

#Réparti les colis dans les UM sortie

#attend une liste de colis en paramètre

Gestion erreurs [Oui]

#récupère la liste des colis à répartir

Définir variable [\$idColis_liste; Valeur :Obtenir(ParamètreScript)]

#récupère si les colis peuvent être cassés (1ère valeur)

Définir variable [\$casse; Valeur :Si(
 ObtenirValeur (
 \$idColis_liste;
 1
)
 = 1;
 True;
 False
)]

Définir variable [\$divisionColis; Valeur :False]

#cherche les colis demandés

Activer modèle ["cs_Colis" (cs_Colis)]

Mode Recherche []

#la variable \$compte commence à 1, de façon à récupérer en premier la valeur n°2 de la liste. Car la valeur n°1 indique si les colis doivent être cassés

Définir variable [\$compte; Valeur :1]

Boucle

Définir variable [\$compte; Valeur :\$compte + 1]

Définir variable [\$idColis; Valeur :ObtenirValeur (
 \$idColis_liste;
 \$compte
)]

Fin de boucle si [EstVide (\$idColis)]

Définir rubrique [cs_Colis::idColis; ObtenirValeur (
 \$idColis_liste;
 \$compte
)]

Nouvel enreg./requête

Fin de boucle

Exécuter la recherche []

#si aucun colis n'est trouvé le script s'arrête

Si [Obtenir (NombreEnregTrouvés) = 0]

Activer modèle [modèle d'origine]

Fin de script []

Fin de si

#boucle sur les colis trouvés.

#! utilisation d'une variable plutôt que "afficher enreg. suivant", car la création de colis entraînerait une discontinuité dans la boucle.

Boucle

Définir variable [\$compte; Valeur :0]

Trier enregistrements [Conserver l'ordre de tri; Ordre de tri défini : cs_Colis::datePeremption; croissant
cs_Colis::fk_UniteManutentionEntree; croissant
cs_Colis::fk_Article; croissant
cs_Colis::quantiteProduit; décroissant]
 [Rétablir; Sans fenêtre]

Définir variable [\$idArticle_ancien; Valeur :0]

Définir variable [\$idBLentree_ancien; Valeur :0]

Définir variable [\$idBCsortie_ancien; Valeur :0]

Boucle

#colis suivant

Activer modèle ["cs_Colis" (cs_Colis)]

Définir variable [\$compte; Valeur :\$compte + 1]

Afficher enreg/requête/page [\$compte]
 [Sans fenêtre]

Fin de boucle si [\$compte > Obtenir (NombreEnregTrouvés)]

#n'essaie jamais de répartir un colis dont l'UM sortie est confirmée ou dont l'UMe est en stock

Si [EstVide(cs_Colis::emplacementConfirme) And cs_UniteManutentionEntree_pour_Colis::stock = 0]

#mémoire les données de l'article

Définir variable [\$idColis; Valeur :cs_Colis::idColis]

Définir variable [\$idArticle; Valeur :cs_Colis::fk_Article]

Définir variable [\$idBLentree; Valeur :cs_BonLivraisonEntree_pour_Colis::idBonLivraisonEntree]

Définir variable [\$idBCsortie; Valeur :cs_BonCommandeSortie_pour_Colis::idBonCommandeSortie]

Définir variable [\$quantite; Valeur :cs_Colis::quantiteProduit]

Définir variable [\$litige; Valeur :cs_Colis::fk_litige]

Définir variable [\$litigeDecision; Valeur :cs_Colis::fk_LitigeDecision]

Définir variable [\$delaiPeremption; Valeur :Si(
 EstVide(cs_Colis::datePeremption);
 1000000;
 (cs_Colis::datePeremption - Obtenir(DateActuelle))
)]

#cherche la ligne de BC sortie auquel le colis est affecté

Si [EstVide (\$idBCsortie)]

Définir variable [\$idLBCsortie; Valeur :0]

Sinon si [\$idBCsortie_ancien ≠ \$idBCsortie Or \$idArticle_ancien ≠ \$idArticle]

Activer modèle ["cs_Colis__LigneBonCommandeSortie" (cs_LigneBonCommandeSortie_pour_Colis)]

Mode Recherche []

Définir rubrique [cs_LigneBonCommandeSortie_pour_Colis::fk_BonCommandeSortie; \$idBCsortie]

Définir rubrique [cs_LigneBonCommandeSortie_pour_Colis::fk_Article; \$idArticle]

Exécuter la recherche []

Si [Obtenir (NombreEnregTrouvés) > 0]

