de projet à chaque avance. Offrez à l'utilisateur un choix de toutes les phases actives du projet actuel.
- Exercice 18** Créez un champ supplémentaire dans la phase de projet ayant le nom **Degré de finition** et le libellé **Degré de finition en %**. Représentez deux nouvelles colonnes dans une liste de projets:
- nombre de phases de projet sans degré de finition
  - degré de finition le plus bas

## 7. Constraints

Une Condition (règle de validité) détermine, quand un enregistrement est valable. Les enregistrements non valables s'affichent en rouge à l'écran, par exemple dans une liste:



	Projet	Phase	Date	Collaborateur	type de prestation	Texte
*	COMINSTALL	1 ANA	28.04.2008	Keller		
			28.04.2008	Keller		

Figure 43 Enregistrement non valable dans cette liste

Dans le masque individuel, cette information est aussi affichée et en plus l'information pourquoi (passez avec la souris sur le drapeau et attendez l'affichage de l'information) :

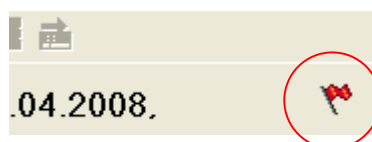


Figure 44 Enregistrement non valable

Si un enregistrement vient d'être créé et affiché dans le masque individuel, la fenêtre ne peut pas être quittée tant que l'enregistrement soit déclarée comme valable ou soit effacé. Si vous fermez la fenêtre d'un enregistrement incomplet, le message suivant s'affiche:



Figure 45 Un enregistrement incomplet ne peut pas être sauvegardé

Si l'on clique sur **Oui** l'enregistrement est effacé. **Non** permet de compléter l'enregistrement.

Si l'on saisi dans une liste (par exemple les prestations) les enregistrements incomplets s'affichent

comme suit, mais ils ne sont pas effacés si l'on quitte la liste:

Prestations ouvertes aujourd'hui									
<input checked="" type="checkbox"/> Jour simple <input type="checkbox"/> Intervalle <span>←</span> lundi , 28. avril 2008 <span>→</span>									
Projet	Phase	Date	Collaborateur	type de prestation	Texte	Heures	Taux	Honoraires	
COMINSTALL	1 ANA	28.04.2008	Keller			0:00	180.00	0.00	
1-TRASTA	1 - PLANNINI	28.04.2008	Keller			0:00	180.00	0.00	
*		28.04.2008	Keller						

Figure 46 Deux enregistrements non valables dans la liste.

Si l'on ferme le programme Vertec, le message suivant s'affiche :

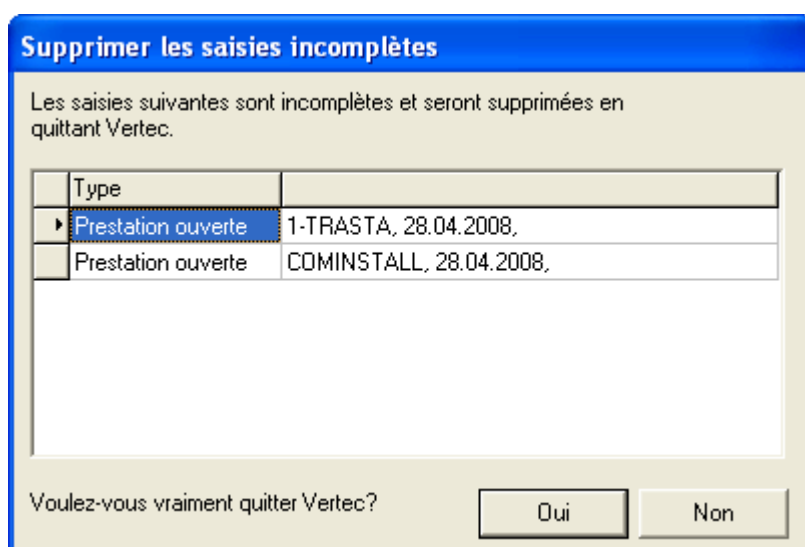


Figure 47 Affichage des enregistrements incomplets

Si vous cliquez sur **Oui**, les enregistrements incomplets sont effacés. **Non** permet de retourner au programme et de compléter les enregistrements. Par un double-clic dans la liste, vous pouvez confortablement ouvrir le détail de l'enregistrement et le modifier.

Quelques conditions sont introduites de manière standard dans Vertec et ne peuvent pas être modifiées. Un exemple est le fait qu'une prestation doit obligatoirement être attribuée à un projet. D'autres conditions ne sont pas obligatoires et vous pouvez en créer vous-mêmes et ainsi définir la validité des données dans Vertec.

Tous les conditions se trouvent dans le dossier **Conditions** du dossier **Paramétrages**:

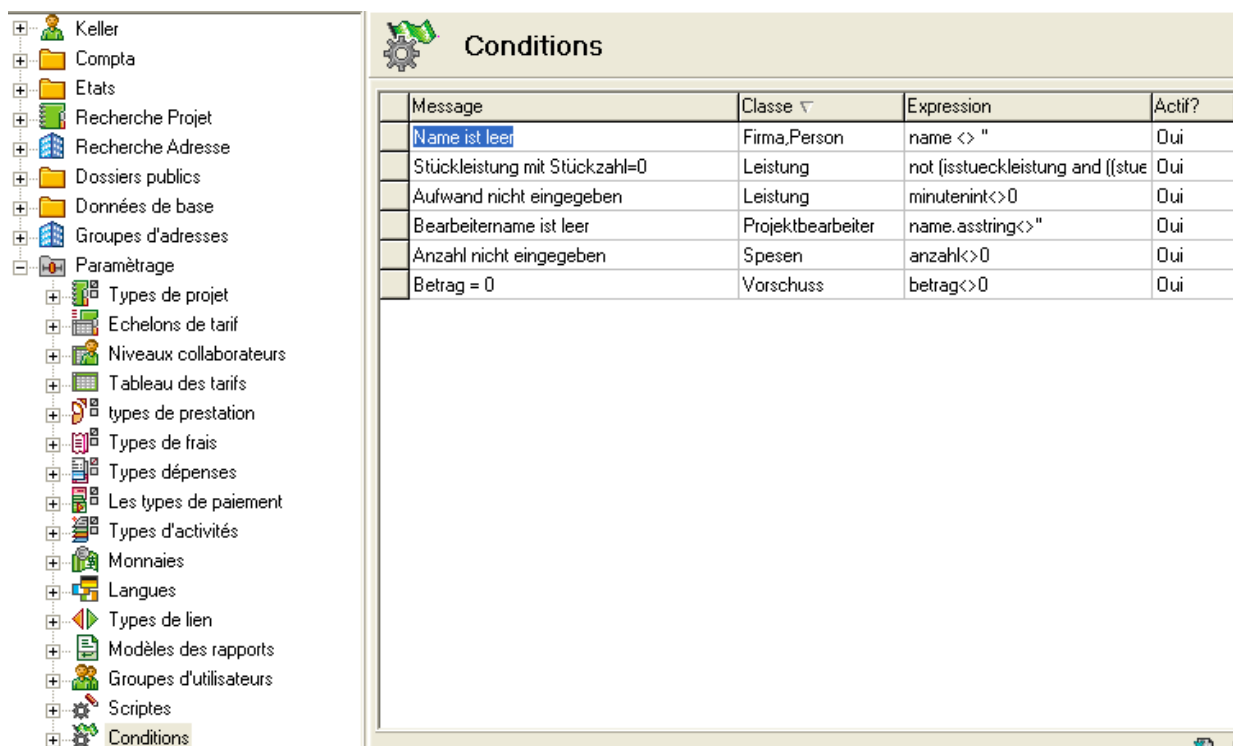


Figure 48 Dossier Conditions dans Paramétrage

Un clique droit de la souris permet de créer une nouvelle. Une condition existante est éditée par un double-clic.

