prologue.R

Fab

Mon Nov 24 23:02:44 2014

```
# prologue
# doit être dans le même répertoire que le programme principal et sa bibliothèque

# dans le cas où l'on ne lance le programme que pour certaines années, il préciser début.période sous r
# Dans ce cas fixer extraire.années en valeur TRUE.
# Sinon le programme travaille sur l'ensemble des années disponibles dans la base : elles sont détectée

# PARAMETRES GLOBAUX BOOLEENS ET ENTIERS


racine                        <- "R-Altair/"
# "Ville Annecy Paye BP-"
# "UTF-8.PDC-"
# "RAG_2009_2012-"
# "UTF-8.RAG_2009_2012-"
# "UTF-8.SIERG-"
# "UTF-8.RAG_2009_2012-"
# "RAG_2009_2012-"

extraire.années      <- F
    début.période.sous.revue    <- 2011
    fin.période.sous.revue      <- 2013

setOSWindows            <- Sys.info()["sysname"] != "Linux"
exec.root               <- ifelse(setOSWindows, ".exe", "")

éliminer.duplications   <- T
enlever.quotités.nulles <- T
enlever.quotités.na     <- T
générer.codes           <- FALSE
paralléliser            <- TRUE
extraire.population     <- FALSE
fusionner.nom.prénom    <- FALSE
charger.bases           <- T
sauvegarder.bases.analyse    <- T
sauvegarder.bases.origine    <- F
générer.table.effectifs      <- F
générer.table.élus           <- F
tester.matricules            <- TRUE
tester.lignes.bulletins.mois <- TRUE
corriger.quotité         <- FALSE
comportement.strict      <- TRUE
etp.égale.effectif       <- FALSE

seuil.troncature         <- 0 # jours
taux.tolérance.homonymie <- 2  # en %
quantile.cut             <- 1  # en %
```

```
minimum.positif          <- 0.5
minimum.quotité          <- 0.1

séparateur.liste.entrée     <- ";"
séparateur.décimal.entrée   <- ","
séparateur.liste.sortie     <- ";"
séparateur.décimal.sortie   <- ","

# FICHIERS EN INPUT
# conventions de nommage : les noms et chemins sont en minuscules ;
# les bases commencent par une majuscule. Un nom de fichier est souvent associé
# à une variable commençant par une majuscule et représentant la base (data.frame ou matrice)

nom.fichier.codes.paiement  <- paste0(racine, "codes.csv")
fichier.personnels          <- "Catégories des personnels"
nom.fichier.personnels      <- paste0(fichier.personnels, ".csv")
nom.fichier.paie            <- paste0(racine, "Lignes de paye")
nom.bulletin.paie           <- paste0(racine, "Bulletins de paye")
nom.table                   <- paste0(racine, "Table_1.csv")
nom.bulletins                   <- paste0(racine, "Bulletins_1.csv")

# DOSSIERS
# Attention, si l'on lance la génération de document pour la première fois sans répertoire Bases exista
# pour les applications à ergonomie facilitée, prévoir de distribuer le logiciel avec un dossier Bases

chemin.dossier              <- getwd()
chemin.dossier.bases        <- file.path(chemin.dossier, "Bases")
chemin.dossier.outils       <- file.path(chemin.dossier, "..", "..", "Outils")
chemin.dossier.données      <- file.path(chemin.dossier, "Donnees")

# Outils

if (setOSWindows) {
  iconv <- file.path(chemin.dossier.outils, paste0("iconv", exec.root))
  sed   <- file.path(chemin.dossier.outils, paste0("sed", exec.root))
} else {
  iconv <- "iconv"
  sed   <- "sed"
}


# ETIQUETTES ET FORMATS

étiquette.matricule         <- "Matricule"
étiquette.Type.rémunération <- "Type rémunération"
étiquette.année             <- "Année"
étiquette.libellé           <- "Libellé"
étiquette.montant           <- "Montant"
étiquette.code              <- "Code"
étiquette.rém.indemn        <- "Rémunération contractuelle ou indemnitaire"
champ.détection.1           <- étiquette.matricule
champ.détection.2           <- "Code"
```

```
ifelse(fusionner.nom.prénom,
       clé.fusion <<- c("Nom", "Prénom"),
       clé.fusion <<- étiquette.matricule)
```

```
## [1] "Matricule"
```

```
colonnes.requises           <- c(union(clé.fusion, étiquette.matricule),
                                 étiquette.année,
                                 "Mois",
                                 "Statut",
