



Crésus Salaires

31 - L'éditeur de calculs

31.1 - Opération

31.2 - Les totalisateurs

31.3 - Les tests

31.4 - Les conversions

31.5 - Les pourcentages

31.6 - Les dates

31.7 - Les chaînes de caractères


31.8 - Spécialités

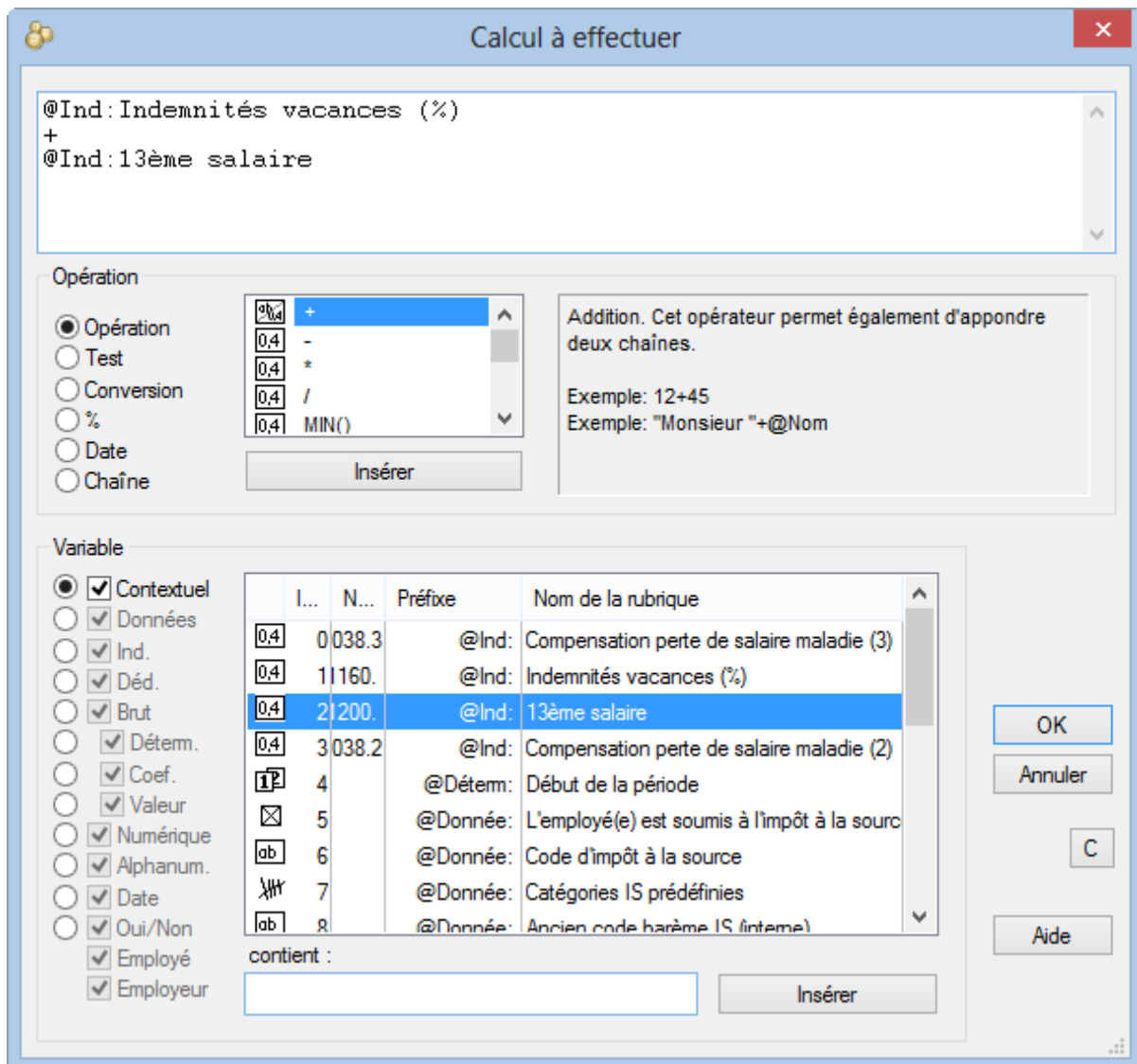
31 - L'éditeur de calculs

Des formules plus ou moins compliquées sont souvent employées, que ce soit pour calculer des montants, générer des phrases et pour toutes sortes d'informations.

Pour définir les calculs associés aux rubriques, utilisez le dialogue *Entreprise – Rubriques* (§19.7.4 Rubriques de type Indemnités et §19.7.7 Création de rubriques personnalisées).

L'éditeur de calculs est le même que vous travaillez dans les rubriques ou dans une présentation.

Dans une présentation, l'objet calcul  vous permet de placer une valeur, une chaîne de caractères, une date autre que les rubriques prévues par défaut. Le résultat d'un calcul dans une présentation peut seulement être affiché, il ne peut pas non plus être mémorisé pour un autre usage. Pour modifier un calcul existant dans une présentation, sélectionnez l'objet et cliquez sur l'icône « calcul » de la barre d'outils, ou double-cliquez sur l'objet.



La première zone du dialogue vous sert à construire votre formule: vous pouvez y taper ce que vous voulez au clavier. Vous devez toutefois scrupuleusement respecter les noms des rubriques, plutôt complexes, y compris leur préfixe : pour vous faciliter la tâche, utilisez de préférence les listes. Indiquez les chaînes de caractères entre guillemets.

La liste *Opération* vous présente toutes les opérations du module de calcul. Elles sont réparties en six sous-groupes :

- *Opération* : les quatre opérations de base et des totalisateurs
- *Tests* : opérateurs conditionnels
- *Conversion* : transformation des formats
- *%* : calculs de pourcentages
- *Date* : traitement des dates et heures

- *Chaîne* : traitement des chaînes de caractères

Chaque fonction sélectionnée affiche un texte d'aide dans la partie droite du dialogue. Les opérations sont détaillées ci-après.

La liste *Variable* affiche toutes les rubriques utilisées par Cresus. Elle est dynamique : lorsque vous tapez une partie du libellé de la rubrique dans le calcul ou dans la case éditable au bas du dialogue, Cresus affiche les rubriques correspondantes.

Vous pouvez utiliser les cases à cocher de la partie de gauche pour affiner votre recherche.

La liste affiche en priorité les rubriques récemment utilisées, et regroupe si possible les déterminants, coefficients et valeurs d'une même rubrique.

Pour ajouter l'élément sélectionné dans la formule en édition, cliquez sur le bouton *Insérer* ou double-cliquez sur l'élément.

Les rubriques vides, sans contenu, sont traitées comme des nombres nuls dans la plupart des cas. Il peut être nécessaire d'utiliser la fonction *DEFAULT* dans certains cas particuliers. Par exemple : $3 * @Coef:Heures\ maladie\ CM$ est égal à zéro s'il n'y a pas d'heures d'absence pour maladie. Il en va de même pour $CM\ 3 * DEFAULT (@Coef:Heures\ maladie\ CM ; 0)$, aussi égal à zéro en l'absence d'heures de congé maladie ; $3 * DEFAULT (@Coef:Heures\ maladie\ CM ; 2)$ donne 6 s'il n'y a pas d'heures d'absence pour maladie.

