



Analyseur de Mail

Par
Pascal PRADIER, Responsable Programme 4D S.A.
Note technique 4D-200001-03-FR
Version 1
Date 1 Janvier 2000

Résumé

Analyser un texte en provenance d'un mail récupéré avec 4D Internet Command ou d'un autre utilitaire afin de rechercher une chaîne spécifiée et d'extraire de celle-ci un certain nombre d'informations si la chaîne est trouvée.

4D Notes techniques

Copyright © 1985-2004 4D SA - Tous droits réservés

Tous les efforts ont été faits pour que le contenu de cette note technique présente le maximum de fiabilité possible. Néanmoins, les différents éléments composant cette note technique, et le cas échéant, le code, sont fournis sans garantie d'aucune sorte. L'auteur et 4D S.A. déclinent donc toute responsabilité quant à l'utilisation qui pourrait être faite de ces éléments, tant à l'égard de leurs utilisateurs que des tiers.

Les informations contenues dans ce document peuvent faire l'objet de modifications sans préavis et ne sauraient en aucune manière engager 4D SA. La fourniture du logiciel décrit dans ce document est régie par un octroi de licence dont les termes sont précisés par ailleurs dans la licence électronique figurant sur le support du Logiciel et de la Documentation afférente. Le logiciel et sa documentation ne peuvent être utilisés, copiés ou reproduits sur quelque support que ce soit et de quelque manière que ce soit, que conformément aux termes de cette licence.

Aucune partie de ce document ne peut être reproduite ou recopiée de quelque manière que ce soit, électronique ou mécanique, y compris par photocopie, enregistrement, archivage ou tout autre procédé de stockage, de traitement et de récupération d'informations, pour d'autres buts que l'usage personnel de l'acheteur, et ce exclusivement aux conditions contractuelles, sans la permission explicite de 4D SA.

4D, 4D Calc, 4D Draw, 4D Write, 4D Insider, 4ème Dimension ®, 4D Server, 4D Compiler ainsi que les logos 4e Dimension, sont des marques enregistrées de 4D SA.

Windows, Windows NT, Win 32s et Microsoft sont des marques enregistrées de Microsoft Corporation.

Apple, Macintosh, Power Macintosh, LaserWriter, ImageWriter, QuickTime sont des marques enregistrées ou des noms commerciaux de Apple Computer, Inc.

Mac2Win Software Copyright © 1990-2002 est un produit de Altura Software, Inc.

4D Write contient des éléments de "MacLink Plus file translation", un produit de DataViz, Inc, 55 Corporate drive, Trumbull, CT, USA.

XTND Copyright 1992-2002 © 4D SA. Tous droits réservés.

XTND Technology Copyright 1989-2002 © Claris Corporation. Tous droits réservés ACROBAT © Copyright 1987-2002, Secret Commercial Adobe Systems Inc. Tous droits réservés. ACROBAT est une marque enregistrée d'Adobe Systems Inc.

Tous les autres noms de produits ou appellations sont des marques déposées ou des noms commerciaux appartenant à leurs propriétaires respectifs.

Utilisation

Lors de l'envoi de mails, il arrive parfois que l'adresse des destinataires soit invalide ou bien que le serveur distant soit en dérangement pour une raison ou une autre : cela générant un message d'erreur en retour. Dans certains cas, il peut être intéressant d'analyser automatiquement ces mails afin de réagir en fonction de l'erreur.

Par exemple, si une adresse nous est retournée comme étant incorrecte, il sera souhaitable d'extraire l'adresse e-Mail du destinataire et de la 'sortir' du carnet d'adresse afin de ne plus lui renvoyer de mails.

