

Correction et mise à jour du gestionnaire de rubrique

Il s'agit ici des modifications qui ont été apportées via la version 62 du gestionnaire de rubrique.

Divers

REDUCRTPAT : Déduction forfaitaire sur rachat de jour de RTT

Modification du taux afin de prendre en compte la déduction forfaitaire de 0.50€ pour les entreprises avec au moins 20 salariés et moins de 250 salariés.

La formule taux employeur prend en compte la constante générale NBSALARIE0101

- entreprise < 20 salariés => taux 1.50€
- 20 salariés <= entreprise < 250 => taux 0.50 €

Si la constante NBSALARIE0101 est égale à zéro le taux sera égal à 1.50

Il est également possible de mettre directement le taux dans la constante générale MTREDUCHEURE.

Pensez à ajouter la cotisation REDUCRTPAT au niveau de la codification de régime pour les entreprises concernées.

Avant :

```
si ( CONSTANTE( GENERAL. MTREDUCHEURE) =0)
alors (
    si ( CONSTANTE( GENERAL. NBSALARIE0101) <=19)
    alors (1.50)
    sinon (0)
```

```

)
sinon ( CONSTANTE( GENERAL. MTREDUCHEURE) )

```

Après

```

si ( CONSTANTE( GENERAL. MTREDUCHEURE) =0)
alors (
    si ( CONSTANTE( GENERAL. NBSALARIE0101) <=19)
    alors (1.50)
    sinon (
        si ( CONSTANTE( GENERAL. NBSALARIE0101) <=249)
        alors (0.50)
        sinon (0)
    )
)
sinon ( CONSTANTE( GENERAL. MTREDUCHEURE) )

```

MALCOMP25 : Complément Maladie si salaire >2.5 SMIC

- Ajout d'un contrôle d'un SMIC positif.
- Prise en compte de la nouvelle moulure AGCP

Avant

```

si (
    (( [ V_AGCP. CUMREMUCOEF] ) > ( 2.5*[ V_AGCP. CUMSMICPRORATE] ))
    et ( [ V_AGCP. CUMREMUCOEF] >0)
)
alors (
    histocumulsitu([ BC_BRUT_URSAFF. MONTANT] ) + [ BC_BRUT_URSAFF. MONTANT]
    - histocumulsitu([ BC_MALCOMP25. MONTANT] )
)
sinon (
    histocumulsitu([ BC_MALCOMP25. MONTANT] ) *(-1)
)

```

Après

```

si ( CONSTANTE( GENERAL. NEWAGCP) = 0)
    alors (
        si (

```

```

        (([ V_AGCP. CUMREMUCOEF]) > (2.5*[ V_AGCP. CUMSMICPRORATE]))
    et ([ V_AGCP. CUMREMUCOEF]>0)
    et ([ V_AGCP. CUMSMICPRORATE]>0)
    )
    alors (
        histocumulsitu([ BC_BRUT_URSAFF. MONTANT]) +
[ BC_BRUT_URSAFF. MONTANT]
        - histocumulsitu([ BC_MALCOMP25. MONTANT])
    )
    sinon (
        histocumulsitu([ BC_MALCOMP25. MONTANT]) *(-1)
    )
    )
    sinon (
        si (
            (([ V_NEWAGCP. CUMREMUCOEF]) > (2.5*[ V_NEWAGCP. CUMSMICPRO]))
            et ([ V_NEWAGCP. CUMREMUCOEF]>0)
            et ([ V_NEWAGCP. CUMSMICPRO]>0)
        )
        alors (
            histocumulsitu([ BC_BRUT_URSAFF. MONTANT]) +
[ BC_BRUT_URSAFF. MONTANT]
            - histocumulsitu([ BC_MALCOMP25. MONTANT])
        )
        sinon (
            histocumulsitu([ BC_MALCOMP25. MONTANT]) *(-1)
        )
    )
    )

```

PL_HRSDUREE : Durée en heures pour les temps partiels

Correctif pour prendre en compte les forfaits jours

Avant

```
CONSTANTE( CONTRAT. HORAIREMENSREEL) +[ NBHRSCOMPL. MONTANT] - [ TPT_NONPAYE. DUREE]
```

Après

```

si ( CONSTANTE( CONTRAT. SITUATIONHORAIRE) =4)
alors (

```

```

CONSTANTE( CONTRAT. HORAIRECOLLECTIF ) *CONSTANTE( CONTRAT. JOUR_FORFAIT )
/
CONSTANTE( CONTRAT. FORFAITJOURCOLLECTIF )
)
SINON(
CONSTANTE( CONTRAT. HORAIREMENSREEL ) + [ NBHRS COMPL. MONTANT ] - [ TPT_NONPAYE. DUREE ]
)

```

V_AGCP.HREMU

Correctif pour l'horaire collectif des forfaits jours

Remplacement de 218 par CONSTANTE(CONTRAT.FORFAITJOURCOLLECTIF)

Avant

```

{Forfait Annuel en jours}
    si ( CONSTANTE( CONTRAT. SITUATIONHORAIRE ) =4)
        alors
( CONSTANTE( CONTRAT. HORAIRECOLLECTIF ) *CONSTANTE( CONTRAT. JOUR_FORFAIT ) /218
    *
    si ( ( [ IJBRUT. MONTANT ] <>0 ) ou ( [ ABSTOTALV. MONTANT ] -
[ ABSTOTALM. MONTANT ] <>0 ) )
        alors ( [ BC_BRUT_URSAFF. MONTANT ]
            /
            ( [ BC_BRUT_URSAFF. MONTANT ] + [ IJBRUT. MONTANT ]
+[ ABSTOTALV. MONTANT ] - [ ABSTOTALM. MONTANT ] )
        )
        sinon ( 1 )
    )
    sinon (