31.1 - Opération

Les quatre opérateurs de base (+, -, *, /) servent comme il se doit à additionner, soustraire, multiplier et diviser des nombres. Les multiplications et les divisions se font avant les additions et les soustractions.

Addition De plus, vous pouvez utiliser le + pour rassembler deux chaînes de

MIN() Cherche Valeur minimale dans une liste de nombres, exemple :

MIN(12;45;-3;28) résultat : -3

MAX() Cherche la valeur maximale dans une liste de nombres, exemple :

MAX(12;45;-3;28) résultat : 45

SOMME() Calcule la somme d'une liste de nombres, exemple :

SOMME(12;45;-3;28) résultat : 82

MOYENNE() Calcule la moyenne d'une liste de nombres, exemple :

MOYENNE(12;45;-3;28) résultat : 20.5

31.2 - Les totalisateurs

Les six fonctions suivantes font aussi partie des opérations.

Elles vous permettent d'obtenir non pas la valeur d'une rubrique pour un employé pour le mois courant, mais le cumul soit pour les employés, soit pour l'année soit encore pour le trimestre courant.

<p><i>TOTEMPLOYES()</i></p>	<p>Total pour tous les employés. <i>TOTEMPLOYES(@Brut:Salaires versé)</i> Vous pouvez aussi procéder au cumul pour les employés décrit au §30.7.9 Cumul sur les employés pour l'objet Calcul dessiné dans la présentation.</p>
<p><i>TOTANNEE()</i></p>	<p>Total pour toute l'année, limité aux périodes avec un salaire calculé. <i>TOTANNEE(@Déd:Cotisation AC)</i> donne le total des cotisations AC pour les salaires versés. Vous pouvez aussi procéder au cumul pour la période décrit au §30.7.8 Cumul sur l'année pour l'objet Calcul dessiné dans la présentation.</p>
<p><i>TOUTANNEE()</i></p>	<p>Total pour toute l'année courante, prend en compte aussi les mois sans salaire calculé. <i>TOUTANNEE(DEFAULT(@Déd:Cotisation AC ; 200))</i> vous donne le total des cotisations AC déjà calculées plus 200.– par période vide.</p>
<p><i>TOTTRIMESTRE()</i></p>	<p>Total pour le trimestre qui comprend le mois sélectionné <i>TOTTRIMESTRE(@Ind:Remboursement de km)</i></p>
<p><i>PRECEDENT()</i></p>	<p>Valeur d'une expression calculée pour la période précédente. <i>PRECEDENT(@Brut:Salaires versé)</i> vous donne le montant du salaire versé le mois passé. Au début de l'année, <i>PRECEDENT()</i> vous donne la valeur du dernier salaire de l'année précédente.</p>

TOTPRECEDENT()

Total de l'ensemble des périodes précédentes.
 TOTPRECEDENT(@Déd:Prime AANP) vous donne le montant total des primes AANP déjà prélevées les mois précédents.
 Cette fonction accepte un second paramètre optionnel pour limiter les périodes additionnées à celles qui ont le même critère.
 TOTPRECEDENT(@Coef:Jours de maternité ; @Déterm:Date de l'accouchement) vous donne le nombre de jours de maternité des périodes précédentes pour la même date d'accouchement.

Dans le calcul d'un salaire, ces opérations de total pour l'année ne dépassent pas le mois courant. Par exemple, un salaire calculé en février ignore les salaires de mars et des périodes suivantes.

CUMUL()

Calcule la somme de toutes les indemnités selon qu'elles sont soumises ou non à divers bruts déterminants. Exemples :
 CUMUL(« AVS ») vous donne le total des montants soumis à l'AVS
 CUMUL(« 2.1 ») vous donne le total des montants affichés au point 2.1 du certificat de salaire

Les termes utilisables avec la fonction CUMUL() sont les suivants :

<i>NET</i>	compris dans le salaire net
AVS	soumis à l'AVS
AC	soumis à l'assurance chômage


<i>LAA</i>	soumis à l'assurance accidents
<i>LAAC</i>	soumis à l'assurance accidents complémentaire
<i>LPP</i>	pris en compte pour la prime LPP
<i>FT</i>	soumis à l'assurance frais de traitement (ancienne)
<i>IJM</i>	soumis à l'assurance indemnité journalière maladie
<i>CPROF</i>	soumis à la contribution professionnelle
<i>IMP</i>	soumis à l'impôt à la source
<i>VAC</i>	pris en compte dans les indemnités vacances
<i>FERIE</i>	pris en compte dans les indemnités pour jours fériés
<i>13E</i>	pris en compte dans le calcul du 13 ^{ème} salaire
<i>FORFAIT</i>	pris en compte dans les frais AVS forfaitaires
<i>ALLOC.</i>	ajouté aux allocations familiales
<i>IJ.</i>	total des indemnités journalières
<i>TIERS</i>	total des prestations de tiers
<i>ACCMAL.</i>	total des indemnités accident et maladie

PTX.

indemnités proportionnelles pour les IS (§32 Calcul des impôts à la source)

Vous pouvez aussi utiliser le *code pour les accumulations* des bases personnalisées (§19.7.6 Rubriques de type Bases).

La fonction *CUMUL()* peut également traiter les positions de 1. à 13.2.3 liées au certificat de salaire.

Vous pouvez aussi utiliser tous ces termes dans le filtre  associé à *@Ind:** et *@Déd:** (§30.7.10 Filtre)

Si vous créez vos propres bases (§19.7.6 Rubriques de type Bases), vous pouvez spécifier le *Code pour les accumulations* et utiliser ce code pour la fonction *CUMUL()*.

Vous pouvez évaluer plusieurs critères en les séparant par le caractère & (fonction ET) qui s'obtient par Shift + 6 ou le caractère « barre verticale » | (fonction OU) qui s'obtient sur un clavier Windows par AltGr + 7.

AVS&LAA&LPP soumis à l'AVS et soumis à la LAA et soumis à la LPP

AVS|LAA|LPP soumis à l'AVS ou soumis à la LAA ou soumis à la LPP

Utiliser simultanément des & (ET) et des | (OU) n'est pas possible.

Vous pouvez cumuler des indemnités **non soumises** à un brut déterminant donné en utilisant le caractère **point d'exclamation !** devant le code.

!AVS indemnités non soumises à l'AVS

BRUT&!LPP indemnités payées à l'employé, mais non soumises à la LPP

Vous pouvez enchâsser les fonctions les unes dans les autres. Exemple :
TOTEMPLOYES(TOTANNEE(@Brut:salaire AC))
 vous donne le total pour tous les employés et pour toute l'année du salaire AC déterminant.

Vous pouvez associer cette procédure avec des tests (§31.3 Les tests) pour obtenir un total partiel sur une catégorie d'employés ou une période de dates. Par exemple pour avoir le total d'une valeur pour toutes les femmes :

TOTEMPLOYES(SI(@Sexe de l'employé=1 ; @Brut:salaire LAA ; 0))

Dans le cas où le sexe de l'employé est de 1 (c'est une femme), on additionne le salaire LAA, dans le cas contraire, on additionne zéro.