Définir variable [\$idLBCsortie; Valeur :cs_LigneBonCommandeSortie_pour_Colis::idLigneBonCommandeSortie]

Définir variable [\$LBCsortie_quantiteRestante; Valeur :cs_LigneBonCommandeSortie_pour_Colis::
 quantiteProduitALivrer - cs_LigneBonCommandeSortie_pour_Colis::quantiteProduitLivre]

Sinon

Définir variable [\$idLBCsortie; Valeur :0]

Fin de si

Fin de si

#si le n° lot du colis est retiré, le met automatiquement en rappel

Activer modèle ["cs_Colis" (cs_Colis)]

Si [Not

EstVide(cs_LotRetire_pour_Colis::idLotRetire)

And

(

cs_Colis::fk_litige ≠ 5

Or

cs_Colis::fk_LitigeDecision ≠ 1

)]

Définir rubrique [Colis pour UniteManutentionEntree::fk_litige; 5]

Définir rubrique [Colis pour UniteManutentionEntree::fk_LitigeDecision; 1]

Valider enreg./requêtes

Fin de si

#COLIS LITIGIEUX

#demande à répartir le colis dans une UM sortie litige s'il est litigieux et n'est pas dans une UM sortie litige

Si [\$litige ≠ 0

And

(

(

\$litigeDecision = 1

And

(

EstVide (cs_BonCommandeSortie_pour_Colis::idBonCommandeSortie)

Or

cs_BonCommandeSortie_pour_Colis::fk_TypeBonCommandeSortie ≠ 1

)

)

Or

(

\$litigeDecision = 2

And

EstVide (cs_ZoneDepot_pour_Colis::idZoneDepot)

)

)]

#met l'ancienne ligne de BC sortie à jour

Si [\$idLBCsortie ≠ 0 And Not EstVide (\$idLBCsortie)]

Activer modèle ["cs_Colis__LigneBonCommandeSortie" (cs_LigneBonCommandeSortie_pour_Colis)]

Mode Recherche []

Définir rubrique [cs_LigneBonCommandeSortie pour Colis::idLigneBonCommandeSortie; \$idLBCsortie]

Exécuter la recherche []

Si [Obtenir (NombreEnregTrouvés) > 0]

Définir rubrique [cs_LigneBonCommandeSortie pour Colis::quantiteProduitPotentielle;

cs_LigneBonCommandeSortie_pour_Colis::quantiteProduitPotentielle - \$quantite]

Définir rubrique [cs_LigneBonCommandeSortie pour Colis::quantiteColisPotentielle;

cs_LigneBonCommandeSortie_pour_Colis::quantiteColisPotentielle - 1]

Valider enreg./requêtes

Fin de si

Fin de si

#retire de toute UM sortie

Exécuter script ["cs_Colis_repartir__lierUMsortie"; Paramètre : 0]

#si la décision pour le litige est de le renvoyer en retour

Si [\$litigeDecision = 1]

#mémoire le destinataire

Définir variable [\$idDestinataire; Valeur :cs_BonLivraisonEntree_pour_Colis::fk_Destinataire__litige]

#il n'y a rien à faire si le destinataire est vide

Si [Not EstVide (\$idDestinataire) And \$idDestinataire ≠ 0]

#vérifie si le destinataire a déjà un bon de commande sortie litige

Activer modèle ["cs_Colis__BonCommandeSortie" (cs_BonCommandeSortie_pour_Colis)]

Mode Recherche []

Définir rubrique [cs_BonCommandeSortie_pour_Colis::fk_Destinataire; \$idDestinataire]

Définir rubrique [cs_BonCommandeSortie_pour_Colis::fk_TypeBonCommandeSortie; 1]

Exécuter la recherche []

#crée le BC sortie si nécessaire

Si [Obtenir (NombreEnregTrouvés) = 0]

Nouvel enreg./requête

Définir rubrique [cs_BonCommandeSortie_pour_Colis::fk_Destinataire; \$idDestinataire]

Définir rubrique [cs_BonCommandeSortie_pour_Colis::fk_TypeBonCommandeSortie; 1]

Valider enreg./requêtes

Fin de si

#met le colis dans l'UM sortie

Définir variable [\$idBonCommandeSortie; Valeur :cs_BonCommandeSortie_pour_Colis::idBonCommandeSortie]

Exécuter script ["cs_Colis_repartir__lierUMsortie"; Paramètre : \$idBonCommandeSortie]

#fin "destinataire" n'est pas vide

Fin de si

#si la décision pour le litige est de le mettre en zone d'attente

Sinon si [\$litigeDecision = 2]

