

Version 62

- [Version 6](#)
- [Version 7 : AGCP](#)
- [Version 8](#)
- [Version 9](#)
- [Version 10](#)
- [Version 11](#)

Version 6

Calcul de plafond

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 ) + [ NBHRSCOMPL. 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. FORFAITJOUR)
    *
    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 (

```

Calcul AGCP et SMICPRORATE

Remplacement de histocumulsitu par histo cumuljoint car le calcul annuel se fait désormais tout contrat confondu

BC MALCOMP25 : Assiette complément maladie si salaire > 2.5 SMIC

Avant

```

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)
  )

```

Après

```

si (
    (([ V_AGCP. CUMREMUCOEUF] ) > ( 2. 5*[ V_AGCP. CUMSMICPRORATE] ))
    et ([ V_AGCP. CUMREMUCOEUF] >0)
    et ([ V_AGCP. CUMSMICPRORATE] >0)
  )
alors (
    histocumuljoint([ BC_BRUT_URSAFF. MONTANT]) + [ BC_BRUT_URSAFF. MONTANT]
    - histocumuljoint([ BC_MALCOMP25. MONTANT])
  )
sinon (
    histocumuljoint([ BC_MALCOMP25. MONTANT]) *(-1)
  )

```

BC REG_ALLOC Base régularisation alloc

Avant

```

si (
    ([ V_AGCP. CUMREMUCOEUF] <=( 3. 5*[ V_AGCP. CUMSMICPRORATE] ))
    ou ([ V_AGCP. CUMREMUCOEUF] <=0)
    ou ([ V_AGCP. CUMSMICPRORATE] <=0)
  )
alors (
    histocumulsitu([ V_ALLOC. BASEALLOC]) *(-1)
    -histocumulsitu([ BC_REG_ALLOC. MONTANT])
  )
sinon (
    histocumulsitu([ V_ALLOC. BASEALLOCRED])
    +histocumulsitu([ BC_REG_ALLOCRE. MONTANT])
  )

```

```
)
```

Après

```
si (
  ([ V_AGCP. CUMREMUCOEF] <=( 3. 5*[ V_AGCP. CUMSMICPRORATE] ))
  ou ([ V_AGCP. CUMREMUCOEF] <=0)
  ou ([ V_AGCP. CUMSMICPRORATE] <=0)
)
alors (
  histocumuljoint([ V_ALLOC. BASEALLOC] )*(-1)
  -histocumuljoint([ BC_REG_ALLOC. MONTANT] )
)
sinon (
  histocumuljoint([ V_ALLOC. BASEALLOCRED] )
  +histocumuljoint([ BC_REG_ALLOCRE. MONTANT] )
)
```

Correctif sur fonction F_AGCP

Avant

```
si ((si (moisfin=0) alors (12) sinon (moisfin))<[ AGCPDEBUT. MONTANT])
alors (0)
sinon (
  si ((moisdebut=0) ou (moisfin=0))
  alors (histocumulsitu(RUBACUMULER))
  sinon (
    histocumulsitu(RUBACUMULER;
      maximum(1;(si (moisdebut=0) alors (1) sinon
(moisdebut));[ AGCPDEBUT. MONTANT]));
      CONSTANTE( GENERAL. ANNEE_PAYE);
      (si (moisfin=0) alors (12) sinon
(moisfin));
      CONSTANTE( GENERAL. ANNEE_PAYE)
    )
  )
)
```

Après

```

si ((si (moisfin=0) alors (12) sinon (moisfin))<[ AGCPDEBUT. MONTANT])
alors (0)
sinon (
si ((moisdebut=0) ou (moisfin=0))
alors (histocumuljoint(RUBACUMULER))
sinon (
histocumuljoint(RUBACUMULER;
maximum(1;(si (moisdebut=0) alors (1) sinon
(moisdebut));[ AGCPDEBUT. MONTANT]));
CONSTANTE( GENERAL. ANNEE_PAYE);
(si (moisfin=0) alors (12) sinon
(moisfin));
CONSTANTE( GENERAL. ANNEE_PAYE)
)
)
)
)

```

Assistantes familiales

66 AFDISPO : Indemnité de disponibilité

A saisir en variable de paye

Indemnité de disponibilité qui « ne peut être inférieure à 2.25 fois le montant du salaire minimum de croissance, sans pouvoir être inférieure à 90% de la rémunération prévue par le contrat de travail ».

Pour chaque journée de disponibilité, une comparaison doit être effectuée entre le salaire légal (2.25 x Smic horaire) et contractuel (90% de la rémunération contractuelle).

66 AFPARTMIXTE : Part accueil mixte

Correctif pour que la rubrique fonctionne même si pas d'enfant au contrat.

Et comparatif entre le légal et le conventionnel

Avant :