Autre exemple pour obtenir le montant d'une rubrique pour le mois précédent :

TOTANNEE(SI(@Mois courant=@Mois sélectionné-1 ; @Ind:salaire; 0))

Pour comprendre cette formule, il faut savoir que *@Mois sélectionné* correspond au mois où se trouve le curseur dans le tableau général. C'est donc le mois pour lequel on imprime le document. *@Mois courant* par contre passe de 1 à 12 (ou plus selon le nombre de périodes de paie) durant le calcul de *TOTANNEE*. La formule entre parenthèses sera donc calculée n fois, en changeant le mois courant à chaque fois. Lorsque le mois courant est égal au mois sélectionné moins un, on additionne l'indemnité voulue, sinon on ajoute zéro. Le résultat est donc la valeur de l'indemnité pour le mois précédent.

PRECEDENT(@Ind:salaire) est similaire, mais cherche la valeur précédemment payée en ignorant les mois sans salaires.

D'une manière similaire, on peut obtenir une valeur pour une période spécifique :

TOTANNEE(SI(@Mois courant=2; @Ind:salaire mensuel + @Ind:salaire horaire; 0))

vous donne le total des salaires pour février, quel que soit le mois actuellement sélectionné dans le tableau général.

31.3 - Les tests

Il vous est possible de comparer des éléments entre eux pour exécuter ensuite une opération ou une autre.

Le résultat d'un test est égal soit à 1 (auquel cas il est vrai) soit à 0 (auquel cas il est faux). Toute valeur supérieure à 0 est considérée comme vraie, et toute valeur nulle ou négative est considérée comme fausse.

SI(test; sivrai; sifaux)

Si le résultat de *test* est vrai, traite l'expression *sivrai*, sinon traite l'expression *sifaux* :

SI(@Brut:Salaires versé > 0; « à verser »; « à retenir »)

Signifie *Si le salaire est supérieur à 0., il faut écrire « à verser », sinon il faut écrire « à retenir »*

=

Teste si deux nombres sont **identiques**.

<>

Teste si deux nombres sont **différents**.

<

Teste si le premier nombre est **plus petit que** le second.

<=

Teste si le premier nombre est **plus petit ou égal** au second.

>

Teste si le premier nombre est **plus grand que** le second.

>=

Teste si le premier nombre est **plus grand ou égal** au second.

ET()

Opère une addition, ou **ET logique**, entre plusieurs expressions pour donner un résultat vrai.

ET(3 > 2 ; 4 < -1 ; 1 >= 0) donne 0 (« faux »), car 4 < -1 est faux.

OU()

Opère une sélection, ou **OU logique**, entre plusieurs expressions pour donner un résultat vrai.

OU(3 > 2 ; 4 < -1 ; 1 >= 0) donne 1 (« vrai »), car 3 > 2 et 1 >= 0 sont vrais.

NON()

Inverse le résultat d'un test. *NON(4<-1)* donne 1 (vrai).

<p><i>SEXE()</i></p>	<p>Donne 0 ou 1 selon le sexe de l'employé : 0 = homme, 1 = femme. Il est aussi possible d'utiliser @Donnée:Sexe de l'employé(e).</p>
<p><i>HOMME()</i></p>	<p>Évalue l'expression entre parenthèses pour la valeur applicable HOMME(@Donnée:Age retraité AVS) Donne l'âge de la retraite AVS pour les hommes.</p>
<p><i>FEMME()</i></p>	<p>Évalue l'expression entre parenthèses pour la valeur applicable FEMME(@Donnée:Age retraité AVS) Donne l'âge de la retraite AVS pour les femmes.</p>
<p><i>SISOUMIS() et SINONSOUUMIS()</i></p>	<p>Ces tests ne figurent pas dans la liste. Ils sont décrits au §31.8</p>

31.4 - Les conversions

ARRONDI(*nombre; arrondi; excès*)

Arrondit un nombre au multiple le plus proche, par excès ou par défaut. Si on omet le dernier paramètre, l'arrondi se fait par excès.
 ARRONDI(12.775) donne 12.80.
 ARRONDI(12.775; 0.05) donne 12.80
 ARRONDI(12.775; 0.05; 0) donne 12.75

Attention : le format de la rubrique affichée ou imprimée entraîne également un arrondi (§30.7.6 Format des rubriques).

INT(*nombre*)

Ne garde que la partie entière du nombre.
 INT(12.775) donne 12.

FRAC(*nombre*)

Ne garde que la partie fractionnaire du nombre.
 FRAC(12.775) donne 0.775.

DEFAULT(*rubrique ; valeur de substitution*)

Si la rubrique est vide, elle est substituée par la valeur spécifiée (voir la note au §31 L'éditeur de calculs).
 DEFAULT(@Ind:Salaire mensuel;100) donne 100 s'il n'y a aucune valeur dans la rubrique Salaire mensuel.

VIDE()

Donne un nombre vide (inexistant), peut être utile en relation avec la fonction DEFAULT() (voir la note au §31 L'éditeur de calculs).
 SI(@Coef:Jours travaillés>0
 ; @Coef:Jours travaillés
 ; VIDE()
)
 n'affiche rien au lieu de 0 si le nombre n'est pas plus grand que 0.

Donne la valeur 1 (vrai) si la rubrique est vide, 0 (faux) dans le cas contraire.

`SI(VIDE(@Ind:Salaire mensuel)`
 ; « à l'heure »
 ; « au mois »
)
 affiche (imprime) « à l'heure » si le salaire mensuel est vide et « au mois » dans le cas contraire.

<p><i>CHAINE(valeur ; décimales)</i></p>	<p>Convertit un nombre ou une date en une chaîne de caractères.</p> <p>Le paramètre optionnel des décimales vous permet de préciser le nombre de chiffres significatifs à afficher (imprimer).</p> <p>« Montant : » + <code>CHAINE(@Ind:Salaire mensuel;2)</code> « Nous sommes le » + <code>CHAINE(AUJOURDHUI())</code></p>
--	---

<p><i>CHAINEMOIS(mois ; langue)</i></p>	<p>Donne le nom du mois (1 à 12). Le second paramètre est optionnel et précise la langue voulue (0 = français, 1 = allemand, 2 = anglais, 3 = italien). Par défaut, la fonction donne le nom dans la langue du document.</p> <p><code>CHAINEMOIS(2)</code> donne « février », <code>CHAINEMOIS(7;1)</code> donne « Juli ».</p>
---	---

<p><i>CHAINEJDLS(jour ; langue)</i></p>	<p>Donne le jour de la semaine (1 à 7). Le second paramètre est optionnel et précise la langue voulue (0 = français, 1 = allemand, 2 = anglais, 3 = italien). Par défaut, la fonction donne le nom dans la langue du document.</p> <p><code>CHAINEJDLS(2)</code> donne « mardi », <code>CHAINEJDLS(1;2)</code> donne « Monday ».</p>
---	---

FINLIGNE() ou \n

Crée une fin de ligne, ce qui vous permet de mettre une chaîne de caractères sur deux lignes.

@E.Donnée:Nom de l'entreprise + FINLIGNE() +

@E.Donnée:Adresse de l'entreprise

affiche (imprime)

« Espitec SA

Chemin du Fontenay »

@E.Donnée:Nom de l'entreprise + « \n « +

@E.Donnée:Adresse de l'entreprise

est équivalent.