```

Après

```

{Forfait Annuel en jours}
    si ( CONSTANTE( CONTRAT. SITUATIONHORAIRE ) =4)
        alors
( CONSTANTE( CONTRAT. HORAIRECOLLECTIF ) *CONSTANTE( CONTRAT. JOUR_FORFAIT ) /CONSTANTE( CONTRAT. FORFAITJOURCOLLECTIF )
    *
    si ( ( [ IJBRUT. MONTANT ] <>0 ) ou ( [ ABSTOTALV. MONTANT ] -
[ ABSTOTALM. MONTANT ] <>0 ) )

```

```

alors ([ BC_BRUT_URSAFF. MONTANT]
/
([ BC_BRUT_URSAFF. MONTANT] +[ IJBRUT. MONTANT]
+[ ABSTOTALV. MONTANT] - [ ABSTOTALM. MONTANT] )
)
sinon (1)
)
sinon (

```

Modification de I_ICP

La formule ne prenait pas en compte l'historique du contrat en changement de situation dans le mois de paye en cours

Avant

```

DECLARER( MEMB)
AFFECTER( MEMB; CONSTANTE( CONTRAT. MOIS_EмбаUCHE) )
DECLARER( AEMB)
AFFECTER( AEMB; CONSTANTE( CONTRAT. ANNEE_EмбаUCHE) )
DECLARER( MPAYE)
AFFECTER( MPAYE; CONSTANTE( GENERAL. MOIS_PAYE) )
DECLARER( APAYE)
AFFECTER( APAYE; CONSTANTE( GENERAL. ANNEE_PAYE) )
SI(
CONSTANTE( CONTRAT. PAIEMENTCP) =
( 0) )
ALORS( 0)
SINON(
SI(
CONSTANTE( CONTRAT. PAIEMENTCP) =2)
ALORS(
HISTOCumulsitu([ B_ICP. MONTANT];: MEMB;: AEMB;: MPAYE;: APAYE) +([ B_ICP. MONTANT] ) -
HISTOCumulsitu([ I_ICP. BASE];: MEMB;: AEMB;: MPAYE;: APAYE) )
SINON(
SI(
CONSTANTE( CONTRAT. CLOTURE) =1)
ALORS(
HISTOCumulsitu([ B_ICP. MONTANT];: MEMB;: AEMB;: MPAYE;: APAYE) +
([ B_ICP. MONTANT] ) -
HISTOCumulsitu([ I_ICP. BASE];: MEMB;: AEMB;: MPAYE;: APAYE) )

```

```
SINON( 0 ) )
```

Après **(Faux) aller sur le code version 63** Version 63

```
SI(
  CONSTANTE( CONTRAT. PAIEMENTCP) =
    ( 0 ) )
ALORS( 0 )
SINON(
  SI(
    CONSTANTE( CONTRAT. PAIEMENTCP) =2)
  ALORS(
    HISTOCumulsitu( [ B_ICP. MONTANT] ; 1, 1980) +( [ B_ICP. MONTANT] ) -
    HISTOCumulsitu( [ I_ICP. BASE] ; 1, 1980) )
  SINON(
    SI(
      CONSTANTE( CONTRAT. CLOTURE) =1)
    ALORS(
      HISTOCumulsitu( [ B_ICP. MONTANT] ; 1, 1980) +
      ( [ B_ICP. MONTANT] ) -
      HISTOCumulsitu( [ I_ICP. BASE] ; 1, 1980) )
    SINON( 0 ) ) )
```

Formule incorrecte, on a des , au lieu de ; dans les histocumulsitu =>
ISTOCumulsitu([I_ICP.BASE];1,1980))

Dans la version 63 du gestionnaire de rubrique, la formule a été corrigée

Version 63

```
SI(
  CONSTANTE( CONTRAT. PAIEMENTCP) =
    ( 0 ) )
ALORS( 0 )
SINON(
  SI(
    CONSTANTE( CONTRAT. PAIEMENTCP) =2)
  ALORS(
    HISTOCumulsitu( [ B_ICP. MONTANT] ; 1; 1980) +( [ B_ICP. MONTANT] ) -
    HISTOCumulsitu( [ I_ICP. BASE] ; 1; 1980) )
  SINON(
```

```

SI(
    CONSTANTE( CONTRAT. CLOTURE) =1)
ALORS(
    HISTOCumulsitu([ B_ICP. MONTANT]; 1; 1980) +
    ([ B_ICP. MONTANT] ) -
    HISTOCumulsitu([ I_ICP. BASE]; 1; 1980))
SINON( 0) ) )

```

Stagiaire

BC_BRUT_STA : Brut stagiaire

Contrôle de la constante contrat DSNATURE et mettre à zéro si différent de 29 convention de stage

Avant

```

si ([ BC_BRUT_URSAFF. MONTANT] >=0)
alors ( MAXIMUM([ BC_BRUT_URSAFF. MONTANT] - [ CONAIDE. SEUILSTAGIAIRE]; 0) )
sinon ( MAXIMUM(( 0-[ BC_BRUT_URSAFF. MONTANT] ) - [ CONAIDE. SEUILSTAGIAIRE]; 0)
        *-1)

```

Après

```

si ( CONSTANTE( CONTRAT. DSNATURE) =29)
alors (
    si ([ B_COT_URSAFF. MONTANT] >=0)
    alors ( MAXIMUM([ B_COT_URSAFF. MONTANT] - [ CONAIDE. SEUILSTAGIAIRE]; 0) )
    sinon ( MAXIMUM(( 0-[ B_COT_URSAFF. MONTANT] ) - [ CONAIDE. SEUILSTAGIAIRE]; 0)
            *-1)
    )
sinon ( 0)

```

BC_BRUT_STA_PL

Idem que précédemment

Avant

```

si ([ BC_BRUT_URSAFF. MONTANT] >=0)
alors ( PLAFONNER([ BC_BRUT_URSAFF. MONTANT] ; 0; [ CONAIDE. SEUILSTAGIAIRE] ) )
sinon ( PLAFONNER( ( 0-[ BC_BRUT_URSAFF. MONTANT] ) ; 0; [ CONAIDE. SEUILSTAGIAIRE] ) *- 1)