                                 "Brut",
                                 "Net.à.Payer",
                                 "Heures.Sup.",
                                 "Heures",
                                 "Emploi",
                                 "Nir",
                                 "Temps.de.travail")

colonnes.input <- c("Année", "Mois", "Nom", "Prénom", "Matricule",
                    "Service", "Statut", "Temps.de.travail", "Heures.Sup.", "Heures",
                    "Indice", "Brut", "Net", "Net.à.Payer", "NBI",
                    "Libellé", "Code", "Base", "Taux", "Nb.Unité",
                    "Montant", "Type", "Emploi", "Grade", "Nir")

colonnes.classes.input    <- c("integer", "integer", "character", "character", "character",
                               "character", "character", "numeric", "numeric", "numeric",
                               "character",  "numeric", "numeric", "numeric", "numeric",
                               "character",  "character", "numeric", "numeric", "numeric",
                               "numeric", "character", "character", "character", "character")

colonnes.bulletins.input <- c("Année", "Mois", "Nom", "Prénom", "Matricule",
                              "Service", "Statut", "Temps.de.travail", "Heures.Sup.", "Heures",
                              "Indice", "Brut", "Net", "Net.à.Payer", "NBI",
                              "Emploi", "Grade", "Nir")

colonnes.bulletins.classes.input <- c("integer", "integer", "character", "character", "character",
                                      "character", "character", "numeric", "numeric", "numeric",
                                      "character",  "numeric", "numeric", "numeric", "numeric",
                                      "character", "character", "character")

lignes.paie.input               <- c("Année", "Mois", "Matricule", "Année", "Mois", "Matricule", "Libell

lignes.paie.input.fallback      <- c("Année", "Mois", "Matricule", "Libellé", "Code", "Base", "Taux", "

lignes.paie.classes.input       <- c("integer", "integer", "character", "integer", "integer", "characte

lignes.paie.classes.input.fallback <- c("integer", "integer", "character", "character", "character", "nu

date.format                      <- "%d/%m/%Y"

# ESPACES DE VALEURS LICITES POUR CERTAINS CHAMPS (modalités)
```

```
#libellés.élus               <- c("Elu", "Elus", "élu", "élus", "maire", "président", "adjoint au maire

########## Problématique ##############
codes.NBI <- c("1012", "101B", "101M", "4652", "4672")


# A priori les deux modes de lectures de tables (rapide et standard) lisent aussi bien le Windows ANSI/
# l'UTF-8 à condition que le Windows ANSI soit encodé par Excel ou l'éditeur de RStudio.

encodage.entrée <- "ISO-8859-1"
  # "WINDOWS-1252"
  # "UTF-8"
  # "ISO-8859-1"
encodage.entrée.xhl2csv <- "UTF-8"

encodage.sortie <- ifelse(setOSWindows, "ISO-8859-15", encodage.entrée)

modalité.traitement           <- "TRAITEMENT"     # s'applique aussi aux NBI
modalité.indemnitaire         <- "INDEMNITAIRE"   # hors vacations


modalité.principal.contractuel <- "PRINCIPAL.CONTRACTUEL" # contractuels qui ne sont pas payés par réfé
modalité.élu                   <- "ELU"
modalité.vacations             <- "VACATIONS"
modalité.autres                <- "AUTRES"         # notamment les remboursements de frais professionne


# expressions régulières


expression.rég.heures.sup <- ".*(I.?H.?T|H.?[SC]|\\bI[[:alpha:]]*.?.*\\bH[[:alpha:]]*.?\\b.*T[[:alpha:]]
expression.rég.iat        <- ".*(\\bi.?a.?t\\b|\\bi[[:alpha:]]*.?\\b.*a[d][[:alpha:]]*.?\\b.*tec[[:alpha
expression.rég.ifts       <- ".*(\\bi.?f.?t.?s\\b|\\bi[[:alpha:]]*.?\\b\\s*f[[:alpha:]]*.?\\b\\s*trav[[
expression.rég.population <- ".*\\bASS(\\b|A).*"
expression.rég.élus       <- "maire|pr[eé]sident|.*([eé]lu[s]?|adj.*maire|v[[:alpha:]]*\\b\\s*pr[eé]sid
expression.rég.nbi        <- ".*\\bN[[:alpha:]]*.?\\s*B[[:alpha:]]*.?\\s*I.*"
```