31.5 - Les pourcentages

$C\%(montant ; taux)$ Calcule le pourcentage d'un montant.
 $C\%(20;7)$ donne 1.40.

$A\%(montant ; taux)$ Ajoute un pourcentage à un montant.
 $A\%(20;7)$ donne 21.40,
 $A\%(20;-7)$ donne 18.60.

$S\%(montant ; taux)$ Calcule la valeur qu'avait un montant avant qu'on ne lui ajoute un pourcentage.
 $S\%(21.4;7)$ donne 20,
 $S\%(18.6;-7)$ donne 20.

$M\%(montant ; taux)$ Calcule la valeur ajoutée à un montant augmenté d'un pourcentage.
 $M\%(21.4;7)$ donne 1.40,
 $M\%(18.6;-7)$ donne 1.40.

$T\%(montant ; montant)$ Calcule le pourcentage entre deux montants.
 $T\%(21.4;20)$ donne 7,
 $T\%(18.6;20)$ donne -7.

31.6 - Les dates

Lorsque vous saisissez une année, si vous n'indiquez que la dizaine et l'unité de l'année, Cresus complète par le millénaire et le siècle, soit 19xx ou 20xx. Selon le type de date, le complément automatique n'est pas le même. Par exemple :

- Pour une date de naissance, les indications d'années 00 à 19 sont transformées dans les années 2000 à 2019, alors que les indications 20 à 99 sont transformées dans les années 1920 à 1999.
- Pour la date d'entrée, les indications d'années 40 à 99 sont transformées dans les années 1940 à 1999, alors que les indications 00 à 39 sont transformées dans les années 2000 à 2039.
- S'il s'agit d'une date de calcul du salaire, toutes les années sont transformées en 20xx.

Vérifiez donc toujours que l'année complète soit correcte.

De manière interne, le programme convertit les dates en nombres de jours depuis le 1.1.1900.

Les fonctions suivantes s'appliquent aux dates.

<p><i>MOIS(date)</i></p>	<p>Donne le numéro du mois (de 1 à 12) de la date en question. MOIS(DATE(3;2;2014)) donne 2. Le résultat est affiché sous forme de texte, à moins que la fonction soit utilisée dans une formule numérique. « Nous sommes en « +MOIS(AUJOURDHUI()) donne « Nous sommes en MOIS(DATE(3;2;2014)) + 3 donne 5. Pour afficher le rang du mois, il faut utiliser la fonction NOMBRE(MOIS(</p>
<p><i>JDLS(date)</i></p>	<p>Donne l'année (1900 à 2099) de la date en question. Si l'année est saisie avec 2 chiffres uniquement</p> <p>Donne le jour de la semaine (1 = lundi à 7 = dimanche) de la date en question. JDLS(DATE(28;3;14)) retourne 5 (vendredi).</p>

<p><i>ADDMOIS(date;nombre)</i></p>	<p>Convertit une date en numéro de semaine (0 à 53). La partie fractionnaire de jours en centièmes depuis le début de la semaine</p> <p>Ajoute un ou plusieurs mois à une date. ADDMOIS(DATE(15;5;18);2) donne 15.07.18 ADDMOIS(DATE(31;12;17);2) donne 29.02.2018 Le paramètre nombre est optionnel. Si on ne le précise pas, ADDMOIS</p>
<p><i>FINMOIS(date)</i></p>	<p>Donne la date du dernier jour du mois. FINMOIS(DATE(3;2;18)) donne 29.02.2018.</p>

<p><i>TEMPS</i>(heure ; minute ; seconde)</p>	<p>Donne le nombre de secondes de l'heure exprimée en heures, minutes et secondes.</p> <p>TEMPS(13;40;35) donne 49'235.</p> <p>Les paramètres absents sont complétés par l'heure courante.</p> <p>TEMPS(;0;0) correspond au début de l'heure courante.</p> <p>Convertit un temps donné en nombre d'heures (0 à 23). La partie fractionnaire est arrondie à l'heure.</p>
<p><i>MINUTE</i>(temps)</p>	<p>Donne le nombre de minutes (0 à 59) écoulées depuis l'heure pleine. Le paramètre optionnel exprime les secondes en 1/100^e de minute.</p> <p>MINUTE(MAINTENANT())</p> <p>MINUTE(TEMPS(13;40;35)) donne 40.58.</p>
<p><i>PERIODE</i>(date)</p>	<p>Donne un temps en nombre de secondes (0 à 59) écoulées depuis l'heure pleine.</p> <p>Limite une date à la période courante. Une date inférieure à la période prend la valeur de la fin de la période, une date supérieure à la période prend la valeur de la fin de la période.</p> <p>PERIODE(AUJOURDHUI())</p>
<p><i>FINPERIODE</i>()</p>	<p>Donne la date de début de la période courante. Le paramètre optionnel est à évaluer.</p> <p>FINPERIODE(11) donne la date de la fin de la 11^{ème} période.</p>
<p><i>JFERIE</i>(n)</p>	<p>Donne le n^e jour férié du mois courant, en fonction de la liste introduite dans les paramètres de configuration (§19.14 Jours fériés). Attention : le 1^{er} jour de la liste est le jour 1.</p>

31.7 - Les chaînes de caractères

<p><i>MAJUSCULE(chaine)</i></p>	<p>Transforme toute la chaîne de caractères en caractères majuscules. MAJUSCULE(« Bonjour ») donne « BONJOUR ».</p>
<p><i>MINUSCULE(chaine)</i></p>	<p>Transforme toute la chaîne de caractères en caractères minuscules. MINUSCULE(« Bonjour ») donne « bonjour ».</p>
<p><i>NOMPROPRE(chaine)</i></p>	<p>Met la première lettre de chaque mot de la chaîne de caractères en lettre majuscule et les autres lettres en minuscules. NOMPROPRE(« Bonjour monsieur louis ») donne « Bonjour Monsieur Louis ».</p>
<p><i>GAUCHE(chaine ; n)</i></p>	<p>Extrait n caractères de la chaîne de caractères en partant depuis la gauche de la chaîne. GAUCHE(« Bonjour »;3) donne « Bon ».</p>
<p><i>DROITE(chaine ; n)</i></p>	<p>Extrait n caractères de la chaîne de caractères en partant depuis la droite de la chaîne. DROITE(« Bonjour »;4) donne « jour ».</p>
<p><i>MILIEU(chaine ; position ; n)</i></p>	<p>Extrait n caractères de la chaîne de caractères depuis la position (0 à n-1). MILIEU(« Bonjour »;1;4) donne « onjo ».</p>
<p><i>LONGUEUR(chaine)</i></p>	<p>Donne le nombre de caractères que comporte la chaîne. LONGUEUR(« Bonjour ») donne 7.</p>
<p><i>TROUVE(chaine ; critere)</i></p>	<p>Indique où dans la chaîne de caractères se trouve le critère recherché. TROUVE(« Bonjour »; »jou ») donne 3. Si le critère est au début, donne la valeur 0. TROUVE(« Bonjour »; »Bon ») donne 0. Si le critère n'est pas trouvé, la fonction donne la valeur -1. TROUVE(« Bonjour »; »x ») donne -1.</p>

Remplace les occurrences du paramètre *critère* par le paramètre *chaîne de substitution*.

