version 104

- Version 1

Version 1

A faire avant de la passer à Clément :

Correction rubrique BC_TR_C_PREV_NC : Utilise BC_TR_C_PREV_C

Avant

```
F_TRANCHECPREV([BC_BRUT_PREV_C.MONTANT];[BC_TR_A_PREV_C.MONTANT]  ;[BC_TR_B_PREV_C.MONTANT]
;[BC_TR_C_PREV_C.MONTANT];[PLAFONDSECU.PLAFOND_OC];[V_PLAFPREV.NB_TRC_NC])
```

Après

```
F_TRANCHECPREV([BC_BRUT_PREV_NC.MONTANT];[BC_TR_A_PREV_NC.MONTANT]
;[BC_TR_B_PREV_NC.MONTANT]
;[BC_TR_C_PREV_NC.MONTANT];[PLAFONDSECU.PLAFOND_OC];[V_PLAFPREV.NB_TRC_NC])
```

Mise à jour formule F_RGDU pour nouveau paramètre RGDU AD

Il s'agit d'ajouter le nouveau paramètre Heure Non exonéré.

Le paramètre Heure contrat qui était utilisé précédemment redevient Heure contrat

je fait également la correction des heures sup : ils les envoie proraté alors que le programme les prorate aussi

Avant

```
declarer (FORFAITJOUR)
affecter (FORFAITJOUR;si (CONSTANTE(CONTRAT.SITUATIONHORAIRE)=4)
                      alors (1)
                      sinon (0)
 )
declarer (AD)
affecter (AD;si (CONSTANTE(CONTRAT.TYPESALARIE)=5)
             alors (1)
             sinon (0)
 )
declarer (NBHEURES)
affecter (NBHEURES;si (:AD=1)
```

```
                  alors ([AD_HNEXO.BASE])
                  sinon (
                        si (:FORFAITJOUR=1)
                        alors (0)
                        sinon (si (CONSTANTE(CONTRAT.TYPESALARIE)=3)
                               alors (CONSTANTE(CONTRAT.HORAIRECOLLECTIF))
                               sinon (CONSTANTE(CONTRAT.HORAIREMENSUEL))
                              )
                        )
          )
declarer (NBJOURS)
affecter (NBJOURS;si ((:FORFAITJOUR=0) ou (:AD=1))
□□□alors (0)
□□□sinon (CONSTANTE(CONTRAT.JOUR_FORFAIT))
          )

declarer (HSUP)
affecter (HSUP;si (:AD=1)

alors([AGCP_HREMUPLUS.MONTANT]*[AD_HNEXO.BASE]/([AD_HNEXO.BASE]+[AD_HEXO.BASE]))
               sinon ([AGCP_HREMUPLUS.MONTANT])
□  )
declarer (HORBASE)
affecter (HORBASE;si ([AGCPHORAN.BASE]=0)
                  alors (
□□□si (LISTEGEN(TPSAN;CONSTANTE(CONTRAT.CATPERSO))=0)
□□□alors (1607)
□□□sinon (LISTEGEN(TPSAN;CONSTANTE(CONTRAT.CATPERSO)))
□□□)
□□  sinon ([AGCPHORAN.BASE])
□  )
declarer (HEXO)
affecter (HEXO;si (:AD=1)
               alors ([AD_HEXO.BASE])
               sinon (0)
         )
declarer (PPV)
affecter (PPV;[P_PPVEXO.MONTANT]+[P_PPVNONEXO.MONTANT])
declarer (ANNEE)
affecter (ANNEE;si ([PLAFONDSECU.NUMCAS]=2)
```

```
                alors (CONSTANTE(CONTRAT.ANNEE_CLOTURE))
                sinon (CONSTANTE(GENERAL.ANNEE_PAYE)))
declarer (MOIS)
affecter (MOIS;si ([PLAFONDSECU.NUMCAS]=2)
                alors (CONSTANTE(CONTRAT.MOIS_CLOTURE))
                sinon (CONSTANTE(GENERAL.MOIS_PAYE)))

rgdu(TYPEMONTANT;
    :ANNEE;
    :MOIS;
    CONSTANTE(GENERAL.FNAL010);
    :NBHEURES;
    :NBJOURS;
    :HSUP;
    [AGCPITREMU.MONTANT];
    [AGCP_NUMHREMU.MONTANT];
    [B_PXHEURE.MONTANT];
    :HORBASE;
    :AD;
    :HEXO;
    :PPV)
```