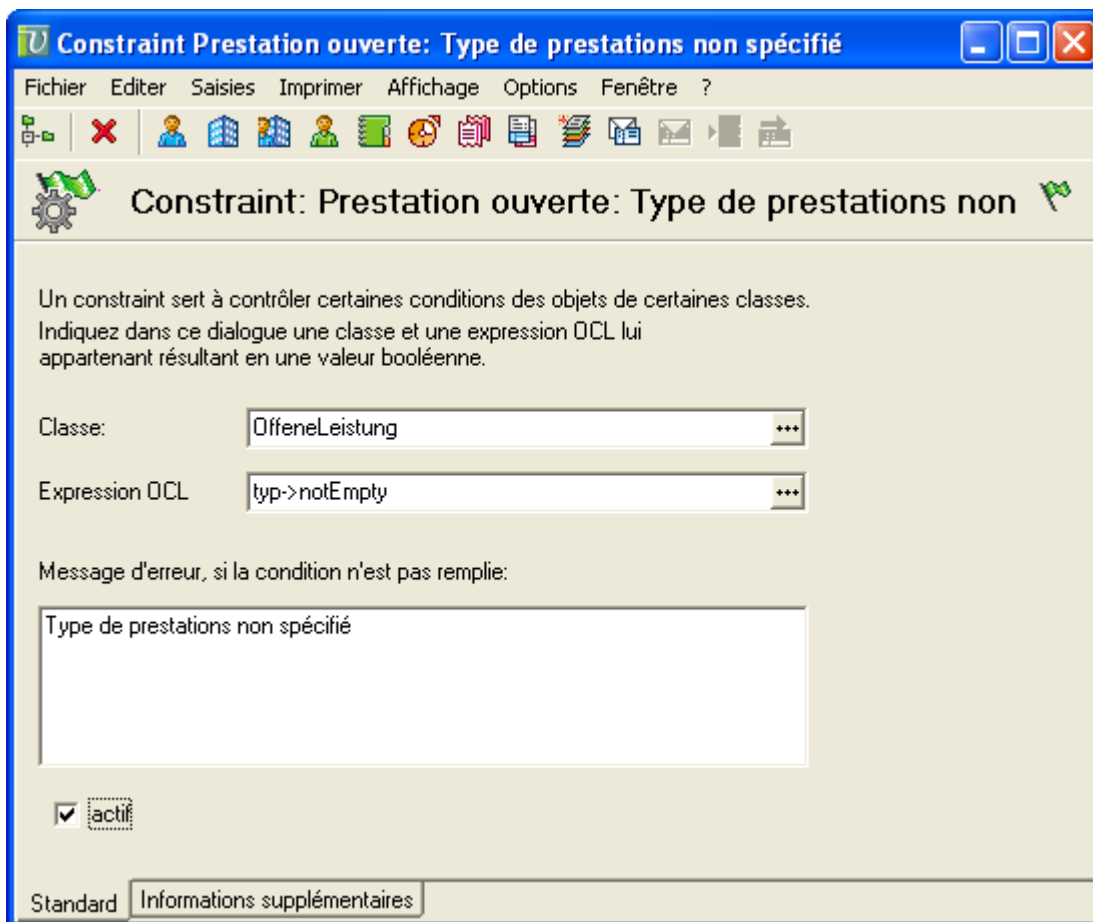


Figure 49 Une condition

Classe	Indique la classe à laquelle la condition s'applique
Expression	Indique l'expression OCL qui définit la validité. L'expression OCL indique doit retourner une valeur booléenne (Vrai/Faux). L'exemple teste si la prestation possède un code de prestation.' L'opérateur <code>-&gt;notEmpty</code> indique si le lien n'est pas vide.
Message d'erreur	Le message qui apparaît à l'écran
Actif	Indique si la condition est actif c'est-à-dire actuellement en utilisation. Si vous n'utilisez plus une condition, il suffit de la désactiver.

## 7.1 Exercices

*Exercice 19* Créez une condition qui définit qu'un texte doit être saisi avec chaque prestation.

*Exercice 20* Vous aimeriez que le code de projet commence par "AB-". Comment procédez-vous? Conseil: l'opérateur `->substring` permet de définir une suite de signes (voir 9.3 Opérateurs à page 52).

## 8. D'autres possibilités de customizing

### 8.1 Générateur de numéros

Le générateur de numéros dans Vertec permet de remplir chaque champ dans Vertec avec un numéro consécutif automatique. Le générateur de numéro remplit le champ si un enregistrement est créé à l'aide de la fonction „Nouveau“.

Il peut s'agir d'un champ déjà existant (p. ex. le champ **Numéro** de la facture), ou il peut s'agir d'un champ complémentaire généré individuellement (p.ex. un numéro de projet automatiquement généré).

Le générateur de numéros est paramétré à l'aide de **Options > Paramètres système**. Le dialogue se présente comme suit :

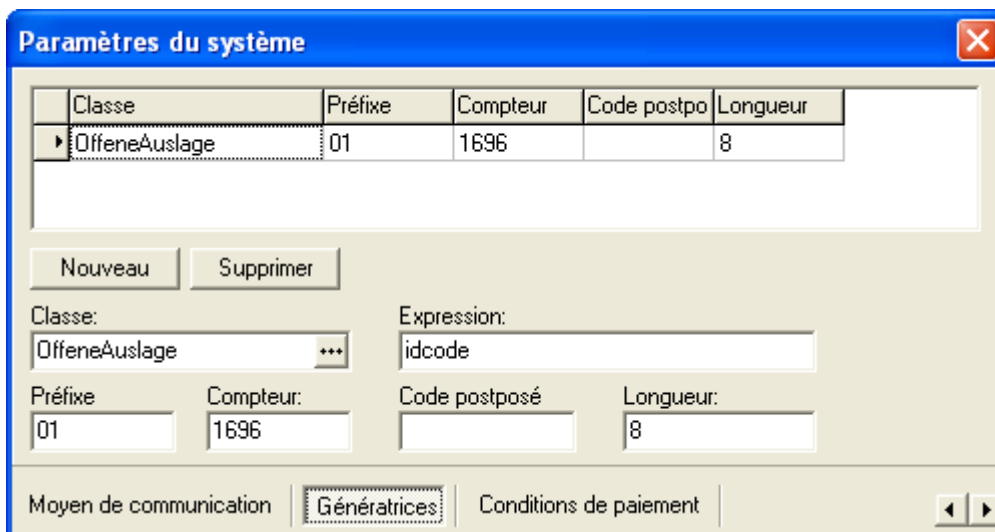


Figure 50 Exemple d'un générateur de numéros

Par **Neu/Nouveau** et **Löschen/supprimer** de nouveaux générateurs sont définis ou effacés. Le générateur sélectionné dans la liste peut être paramétré dans la partie inférieure du masque.

Classe	Nom de classe pour laquelle est prévu le générateur, (p.ex. <b>Projekt</b> ou <b>Projektphase</b> ).
Expression	Expression OCL du champ qui doit être rempli par le générateur de numéros.
Préfixe	Préfixe devant le numéro proprement dit. Vous pouvez aussi utiliser des lettres.
Compteur	Le compteur actuel. Si rien n'est indiqué, le système commence à 1.
Postposé	Un postposé qui suit le numéro. Vous pouvez aussi utiliser des lettres.
Longueur	La longueur totale du numéro (y compris préfixe et post-fixe).

L'exemple ci-dessus produit le numéro **01000015** dans le champ **idcode** sur une nouvelle dépense ouverte.

### 8.2 Reports

Vertec connaît 3 types de rapports :

	AVANTAGES	INCONVENIENTS
Rapports internes (Interneer Bericht)	Totalement intégré dans Vertec, rapide	Pas adaptable et pas enregistrable
Report-Builder (Standard Bericht)	Totalement intégré dans Vertec. Création simple de rapports. Enregistrable	Pas très répandu (le choix de rapports est encore restreint)
Office-Reports (Word- ou Excel)	Layout / contenu adaptable et enregistrable.	Adaptation du contenu complexe. Un peu lent car toujours besoin de Word ou Excel

Tous les rapports qui doivent apparaître dans le menu contextuel ,Imprimer' doivent être enregistrés. Les rapports enregistrés se trouvent sous **Paramétrages > Modèles de rapports**.

Vous trouvez ici un court aperçu des différents types de rapports. Nous vous recommandons de suivre le cours **Rapports et scripts/Berichte und Scripts** pour avoir une formation complète sur le sujet.

### Reports internes

Ils ne peuvent pas être adaptés mais le nom du rapport qui apparaît dans le menu peut être adapté (1), ainsi que le fait que le rapport s'affiche ou pas (2) :

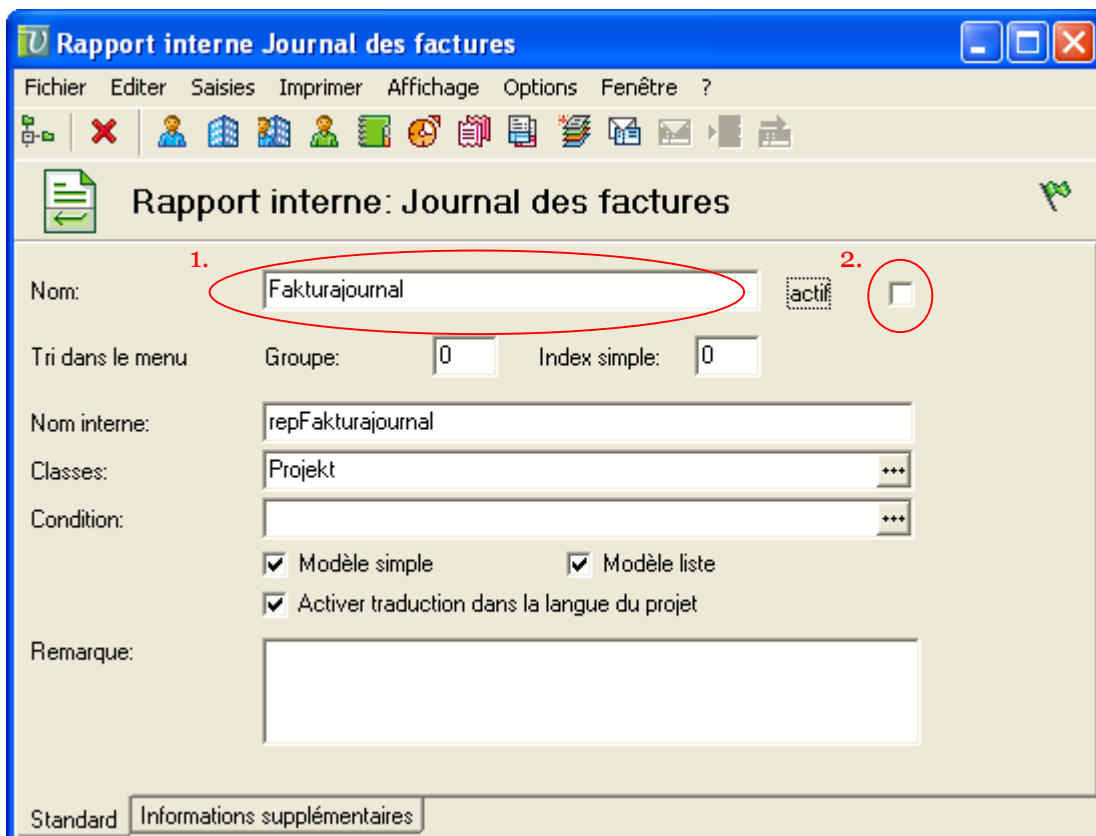


Figure 51 Les propriétés d'un rapport interne

### Report-Builder

Des rapports Report-Builder sont des rapports intégrés dans Vertec qui peuvent être adaptés à

l'aide du Report-Designer intégré. Il est aussi possible de créer de nouveaux rapports. Ces rapports sont définis essentiellement par **OCL**.