REPLACE(« Bonjour »; »jou »; »SOI ») retourne « BonSOIr ».

Il est possible de mettre plusieurs couples *critère – chaîne de substitution*, dans ce cas chaque *critère* trouvé dans la chaîne est

REPLACE(*chaîne ; critère ; chaîne de substitution*)

remplacé par la *chaîne de substitution* correspondante. REPLACE(« Bonjour Mme Bijou »; « jou » ; « soi »; « Mme » ; « M. ») donne « Bonsoir M. Bisoi ».

Le paramètre chaîne peut bien entendu être une rubrique.

REPLACE(@Donnée:Nom de l'employé; »jou »; »soi »).

<p><i>NOMBRE(chaîne)</i></p>	<p>Convertit une chaîne représentant un nombre, en nombre. NOMBRE(« 1'123.45 ») donne 1123.45.</p>
<p><i>MOT(chaîne ; position ; séparateurs)</i></p>	<p>Extrait le n^{ème} élément (selon sa position) d'une chaîne, en considérant le séparateur défini. MOT(« aa/bb/cc/dd »;2; »/ ») donne « bb ». Un séparateur ne peut pas être un mot ou une suite de caractères. Si on spécifie plusieurs séparateurs, chaque séparateur est considéré individuellement. Cette fonction donne la première occurrence trouvée. MOT(« aa+bb/cc/dd »;2; »/+& ») donne « bb ». Si le séparateur à considérer est un caractère spécial, comme un tabulateur, il faut chercher son code ASCII avec la fonction CODECAR() décrite ci-dessous. MOT(« aa<TAB>bb<TAB>cc/dd »;2;CODECAR(9)) retourne « bb ».</p>
<p><i>LIGNE(chaîne ; position)</i></p>	<p>Extrait la n^{ème} ligne (selon le paramètre <i>position</i>) d'une rubrique de plusieurs lignes. LIGNE(@Donnée:Adresse;2)</p>

CODECAR(code_ascii)
 Donne un caractère en fonction de son code ASCII. On peut donner une liste de codes. *CODECAR*(65;66;9;67) donne « AB<TAB>C ».

NOMRUBRIQUE()
 Donne le nom de la rubrique. Utile pour permettre la modification du nom de la rubrique par un calcul (§19.2 Lieux de travail). *NOMRUBRIQUE*()+ » « +@Donnée:Code d'impôt à la source
 Il est également possible de mettre une rubrique en paramètre, la fonction retourne alors le texte correspondant au nom de la rubrique. *NOMRUBRIQUE*(@Ind:Salaire horaire) donne le libellé de la rubrique tel qu'il a été modifié par l'utilisateur (§19.2 Lieux de travail).

TROUVEUNPARMI(chaîne ; caractères)
 Donne la position du premier caractère de la liste trouvé dans le texte. Le paramètre *caractères* est une chaîne donnant la liste des caractères à chercher. *TROUVEUNPARMI*(« A B,C;D » ; « ;, : ») donne 3, soit la position de la première virgule. L'espace n'est pas trouvé car on n'a pas mis d'espace dans la liste des séparateurs.

OTE(*chaîne ; position ; critères*)

Enlève les espaces, tabulateurs ou fins de ligne d'une chaîne de caractères.

OTE(» Bonjour «) donne «Bonjour».

Le paramètre *position* est optionnel :

S'il est omis, les caractères sont supprimés au début et à la fin de la chaîne.

S'il vaut 0, les caractères sont supprimés au début de la chaîne.

S'il vaut 1, les caractères sont supprimés à la fin de la chaîne. .

S'il vaut 2, les caractères sont supprimés au début et à la fin de la chaîne.

OTE(» Bonjour » ; 0) donne «Bonjour » .

OTE(» Bonjour » ; 1) donne « Bonjour».

OTE(» Bonjour » ; 2) donne «Bonjour».

Le paramètre optionnel *critères* est une liste de caractères qui seront tous supprimés s'ils se trouvent au début ou à la fin du texte traité.

Chaque caractère de *critères* est considéré individuellement. Dès que le caractère de la chaîne n'est pas un des caractères listés dans critère, le traitement s'arrête.

OTE(» Bonjour » ; 0 ; « jour ») donne « Bonjour».

OTE(» Bonjour » ; 1 ; « jour ») donne « Bon».

OTE(» Bonjour » ; 2 ; « ujr ») donne « Bon».

OTE(« Bonjour » ; 0 ; « ojr ») donne «Bonjour».

OTE(« Bonjour » ; 1 ; « ojru ») donne «Bonjour».

OTE(» Bonjour » ; 2 ; « jour ») donne « Bonjour » (le caractère espace n'est pas spécifié dans la liste caractères à supprimer.

OTE(« Bonjour » ; 2 ; « jo ur ») donne «Bon».

OTE(» rujo Bonjour » ; 2 ; « ur jo ») donne «Bon».

31.8 - Spécialités

Les fonctions décrites ci-dessous ne se trouvent pas dans les listes des opérations du dialogue. Vous les utilisez en les inscrivant directement dans la formule.

SISOUMIS(« mode » ;
expression)

Évalue l'expression en ne prenant en compte que les rubriques soumises aux bruts déterminants selon le mode (voir la fonction *CUMUL* §31.2 Les totalisateurs).

SISOUMIS(« LPP » ; @Ind:Prime ancienneté + @Ind:Salaire mensuel) calcule la somme des valeurs trouvées dans les champs pour autant qu'elles soient soumises à la base LPP.

Si l'une des valeurs est soumise mais pas l'autre, la valeur soumise sera prise en compte. Il n'est pas nécessaire que les 2 rubriques soient soumises.

SISOUMIS(« AVS » ; @Ind:Prime ancienneté + @Ind:Salaire mensuel) fonctionnera également si l'une des deux rubriques est vide car *vide* est considéré comme zéro pour une addition (voir note §31 L'éditeur de calculs).

SIRUBSOUMISE(« mode » ;
@rubrique)

Rend 1 ou 0 selon que la rubrique donnée en second paramètre est soumise ou non selon le mode (voir la fonction *CUMUL* §31.2 Les totalisateurs).

SIRUBSOUMISE(« LPP » ; @Ind:Prime ancienneté) donne 1 si la prime d'ancienneté est soumise à la LPP.

Évalue l'expression en ne prenant en compte que les rubriques non soumises aux bruts déterminants selon le mode (voir *CUMUL* §31.2 Les totalisateurs).

SINONSOU MIS (« LPP »

; @Ind:Prime ancienneté

; @Ind:Salaire mensuel)

calcule la somme des valeurs si elles ne sont pas soumises à la base LPP.

SINONSOU MIS(« mode » ;
expression)

Si l'une des valeurs est soumise mais pas l'autre, la valeur soumise sera prise en compte. Il n'est pas nécessaire que les 2 rubriques soient soumises.

SINONSOU MIS (« AVS »

; @Ind:Prime ancienneté

; @Ind:Salaire mensuel)

fonctionnera également si l'une des deux rubriques est vide car vide est considéré comme égal à zéro pour une addition (voir note §31 L'éditeur de calculs).

AGELIMITE(limite ; mode)

Détermine si l'âge de l'employé est inférieur à un âge limite.