```

Après

```

si ( CONSTANCE( CONTRAT. DSNATURE) =29)
alors (
    si ([ B_COT_URSAFF. MONTANT] >=0)
    alors ( PLAFONNER([ B_COT_URSAFF. MONTANT] ; 0; [ CONAIDE. SEUILSTAGIAIRE] ) )
    sinon ( PLAFONNER( ( 0-[ B_COT_URSAFF. MONTANT] ) ; 0; [ CONAIDE. SEUILSTAGIAIRE] ) *- 1)
)
sinon ( 0)

```

BC_BRUT_URSAFF

Même chose que précédemment mais à l'inverse

Avant

```

si ( CONSTANCE( CONTRAT. NATURECONTRAT) =0008)
alors (
    si ([ F0. TOTAL] <>0)
    alors ([ F0. TOTAL] )
    sinon ([ B_COT_URSAFF. MONTANT] )
)
sinon ([ B_COT_URSAFF. MONTANT] )

```

Après

```

si ( CONSTANTE( CONTRAT. DSNATURE) =29)
alors ( 0)
sinon (
    si ( CONSTANTE( CONTRAT. NATURECONTRAT) =0008)
    alors (
        si ([ F0. TOTAL] <>0)
        alors ([ F0. TOTAL])
        sinon ([ B_COT_URSAFF. MONTANT])
    )
    sinon ([ B_COT_URSAFF. MONTANT])
)

```

MALCOMP25 : Complément Maladie si salaire >2.5 SMIC

Correction de la formule taux employeur pour prendre en compte les stagiaires qui n'ont pas droit au taux réduit de cotisation maladie

Avant :

```

si (( [ BC_MALCOMP25. MONTANT] >0) ou ([ BC_BRUT_STA. MONTANT] >0) )    alors ( 6)    sinon ( 0)

```

Après

```

si (
    ([ BC_MALCOMP25. MONTANT] >0)
    ou (( [ BC_BRUT_STA. MONTANT] >0) et ( CONSTANTE( CONTRAT. DSNATURE) =29) )
)
alors ( 6)
sinon ( 0)

```

CSG

Voici les rubriques créées ou modifiées pour prendre en compte le calcul du plafond CSG

Deux rubriques itératives pour calculer la base brute qui doit être abattue (avant application du plafond) et la base brute non abattue :

- B_COT_CSG_ABT : Base CSG-CRDS abattue
- B_COT_CSG_NABT : Base CSG-CRDS non abattue

Ces rubriques sont alimentées par la base CSG (B_COT_CSG) en suivant la règle suivante :

- taux=98.25 -> B_COT_CSG_ABT à 100%
- si taux=1.75 -> B_COT_CSG_ABT à -100% et B_COT_CSG_NABT à 100%
- si taux=-98.25% -> B_COT_CSG_ABT à -100%
- si taux=100% -> B_COT_CSG_NABT à 100%
- si taux=-100% -> B_COT_CSG_ABT à -100%
- si taux=-1.75% -> Il s'agit d'une erreur, les réintégrations fiscales n'entrent pas dans la base CSG-CRDS
- Si le taux est négatif et différent de -98.25% ou -1.75% :

il s'agit probablement de remboursement de prévoyance -> B_COT_CSG_ABT au taux indiqué

V_CSG : Rubrique libre pour les éléments de calcul CSG

CUM_AV_ABT : Cumul montant qui doit être abattu

```
minimum(
    (histocumulJOINT([ B_COT_CSG_ABT. MONTANT]) +[ B_COT_CSG_ABT. MONTANT])
    ;
    (histocumulJOINT([ PL_MENSA. MONTANT]) +[ PL_MENSA. MONTANT]) *4
)
```

CUM_ABT Cumul abattu :

```
[ V_CSG. CUM_AV_ABT] *0. 9825
```

CUM_NABT : Cumul non abattu

```
(histocumulJOINT([ B_COT_CSG_ABT. MONTANT]) +[ B_COT_CSG_ABT. MONTANT] - [ V_CSG. CUM_AV_ABT])
+
(histocumulJOINT([ B_COT_CSG_NABT. MONTANT]) +[ B_COT_CSG_NABT. MONTANT])
```

Rubrique BC_CSG et BC_CRDS

Avant

```
si ( CONSTANTE( CONTRAT. NATURECONTRAT) =0008)
alors (
    si ([ F0. TOTAL] <>0)
    alors ([ F0. TOTAL])
    sinon ( F_TESTCSG([ B_COT_CSG. MONTANT]; [ BC_CSG. MONTANT]) )
)
sinon (
    si ( CONSTANTE( CONTRAT. DSNATURE) =29)
```

```

alors ( F_TESTCSG([ BC_BRUT_STA. MONTANT]*0. 9825; [ BC_CSG. MONTANT] ) )
sinon ( F_TESTCSG([ B_COT_CSG. MONTANT]; [ BC_CSG. MONTANT] ) )
)

```

Après

```

si ( ( CONSTANTE( GENERAL. NEWCSG )=1) ou ( CONSTANTE( GENERAL. ANNEE_PAYE ) >=2024 ) )
alors (
    si ( ( CONSTANTE( CONTRAT. NATURECONTRAT )=0008) et ( [ F0. TOTAL ] <>0 ) )
    alors ( [ F0. TOTAL ] )
    sinon (
        [ V_CSG. CUM_ABAT ]
        +[ V_CSG. CUM_NABAT ]
        -histocumulJOINT([ BC_CSG. MONTANT ] )
    )
)
sinon (
    si ( CONSTANTE( CONTRAT. NATURECONTRAT )=0008)
    alors (
        si ( [ F0. TOTAL ] <>0 )
        alors ( [ F0. TOTAL ] )
        sinon ( F_TESTCSG([ B_COT_CSG. MONTANT]; [ BC_CSG. MONTANT] ) )
    )
    sinon (
        si ( CONSTANTE( CONTRAT. DSNATURE )=29)
        alors ( F_TESTCSG([ BC_BRUT_STA. MONTANT]*0. 9825; [ BC_CSG. MONTANT] ) )
        sinon ( F_TESTCSG([ B_COT_CSG. MONTANT]; [ BC_CSG. MONTANT] ) )
    )
)
)