Après

```
declarer (FORFAITJOUR)
affecter (FORFAITJOUR;si (CONSTANTE(CONTRAT.SITUATIONHORAIRE)=4)
                     alors (1)
                     sinon (0)
  )
declarer (AD)
affecter (AD;si (CONSTANTE(CONTRAT.TYPESALARIE)=5)
             alors (1)
             sinon (0)
  )
declarer (NBHEURES)
affecter (NBHEURES;
                       si (:FORFAITJOUR=1)
                       alors (0)
                       sinon (si (CONSTANTE(CONTRAT.TYPESALARIE)=3)
                              alors (CONSTANTE(CONTRAT.HORAIRECOLLECTIF))
```

```
                                sinon (CONSTANTE(CONTRAT.HORAIREMENSUEL))
                               )
          )
declarer (NBJOURS)
affecter (NBJOURS;si ((:FORFAITJOUR=0) ou (:AD=1))
□□□alors (0)
□□□sinon (CONSTANTE(CONTRAT.JOUR_FORFAIT))
          )

declarer (HSUP)
affecter (HSUP;[AGCP_HREMUPLUS.MONTANT]
□  )
declarer (HORBASE)
affecter (HORBASE;si ([AGCPHORAN.BASE]=0)
                  alors (
□□□si (LISTEGEN(TPSAN;CONSTANTE(CONTRAT.CATPERSO))=0)
□□□alors (1607)
□□□sinon (LISTEGEN(TPSAN;CONSTANTE(CONTRAT.CATPERSO)))
□□□)
□□  sinon ([AGCPHORAN.BASE])
□  )
declarer (HEXO)
affecter (HEXO;si (:AD=1)
               alors ([AD_HEXO.BASE])
               sinon (0)
         )
declarer (HNEXO)
affecter (HNEXO;si (:AD=1)
               alors ([AD_HNEXO.BASE])
               sinon (0)
         )
declarer (PPV)
affecter (PPV;[P_PPVEXO.MONTANT]+[P_PPVNONEXO.MONTANT])
declarer (ANNEE)
affecter (ANNEE;si ([PLAFONDSECU.NUMCAS]=2)
                alors (CONSTANTE(CONTRAT.ANNEE_CLOTURE))
                sinon (CONSTANTE(GENERAL.ANNEE_PAYE)))
declarer (MOIS)
affecter (MOIS;si ([PLAFONDSECU.NUMCAS]=2)
```

```
                alors (CONSTANTE(CONTRAT.MOIS_CLOTURE))
                sinon (CONSTANTE(GENERAL.MOIS_PAYE)))

rgdu(TYPEMONTANT;
    :ANNEE;
    :MOIS;
    CONSTANTE(GENERAL.FNAL010);
    :NBHEURES;
    :NBJOURS;
    :HSUP;
    [AGCPITREMU.MONTANT];
    [AGCP_NUMHREMU.MONTANT];
    [B_PXHEURE.MONTANT];
    :HORBASE;
    :AD;
    :HEXO;
    :PPV;
    :HNEXO)
```

Mise à jour de la rubrique 51_DIFFSMIC

La rubrique différentiel SMIC se déclenchait pour quelques centimes lorsque le salaire conventionnel était égale au SMIC à cause des arrondi de 151,67.