AGELIMITE(18) donne zéro si l'employé a 18 ans (dans l'année courante) ou plus. Donne 1 s'il a 17 ans ou moins.

Le paramètre *mode* est optionnel. S'il vaut 1, la fonction tient compte du mois de naissance et donne la valeur zéro dès le mois suivant l'anniversaire de l'âge concerné.

AGELIMITE(65;1) donne zéro dès le mois suivant le 65^{ème} anniversaire.

Donne la valeur de l'expression pour la période donnée.
 La période peut être donnée par un nombre (n° de la colonne) ou une date.
 POURPERIODE(*colonne* ; *expression*)
 POURPERIODE(DATE(15;3;2014);@Ind:Salaire mensuel)
 donne le salaire mensuel du mois de mars 2014.

Donne 1 (=vrai) ou 0 (=faux) selon les réglages faits pour *paramètre* dans *Entreprise – Périodes* pour la *colonne* (1 à n) concernée.
 paramètre = 1 : le 13^{ème} salaire doit être payé.
 paramètre = 2 : les vacances doivent être payées.
 paramètre = 3 : les vacances incluent le mois courant.
 FANIONPERIODE(*colonne* ; *paramètre*)
 FANIONPERIODE(6;1)
 donne 1 si le 13^{ème} doit être versé en juin.

Ramène un montant mensuel au prorata de la période courante.
 Par exemple, si on a fait un salaire du 1^{er} février au 15 mars inclus, MONTANTPERIODE(1000) donne 1500 puisque la période correspond à un mois et demi.

Donne le nom de la *colonne* (1 à n) du tableau général.
 NOMPERIODE(1) donne habituellement « Janvier ».
 NOMPERIODE(*colonne*)

Évalue l'expression dans une langue donnée (0 = français, 1 = allemand, 2 = anglais, 3 = italien).
 POURLANGUE(*langue* ; *expression*)
 POURLANGUE(2;NOMPERIODE(1)) donne « January ».

*ASSURANCE(rubrique ;
type ; mode)*

Donne les valeurs en fonction de l'assurance de l'employé.

Le premier paramètre doit être la rubrique du nom de l'assurance attribué à l'employé (

@Donnée:Assurance LPP, @Donnée:Assurance IJM, @Donnée:Assurance LAAC, ...).

Le second paramètre (valeur 1 à 9) détermine le type d'information souhaitée :

1 : taux à appliquer (selon la tranche d'âge si LPP)

2 : plafond mensuel

3 : valeur de « soumis à partir de » pour la LPP, pour les assurances c'est la valeur de déduction fixe

4 : « montant à déduire » pour la LPP

5 : « montant minimum » pour le salaire coordonné de la LPP

8 : état du fanion « calcule sur base annuelle » pour la LPP

11 : taux de la cotisation employeur

31 : valeur fixe pour l'employeur

Toutes ces valeurs sont adaptées automatiquement à la période traitée et à l'âge de l'employé.

ASSURANCE(@Donnée:Assurance IJM;1)

donne le taux de cotisation à l'assurance IJM de l'employé.

Le troisième paramètre est facultatif et permet de spécifier si la valeur retournée est calculée en mode mensuel ou annuel. Par défaut, le mode est mensuel. Ajoutez le paramètre 1 pour forcer le mode annuel.

ASSURANCE(@Donnée:Assurance IJM ; 4 ; 1)

donne le montant à déduire, rapporté à la durée de la période le cas échéant.

Donne le taux à appliquer pour l'imposition à la source.

Le paramètre *code* doit obligatoirement être la rubrique donnant le code de l'impôt (*@Donnée:Code d'impôt à la source*). Le paramètre *mode* détermine le type d'information souhaité, cela peut être :

1 : taux selon le montant déterminant en 3^{ème} paramètre

2 : donne 1 si le calcul est annuel (0 pour mensuel)

3 : donne 1 si le calcul annuel ne doit se faire qu'en fin de l'année

4 : donne 1 si le calcul annuel n'est possible que si l'employé était présent toute l'année

5 : donne 1 si le calcul final doit se faire selon le code de la dernière période

6 : donne le nombre d'heures par mois à considérer

7 : donne l'adresse de l'office tel qu'introduit sous *Entreprise – Impôt source*

8 : donne le n° de contribuable de l'entreprise

9 : donne le taux de la commission de perception

0 : donne 1 si le calcul mensuel doit se faire rétroactivement.

*IMPOT(code ; mode ;
déterminant annuel)*

ANNUEL(mode ; brut ; occasionnel ; prévision ; déduction)

Procède à une estimation du salaire déterminant annuel.

Le paramètre *mode* peut être :

0 : estimation en cours d'année

1 : détermination lors du dernier salaire

2 : extrapolation à partir du mois courant uniquement

3 : comme 1 mais limité aux dates d'entrée et de sortie de l'employé.

Le paramètre *brut* doit être le montant cumulé des indemnités soumises à l'impôt à la source pour la période courante.

Le paramètre *occasionnel* doit être le cumul des indemnités exceptionnelles soumises. Ces montants ne seront donc pas ramenés proportionnellement à l'année.

Le paramètre *prévision* permet d'ajouter les montants non encore payés, mais qui devraient l'être à la fin de l'année (indemnités vacances, 13^{ème} salaire).

Le paramètre *déduction* doit être les montants payés à l'avance sur les vacances et le 13^{ème} salaire.

Le logiciel prend encore en compte les dates des périodes ainsi que les dates d'entrée et de sortie de l'employé.

LISTEDEF(nombre ; texte)

S'utilise pour définir le contenu à choix d'une rubrique de type liste. Chaque élément de la liste se compose d'une valeur numérique et d'un texte associé. Voyez l'exemple au §6.3 Les bases.

*VARTEMP(nom ;
expression)*

Mémoire le résultat de *expression* dans l'une des 50 variables temporaires à disposition. La valeur mémorisée peut être reprise avec *VARTEMP(nom)* sans second paramètre.

VARTEMP(« essai » ; 250) met la valeur 250 dans la variable « essai ».

VARTEMP(« essai ») donne alors 250.

SI (*VARTEMP*(« essai » ; *CUMUL* (« AVS »)) > 0 ;

VARTEMP(« essai ») ;

VIDE())

Dans cet exemple, le calcul commence par mémoriser le total des montants soumis à l'AVS dans la variable temporaire « essai ».

Puis il teste si le cumul est > 0. Si oui, l'expression donne le contenu de la variable « essai » (=le montant cumulé), sinon elle est vide.

C'est l'équivalent de

SI (*CUMUL* (« AVS ») > 0 ;

CUMUL (« AVS ») ;

VIDE())

mais cela évite de calculer deux fois le cumul.

La variable temporaire (nom) est initialisée à *VIDE*() au début d'une impression ou au début de chaque calcul si elle est utilisée ailleurs que pour une impression.

*VARTEMP(index ;
expression)*

Mémoire le résultat de *expression* dans l'une des 10 variables temporaires à disposition (index = 0 .. 9). La valeur mémorisée peut être reprise avec *VARTEMP(index)* sans second paramètre.

SI (*VARTEMP*(7 ; *CUMUL* (« AVS »)) > 0 ;

VARTEMP(7) ;

VIDE())

Ce calcul mémorise le total des valeurs soumises à l'AVS dans la variable temporaire 7 et donne le contenu si le cumul est > 0.