```

AGCP nouveau calcul

Fonction F_AGCPSMICMENS : Recalcul du smic mensuel proraté d'un mois

Paramètres :

- MDEBUT : Mois de début du cumul (dans le cas d'un apprenti)
- MACALCULER : Mois à calculer
- HREMU : Rubrique heures rémunérées

- HORBASE : Rubrique horaire de base

```

si ((CONSTANTE( GENERAL. MOIS_PAYE)>=MACALCULER) et ( MDEBUT<=MACALCULER))
alors (
    si ( MACALCULER=CONSTANTE( GENERAL. MOIS_PAYE))
    alors (
        (
            HREMU
            /
            CONSTANTE( CONTRAT. HORAIRECOLLECTIF)*CONSTANTE( GENERAL. SMICMENS)
        )
        *
        HORBASE/1607
    )
    sinon (
        (
            histocumulsitu(HREMU;MACALCULER;CONSTANTE( GENERAL. ANNEE_PAYE);MACALCULER;CONSTANTE( GENERAL. ANNEE_PAYE))
            /
            CONSTANTE( CONTRAT. HORAIRECOLLECTIF)*CONSTANTEHIS( GENERAL. SMICMENS;MACALCULER;CONSTANTE( GENERAL. ANNEE_PAYE))
        )
        *
        histocumulsitu(HORBASE;MACALCULER;CONSTANTE( GENERAL. ANNEE_PAYE);MACALCULER;CONSTANTE( GENERAL. ANNEE_PAYE))
        /
        (1607
        *
        si ( OKREMUH=-1)
        alors (1)
        sinon (
            histocumulsitu(OKREMUH;MACALCULER;CONSTANTE( GENERAL. ANNEE_PAYE);MACALCULER;CONSTANTE( GENERAL. ANNEE_PAYE))
        )
    )
)
sinon (0)

```

Rubriques créées

- V_NEWAGCP : Rubrique libre nouveau calcul AGCP
- V_AD_AGCP : Rubrique libre nouveau calcul AGCP pour l'aide à domicile
- AD_HEXO : Heures aide à domicile exonérées
- AD_HNEXO : Heures non exonérées aide à domicile

V_NEWAGCP : Rubrique libre nouveau calcul AGCP

Liste des formules :

- REMUCOEF : Rémunération
- HREMU : Heure rémunérée
- HORBASE : Horaire de base
- SMICPRO : Smic proraté mensuel
- CUMREMUOEF : Cumul rémunération
- CUMSMICPRO : Cumul Smic proraté mensuel total

Si la constante SMICAGCP (recalcule smic chaque mois) est à oui, utilise TOTSMICPRO sinon comme avant