Le cahier des charges

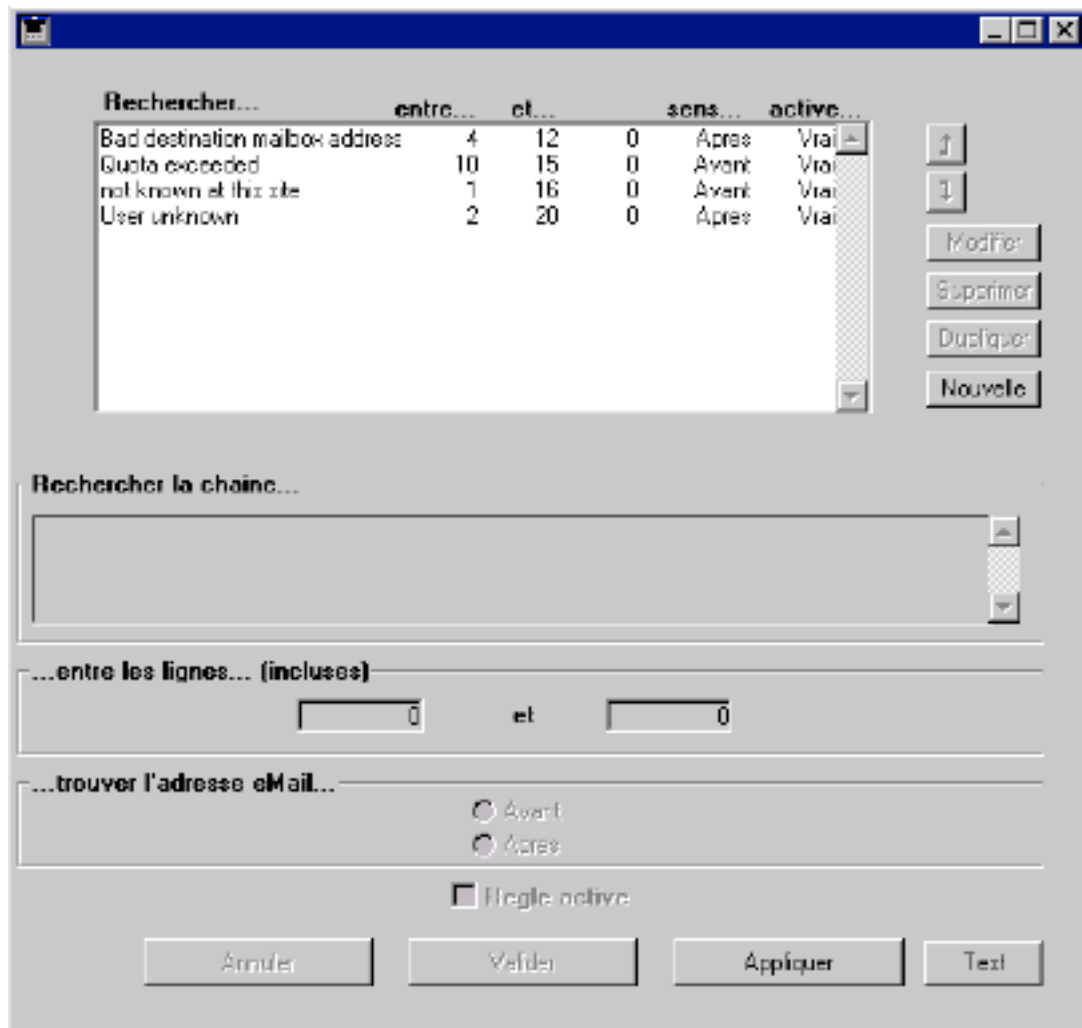
Il s'agit d'un dialogue 4D contenant un éditeur de règles définissant les chaînes de caractères à rechercher, l'intervalle dans lequel effectuer la recherche, le sens de recherche ainsi que l'activation ou non de la règle.

Les règles sont archivées dans un fichier de préférences qu'il sera possible d'éditer avec un simple éditeur de texte.

Il sera possible d'appliquer plusieurs règles pour chaque mail et l'ordre d'application sera paramétrable.

Le résultat de l'analyse pour un mail sera stocké dans un tableau exploitable par une autre partie du programme.

La copie d'écran suivante présente le dialogue de gestion des règles :



La structure des règles

Les règles sont stockées dans un fichier texte à priori à côté de la structure, mais la base gère un déplacement éventuel de ce fichier.