```

si ( ( CONSTANTE( CONTRAT. NATURECONTRAT) =3) et ([ NUMCONV. NUMCONV] =66) et
([ 66_AFNBNF. MONTANT] <>0) )
alors (

```

```

[ 66_AFJRSINTER. MONTANT] /26
*([ 66_AF_1ENF. RESULTAT])
*[ VP. VP]*(1+CONSTANTE( GENERAL. TAUXSUJETION66) /100)
)
sinon (0)

```

Après

```

si ( ( CONSTANTE( CONTRAT. NATURECONTRAT)=3) et ([ NUMCONV. NUMCONV]=66) )
alors (
    maximum(
        [ 66_AFJRSINTER. BASE] *5. 06*CONSTANTE( GENERAL. SMICHOR)
        ;
        [ 66_AFJRSINTER. MONTANT] /26
        *([ 66_AF_1ENF. RESULTAT])
        *[ VP. VP]*(1+CONSTANTE( GENERAL. TAUXSUJETION66) /100)
    )
)
sinon (0)

```

66_AFNBJINT : Nombre de jours d'accueil permanent intermittent

Erreur sur la formule sur le nombre de jour d'accueil enfant 2

Avant

```

si ( ( CONSTANTE( CONTRAT. NATURECONTRAT)=3) et ([ NUMCONV. NUMCONV]=66) et
([ 66_AFNBENF. MONTANT] <>0)
    et ([ 66_AFACCINTER. MONTANT]=1) )
alors (
    [ 66_AFNBJENF1. MONTANT]
    +
    si ([ 66_AFNBENF. MONTANT] >=2)
    alors (
        [ 66_AFNBJENF2. MONTANT]=0
    )
    sinon (0)
    +
    si ([ 66_AFNBENF. 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]=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)  
  )  
sinon (0)
```

66 AFNBABSCP : Nombre de jour absence convenance personnelle

Création de la rubrique pour minorer l'accueil non réalisé

66 AFNBABS1

Création de la rubrique pour minorer l'accueil non réalisé par enfant

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_AFNBABS+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

Augmenter le nombre d'enfant

Afin de simplifier la rubrique 66_AF, on va créer des fonctions qui donneront le résultat.

Création d'une fonction :

66AF (REMU , CONV , NBJACCUEIL,NBJABSCP , PROG , PRECPROG,NUMERO)

- REMU : 1 si calcul de la rémunération 0 si calcul de l'absence
- CONV : 1 si calcul conventionnel, 0 si calcul légal
- NBJACCUEIL : Rubrique contenant le nombre de jour d'accueil de l'enfant
- NBJABSCP : Nombre de jour d'absence à déduire de l'accueil non réalisé
- PROG : Rubrique de progression de l'enfant pour le calcul de la rémunération conventionnelle
- PRECPROG : Rubrique de progression de l'enfant précédent pour le calcul de la rémunération conventionnelle
- NUMERO : Numéro de l'enfant (pour le test avec 66_AFN BENF nombre d'enfant contractuel)

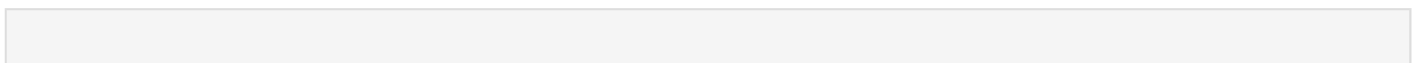
Passage à 5 enfants :

Création des rubriques suivantes sur le même modèle que les autres :

- 66_AF_4ENF,66_AF_5ENF
- 66_AFPART4,66_AFPART5

Il n'y a pas d'indemnité pour accueil non réalisé pour les enfants 4 et 5

66_AFIACCNE : Indemnité compensatrice pour accueil non réalisé