- TOTSMICPRO : Total Smic proraté. Recalcule tous les smic prorates de l'année

```
si ( CONSTANTE( GENERAL. SMICAGCP ) =0)
alors ( 0)
sinon (
    si ( ( CONSTANTE( GENERAL. ANNEE_PAYE ) >=2024 ) ou ( CONSTANTE( GENERAL. AGCP_AD ) =1) )
    alors (

F_AGCPSMICMENS( [ AGCPDEBUT. BASE ] ; 1 ; [ V_NEWAGCP. HREMU ] ; [ V_NEWAGCP. HORBASE ] ; [ V_NEWAGCP. HREMUOK ] )

+F_AGCPSMICMENS( [ AGCPDEBUT. BASE ] ; 2 ; [ V_NEWAGCP. HREMU ] ; [ V_NEWAGCP. HORBASE ] ; [ V_NEWAGCP. HREMUOK ] )

+F_AGCPSMICMENS( [ AGCPDEBUT. BASE ] ; 3 ; [ V_NEWAGCP. HREMU ] ; [ V_NEWAGCP. HORBASE ] ; [ V_NEWAGCP. HREMUOK ] )

+F_AGCPSMICMENS( [ AGCPDEBUT. BASE ] ; 4 ; [ V_NEWAGCP. HREMU ] ; [ V_NEWAGCP. HORBASE ] ; [ V_NEWAGCP. HREMUOK ] )

+F_AGCPSMICMENS( [ AGCPDEBUT. BASE ] ; 5 ; [ V_NEWAGCP. HREMU ] ; [ V_NEWAGCP. HORBASE ] ; [ V_NEWAGCP. HREMUOK ] )

+F_AGCPSMICMENS( [ AGCPDEBUT. BASE ] ; 6 ; [ V_NEWAGCP. HREMU ] ; [ V_NEWAGCP. HORBASE ] ; [ V_NEWAGCP. HREMUOK ] )

+F_AGCPSMICMENS( [ AGCPDEBUT. BASE ] ; 7 ; [ V_NEWAGCP. HREMU ] ; [ V_NEWAGCP. HORBASE ] ; [ V_NEWAGCP. HREMUOK ] )
```

```

+F_AGCPSMICMENS( [ AGCPDEBUT. BASE]; 8; [ V_NEWAGCP. HREMU]; [ V_NEWAGCP. HORBASE]; [ V_NEWAGCP. HREMUOK] )

+F_AGCPSMICMENS( [ AGCPDEBUT. BASE]; 9; [ V_NEWAGCP. HREMU]; [ V_NEWAGCP. HORBASE]; [ V_NEWAGCP. HREMUOK] )

+F_AGCPSMICMENS( [ AGCPDEBUT. BASE]; 10; [ V_NEWAGCP. HREMU]; [ V_NEWAGCP. HORBASE]; [ V_NEWAGCP. HREMUOK] )

+F_AGCPSMICMENS( [ AGCPDEBUT. BASE]; 11; [ V_NEWAGCP. HREMU]; [ V_NEWAGCP. HORBASE]; [ V_NEWAGCP. HREMUOK] )

+F_AGCPSMICMENS( [ AGCPDEBUT. BASE]; 12; [ V_NEWAGCP. HREMU]; [ V_NEWAGCP. HORBASE]; [ V_NEWAGCP. HREMUOK] )
    )
    sinon (
        F_AGCPSMICMENS( [ AGCPDEBUT. BASE]; 1; [ V_AGCP. HREMU]; [ V_AGCP. HORBASE]; - 1)
        +F_AGCPSMICMENS( [ AGCPDEBUT. BASE]; 2; [ V_AGCP. HREMU]; [ V_AGCP. HORBASE]; - 1)
        +F_AGCPSMICMENS( [ AGCPDEBUT. BASE]; 3; [ V_AGCP. HREMU]; [ V_AGCP. HORBASE]; - 1)
        +F_AGCPSMICMENS( [ AGCPDEBUT. BASE]; 4; [ V_AGCP. HREMU]; [ V_AGCP. HORBASE]; - 1)
        +F_AGCPSMICMENS( [ AGCPDEBUT. BASE]; 5; [ V_AGCP. HREMU]; [ V_AGCP. HORBASE]; - 1)
        +F_AGCPSMICMENS( [ AGCPDEBUT. BASE]; 6; [ V_AGCP. HREMU]; [ V_AGCP. HORBASE]; - 1)
        +F_AGCPSMICMENS( [ AGCPDEBUT. BASE]; 7; [ V_AGCP. HREMU]; [ V_AGCP. HORBASE]; - 1)
        +F_AGCPSMICMENS( [ AGCPDEBUT. BASE]; 8; [ V_AGCP. HREMU]; [ V_AGCP. HORBASE]; - 1)
        +F_AGCPSMICMENS( [ AGCPDEBUT. BASE]; 9; [ V_AGCP. HREMU]; [ V_AGCP. HORBASE]; - 1)
        +F_AGCPSMICMENS( [ AGCPDEBUT. BASE]; 10; [ V_AGCP. HREMU]; [ V_AGCP. HORBASE]; - 1)
        +F_AGCPSMICMENS( [ AGCPDEBUT. BASE]; 11; [ V_AGCP. HREMU]; [ V_AGCP. HORBASE]; - 1)
        +F_AGCPSMICMENS( [ AGCPDEBUT. BASE]; 12; [ V_AGCP. HREMU]; [ V_AGCP. HORBASE]; - 1)
    )
)

```

- MAXCOEF : Coefficient maximum AGCP
- COEF : Coefficient Allègement général de cotisation
- TOTALAGCP : Total AGCP
- TOTAGCPURSSAF : Total AGCP part Urssaf
- TOTAGCPAA : Total AGCP part AA
- AGCPURSSAF : Montant AGCP Urssaf mensuel
- AGCPAA : Montant AGCP AA mensuel
- HREMUOK : Egal à 1 si le nombre d'heure rémunéré est différent de zéro. Par la suite , utilisé pour recalculer le smic proraté chaque mois

Les quatre dernières formules permettent de recalculer le total AGCP pour chaque part en fonction du paramètre T actuel. Ainsi, s'il était incorrect en début d'année, le fait de le changer calculera les bons montant AGCP.

V_AD_AGCP : AGCP aide à domicile

Liste des formules :

- ADREMU : Rémunération
- ADSMICPRO : Smic proraté mensuel
- ADCUMSMICPRO : Cumul Smic proraté mensuel total
- ADTOTSMICPRO : Total Smic proraté Aide à domicile
- ADCOEF : Coefficient aide à domicile
- ADTOTALAGCP : Total AGCP Aide à domicile
- ADTOTAGCPURSSAF : Total AGCP Urssaf aide à domicile
- ADTOTAGCPAA : Total AGCP AA Aide à domicile
- ADAGCPURSSAF : Montant AGCP Urssaf aide à domicile mensuel
- ADAGCPAA : Montant AGCP AA aide à domicile mensuel
- ADREMUCOEF : Rémunération utilisée
- ADCUMREMUCOEF : Cumul rémunération
- ADHORBASE : Horaire de base
- ADHREMUOK :Egal à 1 si le nombre d'heure rémunéré est différent de zéro. Par la suite , utilisé pour recalculer le smic proraté chaque mois

AD_HNEXO : Heures non exonérées aide à domicile

Rubrique de paye des heures aide à domicile non exonérées. A saisir en variable de paye.

AD_HEXO : Heures aide à domicile exonérées

Rubrique de paye des heures exonérées aide à domicile. Cette rubrique est calculée en fonction des heures non exonérées saisies par la différence entre les heures contractuelles et les heures non exonérées saisies.

Les heures supplémentaires qui sont saisies sont réparties au prorata des heures exonérées et non exonérées.

CUMUL.COMPTEUR

Rubrique toujours égale à 1 pour comptabiliser le nombre de contrat.