Avant :

```
SI([NUMCONV.NUMCONV] = 51 )
alors (
si (([51_MINCONV.MONTANT]*151.67/CONSTANTE(CONTRAT.HORAIREMENSUEL))<1400)
    alors (

maximum((CONSTANTE(GENERAL.SMICHOR)*MINIMUM(CONSTANTE(CONTRAT.HORAIREMENSUEL);35*52/12))-
[51_SALMINSMIC.MONTANT];0)
            )
    sinon (

maximum((CONSTANTE(GENERAL.SMICHOR)*MINIMUM(CONSTANTE(CONTRAT.HORAIREMENSUEL);35*52/12))-
                  ([51_SALMINSMIC.MONTANT]-[51_SALMINCONV.MONTANT]+[51_MINCONV.MONTANT]);0)
            )
      )
```

```
sinon (0)
```

Après :

```
SI([NUMCONV.NUMCONV] = 51 ET (CONSTANTE(GENERAL.SMICMENS) > CONSTANTE(GENERAL.51MINCONV)) )
alors (
si (([51_MINCONV.MONTANT]*151.67/CONSTANTE(CONTRAT.HORAIREMENSUEL))<1400)
    alors (

maximum((CONSTANTE(GENERAL.SMICHOR)*MINIMUM(CONSTANTE(CONTRAT.HORAIREMENSUEL);35*52/12))-
[51_SALMINSMIC.MONTANT];0)
            )
    sinon (

maximum((CONSTANTE(GENERAL.SMICHOR)*MINIMUM(CONSTANTE(CONTRAT.HORAIREMENSUEL);35*52/12))-
                  ([51_SALMINSMIC.MONTANT]-[51_SALMINCONV.MONTANT]+[51_MINCONV.MONTANT]);0)
            )
      )
sinon (0)
```

Mise à jour de la rubrique BC_TR_C_PRV_NC

Les rubriques utilisées pour le calcul de BC_TR_C_PREV_NC étaient des rubriques de prévoyance Cadre au lieu de rubriques de prévoyance Non-cadre.

Avant :

```
F_TRANCHECPREV(
  [BC_BRUT_PREV_C.MONTANT];
  [BC_TR_A_PREV_C.MONTANT];
  [BC_TR_B_PREV_C.MONTANT];
  [BC_TR_C_PREV_C.MONTANT];
  [PLAFONDSECU.PLAFOND_OC];
  [V_PLAFPREV.NB_TRC_NC]
  )
```

Après :

```
F_TRANCHECPREV(
  [BC_BRUT_PREV_NC.MONTANT];
  [BC_TR_A_PREV_NC.MONTANT];
  [BC_TR_B_PREV_NC.MONTANT];
  [BC_TR_C_PREV_NC.MONTANT];
  [PLAFONDSECU.PLAFOND_OC];
  [V_PLAFPREV.NB_TRC_NC]
  )
```

Correction de la rubrique BC_MACOMP25

Avant

```
      si (
            (([V_NEWAGCP.CUMREMUAF]) >  ([V_NEWAGCP.TXSMICMAL]*[V_NEWAGCP.CUMSMICPRO3112]))
            et ([V_NEWAGCP.CUMREMUAF]>0)
            et ([V_NEWAGCP.CUMSMICPRO3112]>0)
            )
         alors (
                f_histoclotsit([BC_BRUT_URSAFF.MONTANT])+ [BC_BRUT_URSAFF.MONTANT]
                - f_histoclotsit([BC_MALCOMP25.MONTANT])
                )
         sinon (
               f_histoclotsit([BC_MALCOMP25.MONTANT])*(-1)
               )
```

Après

```
si (([PLAFONDSECU.NUMCAS]=2) et (CONSTANTE(CONTRAT.ANNEE_CLOTURE)<2026))
alors (
      si (
            (([V_NEWAGCP.CUMREMUAF]) >  ([V_NEWAGCP.TXSMICMAL]*[V_NEWAGCP.CUMSMICPRO3112]))
            et ([V_NEWAGCP.CUMREMUAF]>0)
            et ([V_NEWAGCP.CUMSMICPRO3112]>0)
            )
         alors (
                f_histoclotsit([BC_BRUT_URSAFF.MONTANT])+ [BC_BRUT_URSAFF.MONTANT]
                - f_histoclotsit([BC_MALCOMP25.MONTANT])
                )
         sinon (
```

```
                f_histoclotsit([BC_MALCOMP25.MONTANT])*(-1)
                )
       )
sinon ( [BC_BRUT_URSAFF.MONTANT])
```