```

si ([ 66_AF. PRENDRELEG1]=1)
    alors ([ 66_AF. ABSLEG1])
    sinon ([ 66_AF. ABSCONV1])
+
si ( [ 66_AFN BENF. MONTANT]>1
)
alors (
    si ([ 66_AF. PRENDRELEG2]=1)
    alors ([ 66_AF. ABSLEG2])
    sinon ([ 66_AF. ABSCONV2])
)
sinon (0)
+
si ( [ 66_AFN BENF. MONTANT]>2
)
alors (
    si ([ 66_AF. PRENDRELEG3]=1)
    alors ([ 66_AF. ABSLEG3])
    sinon ([ 66_AF. ABSCONV3])
)
sinon (0)

```

66_AFBASE : Salaire de base

```

si ( ( CONSTANTE( 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)

```

66_AFNBJRPER : Nombre de jours d'accueil Permanent continu

```

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_AFNBENF. MONTANT] >=2)
  alors (
    [ 66_AFNBJENF2. MONTANT]
    )
  sinon (0)
  +
  si ([ 66_AFNBENF. MONTANT] >=3)
  alors (
    [ 66_AFNBJENF3. MONTANT]
    )
  sinon (0)
  +
  si ([ 66_AFNBENF. MONTANT] >=4)
  alors (
    [ 66_AFNBJENF4. MONTANT]
    )
  sinon (0)
  +
  si ([ 66_AFNBENF. MONTANT] >=5)
  alors (
    [ 66_AFNBJENF5. MONTANT]
    )
  sinon (0)
)
sinon (0)

```

66_AFNBJINT : Nombre de jours d'accueil permanent intermittent

```

si ( ( CONSTANTE( CONTRAT. NATURECONTRAT) =3) et ([ NUMCONV. NUMCONV] =66) et
([ 66_AFNBENF. MONTANT] <>0)
  et ([ 66_AFACCINTER. MONTANT] =1) )
alors (
  [ 66_AFNBJENF1. MONTANT]
  +
  si ([ 66_AFNBENF. MONTANT] >=2)
  alors (

```

```
        [ 66_AFNBJENF2. MONTANT]
    )
sinon (0)
+
si ([ 66_AFNBJENF. MONTANT] >=3)
alors (
    [ 66_AFNBJENF3. MONTANT]
)
sinon (0)
+
si ([ 66_AFNBJENF. MONTANT] >=4)
alors (
    [ 66_AFNBJENF4. MONTANT]
)
sinon (0)
+
si ([ 66_AFNBJENF. MONTANT] >=5)
alors (
    [ 66_AFNBJENF5. MONTANT]
)
sinon (0)

)
sinon (0)
```

Version 7 : AGCP

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 (
    ( histoactifjoint( HREMU; MACALCULER; CONSTANTE( GENERAL. ANNEE_PAYE) ) /
    CONSTANTE( CONTRAT. HORAIRECOLLECTIF ) *CONSTANTEHIS( GENERAL. SMICMENS; MACALCULER; CONSTANTE( GENERAL.
    )
    *histoactifjoint( HORBASE; MACALCULER; CONSTANTE( GENERAL. ANNEE_PAYE) ) /1607
    )
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
- CUMREMUCOEF : 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 proratés de l'année

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

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

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

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

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

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

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

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

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

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

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

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

+F_AGCPSMICMENS( [ AGCPDEBUT. BASE ] ; 12 ; [ V_NEWAGCP. HREMU ] ; [ V_NEWAGCP. HORBASE ] )

        )
sinon (
        F_AGCPSMICMENS( [ AGCPDEBUT. BASE ] ; 1 ; [ V_AGCP. HREMU ] ; [ V_AGCP. HORBASE ] )
        +F_AGCPSMICMENS( [ AGCPDEBUT. BASE ] ; 2 ; [ V_AGCP. HREMU ] ; [ V_AGCP. HORBASE ] )
        +F_AGCPSMICMENS( [ AGCPDEBUT. BASE ] ; 3 ; [ V_AGCP. HREMU ] ; [ V_AGCP. HORBASE ] )
        +F_AGCPSMICMENS( [ AGCPDEBUT. BASE ] ; 4 ; [ V_AGCP. HREMU ] ; [ V_AGCP. HORBASE ] )
        +F_AGCPSMICMENS( [ AGCPDEBUT. BASE ] ; 5 ; [ V_AGCP. HREMU ] ; [ V_AGCP. HORBASE ] )
)
```

```

+F_AGCPSMICMENS( [ AGCPDEBUT. BASE] ; 6; [ V_AGCP. HREMU] ; [ V_AGCP. HORBASE] )
+F_AGCPSMICMENS( [ AGCPDEBUT. BASE] ; 7; [ V_AGCP. HREMU] ; [ V_AGCP. HORBASE] )
+F_AGCPSMICMENS( [ AGCPDEBUT. BASE] ; 8; [ V_AGCP. HREMU] ; [ V_AGCP. HORBASE] )
+F_AGCPSMICMENS( [ AGCPDEBUT. BASE] ; 9; [ V_AGCP. HREMU] ; [ V_AGCP. HORBASE] )
+F_AGCPSMICMENS( [ AGCPDEBUT. BASE] ; 10; [ V_AGCP. HREMU] ; [ V_AGCP. HORBASE] )
+F_AGCPSMICMENS( [ AGCPDEBUT. BASE] ; 11; [ V_AGCP. HREMU] ; [ V_AGCP. HORBASE] )
+F_AGCPSMICMENS( [ AGCPDEBUT. BASE] ; 12; [ V_AGCP. HREMU] ; [ V_AGCP. HORBASE] )
)
)