Mise en place de l'AGCP AD en cours d'année

ATTENTION : Pour activer SMICAGCP il faut faire une modif d'historique de V_NEWAGCP.HREMUOK et l'initialiser à 1 si V_NEWAGCP.HREMU<>0 pour un calcul correct de TOTSMICPRO

- Modifier les salariés pour les mettre en type Travailleurs à domicile.

```
update rhpersonne P set pertypesalarie=5
where exists (select * from rhcontrat C where (ctrempart=1003) and (ctrperid=P.perid))
```

Cette ordre ne peut être fait que par le service développement

- Activer la constante NEWAGCP
- Activer la constante AGCP_AD pour les aides à domicile
- Activer la constante SMICAGCP recalcul smic proraté
- Passer le questionnaire de rubrique 62
- Dans gestion des structures, ajouter le module AGCP aide à domicile
- Ajouter les cotisations AGCP aide à domicile :
AGCPURSSAF_AD,AGCPURSSAFR_AD,AGCPRUAA_AD,AGCPRUAAREG_AD
- Enlever les cotisations AGCP aide à domicile utilisateur
- Ci dessous les modifications d'historique à mettre en place :

A noter que les modifications d'historique doivent être faites pour TOUS les salariés pas uniquement les travailleurs à domicile

AD_HEXO et AD_HNEXO

à remplir par les rubriques utilisées précédemment. On parle ici de la formule BASE

V_NEWAGCP.HREMU

Reprendre AGCP.HREMU.

V_NEWAGCP.REMUCOEF

Reprendre AGCP.REMUCOEF.

V_NEWAGCP.HORBASE

```
si ([ V_AGCP. HREMU] =0)
alors (0)
sinon ([ V_AGCP. HORBASE])
```

V_NEWAGCP.HREMUOK

```
si ([ V_AGCP. HREMU] =0)
alors ( 0)
sinon ( 1)
```

V_AD_AGCP.ADHREMU

Reprendre la rubrique utilisateur AD_AGCP.HREMU.

V_AD_AGCP.ADREMUCOEF

Reprendre la rubrique utilisateur AD_AGCP.REMUCOEF.

V_AD_AGCP.ADHORBASE

```
si ([ _V_AD_AGCP. HREMU] =0)
alors ( 0)
sinon ([ _V_AGCP_AD. HORBASE] )
```

V_AD_AGCP.ADHREMUOK

```
si ([ _V_AGCP_AD. HREMU] =0)
alors ( 0)
sinon ( 1)
```

BC_ADAGCPRUAA

Reprendre la base de cotisation AGCP aide à domicile utilisateur

BC_ADAGCPURSS

Reprendre la base de cotisation AGCP aide à domicile utilisateur

Dans le gestionnaire de rubrique 63, deux nouvelles formules sont ajoutées dans les deux rubriques libres :

HSUP et CUMHSUP.

Elles ne sont utilisées que pour l'édition des réductions de cotisation dans l'édition des charges.

On peut éventuellement faire une modification d'historique pour avoir un état correct mais il faut savoir qu'avant et ce pour toutes les bases, cette donnée était toujours à zéro dans l'édition.

Il faut faire deux modifications d'historique, une avec les salariés travailleurs à domicile et l'autre avec les autres salariés :

Salariés standard :

V_NEWAGCP.HSUP

```
[ AGCP_HREMUPLUS. MONTANT]
```

Travailleurs à domicile :

V_NEWAGCP.HSUP

```
si (( [ AD_HNEXO. BASE] +[ AD_HEX0. BASE] ) =0)
alors (0)
sinon ([ AGCP_HREMUPLUS. MONTANT] * [ AD_HNEXO. BASE] / ([ AD_HNEXO. BASE] + [ AD_HEX0. BASE] ) )
```

V_AGCP_AD.HSUP

```
si (( [ AD_HNEXO. BASE] +[ AD_HEX0. BASE] ) =0)
alors (0)
sinon ([ AGCP_HREMUPLUS. MONTANT] * [ AD_HEX0. BASE] / ([ AD_HNEXO. BASE] + [ AD_HEX0. BASE] ) )
```

Assistants familiaux

Les créations de rubriques:

66_AFDISPO : Indemnité de disponibilité

Rubrique de paie à saisir en paie.

66_AFDISPO.TAUX

```
| CONSTANCE( GENERAL. SMICHOR) *2. 25 |
```

Le montant alimente le brut (BRUT), la base indemnité fin de CDD (B_FINCDD) et le salaire convention CCCNT 66 (CCNT_66)

66_AFNBJABSCP : Nombre de jour absence convenance personnelle

Une rubrique de paie qui permet la saisie du nombre de jour d'absence pour convenance personnelle.

A saisir en variable de paie ; pris en compte dans le calcul d'indemnité d'accueil non réalisé.

Elle viendra minorer le nombre de jour d'absence afin de ne pas déclencher d'indemnité d'accueil non réalisé; car cela n'est pas au fait de l'employeur.

66_AFNBJENF4 : Nombre de jour d'accueil enfant 4

Dans le cas où l'assistant familial peut accueillir au moins 4 enfants.

La base est à saisir ou sera automatiquement calculée au nombre de jour calendaire du mois.

```
si ([ 66_AFNBJENF. MONTANT] >=4) alors ( CONSTANTE( GENERAL. NBJMOISPAYE)) sinon ( 0)
```

66_AFPART4 : Part de l'enfant 4

Part de salaire pour l'enfant 4

```
si ( ( CONSTANTE( CONTRAT. NATURECONTRAT)=3) et ([ NUMCONV. NUMCONV]=66) et  
([ 66_AFNBJENF. MONTANT] >=4) )  
alors (  
    si ([ 66_AF. PRENDRELEG4]=1)  
    alors ([ 66_AF. REMLEG4])  
    sinon ([ 66_AF. REMCONV4])  
)  
sinon ( 0)
```

66_AFNBJENF5 : Nombre de jour d'accueil enfant 5

```
si ([ 66_AFNBJENF. MONTANT] >=5) alors ( CONSTANTE( GENERAL. NBJMOISPAYE)) sinon ( 0)
```

66_AFPART5 : Part de l'enfant 5

Part de salaire pour l'enfant 5

```

si ( ( CONSTANTE( CONTRAT. NATURECONTRAT)=3) et ([ NUMCONV. NUMCONV]=66) et
([ 66_AFNBNF. MONTANT]>=4) )
alors (
    si ([ 66_AF. PRENDRELEG5]=1)
    alors ([ 66_AF. REMLEG5])
    sinon ([ 66_AF. REMCONV5])
)
sinon (0)

```

66_AFNBJABS1 Nombre de jour accueil non réalisé enfant 1

La rubrique permet la gestion de modification de nombre d'enfant au niveau du contrat et neutraliser ou diminuer l'indemnité d'accueil non réalisée.