### Word-Reports

Le mécanisme du Word-Report de Vertec permet de définir des rapports individuels dans MS Word et de les exécuter directement depuis Vertec. Un rapport Word se base toujours sur un modèle de document (.dot) de Word. Ce rapport contient des éléments particuliers qui sont interprétés.

Une multitude de modèles de rapports sont installés automatiquement par Vertec. Vous les trouvez dans le répertoire **Reports** de votre répertoire d'installation de Vertec.

Afin que ce mécanisme puisse fonctionner correctement, un modèle de document Word particulier **VertecReport.dot** doit être chargé automatiquement lors du démarrage de Word. VertecReport.dot contient la macro **DoReport** qui se charge de l'exécution du rapport Word. Ce fichier doit se trouver dans le répertoire Vertec (le setup de Vertec le fait automatiquement).

### Excel-Reports

Contrairement à Word, Vertec ne connaît pas de générateur intégré de rapports pour Excel. Un rapport Excel se base toujours sur un modèle de document (.xlt) et doit contenir une macro qui s'appelle **DoReport** écrit spécialement pour le rapport. Cette macro est appelée lors de l'exécution du rapport et s'occupe de l'insertion des données dans le rapport.

Un exemple pour un rapport Excel dans Vertec est le plan des ressources qui est intégré en standard dans Vertec. Vous le trouvez dans le dossier Paramétrages/Modèles de rapports. Le rapport lui-même se trouve dans le répertoire des rapports et s'appelle Ressourcesplan.xlt. Le menu **Outils > Macro > VisualBasic-Editor** vous permet d'atteindre l'éditeur Visual Basic et le code de la macro.

### Enregistrement de rapports Office dans Vertec

Afin qu'un rapport Office puisse être exécuté dans Vertec, il doit être enregistré. Ceci se fait par la création d'un objet **Office-Bericht**. Un clic droit de la souris sur modèles de rapport dans le dossier Paramétrages de Vertec permet d'en créer un.

Des rapports Word- et Excel sont installés et enregistrés de manière identique dans Vertec. L'extension du fichier détermine si c'est Word ou Excel qui est démarré pour l'exécution du rapport.

**Rapport Office: Facture (Word)**

Nom: Facture (Word) actif

Tri dans le menu Groupe: 0 Index simple: 0

Classes: Rechnung

Condition:

Modèle simple  Modèle liste

Répertoire pour enregistrement: %dokpfad%

Désignation pour enregist. autom.

Français\Facture.dot	%nummer%.doc	FR	Projets
Français>Note d'honorai	%nummer%.doc	FR	Mandats
Italiano\Fattura.dot	%nummer%.doc	IT	Projets

Nouveau... Modifier... Supprimer

Afficher dialogue  Lien sur l'adresse

Lien sur l'objet principal

Proposer par défaut "créer activité"

Type d'activité: Vente

Remarque:

Standard Informations supplémentaires

Figure 52 Enregistrement d'un rapport Office dans Vertec

Les réglages suivants sont possibles :

**Nom:** Nom, sous lequel le rapport s'affiche dans le menu 'Imprimer'.

**Actif:** Seuls les rapports actifs apparaissent dans le menu. Si vous ne voulez pas utiliser un rapport, il suffit de le désactiver ici.

**Classes:** nom de classe pour laquelle le rapport est prévu. Le rapport n'apparaît que dans le menu „Imprimer“ de cette classe.

**Condition:** Il est possible d'indiquer des conditions pour dire quand un rapport s'affiche dans le menu „Imprimer“ ou pas. Ceci se fait à l'aide d'une expression OCL. Un cliquer sur le bouton avec les 3 points ouvre l'éditeur d'expressions.

La condition doit fournir un résultat Oui/Non (booléenne). Si la condition est remplie, le rapport apparaît dans le menu „Imprimer“ selon l'enregistrement, sinon pas.

En cas de rapports de liste, le rapport n'est affiché que si la condition s'applique à tous les enregistrements de la liste.

**Modèle simple/modèle de liste:** indique si un rapport s'exécute sur un seul objet ou sur une liste d'objets de ce type.

**Tri dans le menu:** en cas d'utilisation de beaucoup de rapports, le menu peut rapidement devenir surchargé. Pour améliorer l'organisation, les rapports peuvent être triés et groupés lors de l'enregistrement.

Le **groupe** est activé si un numéro de groupe est saisi. Tous les rapports avec le même numéro de groupe sont réunis dans un seul groupe dans le menu. Les différents groupes sont affichés dans l'ordre des numéros de groupe.

En standard, le tri à l'intérieur du groupe est alphabétique. Si vous souhaitez un autre tri, vous pouvez le faire par la saisie de chiffres dans le champ **index simple**.

**Chemin pour l'enregistrement automatique :** proposition de chemin pour l'enregistrement du document du rapport. La valeur **%dokpfad%** introduit le chemin défini pour les documents (valeur de l'attribut "dokpfad") de l'objet sur lequel est exécuté le rapport. Vous avez aussi la possibilité d'y introduire une expression OCL convenante (entourée par **%** ).

**Concerne pour enregistrement automatique :** Vous pouvez y indiquer un ‚concerne‘ qui est introduit lors de l'enregistrement automatique du document. Là également, l'utilisation d'une expression OCL (entourée de **%**) est possible.

**Afficher dialogue:** si oui, un dialogue est affiché avant l'exécution du rapport. Vous pouvez alors modifier les informations concernant le chemin d'enregistrement, les remarques et les autres options.

**Proposer par défaut "créer activité" :** si **Oui**, lors de l'exécution du rapport une activité est automatiquement créée. Celle-ci est reliée en fonction des réglages avec l'objet suivant:

**Lien sur objet principal:** l'activité est reliée avec l'objet à partir duquel elle a été démarrée.

- **Lien sur l'adresse:** l'activité est liée avec l'adresse à laquelle le document est envoyé.

**Type d'activité:** Vous pouvez mettre ici un type d'activité standard pour l'activité rapport. Celle-ci remplace la valeur définie sous Paramètre du système > CRM/Activités > **Type d'activité standard pour documents (Rapports Word)** pour ce rapport. Le type d'activité peut aussi être remplacé manuellement lorsqu'apparaît la fenêtre de dialogue du processus d'impression.

## Modèles de documents

Sous **Modèles de documents** vous trouverez une série de fichiers qui servent comme modèles pour des rapports. Chaque langue définie dans le système aura ses propres rapports. Le réglage des mo-

dèles sont modifiables par **Edit**.

En plus du chemin vers le modèle de document, vous pouvez aussi modifier le nom proposé d'enregistrement (Nom d'enregistrement) pour un nouveau document créé. En plus de textes fixes, vous pouvez aussi utiliser des expressions OCL (entouré de %). A la place des valeurs témoins, vous pouvez utiliser des expressions OCL quelconques, basées sur l'objet actuel du rapport (dans l'exemple de la lettre de projet depuis le projet). En plus, on peut utiliser dans la valeur-témoin OCL la variable ,concerne', qui contient la valeur actuelle du concerne du dialogue d'impression du rapport/lettre en série.

### Modèle Excel standard

Pour l'exportation d'affichages de listes Vertec vers Excel, vous pouvez enregistrer un modèle Excel standard. Ce modèle doit s'appeler VertecExport.xlt et se trouver dans le répertoire des modèles de rapports (en standard le répertoire Reports de l'installation Vertec). Si un domaine appelé VertecS-tard est défini dans le modèle, le rapport commence à cet endroit.

## 8.3 Scripts

Vertec utilise la Microsoft Scripting Engine (VBScript), afin d'exécuter des scripts définis par l'utilisateur. Depuis le VertecScript, le modèle entier des objets Vertec est accessible. Ceci se passe comme l'accès depuis Word à travers une interface (interne) COM /ActiveX.