La variable temporaire (index) est initialisée à *VIDE*() au début de chaque calcul, même lors d'une impression.

<p><i>RECURSIF(rubriques)</i></p>	<p>S'utilise pour forcer Crésus à faire un calcul itératif sur une valeur dépendante d'elle-même.</p> <p>Exemple : pour déterminer la part employeur des charges sociales en partant d'un salaire brut-brut, qui inclut donc ces charges, il faut évaluer le total de ces charges, les déduire du salaire brut-brut, pour déterminer le salaire brut sur lequel on calculera les charges sociales, puis recommencer. Les charges sociales vont donc varier en fonction d'elles-mêmes.</p> <p>RECURSIF(@Brut:Total des cotisations employeur)</p>
<p><i>DEBUG(expression)</i></p>	<p>Affiche la formule et les étapes de son évaluation.</p> <p><i>Expression</i> peut être un calcul ou une rubrique.</p> <p>DEBUG(C%(@Brut:Base AVS;@Coef:Taux AVS)) affiche</p> <p>@Brut:Base AVS[Janvier] ==> 6500.</p> <p>@Coef:Taux AVS[Janvier] ==> 5.15</p> <p>C%(6500. ; 5.15) ==> 334.75</p>
<p><i>CACHE(expression)</i></p>	<p>Cache l'objet entier si le résultat de l'expression est nul.</p> <p>CACHE(SI(@Donnée:Sexe de l'employé(e) = 0 ; « Monsieur »))</p> <p><i>S'il s'agit d'une employée, tout l'objet est caché, y compris son cadre et/ou fond.</i></p>
<p><i>COMPTE(chaine ; critère)</i></p>	<p>Compte le nombre d'occurrences du critère dans la chaîne.</p> <p>COMPTE(@Donnée:Divers1 ; » ! ») compte le nombre de points d'exclamation trouvé dans la rubrique <i>Divers1</i>.</p>

<p><i>SWITCH(condition 0 ; 0 ; condition 1 ; 1 ; condition 2 ; 2 ; ... ; condition n ; n)</i></p>	<p>Donne le résultat de l'expression 0 à n selon l'état de la dernière condition 0 à n évaluée. Si la condition 0 est vraie, donne le résultat de l'expression 0, si la condition 1 est vraie, donne le résultat de l'expression 1, etc. C'est le résultat de l'expression de la dernière condition vraie évaluée qui est donné. SWITCH(@Donnée:Mois sélectionné >=1 ; « 1er trimestre » ; @Donnée:Mois sélectionné >=4 ; « 2e trimestre » ; @Donnée:Mois sélectionné >=7 ; « 3e trimestre » ; @Donnée:Mois sélectionné >=10 ; « 4e trimestre ») <i>Donne le 1^{er}, le 2^{ème}, le 3^{ème} ou le 4^{ème} trimestre selon le mois où se trouve le curseur.</i></p>
<p><i>CASE(condition ; 0 ; 1 ; .. ; n)</i></p>	<p>Donne le résultat de l'opération 0 à n en fonction de la valeur de condition. Si la condition vaut 0, donne le résultat de l'opération 0, si la condition vaut 1, donne le résultat de l'opération 1, etc.</p>
<p><i>POURCHAQUE(objet ; expression)</i></p>	<p>Répète l'expression pour toutes les occurrences de l'<i>objet</i>, par exemple pour chacune des assurances LAA définies ou pour chaque code d'impôt à la source. POURCHAQUE(@Donnée:Assurance LPP ; expression) L'objet doit être la rubrique qui fait référence à l'assurance, au code d'impôt à la source ou au lieu de travail.</p>

Les lieux de travail définis pour l'entreprise – anciennement « succursales » – sont référencés par un n° d'identification supérieur ou égal à 300.

@Donnée:Lieu de travail de l'employé affiche ou imprime ce n° d'identification.

LIEUTRAVAIL(@E.Donnée:Lieux de travail) donne le nom du lieu de travail de l'employé.

Pour imprimer l'adresse et la localité du lieu de travail :

LIEUTRAVAIL(@E.Donnée:Adresse de l'entreprise)

+ « \n »

+ LIEUTRAVAIL(@E.Donnée:Localité de l'entreprise)

Cette expression peut également être formulée ainsi :

LIEUTRAVAIL(@E.Donnée:Adresse de l'entreprise

+ « \n »

+ @E.Donnée:Localité de l'entreprise)

Si le document concerne toute l'entreprise, la fonction *LIEUTRAVAIL()* donne les données du siège principal de l'entreprise.

LIEUTRAVAIL(expression)

LIMITEPERIODES(condition ; expression)

Permet de ne traiter que les mois où l'employé garde la même condition.

LIMITEPERIODES(@Donnée:Date d'entrée de l'employé

; TOTANNEE(@Brut:Salaires versés))

donne le total du salaire versé pour la période d'engagement courante (jusqu'au mois courant).

Donne les définitions comptables d'une rubrique (§19.11 Comptabilisation des rubriques).

La valeur donnée dépend du *paramètre* :

COMPTA(rubrique ; 1) : compte débit

COMPTA(rubrique ; 1.1) : compte débit de l'employeur

COMPTA(rubrique ; 2) : compte crédit

COMPTA(rubrique ; 2.1) : compte crédit de l'employeur

COMPTA(rubrique ; 3) : libellé

COMPTA(rubrique ; 3.1) : libellé de la part de l'employeur

COMPTA(rubrique ; 4) : mode

COMPTA(rubrique ; 5) : code analytique

COMPTA(rubrique ; 5.1) : code analytique de la part de l'employeur

COMPTA(rubrique ; 6) : code TVA

Dans un bloc d'exportation « pour toutes les indemnités » (§20.5 Exporter des données), si le critère *rubrique* est vide, la fonction traite la rubrique courante de la boucle :

COMPTA(; 1.1) donne le compte débit employeur de chaque rubrique traitée dans la boucle.

COMPTA(rubrique ; paramètre)

SPECIALOP(paramètre)

Opérations particulières effectuées lors d'une impression ou lors d'une exportation.

SPECIALOP() traite toutes les rubriques @Ind: ou @Déd:* d'une ligne imprimée ou d'un bloc exporté « répété pour » (§20.5 Exporter des données).*

SPECIALOP() s'applique à la rubrique courante, en fonction de la valeur de paramètre.

SPECIALOP(90)

donne le type du coefficient de la rubrique :

-1 : type vide

0 : jours

1 : heures

2 : %

3 : sans type (-)

SPECIALOP(91) donne la valeur de la rubrique.

SPECIALOP(92) donne le déterminant de la rubrique.

SPECIALOP(93) donne le coefficient de la rubrique.

SPECIALOP(94) donne le coefficient employeur de la rubrique.

SPECIALOP(95) donne la valeur employeur de la rubrique.

SPECIALOP() est une valeur positive, même pour les déductions. Pour imprimer un signe « + » devant une déduction négative (un remboursement), il faut faire un calcul comme $REPLACE(« + » + CHAINE(0 - SPECIALOP(91) ; 2) ; « +- » ; « - » ; « +0.00 » ; « »)$

Retourne les diverses informations associées à l'assurance.

Tous les éléments donnés sont des textes.