```

- 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

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

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

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 perypesalarie=5
where exists (select * from rhcontrat C where (ctrempart=1003) and (ctrperid=P.perid))
```

- Activer la constante NEWAGCP
- Activer la constante AGCP_AD pour les aides à domicile
- Activer la constante SMICAGCP recalcul smic proraté
- Passer le gestionnaire 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 :

AD_HEXO et AD_HNEXO

à remplir par les rubriques utilisées précédemment.

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_NEWAGCP. HREMU] =0)
alors (0)
sinon (1)
```

V_AD_AGCP.HREMU

Reprendre la rubrique utilisateur AD_AGCP.HREMU.

V_AD_AGCP.REMUCOEF

Reprendre la rubrique utilisateur AD_AGCP.REMUCOEF.

V_AD_AGCP.HORBASE

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

V_AD_AGCP.HREMUOK

```
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

Version 8

Ajout TH_DIM : paiement des dimanches et fériés pour les TH

Version 9

BC_MALCOMP25

Prise en compte de la constante générale FILLONCONTRAT

Prise en compte de la nouvelle rubrique V_NEWAGCP

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. FILLONCONTRAT) =1)
alors (
  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)
  )
)
```

```

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

```

Allocation familiale

Prise en compte de la nouvelle rubrique V_NEWAGCP

ALLOCFAM.TAUX

Avant

```
si (
  ([ V_AGCP. CUMREMUCOEF ] > 3. 5 * [ V_AGCP. CUMSMICPRORATE ] )
  et ([ V_AGCP. CUMREMUCOEF ] > 0)
  et ([ V_AGCP. CUMSMICPRORATE ] > 0)
)
alors ( LISTEGEN( TX_ALLOCF; 0001 ) )
sinon ( 0 )
```

Après

```
si ( CONSTANTE( GENERAL. NEWAGCP ) = 0 )
alors (
  si (
    ([ V_AGCP. CUMREMUCOEF ] > 3. 5 * [ V_AGCP. CUMSMICPRORATE ] )
    et ([ V_AGCP. CUMREMUCOEF ] > 0)
    et ([ V_AGCP. CUMSMICPRORATE ] > 0)
  )
  alors ( LISTEGEN( TX_ALLOCF; 0001 ) )
  sinon ( 0 )
)
sinon (
  si (
    ([ V_NEWAGCP. CUMREMUCOEF ] > 3. 5 * [ V_NEWAGCP. CUMSMICPRO ] )
    et ([ V_NEWAGCP. CUMREMUCOEF ] > 0)
    et ([ V_NEWAGCP. CUMSMICPRO ] > 0)
  )
  alors ( LISTEGEN( TX_ALLOCF; 0001 ) )
  sinon ( 0 )
)
```