Des scripts peuvent être créés et exécutés individuellement dans l'éditeur ou chargés et ensuite exécutés ou être enregistrés comme objets de script dans la base de données. Les objets script sont enregistrés pour un certain type de données et apparaissent ensuite dans le menu du type d'enregistrement. L'utilisateur les verra donc comme une fonctionnalité de Vertec.

Une multitude de scripts utiles se trouvent dans notre Knowledgebase <http://www.vertec.ch/support>. Nous vous recommandons notre cours **Rapports et scripts/Berichte und Scripts** afin d'obtenir une formation approfondie sur cette fonctionnalité.

## 8.4 Interface COM

L'interface COM (ou ActiveX) permet de commander Vertec depuis l'extérieur de manière analogue qu'un VBScript. Vertec Script utilise la même interface COM pour les objets Vertec

L'application Vertec (Vertec.exe) peut être utilisée depuis d'autres applications comme out-of-process COM Server. La Typelibrary est contenu dans Vertec.exe. Elle s'appelle **Vertec Typelibrary** et doit être référencée (p.ex. pour MS Excel dans l'éditeur VB **par Outils > Macro > Visual Basic-Editor** et **Outils > Verweise**).

Nous vous recommandons notre cours **Rapports et scripts/Berichte und Scripts** pour apprendre plus de l'interface COM.

## 9. Références: Connaissance de base OCL

Le genre d'accès le plus utilisé aux données (objets) du noyau Vertec se fait à l'aide de **OCL**. OCL représente **Object Constraint Language** et a été développé initialement dans le cadre du standard ULM afin de formuler des conditions de validité pour des objets modélisés. A cause de ses possibilités orientées objet, OCL peut aussi être utilisé comme langage d'interrogation.

Le noyau Vertec contient un interpréteur OCL qui traite des expressions OCL comme texte et qui retourne des résultats depuis les objets Vertec. L'interpréteur OCL utilise les informations du modèle de base Vertec pour contrôler et analyser les expressions.

OCL est utilisé dans Vertec à différents endroits :

- Réglages de listes
- Dossiers à expressions
- Scripts
- Reports Word
- Reports dans Report-Builder
- HTML Templates (Vertec Web Edition)
- Accès à Vertec depuis l'extérieur via l'interface COM
- Dans le serveur XML SOAP

Une expression OCL est toujours analysée en fonction d'un objet déterminé et retourne un résultat d'un type donné. Des types possibles de résultats sont :

- Nombre
- String (chaîne de caractères)
- Boolean (valeur logique respectivement valeur Oui / Non)
- Objet
- Liste de chiffres, Strings, Booleans ou objets

Pour l'affichage de résultats (p.ex. dans colonnes de listes, voir ci-dessous), le résultat est automatiquement transformé en un string. En cas de nombres et strings, cette transformation est simple. Des valeurs booléennes sont représentées comme **Y** (yes) ou **N** (no). Pour des objets, l'affichage standard String apparaît dépendant de la classe d'objet (pour les projets par exemple le code). La représentation de listes comme string affiche la taille de la liste.

En cas d'expressions dans un dossier à expressions, le type de résultat doit impérativement être une liste d'objets.

### 9.1 Members

Un Member est une propriété d'un objet. C'est soit une liaison (association) avec un ou plusieurs objets ou un champ de données (attribute) de l'objet.

Partant d'un objet, un Membername pour l'accès à un Member peut être indiqué. Si une telle référence vers le Member n'est pas placée au début de l'expression, elle doit être séparée par un point. Les noms des Members doivent toujours être écrits en minuscules.

```
code
```

dans une expression de liste d'une liste de projets affiche le code du projet dans la colonne correspondante.

```
kunde.name
```

également dans une liste de projets affiche chaque fois le nom du client. Il s'agit ici d'une référence de liaison (kunde/client) avec un attribut de référence qui suit (Attribut **name** de la classe **Adresseintrag**).

Par l'alignement successif de références de Members dans une expression OCL, de multiples liaisons dans le modèle d'objets Vertec peuvent être interrogés.

Un genre un peu spécial de Members dans Vertec sont les attributs calculés de listes. Ceux-ci sont représentés comme type **ObjectList** dans l'éditeur OCL. Si une référence de Member se réfère sur une telle liste dans une expression, il faut alors utiliser en plus l'opérateur **list** vers cette référence.

Par exemple les prestations d'une phase de projet sont attribuées par la liaison `offeneLeistungen` et `verrechnetLeistungen`. En plus existe la liste calculée `leistungen`, qui affiche le contenu combiné des deux liaisons. Pour l'affichage du nombre de prestations d'une phase de projet (colonne d'une liste de phase) l'expression suivante est appropriée:

```
leistungen.list->size
```

## 9.2 Listes de classes

Une liste de classe est une liste de tous les objets d'un certain type (classe), p.ex. tous les projets ou collaborateurs d'une installation Vertec.

OCL fait la différence entre majuscules et minuscules. Tout ce qui est en majuscules sera compris par l'interpréteur OCL comme nom de classe (type). OCL connaît l'opérateur **allInstances** sur une classe qui donne une liste de toutes les instances d'une classe.

```
Projekt->allInstances
```

fournit une liste de tous les projets.

Puisque les listes de classe sont souvent utilisées dans Vertec, nous avons créé un raccourci. Si un nom de classe est écrit en minuscules, celui-ci est directement interprété comme liste de classe.

```
projekt
```

signifie aussi une liste de tous les projets (par opposition à `Projekt` qui signifie le type proprement dit).

Les listes de classe sont souvent utilisées comme point de départ dans les expressions de dossiers à expressions. La plupart des dossiers à expressions prédéfinis dans Vertec (p.ex. **Données de base-Stammdaten > collaborateurs-Bearbeiter**) contiennent une simple liste de classes comme expression OCL.

**ATTENTION!** Il faut faire attention avec les listes de classes, car il y a des classes dans Vertec qui peuvent comporter un nombre important d'objets. L'indication d'une liste de classe dans OCL provoque le chargement de tous les objets de cette classe dans la mémoire. Ceci peut donc provoquer un besoin énorme en mémoire et amener une très mauvaise performance surtout dans les classe s **OffeneLeistung/prestations ouvertes** ou **VerrechnetLeistung/prestations facturées**.

## 9.3 Opérateurs

Les opérateurs exécutent certaines transformations des valeurs OCL. L'opérateur dans une expression est séparé par une flèche qui le précède (->). Si aucun danger de confusion avec des Members existe, le point comme séparateur est aussi accepté.

Basés sur une liste, différents opérateurs qui modifient la liste, existent dans OCL. D'autres opérateurs qui ne sont pas mentionnés ici, apparaissent dans l'éditeur OCL dans la liste **champs disponibles**.

L'opérateur **size** fournit la taille (nombre d'enregistrements) d'une liste. L'expression

```
projekte->size
```

comme colonne d'une liste dans une liste d'adresses donne le nombre de projets de chaque client.

L'opérateur **select** permet de filtrer le contenu d'une liste sur la base d'une condition.

```
adresseintrag->select(projekte->size > 0)
```

Comme expression d'un dossier à expressions fournit toutes les adresses auxquelles sont attribués des projets (c.-à.-d. des clients). L'argument de l'opérateur **select** doit donner une valeur logique (vrai/faux).

L'opérateur **first** fournit le premier enregistrement d'une liste. L'opérateur **first** est le plus souvent utilisé si – sur la base d'une sélection - un objet doit être utilisé comme résultat. Puisque l'opérateur **select** fournit toujours une liste comme résultat, il faut encore utiliser **first** afin d'obtenir un seul objet comme résultat.

Exemple:

```
phasen->select(code='OFFERTE')->first.offeneleistungen->size
```

affiche dans une une liste de projets , le nombre de prestations ouvertes de la phase **OFFERTE**

L'opérateur **orderby** permet de trier les listes.:

```
adresseintrag->orderby(projekte->size)
```

Remarque concernant l'opérateur **orderby**: Utilisé comme expression dans un dossier à expression, l'opérateur n'a aucune signification, puisque la liste s'occupe elle-même du tri (voir aussi 3.1).

Partant d'une liste, on peut aussi indiquer une référence vers un Member. Une référence vers un Member est ensuite utilisée sur chaque objet de la liste et le résultat est une liste avec des éléments du type de Member. Exemple.

```
projekt.kunde
```

fournit, dans une dossier à expressions, une liste de tous les clients. Le désavantage de cette liste est que les clients apparaissent partiellement à double puisque l'expression amène tous les clients par projet.

Ceci peut être évité en utilisant l'opérateur **asSet**. Celui-ci transforme une liste vers un ensemble (dans le sens de la théorie des ensembles) :

```
projekt.kunde->asSet
```

fournit une liste univoque de tous les clients.

On a souvent la tâche de transformer une suite de signes dans un chiffre etc. Ci-après une liste

d'opérateurs de conversion disponibles :

<code>-&gt;asString</code>	(aussi <code>.asString</code> ): transforme un chiffre ou une date en un string (suite de signes).
<code>-&gt;stringToFloat</code>	Transforme une suite de signes en une date. Si la suite de signes n'est pas un nombre, le résultat sera 0.

L'opérateur `isEmpty` indique si un Linkmember est vide. Ainsi l'expression

```
projektleiter->isEmpty
```

utilisée sur un projet avec une valeur booléenne (vrai, faux) donne comme résultat si un chef de projet est attribué (faux) ou pas (vrai).