#mémoire la zone dans laquelle placer le colis

Définir variable [\$idZoneDepot; Valeur :cs_BonLivraisonEntree_pour_Colis::fk_ZoneDepot__litige]

#met le colis en zone d'attente, si une zone d'attente est définie

Si [\$idZoneDepot ≠ 0 And Not EstVide (\$idZoneDepot)]

Exécuter script ["cs_Colis_repartir_ZoneDepot"; Paramètre : \$idZoneDepot]

Fin de si

#fin des décisions sur litige

Fin de si

#met à jour la variable \$idLBCsortie

Définir variable [\$idLBCsortie; Valeur :0]

fin colis litigieux

Fin de si

#COLIS NON LITIGIEUX

#demande à répartir le colis dans une UM sortie normale s'il est non litigieux et n'est pas dans une UM sortie normale

Activer modèle ["cs_Colis" (cs_Colis)]

Si [\$litige = 0

And

(

EstVide (cs_BonCommandeSortie_pour_Colis::idBonCommandeSortie)

Or

cs_BonCommandeSortie_pour_Colis::fk_TypeBonCommandeSortie ≠ 0

)]

#cherche les Article_Destinataire qui ont le délai de péremption demandé

Définir variable [\$idDestinataire_liste; Valeur : ""]

Activer modèle ["cs_Colis__Article_Destinataire" (cs_Article_Destinataire_pour_Colis)]

Mode Recherche []

Définir rubrique [cs_Article_Destinataire_pour_Colis::fk_Article; \$idArticle]

Définir rubrique [cs_Article_Destinataire_pour_Colis::delaiPeremption; "<" & \$delaiPeremption]

Exécuter la recherche []

Trier enreg. par rubrique [Croissant; cs_Article_Destinataire_pour_Colis::fk_Destinataire]

#liste les destinataires de Article_Destinataire trouvés (liste de valeurs uniques)

Définir variable [\$idDestinataire_ancien; Valeur : 0]

Afficher enreg/requête/page

[Premièr(e)]

Si [Obtenir (NombreEnregTrouvés) > 0]

Boucle

Définir variable [\$idDestinataire; Valeur : cs_Article_Destinataire_pour_Colis::fk_Destinataire]

Si [\$idDestinataire_ancien ≠ \$idDestinataire]

Définir variable [\$idDestinataire_liste; Valeur : Liste(
\$idDestinataire_liste;
\$idDestinataire
)]

Définir variable [\$idDestinataire_ancien; Valeur : \$idDestinataire]

Fin de si

Fin de boucle si [Obtenir (NuméroEnreg) = Obtenir (NombreEnregTrouvés)]

Afficher enreg/requête/page

[Suivant(e); Sortie après dernière]

Fin de boucle

Fin de si

#cherche les ligne de BC sortie qui demandent l'article

Activer modèle ["cs_Colis__LigneBonCommandeSortie" (cs_LigneBonCommandeSortie_pour_Colis)]

Mode Recherche []

Définir variable [\$compte2; Valeur : 1]

Définir variable [\$maxCompte2; Valeur : DecompteValeurs (\$idDestinataire_liste)]

Boucle

Fin de boucle si [\$compte2 > \$maxCompte2]

Si [\$compte2 > 1]

Nouvel enreg./requête

Fin de si

Définir variable [\$idDestinataire; Valeur : ObtenirValeur(
\$idDestinataire_liste;
\$compte2

)]

Définir rubrique [cs_LigneBonCommandeSortie pour Colis::fk_Article; \$idArticle]

Définir rubrique [cs_LigneBonCommandeSortie pour Colis::termine; 0]

Définir rubrique [cs_LigneBonCommandeSortie pour Colis::differenceCommandePotentiel; "<0"]

Définir rubrique [cs_Destinataire pour Colis::idDestinataire; \$idDestinataire]

Définir variable [\$compte2; Valeur :\$compte2 + 1]

Fin de boucle

Exécuter la recherche []

#si une ligne a été trouvée on traite les colis

Si [Not EstVide (\$idDestinataire_liste) And Obtenir (NombreEnregTrouvés) > 0]

Trier enregistrements [Conserver l'ordre de tri; Ordre de tri défini : cs_LigneBonCommandeSortie pour Colis::priorite; décroissant

cs_LigneBonCommandeSortie pour Colis::differenceCommandePotentiel; croissant]

[Rétablir; Sans fenêtre]

Afficher enreg/requête/page

[Premièr(e)]