La *donnée:assurance* peut être l'une des rubriques

@Donnée:Code LAA de l'employé

@Donnée:Assurance LAAC

@Donnée:Assurance LAAC (2ème)

@Donnée:Assurance IJM

@Donnée:Assurance IJM (2ème)

@Donnée:Assurance FT (Frais de traitement)

@Donnée:Assurance LPP

@Donnée:Assurance LPP (2ème)

La valeur donnée dépend du **paramètre**.

Dans les exemples ci-dessous, on teste

l'assurance AA complémentaire d'un

employé, dont le code est A et qui

concerne le groupe Personnel du bureau.

0 : donne le nom complet avec le code.

`INFOASSURANCE(@Donnée:Assurance LAAC (2ème) ; 0)`

donne A2 : Personnel du bureau, cat. 2

1 : donne le code seul.

`INFOASSURANCE(@Donnée:Assurance LAAC (2ème) ; 1)`

donne A2

2 : donne le nom et la catégorie.

`INFOASSURANCE(@Donnée:Assurance LAAC (2ème) ; 2)`

donne Personnel du bureau, cat. 2

3 : donne le code et le nom seul.

`INFOASSURANCE(@Donnée:Assurance LAAC (2ème) ; 3)`

donne A2 : Personnel du bureau

4 : donne la catégorie seulement, par exemple

`INFOASSURANCE(@Donnée:Assurance LAAC (2ème) ; 4)`

donne cat. 2

5 : donne le nom seul, par exemple

`INFOASSURANCE(donnée:assurance ; paramètre)`

<p>TOTEMPLOYESCSV(<i>critère ; colonne 1 ; colonne 2 ; ... ; colonne n</i>) et TOTANNEECSV(<i>critère ; colonne 1 ; colonne 2 ; ... ; colonne n</i>)</p>	<p>S'utilisent lors d'une exportation pour produire un tableau indiquant le cumul des données qui répondent au critère (§20.5.2 Les opérateurs spéciaux TOTEMPLOYESCSV() et TOTANNEECSV())</p>
<p>ADRESSE(<i>mode ; rubrique</i>)</p>	<p>Donne l'élément mode du champ rubrique de type NPA/Rue/Localité (§19.7.3 Rubriques de type Données). La valeur donnée dépend de <i>mode</i> :</p> <ul style="list-style-type: none"> 0 : NPA Localité 1 : Rue 2 : NPA 3 : Localité 4 : Canton 5 : Pays (code ISO) 6 : Pays (en toutes lettres) <p>ADRESSE(0 ; @Donnée:Adresse du mandataire) donne le NPA et localité du mandataire.</p>
<p>SETDATA(<i>rubrique ; expression ; paramètre</i>)</p>	<p>S'utilise pour affecter une valeur à une rubrique. Peut s'utiliser dans des expressions complexes. Le paramètre est facultatif et spécifie la période de paie concernée. S'il est omis, l'expression est affectée à la valeur globale de la rubrique. SETDATA(@Ind:Autres frais forfaitaires ; 200 ; 2) met 200.– dans les frais forfaitaires de l'employé en février. TOTEMPLOYES(SETDATA(@Donnée:Jours vacances par année;25)) met 25 jours dans la valeur globale de tous les employés.</p>

Attention : aucun contrôle n'est fait lors d'un SETDATA(), il est possible de modifier des salaires existants, mais ils ne sont pas recalculés.

S'utilise pour effectuer plusieurs calculs dans la même formule.

OP(VARTEMP(« base »
 OP (expression 1 | ; DEFAULT(@Ind:Salaire mensuel
 expression 2 | ... | ; @Ind:Salaire horaire))
 expression n) | VARTEMP(« taux » ; 12)
 | VARTEMP(« bonus » ; C%(VARTEMP(« base »
 ; VARTEMP(« taux »)))
 | SETDATA(@Ind:Bonus ; VARTEMP(« bonus »)))

S'utilise dans le calcul *Obligatoire selon* de la définition d'une rubrique pour affiche un message dans une bulle d'aide ou après avoir saisi une valeur lors de la saisie de données.

Le paramètre *mode* détermine l'icône affichée dans le champ et le niveau du message :

« INFO » : *le message est une information.*

« WARNING » : *c'est un avertissement, mais il n'est pas impératif de s'y conformer pour quitter la saisie. Le point d'exclamation est rouge pâle, mais le champ de saisie est blanc.*

« ERROR » : *c'est un avertissement important, mais il n'est pas impératif de s'y conformer pour quitter la saisie. Le point d'exclamation et le champ de saisie sont rouge pâle.*

« KO » : *c'est un avertissement incontournable, tant que la valeur entrée ne répond pas au critère, il ne sera pas possible de valider la saisie. Le point d'exclamation est rouge vif.*

MESSINFO(mode ;
 texte)

Permet d'ouvrir un fichier de type texte et d'en traiter le contenu.

expression_début est évaluée une fois au début du traitement

expression_ligne est appliquée sur chaque ligne du fichier

expression_fin est évaluée à la fin du traitement.

La fonction n'est évaluée qu'une seule fois par fichier, ensuite elle retourne toujours le résultat évalué cette fois-là.

Si on souhaite plusieurs résultats, il faut mettre un critère à la suite du nom de fichier après l'un des caractères |;!?*/ par exemple

« %EXEPATH%ListeCommunes.txt; » +

@Donnée:NPA de l'employé

Dans ce cas *POURFICHIER* sera évalué pour chaque NPA différent rencontré dans le fichier Crésus.

expression_ligne est évaluée pour chaque ligne du fichier lu.

Le contenu de la ligne est donné dans

VARTEMP(« #Line »).

On peut utiliser les fonctions habituelles sur la variable, comme

MOT(*VARTEMP*(« #Line »); 5 ; « ; ») dans un fichier CSV pour trouver le 5^{ème} élément.

Le résultat de l'expression (toujours une chaîne de caractères) est ajouté à la chaîne donnée par la fonction.

Par exemple :

POURFICHIER

(« %EXEPATH%\NuPost.TXT » ;

« villes: » ;

SI(*MOT*(*VARTEMP*(« #Line ») ; 3 ; « \t ») = « 1400 » ;

MOT(*VARTEMP*(« #Line ») ; 5 ; « \t ») + « , » ;

« npa = 1400 »

)

ouvre le fichier NuPost.txt

met le texte villes: au début de la chaîne à donner

évalue chaque ligne du fichier : si le 3^e mot de la ligne est 1400, ajoute le 5^e mot de la ligne et une

POURFICHIER(fichier ;
expression_debut ;
expression_ligne ;
expression_fin)

DONNEE(mode ; expression)

Donne la valeur calculée par Cresus ou la valeur introduite par l'utilisateur en fonction de *mode*.

mode = 1 : donne la valeur calculée par Cresus

mode = 2 : donne la valeur saisie par l'utilisateur

La valeur donnée est vide si elle n'existe pas.

DONNEE (1 ; @Ind:Salaire mensuel) donne le salaire mensuel calculé pour le mois, même s'il a été modifié manuellement au moment de sa saisie.

POURLANGUE(paramètre ; expression)

Donne l'expression dans la langue en fonction de paramètre.

paramètre peut être

0 : français

1 : allemand

2 : anglais

3 : italien