Un autre opérateur très utile est l'opérateur `listToString`. Cet opérateur permet de présenter des listes comme strings. Ainsi l'expression

```
phasen.code->listToString(',')
```

exécuté sur un projet (p.ex. via les paramétrages de listes) fournit la liste des codes de phases, séparés par virgule.

Le résultat d'une expression est souvent un string (suite de signes). L'opérateur `toLowerCase` transforme la suite de signes en minuscules, l'opérateur `toUpperCase` en majuscules. Un opérateur très important dans ce contexte est l'opérateur `->substring('Anfang', 'Ende')`. Cet opérateur découpe les signes d'une suite de signes entre le début et la fin. Ainsi l'expression

```
code->substring(1,1)
```

exécutée sur un projet fournit la première lettre du code de projet.

## Opérateurs de date

Une série d'opérateurs de date existent pour des calculs de date souvent employés.

Par exemple pour l'affichage de l'année de saisie de l'enregistrement actuel. Vertec enregistre toujours la date de création d'un Member `creationdatetime`. L'expression suivante prend la `creationdatetime` et renvoie l'année:

```
creationdatetime->year
```

Les opérateurs de date peuvent aussi être très utiles en liaison avec des chemins d'enregistrements de rapports Office, voir aussi chapitre 8.2)

Ci-après une liste d'opérateurs de date disponibles :

<code>date</code>	L'opérateur <code>date</code> fournit la date du jour
<code>dateToFloat</code>	Transforme une valeur de date/temps en un chiffre à virgule flottante. La part entière du chiffre est le nombre de jours depuis le 30.12.1899. La part décimale indique l'heure.
<code>floatToDate</code>	Transforme un chiffre en une valeur date/temps.
<code>strToDate</code>	Transforme une date comme suite de signes en une valeur de date.

Year	Calcule l'année d'une valeur de date.
Month	Calcule le mois (1-12) d'une valeur de date.
Day	Calcule le jour d'une valeur de date.
incMonth(anzahl)	Ajoute le nombre de mois à une date.
incDay(anzahl)	Ajoute le nombre de jours à une date.
firstOfYear	Calcule pour une date la date du premier jour de l'année (1er janvier).
lastOfYear	Calcule pour une date la date du dernier jour de l'année (31 décembre).
firstOfQuarter	Calcule le premier jour du trimestre.
lastOfQuarter	Calcule le dernier jour du trimestre.
firstOfMonth	Calcule le premier jour du mois.
lastOfMonth	Calcule le dernier jour du mois.
mondayOfWeek	Calcule la date du lundi d'une semaine.
encodeDate(year, month, day)	Calcule une date sur la base de chiffres pour année, mois et jour.

## Comparaisons

Comparaison de 2 valeurs (dans la plupart des cas une référence vers un member et une constante, mais aussi member-member) se présentent souvent dans les expressions OCL, le plus souvent comme condition dans un opérateur `select`.

La forme la plus simple d'une comparaison est la comparaison de l'identité sous forme du signe `=`.

```
projekt->select (code='ABC')
```

fournit une liste de tous les projets, dont le code correspond exactement au string `ABC`. Normalement, cette liste ne devrait contenir qu'un projet. La requête n'est donc pas vraiment utile.

Des comparaisons qui n'analysent que des parties de strings sont plus intéressantes. Pour ce faire, l'opérateur `->sqllike` est utile. En principe, il n'a rien à faire avec SQL, sauf que le format du string-modèle respecte le standard SQL. `->sqllike` utilise pour la comparaison un string-modèle qui peut contenir de signes 'Platzhalter'. Le signe `%` indique toujours un string partiel quelconque.

```
projekt->select (code->sqllike('AB%'))
```

fournit tous les projets, dont le code commence par `AB`.

D'autres possibilités de comparaison sont fournies par les opérateurs connus pour plus grand, plus petit ou différent (`<`, `>`, `<>`). Des comparaisons sont possibles à l'aide de strings et de chiffres. La comparaison de valeurs booléennes n'a pas beaucoup de sens, puisqu'une valeur booléenne peut déjà être indiquée en principe comme condition:

```
projekt->select (aktiv.asstring='Y')
```

fonctionne certe, mais est plus compliqué que la formulation plus simple:

```
projekt->select(aktiv)
```

## Conditions

Une expression OCL peut aussi contenir des conditions. Puisqu'une expression doit toujours avoir une certaine valeur, seules les constructions `if ....then.... else .... endif` sont possibles ceci en opposition à des langages de programmation. Un exemple d'une condition **if** suit dans le prochain chapitre concernant les opérateurs-types.

## Opérateurs-type

Toute une série d'opérateurs dans OCL servent à la conversion de types d'objets. Il va de soi que le type d'un objet ne peut pas être modifié, OCL n'est qu'un langage d'interrogation (heureusement). Par contre, il peut être nécessaire d'indiquer une conversion de type afin de rendre valable une expression OCL.

Un exemple fréquent pour cela sont des listes d'enregistrements d'adresses. La classe **Adresseintrag** est la classe de base pour les classes **Person (personne)**, **Firma (entreprise)**, **Kontakt (contact)**, **EinfacheAdresse (adresse simple)**. Dans la plupart des cas, les listes d'adresses sont du type **Collection(Adresseintrag)**. Nous voudrions maintenant représenter dans une liste d'adresses, l'attribut **vorname (prénom)** d'un objet **EinfacheAdresse (adresse simple)**. L'expression OCL

```
vorname
```

dans les colonnes d'une liste d'adresses provoque le message d'erreur `vorname is not a member of Adresseintrag`, ce qui est vrai.

Afin d'afficher malgré tout l'information souhaitée (seulement depuis des adresses simples). On pourrait indiquer l'expression suivante:

```
if oclIsTypeOf(EinfacheAdresse) then oclAsTpe(EinfacheAdresse).vorname else '' endif
```

L'expression se réfère à l'objet en question dans la liste et contient directement deux opérateurs-type. **oclIsTypeOf** contrôle si l'objet est d'un type précis et **oclAsType** amène l'interpréteur OCL à considérer cet enregistrement d'adresse comme adresse simple et d'accepter le member **vorname (prénom)**.

## 9.4 Constantes

La constante

```
now
```

affiche la date système du jour avec l'élément de temps (partie après la virgule), la constante

```
date
```

la date système du jour sans l'élément de temps.

Malheureusement, OCL ne connaît pas les constantes telles que `TRUE` et `FALSE`. Celles-ci peuvent par contre être créées facilement. Ainsi

```
1=1
```

donne toujours `TRUE`, tandis que

```
1=0
```

donne toujours FALSE .

## 9.5 Les classes les plus importantes

Ce chapitre de références indique des classes les plus importantes, les Members les plus importants avec un descriptif court de la signification de l'expression correspondante.

### Classe „UserEintrag“ (toutes les classes)

Tous les objets Business dans Vertec comme projets, factures, adresses etc. possèdent une classe de base commune nommée **UserEintrag**. Ceci signifie que toutes les classes héritent les propriétés (Links, Members) de cette classe et par conséquent possèdent ces propriétés.

#### LINKS

aktivitaeten	Les activités liées avec cet enregistrement (Liste de <b>Aktivitaet/Activité</b> )
ordner	Liste des dossiers dans lesquels se trouve cet enregistrement (Liste de <b>Ordner/Dossier</b> )

#### MEMBERS

dokpfad	chemin (calculé), vers lequel un document exécuté sur cet objet doit être enregistré (string).
---------	--

### Classe „Projekt“

#### LINKS

kunde	Lien vers le client du projet. Attention: Client est de type <b>AbstractAdresseintrag</b> , pour l'utilisation normale, il faut le transformer en <b>Adresseintrag</b> (avec l'opérateur <b>-&gt;olcAsType</b> )
projektleiter	Lien vers le chef de projet ( <b>Projektbearbeiter</b> )
hb	Lien vers le collaborateur principal ( <b>Projektbearbeiter</b> )
hbstv	Lien vers le remplaçant du collaborateur principal ( <b>Projektbearbeiter</b> )
sekr	Lien vers secrétaire ( <b>Projektbearbeiter</b> )
bearbeiter	Lien vers les collaborateurs liés à ce projet (Liste de <b>Projektbearbeiter</b> )
offeneLeistungen	Lien vers les prestations ouvertes du projet (Liste de <b>OffeneLeistung</b> )
verrechneteLeistungen	Lien vers les prestations facturées du projet (Liste de <b>VerrechneteLeistung</b> )
--- idem pour frais, dépenses ---	offeneSpesen, verrechneteSpesen; offeneAuslagen, verrechneteAuslagen
Rechnungen	Lien vers les factures du projet (Liste de <b>Rechnung</b> )

Phasen	Lien vers les phases de projet du projet (Liste de <a href="#">Projektphase</a> )
Vorschuesse	Lien vers les avances du projet (Liste de <a href="#">Vorschuss</a> )
Tarifstufe	Lien vers les échelons de tarifs ( <a href="#">Tarifstufe</a> )
Waehrung	Lien vers la devise du projet ( <a href="#">Waehrung</a> )