Par exemple pour les cas, d'arrivée ou de départ d'enfant et que le nombre de jour de présence de l'enfant ne correspond pas au moins sans pour autant déclencher une indemnité d'accueil non réalisée.

L'équivalence pour l'enfant 2 et 3 existe également : 66_AFNBJABS2 et 66_AFNBJABS3

Comment faire lorsque le nombre d'enfant contractuel change en cours de mois ?

- Toujours laisser le nombre d'enfant contractuel le plus haut.
- Saisir le nombre de jour de chaque enfant
- Neutraliser l'indemnité d'accueil non réalisé par la saisie de la rubrique 66_AFNBJABS+Numéro de l'enfant

Exemple 1 : Du 1er au 10 -> 2 enfants et du 11 au 30 -> 1 enfant

- Laisser 2 enfants au contrats
- Saisir le nombre de jour de l'enfant 1 : 30j jours (si pas de saisie il prend automatiquement 30 jours)
- Saisir le nombre de jour de l'enfant 2 : 10 jours.
- Saisir la rubrique nombre de jour d'absence enfant 2 pour neutraliser l'indemnité d'accueil : 20 jours

Exemple 2 : Du 1er au 10 -> 1 enfant et du 11 au 30 -> 2 enfant

Laisser 2 enfants au contrats

- Saisir le nombre de jour de l'enfant 1 : 30j jours (si pas de saisie il prend automatiquement 30 jours)
- Saisir le nombre de jour de l'enfant 2 : 20 jours.

- Saisir la rubrique nombre de jour d'absence enfant 2 pour neutraliser l'indemnité d'accueil : 10 jours

Les modifications de rubrique

66_AFBASE : Salaire de base

Avant :

```
si ( ( CONSTANCE( CONTRAT. NATURECONTRAT) =3) et ( [ NUMCONV. NUMCONV] =66) et  
([ 66_AFN BENF. MONTANT] <>0) )  
alors (  
    [ 66_AFPART1. MONTANT]  
    +[ 66_AFPART2. MONTANT]  
    +[ 66_AFPART3. MONTANT]  
    +[ 66_AFPARTMIXTE. MONTANT]  
)  
sinon ( 0)
```

Après

```
si ( ( CONSTANCE( CONTRAT. NATURECONTRAT) =3) et ( [ NUMCONV. NUMCONV] =66) et  
([ 66_AFN BENF. MONTANT] <>0) )  
alors (  
    [ 66_AFPART1. MONTANT]  
    +[ 66_AFPART2. MONTANT]  
    +[ 66_AFPART3. MONTANT]  
    +[ 66_AFPART4. MONTANT]  
    +[ 66_AFPART5. MONTANT]  
    +[ 66_AFPARTMIXTE. MONTANT]  
)  
sinon ( 0)
```

Prise en compte des nouvelles rubriques pour la part de salaire enfant 4 et enfant 5.

66_AFNBJINT : Nombre de jours d'accueil permanent intermittent

Avant

```
si ( ( CONSTANCE( CONTRAT. NATURECONTRAT) =3) et ( [ NUMCONV. NUMCONV] =66) et  
([ 66_AFN BENF. MONTANT] <>0)  
et ( [ 66_AFACCINTER. MONTANT] =1))
```

```

alors (
    [ 66_AFNBJENF1. MONTANT]
    +
    si ([ 66_AFN BENF. MONTANT] >=2)
    alors (
        [ 66_AFNBJENF2. MONTANT] =0
    )
    sinon ( 0)
    +
    si ([ 66_AFN BENF. MONTANT] >=3)
    alors (
        [ 66_AFNBJENF3. MONTANT]
    )
    sinon ( 0)
)
sinon ( 0)

```

Après

```

si ( ( CONSTAT E( CONTRAT. NATURECONTRAT) =3) et ([ NUMCONV. NUMCONV] =66) et
([ 66_AFN BENF. MONTANT] <>0)
    et ([ 66_AFACCINTER. MONTANT] =1) )
alors (
    [ 66_AFNBJENF1. MONTANT]
    +
    si ([ 66_AFN BENF. MONTANT] >=2)
    alors (
        [ 66_AFNBJENF2. MONTANT]
    )
    sinon ( 0)
    +
    si ([ 66_AFN BENF. MONTANT] >=3)
    alors (
        [ 66_AFNBJENF3. MONTANT]
    )
    sinon ( 0)
    +
    si ([ 66_AFN BENF. MONTANT] >=4)
    alors (
        [ 66_AFNBJENF4. MONTANT]
    )

```

```

        )
    sinon (0)
+
    si ([ 66_AFN BENF. MONTANT] >=5)
    alors (
        [ 66_AFNBJENF5. MONTANT]
    )
    sinon (0)

)
sinon (0)

```

Prise en compte du nombre de jours enfant 4 et 5.

Correction de la petite coquille sur la prise en compte du nombre de jour de l'enfant 2.