Le contenu type est le suivant :

```
[REGLE]
[Criteria]La chaîne à rechercher[Fin_Criteria]
[Start]Ligne de début[Fin_Start]
[Stop]Ligne de fin[Fin_Stop]
[Offset]0[Fin_Offset]
[Actif]Règle active ou non[Fin_Actif]
[Dir]Sens de recherche[Fin_Dir]
[Fin_REGLE]
```

Chaque règle est balisée par [REGLE]... ..[Fin_REGLE] et un caractère 1 est ajouté entre chaque règle. La chaîne de caractère à rechercher est balisée par [Criteria]... ..[Fin_Criteria].

Afin de réduire le domaine de recherche pour ne pas ralentir inutilement l'analyse, une ligne de début d'analyse et de fin d'analyse est spécifiée. C'est le rôle des balises [Start]... ..[Fin_Start] et [Stop]... ..[Fin_Stop].

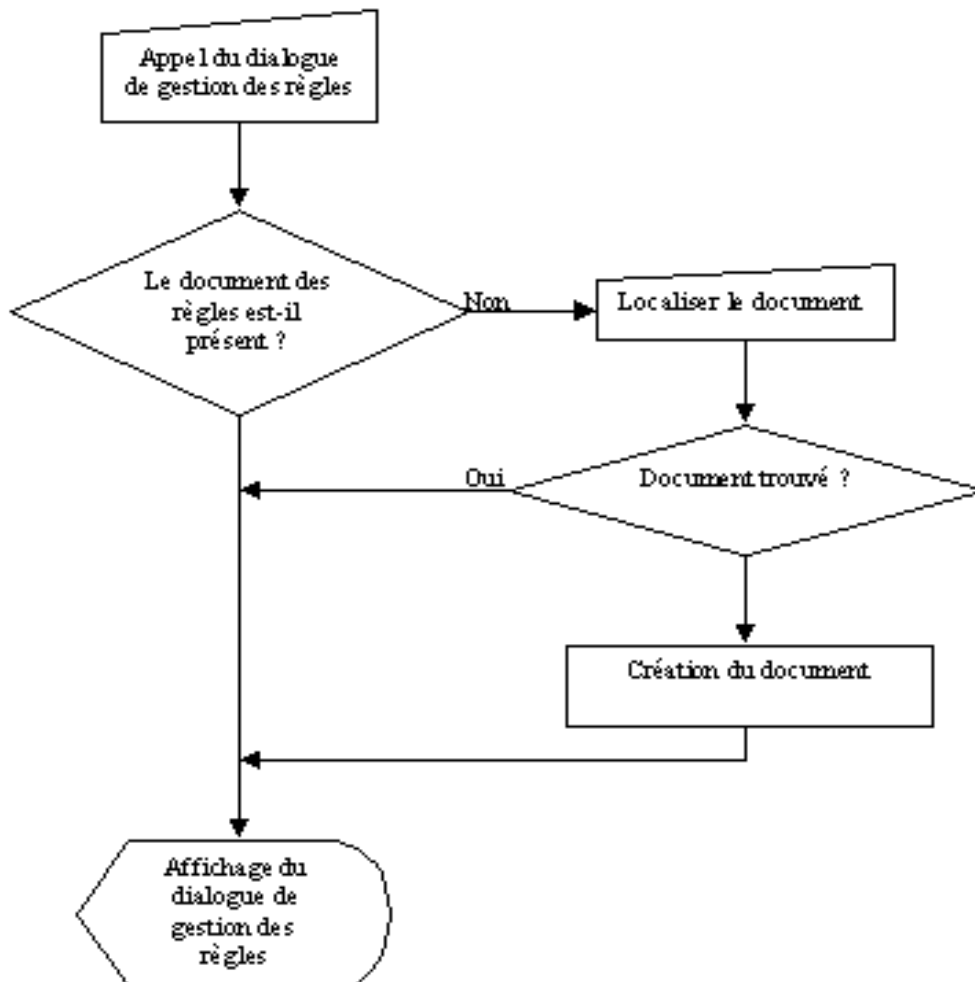
La balise [Offset]... ..[Fin_Offset] n'est pas utilisée.