## MEMBERS

Aktiv	Est-ce que le projet est-il actif ? (boolean)
Code	Projektcode (string) code du projet
Beschrieb	Projektbeschreibung (string) descriptif du projet
Betreffend	Betreffend (string) concernant du projet
Bemerkung	Bemerkung (memo) remarque
Rechnungsadresstext	Texte de l'adresse qui est utilisée pour la facture (soit du client soit de l'adresse de facturation (string)
summeoffeneleistung	Somme des honoraires des prestations ouvertes (correspond à <a href="#">offeneleistungen.wertext-&gt;sum</a> ) (currency)
summinutenint	Somme des heures saisies des prestations ouvertes (correspond à <a href="#">offeneleistungen.wertext-&gt;sum</a> ) (currency)

--- tous les sum. Member représentent de telles sommes ---

## Classes „OffeneLeistung“, „VerrechneteLeistung“, „Leistung“

Les deux classes [OffeneLeistung](#) et [VerrechneteLeistung](#) sont des sous-classes de la classe [Leistung](#) et possèdent à peu près les mêmes propriétés. Des liens individuels comme par exemple [projekt](#) ne sont définis seulement [OffeneLeistung](#) et [VerrechneteLeistung](#), mais pas sur [Leistung](#). Si vous souhaitez lire ces liens sur une liste de prestations, il faut transformer l'objet dans la classe effective ([OffeneLeistung](#) ou [VerrechneteLeistung](#)) (avec [->oclIsTypeOf](#) contrôler et avec [->oclAsType](#) transformer).

## LINKS

projekt	Lien vers le projet ( <a href="#">Projekt</a> )
phase	Lien vers la phase de projet ( <a href="#">Projektphase</a> )
typ	Lien vers le type de prestation ( <a href="#">Taetigkeit</a> )
rechnung	Lien vers la facture ( <a href="#">Rechnung</a> )

## MEMBERS

datum	Date de la prestation (DateTime)
-------	----------------------------------

text	Texte de la prestation (string)
minutenint	Temps saisi en minutes (Aufwand) (integer)
minutenext	Temps facturable, temps externe (Aufwand) en minutes (integer)
wertint	Honoraires saisis dans la devise du projet (currency)
wertext	Honoraires facturables, honoraires externes (currency)
ansatzint	Taux horaire saisi (currency)
ansatzext	Taux horaire facturable, taux horaire externe (currency)
pauschal	Une prestation forfaitaire ? (boolean)
ansatzkosten	Le taux des coûts normalisé (currency)
kosten	Les coûts qui ont été générés par cette prestation (currency)
extfixiert	Les valeurs externes sont fixées (boolean)

### Classes "Spesen" (frais), "Auslagen" (dépenses)

Ces deux classes frais (OffeneSpesen et VerrechneteSpesen) et dépenses (OffeneAuslagen et VerrechneteAuslagen) possèdent pratiquement les mêmes liens et membres comme la **prestation** et ne sont pas traitées ici de manière détaillée.

### Classe "Projektphase" (phase de projet)

#### LINKS

Projekt	Lien vers le projet ( <b>Projekt</b> )
offeneLeistungen	Lien vers les prestations ouvertes ( <b>Leistung</b> )
verrechneteLeistungen	Lien vers les prestations facturées ( <b>Leistung</b> )
--- idem pour frais, dépenses:	offeneSpesen, verrechneteSpesen, offeneAuslagen, verrechneteAuslagen
bearbeiterphasen	Les collaborateurs attribués directement à cette phase. Le lien ne se dirige pas vers le collaborateur, mais vers la classe Bearbeiterphase, puisque celle-ci peut contenir le budget etc. (Liste de <b>Bearbeiterphase</b> )
parentphase	Lien vers la phase supérieure, si cette phase est une sous-phase ( <b>Projektphase</b> )
subphasen	Lien vers toutes les sous-phases de cette phase (Liste de <b>Projektphase</b> )
rechnung	Lien vers la facture, si cette phase est facturée comme phase forfaitaire ( <b>Rechnung</b> )

taetigkeiten

Les types de prestations attribués à cette phase (**Taetigkeit**). Par contre, si vous voulez accéder aux objets liés, sur lesquels sont définis les budgets etc., vous utilisez **taetigkeitphaselink**.

## MEMBERS

code	Code de la phase (string)
beschreibung	Description de la phase (string)
startdatum	Date de départ de la phase (DateTime)
enddatum	Date de fin de la phase (DateTime)
abgeschlossen	Est-ce que la phase est clôturée? (boolean)
abschlussdatum	Date à laquelle la phase a été clôturée (DateTime)
verrechenbar	Est-ce que la phase est-elle facturable ?
planwerttext	Budget d'honoraires (Currency)
sumwerttext	Honoraires effectivement accumulés (Currency)
planminutenint	Budget d'heures en minutes (integer)
summinutenint	Temps effectivement accumulé (integer)

## Classe "Projektbearbeiter" (collaborateur)

### LINKS

eigprojekte	Projets pour lesquels ce collaborateur est chef de projet (Liste de <b>Projekt</b> )
erfprojekte	Projets sur lesquels ce collaborateur a le droit de saisir des prestations (Liste de <b>Projekt</b> ). Attention: <b>erfprojekte</b> est de type <b>ObjecList</b> . Pour la liste de projets, il faut utiliser l'expression suivante : <b>erfprojekte.list-&gt;oclAsType(Projekt)</b>
adresse	Lien vers l'adresse de ce collaborateur ( <b>Adresseintrag</b> )
gruppen	Groupes d'utilisateurs auxquels ce collaborateur est attribué ( <b>BenutzerGruppe</b> )
offeneleistungen	Lien vers les prestations ouvertes de ce collaborateur ( <b>OffeneLeistung</b> )
verrechneteleistungen	Lien vers les prestations facturées de ce collaborateur ( <b>VerrechneteLeistung</b> )
--- idem pour frais :	offeneSpesen, verrechneteSpesen, offeneAuslagen, verrechneteAuslagen
aktivitaeten	Lien vers les activités de ce collaborateur ( <b>Aktivitaet</b> )

stufe                      Lien vers le niveau de collaborateur (**Bearbeiterstufe**)

## MEMBERS

aktiv                      Ce collaborateur est-il actif? (boolean)

ansatzext                Le taux horaire externe défini pour ce collaborateur (également les autres Members **ansatz..**) (Currency)

name                      Nom du collaborateur (string)

loginname                Loginname du collaborateur (string)

kuerzel                    Le nom court du collaborateur (string)

## 9.6 Aperçu des classes d'adresses (Adressklassen)

Les liens des différentes classes d'adresses pour la gestion des adresses sont un peu complexes, c'est pour cela que ces liens et dépendances sont présentés dans un diagramme de classes:

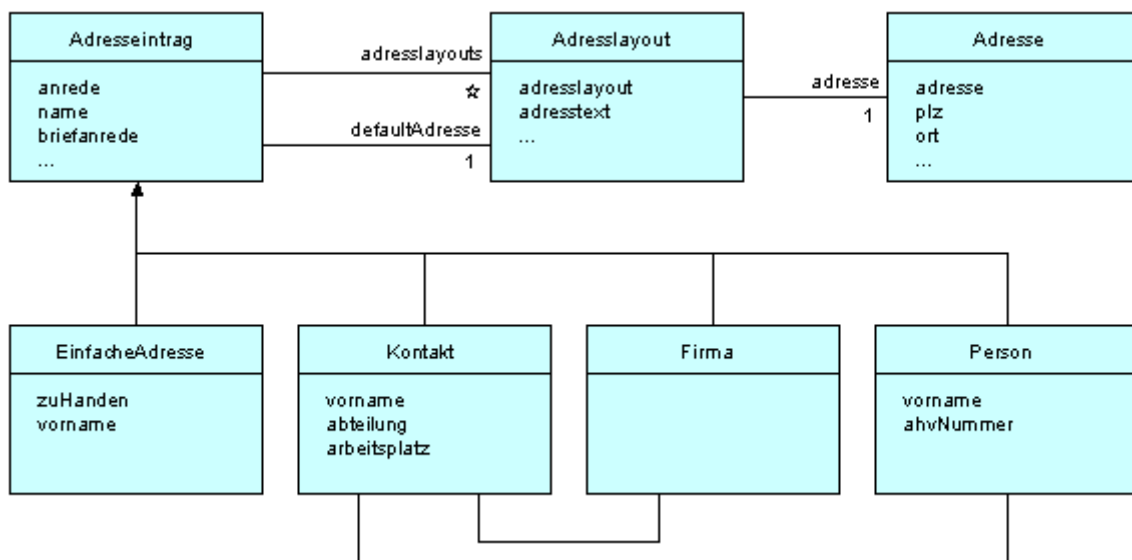


Figure 53      Diagramme de classe Adresseintrag

La classe **Adresseintrag** contient tous les 4 types d'adresses **EinfacheAdresse** (adresse simple), **Kontakt** (contact), **Firma** (entreprise), **Person** (personne). Le contact possède un lien vers l'entreprise et la personne qui le composent.