ALLOCFAMREDUIT.TAUX

Avant

```
si (( [ V_AGCP. CUMREMUCOEF ] > 3. 5 * [ V_AGCP. CUMSMICPRORATE ] ) et ([ V_AGCP. CUMREMUCOEF ] > 0 ) )
alors ( 0 )
sinon ( LISTEGEN( TX_ALLOCF; 0002 ) )
```

Après

```
si ( CONSTANTE( GENERAL. NEWAGCP) =0)
alors (
  si (
    ([ V_AGCP. CUMREMUCOEF] >3. 5*[ V_AGCP. CUMSMICPRORATE] )
    et ([ V_AGCP. CUMREMUCOEF] >0)
    et ([ V_AGCP. CUMSMICPRORATE] >0)
  )
  alors ( 0)
  sinon ( LISTEGEN( TX_ALLOCF; 0002) )
)
sinon (
  si (
    ([ V_NEWAGCP. CUMREMUCOEF] >3. 5*[ V_NEWAGCP. CUMSMICPRO] )
    et ([ V_NEWAGCP. CUMREMUCOEF] >0)
    et ([ V_NEWAGCP. CUMSMICPRO] >0)
  )
  alors ( 0)
  sinon ( LISTEGEN( TX_ALLOCF; 0002) )
)
```

ALLOCFAMCOMP.TAUX

Avant

```
si (([ V_AGCP. CUMREMUCOEF] >3. 5*[ V_AGCP. CUMSMICPRORATE] ) et ([ V_AGCP. CUMREMUCOEF] >0))
alors ( 0) sinon ( 1. 80)
```

Après

```
si ( CONSTANTE( GENERAL. NEWAGCP) =0)
alors (
  si (
    ([ V_AGCP. CUMREMUCOEF] >3. 5*[ V_AGCP. CUMSMICPRORATE] )
    et ([ V_AGCP. CUMREMUCOEF] >0)
    et ([ V_AGCP. CUMSMICPRORATE] >0)
  )
  alors ( 0) sinon ( 1. 80)
)
sinon (
```

```

si (
  ([ V_NEWAGCP. CUMREMUCOEF ] > 3. 5 * [ V_NEWAGCP. CUMSMICPRO ])
  et ([ V_NEWAGCP. CUMREMUCOEF ] > 0)
  et ([ V_NEWAGCP. CUMSMICPRO ] > 0)
)
alors ( 0 ) sinon ( 1. 80 )
)

```

ALLOCFAMCOMPNEG.TAUX

Avant

```

si (( [ V_AGCP. CUMREMUCOEF ] > 3. 5 * [ V_AGCP. CUMSMICPRORATE ]) et ([ V_AGCP. CUMREMUCOEF ] > 0))
alors ( 0 ) sinon ( -1. 80 )

```

Après

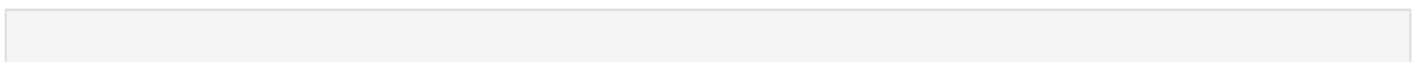
```

si ( CONSTANTE( GENERAL. NEWAGCP ) = 0)
alors (
  si (
    ([ V_AGCP. CUMREMUCOEF ] > 3. 5 * [ V_AGCP. CUMSMICPRORATE ])
    et ([ V_AGCP. CUMREMUCOEF ] > 0)
    et ([ V_AGCP. CUMSMICPRORATE ] > 0)
  )
  alors ( 0 ) sinon ( -1. 80 )
)
sinon (
  si (
    ([ V_NEWAGCP. CUMREMUCOEF ] > 3. 5 * [ V_NEWAGCP. CUMSMICPRO ])
    et ([ V_NEWAGCP. CUMREMUCOEF ] > 0)
    et ([ V_NEWAGCP. CUMSMICPRO ] > 0)
  )
  alors ( 0 ) sinon ( -1. 80 )
)

```

BC REG ALLOC

Avant



```

si (([ V_AGCP. CUMREMUCOEF] <=( 3. 5*[ V_AGCP. CUMSMICPRORATE] ))
    ou ([ V_AGCP. CUMREMUCOEF] <=0))
alors (
    histocumulsitu([ V_ALLOC. BASEALLOC])*(-1)
    -histocumulsitu([ BC_REG_ALLOC. MONTANT])
)
sinon (
    histocumulsitu([ V_ALLOC. BASEALLOCRED])
    +histocumulsitu([ BC_REG_ALLOCRE. MONTANT])
)