Il peut être utile de désactiver une règle et c'est le rôle de la balise [Actif]... ..[Fin_Actif].

La balise [Dir]... ..[Fin_Dir] précise dans quel sens chercher l'adresse email par rapport au critère de recherche.

Le fonctionnement

Gestion du dialogue



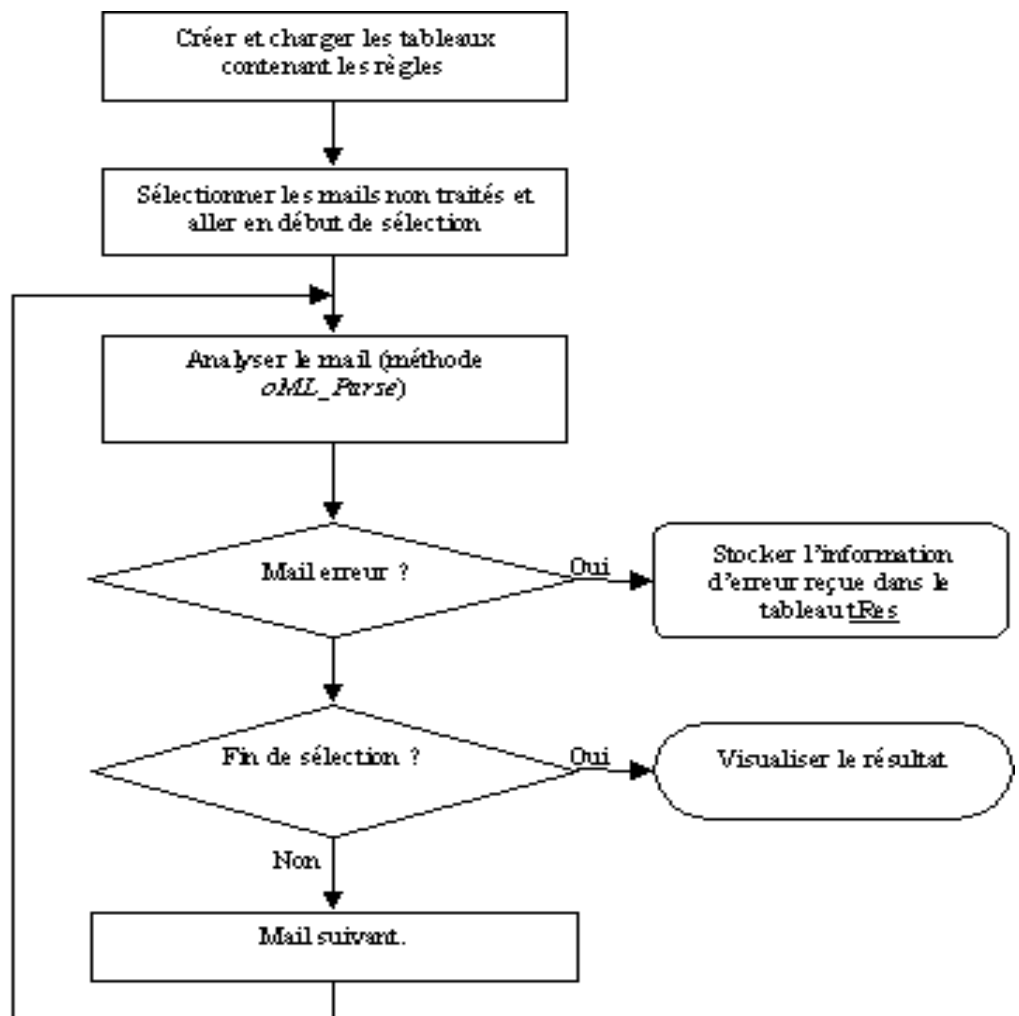
Ce diagramme présente la gestion de la localisation et de l'ouverture du document contenant les règles.