Définir variable [\$quantiteRestante; Valeur :cs_LigneBonCommandeSortie_pour_Colis::differenceCommandePotentiel * -1]

#Dans le cas où la quantité dans le colis est plus importante que la quantité restante pour la ligne de BC sortie

Si [\$quantite > \$quantiteRestante]

#divise le colis si possible

Si [\$divisionColis]

Définir variable [\$reste; Valeur :\$quantite - \$quantiteRestante]

Définir variable [\$quantite; Valeur :\$quantiteRestante]

#met à jour la quantité du colis actuel

Activer modèle ["cs_Colis" (cs_Colis)]

Définir rubrique [cs_Colis::quantiteProduit; \$quantite]

Définir variable [\$datePeremption; Valeur :cs_Colis::datePeremption]

Définir variable [\$numeroLot; Valeur :cs_Colis::numeroLot]

Définir variable [\$fk_Colis; Valeur :cs_Colis::fk_Colis]

Définir variable [\$fk_Article; Valeur :cs_Colis::fk_Article]

Définir variable [\$fk_UMentree; Valeur :cs_Colis::fk_UniteManutentionEntree]

#crée le nouveau colis avec le reste de la quantité

Nouvel enreg./requête

Définir rubrique [cs_Colis::quantiteProduit; \$reste]

Définir rubrique [cs_Colis::datePeremption; \$datePeremption]

Définir rubrique [cs_Colis::numeroLot; \$numeroLot]

Définir rubrique [cs_Colis::fk_Colis; \$fk_Colis]

Définir rubrique [cs_Colis::fk_Article; \$fk_Article]

Définir rubrique [cs_Colis::fk_UniteManutentionEntree; \$fk_UMentree]

Valider enreg./requêtes

#revient au colis en cours

Afficher enreg/requête/page [\$compte]
[Sans fenêtre]

#si le colis ne peut pas être divisé, la quantité est à 0 (indiquant qu'il n'y a aucun traitement à effectuer)

Sinon

Définir variable [\$quantite; Valeur :0]

Fin de si

Fin de si

#si aucune ligne n'est trouvée, aucune quantité n'est à gérer

Sinon

Définir variable [\$quantite; Valeur :0]

Fin de si

#s'il y a une quantité à répartir

Si [\$quantite > 0]

#récupère le bon de commande de la première ligne

Activer modèle ["cs_Colis__LigneBonCommandeSortie" (cs_LigneBonCommandeSortie_pour_Colis)]

Définir variable [\$idBonCommandeSortie; Valeur :cs_LigneBonCommandeSortie_pour_Colis::
fk_BonCommandeSortie]

#met à jour la quantité livrée sur cette ligne de BC sortie

Définir rubrique [cs_LigneBonCommandeSortie_pour_Colis::quantiteProduitPotentielle;
cs_LigneBonCommandeSortie_pour_Colis::quantiteProduitPotentielle + \$quantite]

Définir rubrique [cs_LigneBonCommandeSortie_pour_Colis::quantiteColisPotentielle;
cs_LigneBonCommandeSortie_pour_Colis::quantiteColisPotentielle + 1]

#aucun BC sortie ne demande l'article, on le sort de toute UM sortie (au cas où il serait dans une UM sortie litige)

Sinon

Définir variable [\$idBonCommandeSortie; Valeur :0]

Fin de si

#appel le script de liaison à l'UM sortie

Exécuter script ["cs_Colis_repartir__lierUMsortie"; Paramètre : \$idBonCommandeSortie]

Exécuter script ["cs_Colis_repartir_ZoneDepot"; Paramètre : 0]

#met à jour la variable \$idLBCsortie

Définir variable [\$idLBCsortie; Valeur :\$idBonCommandeSortie]

Fin de si

#les identifiants actuels deviennent les anciens

Définir variable [\$idArticle_ancien; Valeur :\$idArticle]

Définir variable [\$idBLentree_ancien; Valeur :\$idBLentree]

Définir variable [\$idBCsortie_ancien; Valeur :\$idBCsortie]

#fin si l'UM sortie du colis est déjà confirmée

Fin de si

Fin de boucle

#ne recommence pas la répartition si on les colis ne doivent pas être cassés ou s'ils ont déjà été cassés (on a déjà fait deux boucles)

Fin de boucle si [Not \$casse Or \$divisionColis]

#si les colis peuvent être cassés, recommence la boucle en indiquant que l'on essaie de diviser les colis

Si [\$casse]

Définir variable [\$divisionColis; Valeur :True]

Fin de si

#fin de la boucle avec division des colis

Fin de boucle