```

Après

```

si (CONSTANTE( GENERAL. NEWAGCP)=1)
alors (
    si (CONSTANTE( GENERAL. FILLONCONTRAT)=1)
    alors (
        si (
            ([ V_NEWAGCP. CUMREMUCOEF] <=( 3. 5*[ V_NEWAGCP. CUMSMICPRO] ))
            ou ([ V_NEWAGCP. CUMREMUCOEF] <=0)
            ou ([ V_NEWAGCP. CUMSMICPRO] <=0)
        )
        alors (
            histocumulsitu([ V_ALLOC. BASEALLOC])*(-1)
            -histocumulsitu([ BC_REG_ALLOC. MONTANT])
        )
        sinon (
            histocumulsitu([ V_ALLOC. BASEALLOCRED])
            +histocumulsitu([ BC_REG_ALLOCRE. MONTANT])
        )
    )
    sinon (
        si (
            ([ V_NEWAGCP. CUMREMUCOEF] <=( 3. 5*[ V_NEWAGCP. CUMSMICPRO] ))
            ou ([ V_NEWAGCP. CUMREMUCOEF] <=0)
            ou ([ V_NEWAGCP. CUMSMICPRO] <=0)
        )
        alors (
            histocumuljoint([ V_ALLOC. BASEALLOC])*(-1)
            -histocumuljoint([ BC_REG_ALLOC. MONTANT])
        )
    )
)

```

```

        )
    sinon (
        histocumuljoint([ V_ALLOC. BASEALLOCRED])
        +histocumuljoint([ BC_REG_ALLOCRE. MONTANT])
    )
)
)
sinon (
    si ( CONSTANCE( GENERAL. FILLONCONTRAT) =1)
    alors (
        si (
            ([ V_AGCP. CUMREMUCOEF] <=( 3. 5*[ V_AGCP. CUMSMICPRORATE] ))
            ou ([ V_AGCP. CUMREMUCOEF] <=0)
            ou ([ V_AGCP. CUMSMICPRORATE] <=0)
        )
        alors (
            histocumulsitu([ V_ALLOC. BASEALLOC]) *(-1)
            -histocumulsitu([ BC_REG_ALLOC. MONTANT])
        )
        sinon (
            histocumulsitu([ V_ALLOC. BASEALLOCRED])
            +histocumulsitu([ BC_REG_ALLOCRE. MONTANT])
        )
    )
)
sinon (
    si (
        ([ V_AGCP. CUMREMUCOEF] <=( 3. 5*[ V_AGCP. CUMSMICPRORATE] ))
        ou ([ V_AGCP. CUMREMUCOEF] <=0)
        ou ([ V_AGCP. CUMSMICPRORATE] <=0)
    )
    alors (
        histocumuljoint([ V_ALLOC. BASEALLOC]) *(-1)
        -histocumuljoint([ BC_REG_ALLOC. MONTANT])
    )
    sinon (
        histocumuljoint([ V_ALLOC. BASEALLOCRED])
        +histocumuljoint([ BC_REG_ALLOCRE. MONTANT])
    )
)
)

```


Version 10

Je supprime la constante FILLONCONTRAT et modifie toutes les formules y faisant référence

Version 11

Modification de I_ICP

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

Avant

```
DECLARER( MEMB)
AFFECTER( MEMB; CONSTANTE( CONTRAT. MOIS_EMBAUCHE) )
DECLARER( AEMB)
AFFECTER( AEMB; CONSTANTE( CONTRAT. ANNEE_EMBAUCHE) )
DECLARER( MPAYE)
AFFECTER( MPAYE; CONSTANTE( GENERAL. MOIS_PAYE) )
DECLARER( APAYE)
AFFECTER( APAYE; CONSTANTE( GENERAL. ANNEE_PAYE) )
SI(
  CONSTANTE( CONTRAT. PAIEMENTTCP) =
    ( 0) )
ALORS( 0)
SINON(
  SI(
    CONSTANTE( CONTRAT. PAIEMENTTCP) =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

```
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 ) ) )
```