Le chemin d'accès à ce document est stocké en ressource STR# 17000. C'est la méthode oRul_File qui gèrera la présence du document (appelé depuis la méthode oRul_Show). Le détail de cette méthode est visible en annexe. La seule particularité de cette méthode est d'être réentrante. C'est-à-dire qu'elle s'appelle elle-même si le document n'est pas trouvé.

Une fois le document trouvé (ou créé), la méthode oRul_Show crée les tableaux qui recevront les règles et les affecte. Cette méthode reçoit en paramètre le chemin d'accès du document, l'ouvre, boucle sur chacune de règles présentes et affecte les tableaux. Le détail de cette méthode est visible en annexe.

À ce niveau, nous avons un dialogue présent à l'écran, présentant les règles existantes et permettant la saisie de nouvelles règles ou bien la modification d'anciennes.

Application des règles



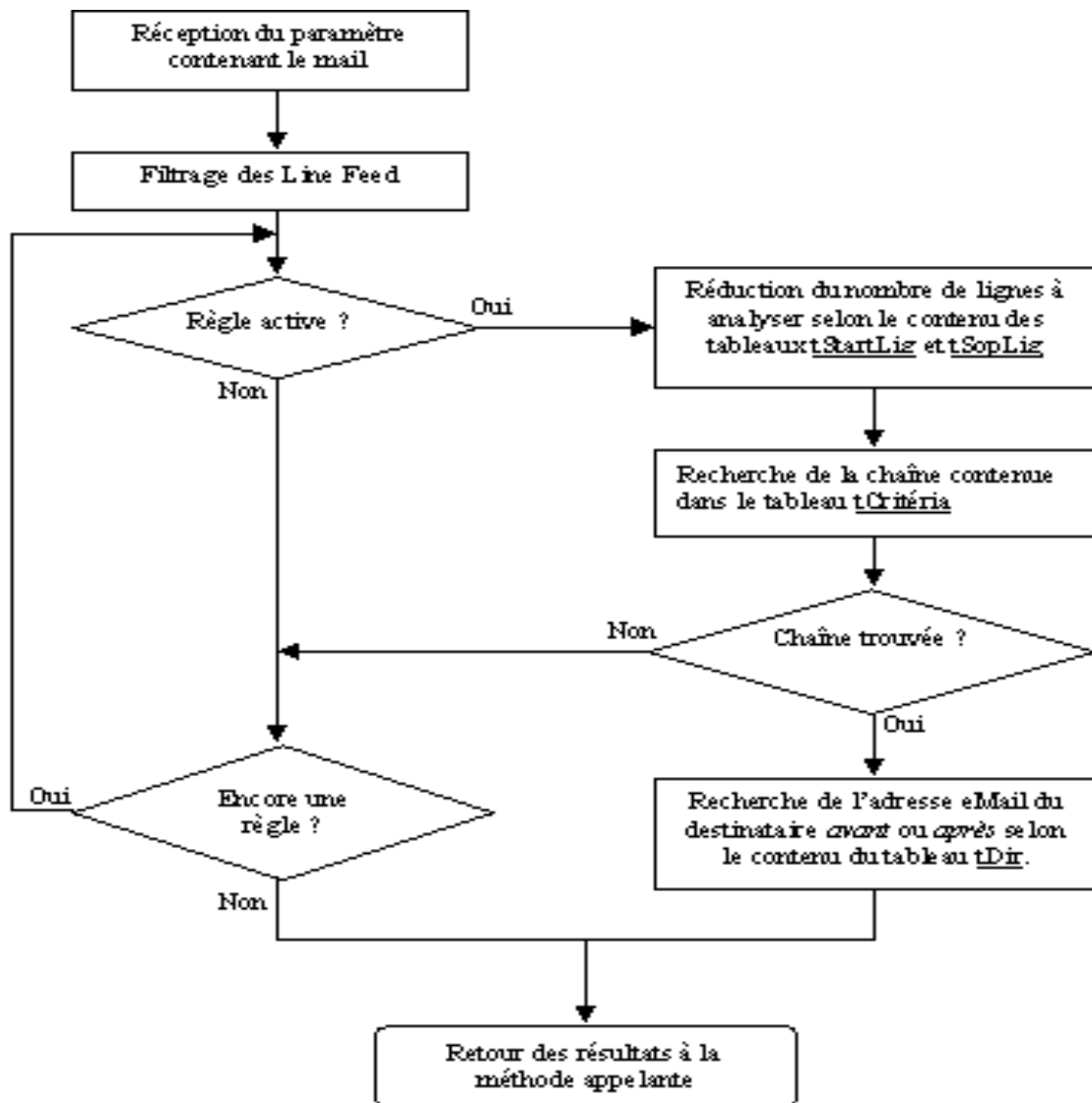
Le traitement des mails est effectué dans un process indépendant afin de ne pas bloquer les autres traitements éventuels. C'est le rôle de la méthode oML_Traite_erreurs (visible en annexe).