### Classe „Adresseintrag“

## LINKS

projekte                    Lien vers les projets de cette adresse comme adresse de client (Liste de **Projekt**)

bearbeiter                Lien vers le collaborateur comme adresse de ce collaborateur (**Projektbearbeiter**)

defaultadresse Lien vers l'adresse par défaut de cette adresse (**Adresse**)

### MEMBERS

adresstext Le layout de l'adresse (indique les informations complètes de cette adresse (string))

alias Alias, ce champ se trouve sur l'onglet **informations supplémentaires** est rempli automatiquement par Vertec pour la recherche (string)

anrede Civilité (Anrede) (string)

briefanrede Intitulé de lettre (string)

grussformel Formule de politesse (string)

name Le nom de l'adresse (Nom de l'entreprise, Nom de famille) (string)

standardadresse L'adresse (Rue) de l'adresse par défaut (string)

--- idem pour NPA, lieu, pays, canton: standardplz, standardort, standardland, standardkanton

standardemail L'adresse e-mail du moyen de communication standard **E-Mail** (string)

--- idem pour téléphone, fax et homepage : standardtelefon, standardfax, standardhomepage

stellung Position de l'adresse (important pour contacts et personnes) (string)

titel Titre (p.ex. Dr.) (string)

### Classe "Person" (personne)

Puisque la classe **Person** est une sous-classe de la classe **Adresseintrag**, elle possède tous les liens et membres de cette classe. A côté de cela, elle possède encore les siens :

### LINKS

kontakte Lien vers les contacts de cette personne (Liste de **Kontakt**)

### MEMBERS

vorname Prénom de la personne (string)

### Classe "Firma" (entreprise)

Puisque la classe **Firma** est une sous-classe de la classe **Adresseintrag**, elle possède tous les liens et

members de cette classe. A côté de cela, elle possède encore les siens :

### LINKS

kontakte Liens vers les contacts de cette entreprise (Liste de **Kontakt**)

### MEMBERS

mwstnr Le numéro de TVA (string)

### Classe "Kontakt" (contact)

Puisque la classe **Kontakt** est une sous-classe de la classe **Adresseintrag**, elle possède tous les liens et members de cette classe. A côté de cela, elle possède encore les siens :

### LINKS

firma Entreprise de ce contact (**Firma**)

person Personne de ce contact (**Person**)

### MEMBERS

vorname Prénom de la personne de contact si aucune personne n'est sélectionnée (string)

abteilung Département du contact (string)

arbeitsplatz Place de travail du contact (string)

## 9.7 Aperçu des classes de dossiers

L'organisation des données dans Vertec se base sur une structure de dossiers dont la présentation rappelle un peu l'explorateur de Windows. Les objets, qui se présentent comme des répertoires (dossiers) dans Windows sont tous des sous-classes de **Container**. La propriété essentielle d'un container est le fait qu'il peut contenir un ou plusieurs **Usereinträge**.

Les sous-classes les plus importantes de Container sont **Ordner**, **ExpressionOrdner** et **LinkContainer**.

L'interface calcule le contenu d'un container toujours par l'attribut de liste sur la classe Container **eintraege** des données. Cet attribut n'est pas enregistré directement dans la base de données, mais est calculé chaque fois en fonction du type proprement dit du container. Seule la classe **Ordner** possède un lien vers **UserEintrag** dans le modèle de données.

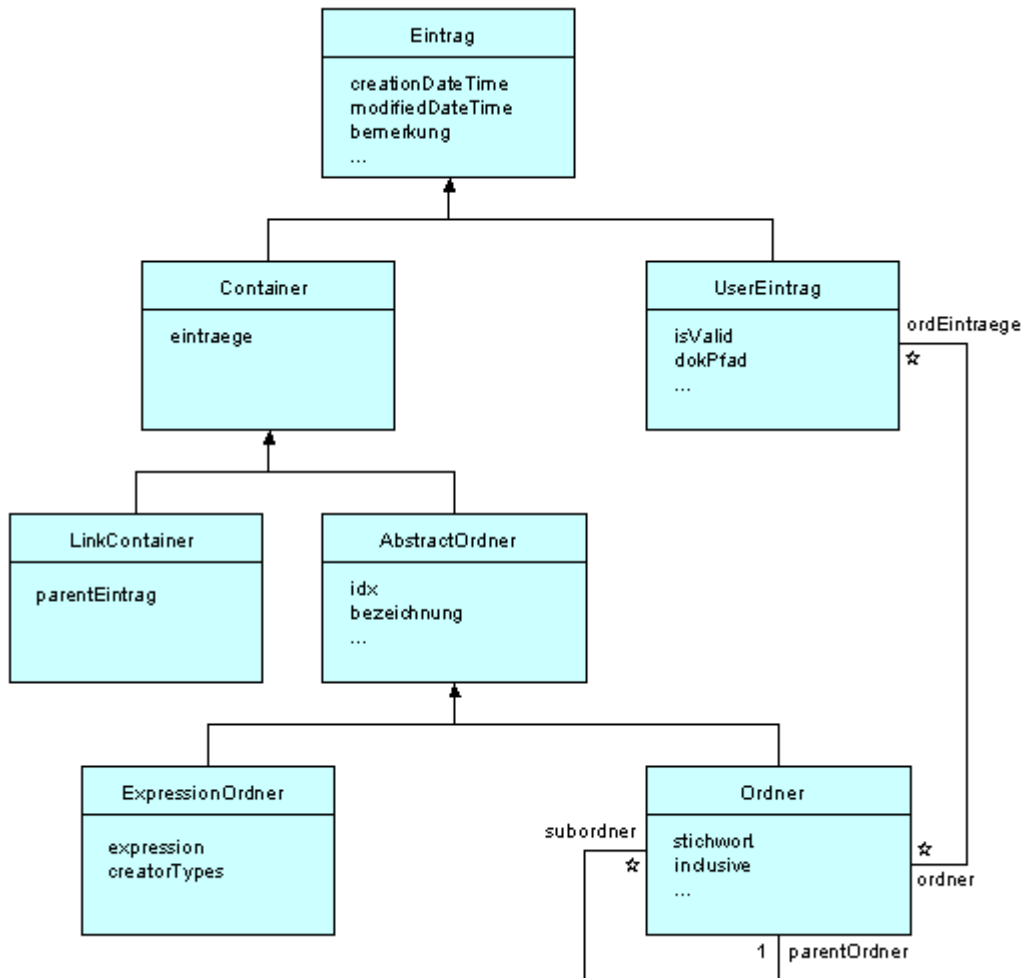


Figure 54 Diagramme de classe dossiers

Les dossiers (Ordner) sont des containers particuliers qui peuvent former une hiérarchie. C'est à dire qu'un dossier se trouve (presque) toujours dans un autre dossier. Les liens y relatifs s'appellent **subordner** – **parentordner**.

Un dossier détermine ses enregistrements sur la base des liens **ordner** – **ordEintraege**. C'est à dire : des enregistrements doivent être attribués à un dossier.

Un dossier à expression détermine la valeur de **eintraege** sur la base d'une **Expression OCL**. C'est à dire : Un dossier à expression calcule son contenu de manière dynamique sur la base des critères et compose ainsi soi-même ses enregistrements.

## 9.8 Exemples de modèles de classes

Projet – chef du projet - facture – client

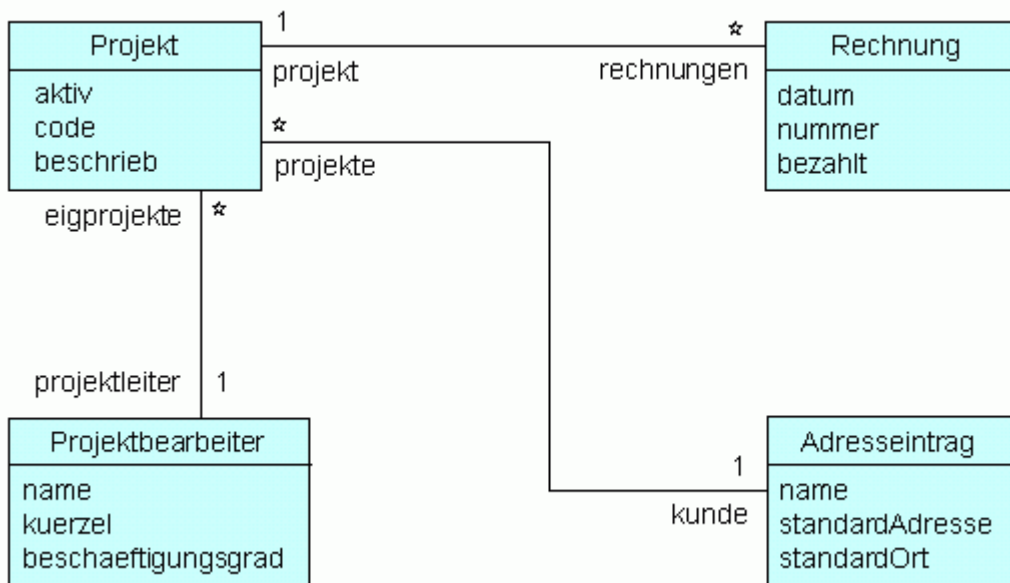


Figure 55 Relations Projekt - Projektleiter - Rechnung – Kunde.