66_AFNBJRPER : Nombre de jours d'accueil permanent continu

Avant

```

si ( ( CONSTATANTE( CONTRAT. NATURECONTRAT) =3) et ([ NUMCONV. NUMCONV] =66) et
([ 66_AFN BENF. MONTANT] <>0)
    et ([ 66_AFACCINTER. MONTANT] =0) )
alors (
    [ 66_AFNBJENF1. MONTANT]
+
    si ([ 66_AFN BENF. MONTANT] >=2)
    alors (
        [ 66_AFNBJENF2. MONTANT]
    )
    sinon (0)
+
    si ([ 66_AFN BENF. MONTANT] >=3)
    alors (
        [ 66_AFNBJENF3. MONTANT]
    )
    sinon (0)
)
sinon (0)

```

Après

```
si ( ( CONSTANTE( CONTRAT. NATURECONTRAT)=3) et ([ NUMCONV. NUMCONV]=66) et
([ 66_AFN BENF. MONTANT] <>0)
  et ([ 66_AFACCINTER. MONTANT]=0) )
alors (
  [ 66_AFNBJENF1. MONTANT]
  +
  si ([ 66_AFN BENF. MONTANT] >=2)
  alors (
    [ 66_AFNBJENF2. MONTANT]
    )
  sinon ( 0)
  +
  si ([ 66_AFN BENF. MONTANT] >=3)
  alors (
    [ 66_AFNBJENF3. MONTANT]
    )
  sinon ( 0)
  +
  si ([ 66_AFN BENF. MONTANT] >=4)
  alors (
    [ 66_AFNBJENF4. MONTANT]
    )
  sinon ( 0)
  +
  si ([ 66_AFN BENF. MONTANT] >=5)
  alors (
    [ 66_AFNBJENF5. MONTANT]
    )
  sinon ( 0)
  )
sinon ( 0)
```

Prise en compte du nombre de jours enfant 4 et 5.

Correction de la petite coquille sur la prise en compte du nombre de jour de l'enfant 2.

**Modification de quelques éléments de la rubrique libre : 66_AF
(Données pour calcul salaire des AF (Avt 305))**

66_AF.REMLEG1

Faire appel à la fonction de calcul F_66AF

Même modification pour 66_AF.REMLEG2 , 66_AF.REMLEG3 ,66_AF.REMCONV1 , REMCONV2 et REMCONV3

Avant

```
si ( ( CONSTANTE( CONTRAT. NATURECONTRAT)=3) et ([ NUMCONV. NUMCONV]=66) et
([ 66_AFN BENF. MONTANT] <>0) )
alors (
    si ([ 66_AF ACCINTER. MONTANT]=0)
    alors (
        [ 66_AFNBJENF1. MONTANT]
        /CONSTANTE( GENERAL. NBJMOISPAYE)
        *CONSTANTE( GENERAL. SMICMENS)
    )
    sinon ( 0)
)
sinon ( 0)
```

Après

```
si ( CONSTANTE( GENERAL. OLDAF_61)=0)
alors (
    F_66AF(1; 0;
    [ 66_AFNBJENF1. MONTANT]; [ 66_AFNBJABSCP. BASE]+[ 66_AFNBJABS1. BASE]; [ 66_AF_1ENF. RESULTAT]; [ 66_AF_BA
; 1)
)
sinon (
    si ( ( CONSTANTE( CONTRAT. NATURECONTRAT)=3) et ([ NUMCONV. NUMCONV]=66) et
    ([ 66_AFN BENF. MONTANT] <>0) )
    alors (
        si ([ 66_AF ACCINTER. MONTANT]=0)
        alors (
            [ 66_AFNBJENF1. MONTANT]
            /CONSTANTE( GENERAL. NBJMOISPAYE)
            *CONSTANTE( GENERAL. SMICMENS)
        )
    )
)
```

```

        sinon ( 0)
    )
sinon ( 0)
    )

```

V_66AF.ABSLEG1

Faire appel à la fonction de calcul F_66AF

Même modification dans 66_AF.ABSLEG2 ,66_AF.ABSLEG3 , 66_AF.ABSCONV1 , 66_AF.ABSCONV2 et 66_AF.ABSCONV13

Avant

```

si ( [ 66_AFN BENF. MONTANT] >0)
alors (
    maximum( 0; ( CON STANTE( CONTRAT. CIVJOURSTR AVAIL _EMB)
        - [ 66 _AFNBJENF1. MONTANT] )
    )
    /CONSTANTE( GENERAL. NBJMOISPAYE)
    *0. 8
    *CONSTANTE( GENERAL. SMICMENS)
    )
sinon ( 0)

```

Après

```

si ( CON STANTE( GENERAL. OLDAF _61) =0)
alors (
    F_66AF( 0; 0; [ 66 _AFNBJENF1. MONTANT]; [ 66 _AFNBJABSCP. BASE] -
    [ 66 _AFNBJABS1. BASE]; [ 66 _AF _1ENF. RESULTAT]; [ 66 _AF _BASE. RESULTAT] ; 1)
    )
sinon (
    si ( [ 66_AFN BENF. MONTANT] >0)
    alors (
        maximum( 0; ( CON STANTE( CONTRAT. CIVJOURSTR AVAIL _EMB)
            - [ 66 _AFNBJENF1. MONTANT] )
        )
        /CONSTANTE( GENERAL. NBJMOISPAYE)
        *0. 8
    )

```

```
*CONSTANTE( GENERAL. SMICMENS)  
)  
sinon ( 0)  
)
```

Création de 66_AF.REMLEG4 : Rémunération légale 4

```
F_66AF( 1; 0; [ 66_AFNBJENF4. MONTANT]; 0; [ 66_AF_3ENF. RESULTAT]; [ 66_AF_2ENF. RESULTAT] ; 4)
```

Création de 66_AF.REMCONV4 : Rémunération conventionnelle 4

```
F_66AF( 1; 1; [ 66_AFNBJENF4. MONTANT]; 0; [ 66_AF_3ENF. RESULTAT]; [ 66_AF_2ENF. RESULTAT] ; 4)
```

Idem pour la rémunération pour l'enfant 5 66_AF.REMLEG5 et 66_AF.REMCONV5

Pas d'indemnité d'accueil non réalisé pour les enfants 4 et 5

Revision #46

Created 7 March 2023 15:56:52 by Valéry HUMEZ

Updated 29 August 2023 14:37:49 by Valéry HUMEZ