Le principe est simple : on boucle sur une sélection de mails stockés dans des enregistrements 4D. À l'intérieur de cette boucle, on appelle la procédure d'analyse du mail (oML_Parse) et on lui passe le texte du

mail en paramètre. En retour, cette procédure nous renvoie un code d'erreur indiquant si le mail est correct ou bien erroné. Dans le cas où il est erroné, le tableau tRes contiendra les informations demandées. Pour le moment, seule l'adresse email est retournée, mais on pourrait aussi retourner le type de l'erreur rencontrée, la date du mail etc.

Analyse d'un mail

Méthode oML_Parse



Pas de particularité concernant cette méthode d'analyse de chaîne.

Elle reçoit en paramètre des pointeurs sur les tableaux contenant les règles, le texte contenant le mail à analyser. Elle retourne un code d'erreur en fonction de l'erreur trouvée et remplit un tableau passé en paramètre avec l'adresse eMail du destinataire et toutes les infos que VOUS jugerez utiles...

Le principe de recherche de l'adresse eMail est basé sur le fait que les adresses sont balisées par '<' et '>' . Il

suffit de chercher le caractère '<' puis de concaténer caractère par caractère jusqu'au moment où on rencontre le caractère '>'.

Conclusion

Il devient possible grâce à ce système de gérer les erreurs dans les mails sans avoir à 'hardcoder' de façon exhaustive les erreurs à analyser. Si dans le futur, un nouveau type d'erreur envoyé par un serveur fait son apparition, il suffira de l'ajouter aux règles existantes et il sera automatiquement traité.

Annexe

Méthode oRul_Show

C_TEXTE(\$Rule;\$PathRules)

\$PathRules:=oRul_File

Si (\$PathRules#"")

TABLEAU TEXTE(tCriteria;0)

TABLEAU ENTIER(tStartLig;0)

TABLEAU ENTIER(tStopLig;0)

TABLEAU ENTIER(tOffset;0)

TABLEAU ALPHA(5;tAvant;0)

TABLEAU BOOLEEN(tActif;0)

`recupération de règles

\$RefDoc:=**Ouvrir document**(\$PathRules)

Si (ok=1)

Tant que (ok=1)

RECEVOIR PAQUET(\$RefDoc;\$Rule;**Caractere**(1))

Si (\$Rule#"")

\$From:=**Position**("[Criteria]";\$Rule)+10

\$Long:=**Position**("[Fin_Criteria]";\$Rule)-\$From

\$Crit:=**Sous chaine**(\$Rule;\$From;\$Long)

\$From:=**Position**("[Start]";\$Rule)+7

\$Long:=**Position**("[Fin_Start]";\$Rule)-\$From

\$Start:=**Sous chaine**(\$Rule;\$From;\$Long)

\$From:=**Position**("[STOP]";\$Rule)+5

\$Long:=**Position**("[Fin_Stop]";\$Rule)-\$From

STOP:=**Sous chaine**(\$Rule;\$From;\$Long)

\$From:=**Position**("[Offset]";\$Rule)+8

\$Long:=**Position**("[Fin_Offset]";\$Rule)-\$From

\$Offset:=**Sous chaine**(\$Rule;\$From;\$Long)

```
$From:=Position("[Actif]";$Rule)+7
$Long:=Position("[Fin_Actif]";$Rule)-$From
Si (Sous chaîne($Rule;$From;$Long)="Yes")
  $Actif:=Vrai
Sinon
  $Actif:=Faux
Fin de si
```

```
$From:=Position("[Dir]";$Rule)+5
$Long:=Position("[Fin_Dir]";$Rule)-$From
$Dir:=Sous chaîne($Rule;$From;$Long)
```

```
INSERER LIGNES(tCriteria;Taille tableau(tCriteria)+1)
tCriteria{Taille tableau(tCriteria)}:=$Crit
INSERER LIGNES(tStartLig;Taille tableau(tStartLig)+1)
tStartLig{Taille tableau(tStartLig)}:=Num($Start)
INSERER LIGNES(tStopLig;Taille tableau(tStopLig)+1)
tStopLig{Taille tableau(tStopLig)}:=Num($STOP)
INSERER LIGNES(tOffset;Taille tableau(tOffset)+1)
tOffset{Taille tableau(tOffset)}:=Num($Offset)
INSERER LIGNES(tAvant;Taille tableau(tAvant)+1)
tAvant{Taille tableau(tAvant)}:=$Dir
INSERER LIGNES(tActif;Taille tableau(tActif)+1)
tActif{Taille tableau(tActif)}:=$Actif
```

```
Fin de si
Fin tant que
FERMER DOCUMENT($RefDoc)
Fin de si
```

```
vCriteria:=""
vStartLig:=0
vStopLig:=0
vOffset:=0
```

```
Fin de si
```

Méthode oRul_File

```
C_ALPHA(255;$PathRules;$0)
C_BOOLEAN($OkOuvrirDoc)
C_HEURE($RefDoc)
```

```
$PathRules:=Lire ressource chaîne(17000)
```

```
Si (ok=0) `la ressource n'a pas été trouvée
```

```
$RefDoc:=Ouvrir document("";"PRF")
```

```
Si (ok=1)
```

```
FERMER DOCUMENT($RefDoc)
```

```
ECRIRE RESSOURCE CHAINE(17000;Document)
```

```
$PathRules:=Document
```

```
$OkOuvrirDoc:=Vrai
```


Sinon `le document n'a pas été ouvert
CONFIRMER("Le document contenant les règles n'a pas été trouvé. Souhaitez vous le créer ?")
Si (ok=1)
 \$RefDoc:=**Creer document**("";"prf")
 Si (ok=1)
 FERMER DOCUMENT(\$RefDoc)
 ECRIRE RESSOURCE CHAINE(17000;Document)
 \$PathRules:=Document
 \$OkOuvrirDoc:=**Vrai**
 Sinon
 ALERTE("Il sera impossible de traiter les mails en erreur...")
 \$OkOuvrirDoc:=**Faux**
 Fin de si
Sinon
 \$OkOuvrirDoc:=**Faux**
Fin de si
Fin de si

Sinon

Si (**Tester chemin acces**(\$PathRules)<0) `le document n'est pas là où il devrait être
 \$OkOuvrirDoc:=**Faux**
 SUPPRIMER RESSOURCE("STR ";17000)
 oRul_File
Sinon
 \$OkOuvrirDoc:=**Vrai**
Fin de si
Fin de si

\$0:=\$PathRules

Méthode oML_Traite_erreurs

Si (**Nombre de parametres**=0)

 ◇ProcTraiteErr:=**Nouveau process**("oML_Traite_erreurs";32000;"Gestion des erreurs";"Bidon";*)

Sinon

oRul_Show
TABLEAU TEXTE(tRes;3)
CHERCHER([Temp_Mess];[Temp_Mess]Traité=**Faux**)
DEBUT SELECTION([Temp_Mess])
 \$Nb:=**Enregistrements trouvés**([Temp_Mess])
 Boucle (\$i;1;\$Nb)
 vToParse:=**BLOB vers texte**([Temp_Mess]Msg;Texte sans longueur)
 \$error:=oML_Parse (vToParse;->tCriteria;->tStartLig;->tStopLig;->tOffset;->tAvant;->tActif;->tRes)
 [Temp_Mess]Traité:=**Vrai**
 [Temp_Mess]Mailto:=tRes{1}
 STOCKER ENREGISTREMENT([Temp_Mess])
 ENREGISTREMENT SUIVANT([Temp_Mess])
 Fin de boucle
 ALERTE("Traitement des mails effectués !")
 TOUT SELECTIONNER([Temp_Mess])
 VISUALISER SELECTION([Temp_Mess])
Fin de si

Méthode oML_Parse

```
C_TEXTE($1) ` la chaine à analyser
C_TEXTE($ToParse;$TextRduit;$ToSearch;$Return)
C_ALPHA(80;$Adr)
C_POINTEUR($2;$3;$4;$5;$6;$7) `les tableaux contenant les règles
C_POINTEUR($8) `le tableau de retour
C_ENTIER_LONG($0;$Nb;$i;$j;$NbLig;$Pos;$Sens;$k)
C_BOOLEEN($Flag;$Flag2)

` filtrer les LF
$ToParse:=Remplacer chaine($1;Caractere(13)+Caractere(10);Caractere(13))
$ToParse:=Remplacer chaine($ToParse;Caractere(10)+Caractere(13);Caractere(13))

` on récupere la date du mail
$Pos:=Position("Envoyé";$1)
Si ($Pos#0)
  $TextRduit:=Supprimer chaine($1;1;$Pos+7)
  $8->{3}:=-Sous chaine($TextRduit;1;Position(":";$TextRduit)-4)
Fin de si

$Nb:=Taille tableau($2->)
Boucle ($i;1;$Nb) ` pour chaque règle
  Si ($7->{$i}) ` si la règle est active...
    $Dir:=$6->{$i}
    $ToSearch:=$ToParse
    $NbLig:=0
    $TextRduit:=""
    `... on réduit le nombre de lignes à analyser selon les critères transmis.
  Repeter
    $Pos:=Position(Caractere(13);$ToSearch)
    $NbLig:=$NbLig+1
    Si ($NbLig>=$3->{$i}) & ($NbLig<=$4->{$i})
      $TextRduit:=$TextRduit+Sous chaine($ToSearch;1;Position(Caractere(13);$ToSearch))
    Fin de si
    $ToSearch:=-Supprimer chaine($ToSearch;1;$Pos)
  Jusque ($NbLig>$4->{$i})

` recherche de la chaîne
$Pos:=Position($2->{$i};$TextRduit)
Si ($Pos#0) ` si trouvé

  $Flag:=Vrai
  $j:=0
  Tant que ($Flag)
    $j:=$j+1
    ` gestion du sens de recherche de l'adresse email (avant ou après)
    Si ($6->{$i}="Avant")
      $Sens:=$Pos-$j
    Sinon
      $Sens:=$Pos+$j
    Fin de si

  Si ($TextRduit≤$Sens≥"<")
    $Flag:=Faux
    $Adr:=""
```

```

$Car:=""
$k:=0
$Flag2:=Vrai

Tant que ($Flag2) `à partir d'ici, on recupère l'adresse caractère par caractère
  $k:=$k+1
  $Car:=$TextReduit≤$Sens+$k≥
  Si ($Car=">")
    $Flag2:=Faux `l'adresse est construite donc on sort
  Sinon
    $Adr:=$Adr+$Car `on construit l'adresse
  Fin de si
Fin tant que

$Return:=$Adr `on renvoie l'adresse
$0:=0
$i:=$nb
Sinon
  $Return:="l'adresse n'est pas trouvée !" `le car "<" n'est pas trouvé donc pas d'adresse
  $0:=-1
  Fin de si
Fin tant que

Sinon
  $Return:="Le critère n'est pas trouvé !" `la chaîne n'est pas trouvée
  $0:=-2
  Fin de si

Fin de si `fin de règle active
Fin de boucle

$8->{1}:=$Return

```