**Activités**

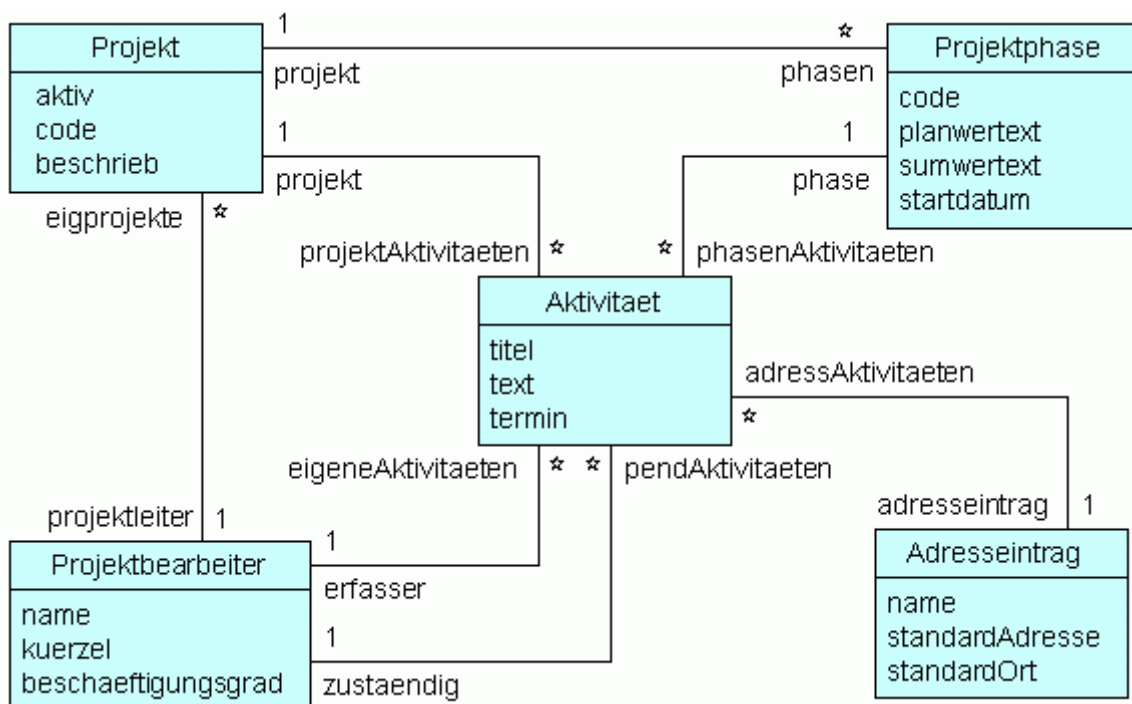
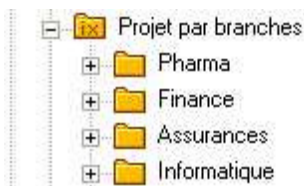


Figure 56 Schéma du modèle des activités

Vous trouverez d’autres exemples de modèles auprès de notre support Online sous <http://www.vertec.ch/de/support/kb/handbuch/modell/uml>

## 10. Solutions

Exercice	Solution
Exercice 1	Expressionordner mit der Expression: <code>projekt-&gt;select(aktiv)</code>
Exercice 2	Expressionordner mit der Expression: <code>projekt-&gt;select(aktiv).phasen</code>
Exercice 3	Expressionordner mit der Expression: <code>projektbearbeiter-&gt;select(eigprojekte-&gt;size &gt; 0)</code>
Exercice 4	Expression de colonne: <code>beschaeftigungsGrad</code> Pour le total général, mettre une coche à <code>totalisé</code> .
Exercice 5	Expression de colonne: <code>summeoffeneleistungen</code>
Exercice 6	Expression de colonne: <code>planWertExt – sumWertExt</code>
Exercice 7	Expression de colonne: <code>eigProjekte-&gt;select(aktiv)-&gt;size</code>
Exercice 8	Expression de colonne: <code>offeneLeistungen-&gt;select(bearbeiter = self.projektleiter).wertext-&gt;sum</code>
Exercice 9	Dossier à expression avec l'expression: <code>adresseintrag-&gt;select(projekte-&gt;size &gt; 0)</code>
Exercice 10	<b>Dossiers normaux</b> . Le dossier en dessus (Branchen) est marqué comme <b>dossier à rubriques</b> . Les sous-dossiers possèdent le type de données <b>Adresseintrag</b> . Exemple :



L'attribution se fait à l'aide de l'onglet **Rubriques** sur les adresses correspondantes.

Exercice 8 Dossier de recherche. Le dossier Kundenadressen est le dossier à expression de l'exercice 6, le dossier Banken est le dossier correspondant de l'exercice 7:



Exercice 12 Dossier à expression avec l'expression :  
 projekt->select(offeneLeistungen->size > 0)

Exercice 13 Sur le type de lien **Projekt > Vorschüsse** il faut mettre une coche du côté des avances pour **afficher toujours le container**.

## Exercice 14

Créer le Wrapper-Link-Typen suivant :

Expression: `eigProjekte.rechnungen->select(not bezahlt)`

## Exercice 15

Créez un generic Linktype:

Type de lien générique: RolleB - Conseillers projets

Fichier Editer Saisies Imprimer Affichage Options Fenêtre ?

Type de lien générique: RolleB - Conseillers projets

de

Désignation:  Tri:  Icône:

Classe:  Membre lien:

Expression:  Classe récipient:

Menu 'nouveau'  Afficher le récipient  
 Toujours afficher le récipient

à

Désignation:  Tri:  Icône:

Classe:  Membre lien:

Expression:  Classe récipient:

Menu 'nouveau'  Afficher le récipient  
 Toujours afficher le récipient

actif  Permettre des liens

Standard Informations supplémentaires

## Exercice 16

Créez le champ complémentaire suivant :

Classe de champ complémentaire: Prévion de marge c

Nom:

Désignation:

Type de champ:

Sélection type:  Expression:

Placement:

Classes:

Visibilité:  caché Condition:

L'accès à la liste de projets se fait à l'aide de l'expression **zusatzfeldcurr('Deckungsbeitrag')**.

Exercice 17 Créez le champ complémentaire suivant :

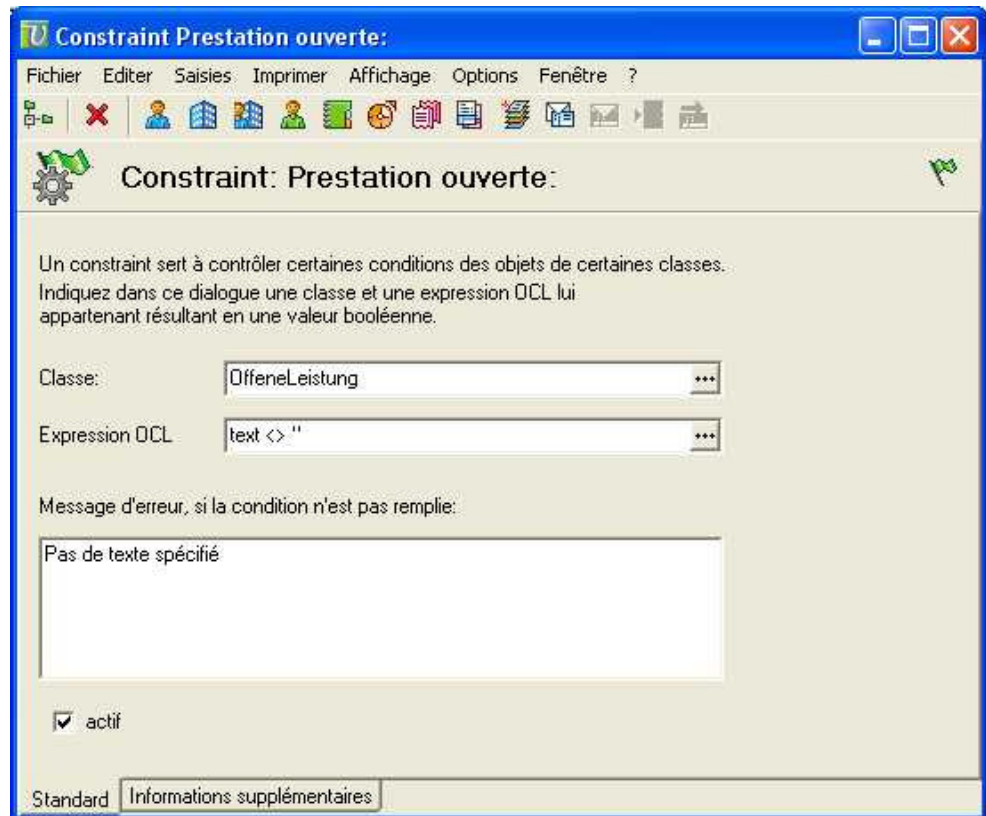
Exercice 18 Créez le champ supplémentaire suivant :

Les deux expressions de la liste de projet sont:

- nombre de phases de projet sans degré de finition :  
`phasen->select(zusatzfeldint("Degré de finition") = 0)->size`

- degré de finition le plus bas:  
`phasen->orderby(zusatzfeldint("Degré de finition"))->first.zusatzfeldint("Degré de finition")`

Exercice 19 Créez la condition suivante:



Exercice 20 Créez la condition suivante:

