<%SUMMARY%>

This Word layout template for Help & Manual defines the entire layout - front page, table of contents, chapter headings, keyword index.

To **customize this template**, use the original file ("Word-Template-A4.docx") from the H&M installation folder, make a copy of it, then tailor it with MS Word to your requirements. You **can use variables** in every place of the document. To insert a variable, insert a COMMENT field in Word, then type the variable name as the field content. Like the  field above this text.

Move on to the next page in this document to customize the table of contents (a placeholder for the real TOC later) and modify text and paragraph format of the chapter headings in the middle of this template. You might also want to change the numbering style.

**Important:** before you use the template for publishing in H&M, make sure that general page format and margins are set correctly. H&M will export images and tables to make them fit in the space available.

and EC Software GmbH

**[Type the company address]**

**[Type the phone number]**

**[Type the fax number]**

**2017**

**Expression Réguličre Pour Delphi**

Contents

<%CHAPTER\_HEADING1%> 3

<%CHAPTER\_HEADING2%> 3

<%CHAPTER\_HEADING3%> 3

<%CHAPTER\_HEADING4%> 3

<%CHAPTER\_HEADING5%> 3

<%CHAPTER\_HEADING6%> 3

# A propos

TRegExpr est facile а utiliser et un outil trиs puissant pour vйrifier les entrйes de chaоne de caractиres dans les champs (dans les DBMS et les applications web), recherche/substitution de texte, utilitaire comme egrep & etc...

Vous pouvez aisйment vйrifier la syntaxe d'adresse e-mail, extraire un numйro de tйlйphone ou un code ZIP d'un texte non formatй ou n'importe quel autre information d'une page web et tout ce que vous pouvez imaginer! Les modиles peuvent кtre changйes sans recompilation du programme !

Cette librairie gratuite est une version йtendue des routines de Henry Spencer V8-routins pour travailler avec un sous-ensemble de Perl ; [Expressions Réguličres](#RegExp_Syntax).

TRegExpr est йcris en objet pascal avec les fichiers source disponible gratuitement.

Le fichier source original en C a йtй amйliorй et encapsulй complиtement dans la classe [TRegExpr](#tregexpr_interface) en un seul fichier: RegExpr.pas.

Aussi, vous n'aurez pas besoin de fichier DLL!

Prenez un simple regard а une [illustration](#Demos) et йtudier la [syntaxe](#RegExp_Syntax) des expressions rйguliиres. (Vous pouvez utiliser le [projet demo](#tregexpr_testrexp) pour explorer et apprendre vos propres expressions rйguliиres).

Vous pouvez utiliser les (WideString Delphi) - voir "Comment utiliser les unicode".

Voir la web section [Quoi de Neuf](http://RegExpStudio.com/TRegExpr/Help/Whats_New.html) pour les rйcents changements.

N'importe quel rapport d'erreur (bug), commentaires et idйes sont apprйciйs.

# Question Légale

Copyright (c) 1999-2004 par Andrey V. Sorokin <[anso@mail.ru](`mailto:anso@mail.ru',`',1)>

Ce logiciel est fourni comme il est, sans aucune garantie. Utiliser le а vos propre risque.

Vous pouvez utiliser ce logiciel dans n'importe quel dйveloppement, incluant l'usage commercial, la redistribution, et le modifier comme bon vous semble, mais en respectant ces restrictions :

1. L'origine de ce logiciel ne doit pas кtre mal reprйsentй. Vous ne pouvez proclamer que vous avez йcris l'original. Si vous utiliser ce logiciel dans n'importe quel produit, il serait grandement apprйciй qu'il y ait une boоte d'information ou qu'il y soit mentionnй dans la documentation cette inscription ; Copyright (c) 2004 partiel par Andrey V. Sorokin.

2. Vous ne pouvez soutirer des revenues pour la distribution des fichiers source aux autre dйveloppeurs. Quand le logiciel est dans un produit commercial, le fichier source ne peut-кtre chargй sйparйment.

--------------------------------------------------------------------------

Question Lйgale pour les fichiers source en C

--------------------------------------------------------------------------

Copyright (c) 1986 par l'Universitй de Toronto.

Йcris par Henry Spencer. N'est pas dйrivй de logiciels licenciйs.

La permission est accordйe а n'importe qui d'utiliser ce logiciel pour n'importe quel utilisation sur n'importe quel ordinateur, et de le redistribuer librement, tout en suivant ces restrictions :

1. L'auteur n'est pas responsable des consйquences de l'utilisation de ce logiciel, sans йgard а la sйvйritй, mкme s'il s'agit de dйfectuositй interne du logiciel.

2. L'origine de ce logiciel ne doit pas кtre mal reprйsentй, soit par omission ou par inscription explicite.

3. Les versions altйrйs doivent кtre clairement identifiйes, et ne doivent pas comporter d'omission ou d'inscription explicite quand а la provenance de ce logiciel.

# Installation et liste de fichier

Pour installer la copie de la librairie installer RegExpr.pas dans le rйpertoire de votre projet ou tout simplement l'ajouter au rйpertoire de votre projet dans votre manageur de projet. C'est aussi simple que cela.

Maintenant utiliser les objets ou les routines de TRegExpr dans votre projet (voir les simples [illustrations](#Demos)). Ne pas oublier d'ajouter l'usage de l'unitй de 'RegExpr' dans votre projet.

Fichiers d'aide (distribuйs sйparйment dans quelques versions de TRegExpr)

· RegExpE.hlp, RegExpE.cnt - Aide Anglaise

· RegExpRu.hlp, RegExpRu.cnt - Aide Russe

· RegExpBg.hlp, RegExpBg.cnt - Aide Bulgare (par Simeon Lilov)

· RegExpG.hlp, RegExpG.cnt - Aide Allemande (par Martin Baur)

· RegExpF.hlp, RegExpF.cnt - Aide en Franзais (par Martin Ledoux)

· emplacement pour l'aide dans votre langue ;) SVP, m'aider pour la traduction !

[Projet Demo](#tregexpr_testrexp) consiste en ces fichiers:

· TestRExp.dpr (fichier de projet).

· TestRE.pas (forme principale).

· TestRE.dfm

· PCode.pas (visionneuse de pseudocode (compilation de l'e.r.)).

· PCode.dfm

· FileViewer.pas (visionneuse pour la recherche dans un fichier comme egrep).

· FileViewer.dfm

· [HyperLinkDecorator.pas](#HyperLinksDecorator) (exemples plus complexe).

· test.htm (fichier demo pour la visionneuse de FileViewer).

# Comment ça Fonctionne

**Simple illustrations**

Si vous n'кtes pas familier avec les expressions rйguliиres, svp, aller au sujet [syntaxe](#RegExp_Syntax).

**Utiliser les routines globales**

C'est simple mais pas une faзon flexible et pratique.

**ExecRegExpr ('\d{3}-(\d{2}-\d{2}|\d{4})', 'Tйlйphone: 555-1234')**;

retourne Vrai

**ExecRegExpr ('^\d{3}-(\d{2}-\d{2}|\d{4})', 'Tйlйphone: 555-1234')**;

retourne Faux, parce qu'il y a des symboles avant le numйro de tйlйphone on utilise le mйtasymbole '^' (qui signifie dйbut de ligne).

**ReplaceRegExpr ('au produit', 'Tendez un regard au produit. TRegExpr est le meilleur!', 'а TRegExpr');**

retourne 'Tendez un regard а TRegExpr. TRegExpr est le meilleur !'.

**Utiliser la classe TRegExpr**

Vous avez tout le pouvoir de la librairie.

// Cette simple fonction extrait tous les e-mail de la chaоne d'entrйe.

// et place la liste de tous les e-mail dans la chaоne sortante.

**function ExtractEmails (const AInputString : string) : string;**

**const**

**EmailRE = '[\_a-zA-Z\d\-\.]+@[\_a-zA-Z\d\-]+(\.[\_a-zA-Z\d\-]+)+'**

**var**

**r : TRegExpr;**

**begin**

**Result := '';**

**r := TRegExpr.Create;** // Crйe L'objet

**try** // s'assure de la relвche de mйmoire en cas d'erreurs d'exceptions.

**r.Expression := EmailRE;**

// Assigne le code source а l'e.r. Il sera compilй quand ce sera nйcessaire

// (par exemple quand Exec sera appelй). S'il y a des erreurs dans l'e.r.

// Des exceptions seront levйes durant la compilation de l'e.r.

**if r.Exec (AInputString) then**

**REPEAT**

**Result := Result + r.Match [0] + ', ';**

**UNTIL not r.ExecNext;**

**finally r.Free;**

**end;**

**end;**

**begin**

**ExctractEmails ('My e-mails is anso@mail.ru and anso@usa.net');**

// retourne 'anso@mail.ru, anso@usa.net, '

**end.**

// Noter: La compilation de l'e.r. durant l'attribution de l'expression

// prend quelque temps , si vous voulez utiliser cette fonction plusieurs fois

// ce sera du travail inutile...

// Pour l'optimiser de faзon significative, crйer TRegExpr

// et prйcompiler l'expression durant la phase d'initialisation du programme.

// Ce simple exemple extrait les numйros de tйlйphone et

// l'analyse en partie (code rйgional, ville, numйro interne).

// Ensuite il substitut les parties en gabarit.

**function ParsePhone (const AInputString, ATemplate : string) : string;**

**const**

**IntPhoneRE = '(\+\d \*)?(\(\d+\) \*)?\d+(-\d\*)\*';**

**var**

**r : TRegExpr;**

**begin**

**r := TRegExpr.Create;** // Crйe l'objet

**try** // s'assure de la relвche de mйmoire en cas d'erreurs d'exceptions.

**r.Expression := IntPhoneRE;**

// Assigne le code source а l'e.r. Il sera compilй quand nйcessaire

// (par exemple quand Exec sera appelй). S'il y a des erreurs dans l'e.r.

// Des exceptions seront levйes durant la compilation de l'e.r.

**if r.Exec (AInputString)**

**then Result := r.Substitute (ATemplate)**

**else Result := '';**

**finally r.Free;**

**end;**

**end;**

**begin**

**ParsePhone ('Phone of AlkorSoft (project PayCash) is +7(812) 329-44-69',**

**'Zone code $1, city code $2. Whole phone number is $&.');**

// retourne 'Code Rйgional +7, Code de Ville (812) . Tйlйphone complet +7(812) 329-44-69.'

**end.**

**Ilustrations plus complexes**

Vous pouvez trouver des illustrations plus complexes pour utiliser TRegExpr dans le projet [TestRExp.dpr](#tregexpr_testrexp) et [HyperLinkDecorator.pas](#HyperLinksDecorator).

Voir aussi mon petit article а [Delphi3000.com](`http://www.delphi3000.com/member.asp?ID=1300',`',1) (en Anglais) et [Delphi Kingdom](`http://delphi.vitpc.com/mastering/strings_birds_eye_view.htm',`',1) (en Russe).

**Explication plus dйtaillйe**

Svp, voir la [description](#tregexpr_interface) d'interface de TregExpr.

# Syntaxe des Expressions Réguličres

**Introduction**

Les Expressions Rйguliиres sont grandement utilisйes pour spйcifier des type de recherches pour le texte. Les **mйtacaractиres** spйciaux vous permettent de spйcifier, par exemple, qu'une chaоne particuliиre que vous chercher se trouve au dйbut ou а la fin d'une ligne, ou contient **n** rйcurrence d'un certain caractиre.

Les Expressions Rйguliиres ressemblent vraiment а du charabia pour les dйbutants, mais elles sont rйellement simples (bien, habituellement), et sont un outil pratique et puissant.

Je recommande fortement de vous amuser avec le demo [TestRExp.dpr](#tregexpr_testrexp) - il vous aidera а comprendre le concept principal. De plus, il y a plusieurs exemple prйdйfinis avec des commentaires inclus dans TestRExp.

Commenзons notre voyage d'apprentissage!

**Simple comparaison**

Un simple se compare а lui-mкme, sauf s'il est un **mйtacaractиre** avec une spйcification spйciale dйcris plus bas.

Une sйrie de caractиre se compare а la mкme sйrie de caractиre dans la chaоne de destination, aussi le gabarit "bluh" se compare а "bluh'' dans la chaоne de destination. Relativement simple, n'est-ce pas ?

Vous pouvez obliger les **mйtacaractиres** ou les **Sйquences d'йchappements** а кtre interprйtйs littйralement avec un 'йchappement' en les prйcйdents avec une barre oblique inverse "\", par exemple: le mйtacaractиre "^" normalement se compare au dйbut de ligne, mais "\^" se compare au caractиre "^", "\\" se compare а "\" et ainsi de suite.

**Exemples:**

foobar *compare la chaоne 'foobar'*

\^FooBarPtr *compare la chaоne '^FooBarPtr'*

**Sйquences d'Йchappements**

Les caractиres peuvent кtre spйcifiйs avec une **Sйquence d'Йchappement** comme celles utilisй dans le langage C et Perl: *"\n'' se compare а une nouvelle ligne*, "\t'' а une tabulation, etc... Plus gйnйralement, \xnn, oщ nn est un nombre hexadйcimal, se compare aux caractиres ASCII avec une valeur dans nn. Si vous avez besoin des caractиres large (wide, ou Unicode), vous pouvez utiliser '\x{nnnn}', d'oщ 'nnnn' - un nombre de plus ou moins 4 caractиres numйrique hexadйcimal.

\xnn *caractиre hexa avec le code nn.*

\x{nnnn} *caractиre hexa avec un code nnnn (un octet pour le texte ordinaire et 2 octets pour l'Unicode).*

\t *Tabulation horizontale (HT/TAB), mкme chose que \x09.*

\n *Nouvelle ligne (NL), mкme chose que \x0a.*

\r *Retour de chariot (CR), mкme chose que \x0d.*

\f *Avance page (FF), mкme chose que \x0c.*

\a *Alarme (bell) (BEL), mкme chose que \x07.*

\e *Йchappement (ESC), mкme chose que \x1b.*

**Exemples:**

foo\x20bar *Se compare а 'foo bar' (noter l'espace dans le milieu).*

\tfoobar *Se compare а 'foobar' prйdйfinis avec une tabulation.*

**Classes de Caractиres**

Vous pouvez spйcifier une **Classe de caractиres**, en insйrant une liste de caractиres dans [], lequel comparera **tous les caractиres inclus** dans la liste.

Si le premier caractиre aprиs "['' est "^'', la classe se compare comme une **nйgation** en comparant tous les caractиres qui **ne sont pas** dans la liste.

**Exemples:**

foob[aeiou]r *Trouve les chaоnes 'foobar', 'foober' etc... Mais pas 'foobbr', 'foobcr' etc...*

foob[^aeiou]r *Trouve les chaоnes 'foobbr', 'foobcr' etc. Mais pas 'foobar', 'foober' etc...*

Dans une liste, le caractиre "-'' est utilisй pour spйcifier une **distance (range)**, aussi a-z reprйsente tous les caractиres entre ''a'' et "z'', inclusivement.

Si vous voulez que "-'' soit membre de la classe, veuillez le mettre au dйbut ou а la fin de la liste, ou encore placer un йchappement ("\") devant. Si vous voulez un ']' vous pouvez le placer au dйbut de la liste ou le placer avec un йchappement "\".

**Exemples:**

[-az] *Compare 'a', 'z' et '-.'*

[az-] *Compare 'a', 'z' et '-.'*

[a\-z] *Compare 'a', 'z' and '-.'*

[a-z] *Trouve tous les 26 petits caractиres de 'a' а 'z'.*

[\n-\x0D] *Trouve tous les #10,#11,#12,#13.*

[\d-t] *Trouve n'importe quel caractиre numйrique, '-' ou 't'.*

[]-a] *Trouve n'importe quel caractиre de ']' а 'a'.*

**Mйtacaractиres**

Les Mйtacaractиres sont des caractиres spйciaux qui sont l'essence mкme des expressions rйguliиres. Il y a diffйrents types de mйtacaractиres, dйcris plus bas.

**Mйtacaractиres - Sйparateurs de ligne**

^ *Au dйbut.*

$ *А la fin.*

\A *Dйbut du texte.*

\Z *Fin du texte.*

. *N'importe quel caractиre dans une ligne.*

**Exemples:**

^foobar *Trouve la chaоne 'foobar' seulement s'il est au dйbut.*

foobar$ *Trouve la chaоne 'foobar' seulement s'il est а la fin.*

^foobar$ *Trouve la chaоne 'foobar' seulement s'il y a ce mot.*

foob.r *Trouve les chaоnes comme 'foobar', 'foobbr', 'foob1r' et ainsi de suite.*

Le mйtacaractиre "^" par dйfaut garantie de trouver le mot seulement s'il est au dйbut de la chaоne ou du texte, le mйtacaractиre "$" seulement а la fin. Les sйparateurs de ligne inclus dans le texte ne sont pas considйrйs comme valable par "^'' ou "$'', donc la condition est fausse et la recherche n'est pas valide.

Vous pouvez, toutefois, dйsirer traiter une chaоne comme une chaоne de plusieurs ligne de texte, de cette faзon "^'' sera valable aprиs le sйparateur de ligne, et "$'' sera valable avant un sйparateur de ligne. Vous pouvez faire ceci avec le modifier /m.

Les mйtacaractиres \A et \Z sont comme "^'' et "$'', exceptй qu'ils fonctionne seulement qu'une seule fois pour tout le texte quand le modifier /m est en usage, pendant que "^'' et "$'' recherchera а chaque sйparateur de ligne.

Le mйtacaractиre ".'' par dйfaut se compare а n'importe quel caractиre, mais si vous mettez а off le modifier /s, les sйparateurs de ligne ne seront plus inclus pour '.'.

TRegExpr travaille avec les sйparateurs de lignes comme recommandй au site web www.unicode.org ( http://www.unicode.org/unicode/reports/tr18/ ):

"^" est au dйbut de la chaоne d'entrйe, et si le modifier /m est а On, aussi suivant immйdiatement n'importe quelle occurrence de \x0D\x0A ou \x0A ou \x0D (si vous utiliser la Version Unicode de TRegExpr, et ensuite \x2028 ou \x2029 ou \x0B ou \x0C ou \x85). Noter qu'il n'y a pas de ligne vide dans la sйquence \x0D\x0A.

"$" est а la fin de la chaоne d'entrйe, et si le modifier /m est а On, aussi prйcйdant immйdiatement n'importe quelle occurrence de \x0D\x0A ou \x0A ou \x0D (si vous utiliser la Version Unicode de TRegExpr, et ensuite \x2028 ou \x2029 ou \x0B ou \x0C ou \x85). Noter qu'il n'y a pas de ligne vide dans la sйquence \x0D\x0A.

"." se compare а n'importe quel caractиre, mais si le modifier /s est a Off "." ne correspondra plus а \x0D\x0A et \x0A et \x0D (si vous utiliser la Version Unicode de TRegExpr, et ensuite \x2028 et \x2029 et \x0B et \x0C et \x85).

Noter que "^.\*$" (un gabarit de ligne vide) ne correspond pas а une chaоne vide, mais se compare а une chaоne vide contenant la sйquence \x0A\x0D.

Le traitement Multiligne peut facilement кtre ajustй selon vos besoins avec l'aide des propriйtйs LineSeparators et LinePairedSeparator de TregExpr. Vous pouvez utiliser le style Unix "\n" ou seulement le style DOS/Windows "\r\n" ou un mйlange des deux (comme dйcris plus haut et utilisй par dйfaut) ou dйfinir vos propres sйparateurs!

**Mйtacactиres - classes prйdйfinies**

\w *Un caractиre alphanumйrique (incluant "\_").*

\W *Un caractиre non alphanumйrique.*

\d *Un caractиre numйrique.*

\D *Un caractиre non numйrique.*

\s *N'importe quel espace (mкme chose que [ \t\n\r\f]).*

\S *Tout ce qui n'est pas un espace.*

Vous pouvez utiliser \w, \d et \s а l'intйrieur de la **classe de caractиres**.

**Exemples:**

foob\dr *Trouve les chaоne comme 'foob1r', ''foob6r', ... Mais pas 'foobar', 'foobbr', ...*

foob[\w\s]r *Trouve les chaоne comme 'foobar', 'foob r', 'foobbr', ... Mais pas 'foob1r', 'foob=r', ...*

TRegExpr utilise les propriйtйs SpaceChars et WordChars pour dйfinir les classes de caractиres \w, \W, \s, \S, aussi vous pouvez aisйment les redйfinir.

**Mйtacaractиres - itйrateurs**

N'importe quel item d'une expression rйguliиre peut-кtre suivi par un autre type de mйtacaractиre - les **itйrateurs**. En utilisant ces mйtacaractиres vous pouvez spйcifier le nombre de fois que le caractиre prйcйdent sera reprйsentй, **mйtacaractиres** ou **sous expression**.

\* *Zйro ou plus ("vorace"), similaire а {0,}.*

+ *Un ou plus ("vorace"), similaire а {1,}.*

? *Zйro or un ("vorace"), similaire а {0,1}.*

{n} *Exactement n fois ("vorace").*

{n,} *Au moins n fois ("vorace").*

{n,m} *Au moins n fois mais pas plus de m fois ("vorace").*

\*? *Zйro ou plus ("non-vorace"), similaire а {0,}?.*

+? *Un ou plus ("non-vorace"), similaire а {1,}?.*

?? *Zйro ou un ("non-vorace"), similaire а {0,1}?.*

{n}? *Exactement n fois ("non-vorace").*

{n,}? *Au moins n fois ("non-vorace").*

{n,m}? *Au moins n fois mais pas plus de m fois ("non-vorace").*

Donc, les nombres dans les accolades de la forme {n,m}, spйcifie le nombre de fois minimum avec la lettre n et le nombre maximum avec la lettre m. La forme {n} est йquivalente а {n,n} et correspond exactement а n fois. La forme {n,} correspond а n ou plus. Il n'y a aucune limite quand а la grosseur de n et m, mais les grands nombres prendront beaucoup plus de mйmoire et vont ralentir l'exйcution de l'e.r.

Si les accolades apparaissent dans un autre contexte, ils sont traitйes comme un caractиre rйgulier.

**Exemples:**

foob.\*r *Se compare а 'foobar', 'foobalkjdflkj9r' et 'foobr'.*

foob.+r *Se compare а 'foobar', 'foobalkjdflkj9r' mais pas 'foobr'.*

foob.?r *Se compare а 'foobar', 'foobbr' et 'foobr', mais pas 'foobalkj9r'.*

fooba{2}r *Se compare а 'foobaar'.*

fooba{2,}r *Se compare а 'foobaar', 'foobaaar', 'foobaaaar', etc...*

fooba{2,3}r *Se compare а 'foobaar', ou 'foobaaar', mais pas 'foobaaaar'.*

Une petite explication а propos de l'utilisation des termes "non-vorace" et "vorace". "Vorace" prend autant que possible, "non-vorace" prend aussi peu que possible. Par exemple, 'b+' et 'b\*' appliquй а la chaоne 'abbbbc' retourne 'bbbb', 'b+?' retourne 'b', 'b\*?' retourne une chaоne vide, 'b{2,3}?' retourne 'bb', 'b{2,3}' retourne 'bbb'.

Vous pouvez changer tous les itйrateurs en mode "non-vorace" en utilisant le modifier /g.

**Mйtacaractиres - Alternatifs**

Vous pouvez spйcifier des **alternatifs** pour le modиle en utilisant "|'' pour les sйparer, donc fee|fie|foe correspond а "fee'', "fie'', ou "foe'' dans la chaоne de destination (comme f(e|i|o)e le ferait). La premiиre alternative inclus tout du dйlimiteur prйcйdent ("('', "['', ou le dйbut du modиle) jusqu'au premier "|'', et la derniиre alternative contient tout du dernier "|'' jusqu'au dernier dйlimiteur. Pour cette raison, il est de pratique courante d'inclure les alternatives dans des parenthиses, pour minimiser le risque de confusion pour savoir quand c'est le dйpart et quand c'est la fin.

Les alternatifs sont йvaluйs de gauche а droite, donc la premiиre alternative trouvй pour la correspondance est celle qui est choisi. Ceci signifie que les alternatives ne sont pas nйcessairement vorace. Par exemple : quand vous faites correspondre foo|foot а "barefoot'', seulement la partie "foo'' correspond, comme c'est la premiиre alternative essayйe, elle correspond exactement а la chaоne de destination. (Ceci ne semble pas important, mais ceci le deviens quand vous capturer du texte correspondant en utilisant les parenthиses.)

Aussi rappeler vous que "|'' est interprйtй comme un littйral entre "[]", donc si vous йcrivez [fee|fie|foe] Vous rйellement rechercher pour [feio|].

**Exemples:**

foo(bar|foo) *Trouve la chaоne 'foobar' ou 'foofoo'.*

**Mйtacaractиres - sous expressions**

Les parenthиses ( ... ) peuvent aussi кtre utilisйes pour construire des sous expression rйguliиre. (aprиs l'analyse, vous pouvez trouver les positions des sous expressions, longueurs et valeurs actuelles dans MatchPos, MatchLen et les propriйtйs de Match dans TRegExpr, et les substituer dans les chaоnes du gabarit de TRegExpr.Substitute).

Les Sous expressions sont numйrotйs de gauche а droite selon les ouvertures des parenthиses. La premiиre sous expression а le numйro '1' (l'e.r. complиte a le numйro '0' - vous pouvez le substituer dans TRegExpr.Substitute comme '$0' ou '$&').

**Exemples:**

(foobar){8,10} *Se compare aux chaоnes qui contiennent 8, 9 ou 10 instances de 'foobar'.*

foob([0-9]|a+)r *Trouve la chaоne 'foob0r', 'foob1r' , 'foobar', 'foobaar', 'foobaar', etc...*

**Mйtacaractиres - Rйfйrences Prйcйdentes**

Les **Mйtacaractиres** \1 jusqu'а \9 sont interprйtйs comme des rйfйrences prйcйdentes. \<n> compare la **sous expression** #<n> prйcйdente trouvй.

**Exemples:**

(.)\1+ *Trouve 'aaaa' et 'cc'.*

(.+)\1+ *Aussi se compare а 'abab' et '123123'.*

*(['"]?)(\d+)\1 Trouve '"13" (entre guillemets), ou '4' (en apostrophe) ou 77 (sans guillemet ou apostrophe), etc...*

**Modifier**

Les Modifier existe dans le but de changer le comportement de TRegExpr.

Il y a plusieurs faзon d'ajuster ces modifier. N'importe quel de ces modifier peuvent кtre incorporй dans l'expression rйguliиre elle-mкme en utilisant la construction de (?...).

Aussi, vous pouvez changer la propriйtй adйquate de TRegExpr (ModifierX par exemple pour changer /x, ou ModifierStr pour changer tous les modificateurs ensemble). Les valeurs par dйfaut de la nouvelle instance de l'objet TRegExpr sont dйfinis dans les variables globales, par exemple la variable globale RegExprModifierX dйfinie la valeur (ModifierX) d'une nouvelle instance de TRegExpr.

**i**

Faire des recherche sans йgard а la casse des caractиres (utilisant les ajustements locaux dйfinis dans votre systиme), voir aussi CasInversé.

**m**

Traite les chaоnes comme des ligne multiples. Change la fonction de "^'' et "$'' pour chercher uniquement а partir du dйbut ou de la fin de la chaоne, ce sera maintenant а partir du dйbut d'une ligne ou а la fin de la ligne, voir aussi Séparateurs de Ligne.

**s**

Traite les chaоnes comme une simple ligne de texte. Change la fonction de "." pour qu'il se compare а n'importe quel caractиre, mкme un sйparateur de ligne (voir aussi Séparateur de Ligne), normalement il ignorerait les sauts de ligne.

**g**

Modifier non standard. En le mettant а Off vous spйcifier de mettre tous les opйrateurs en mode non-vorace (par dйfaut, ce modifier est а On). Aussi, si le modifier /g est а Off, alors '+' fonctionne comme '+?', '\*' comme '\*?' et ainsi de suite...

**x**

Йtend la lisibilitй du modиle en vous permettant des espaces et des commentaires (voir l'explication plus bas).

**r**

Modificateur non standard. Si ajustй, les distances additionnelles de а-я inclus les lettres russe 'ё', А-Я inclus additionnellement 'Ё', et а-Я inclus tous les symboles russe.

Dйsolй pour les utilisateurs de l'extйrieur, mais ces valeurs sont ajustй par dйfaut. Si vous voulez les mettre а off par dйfaut - changer la valeur de la variable globale RegExprModifierR.

Le modifier /x requiert des explications. Il dit а TRegExpr d'ignorer les espaces qui ne sont pas avec un йchappement ou qui ne sont pas dans une classe. Vous pouvez utiliser ceci pour casser l'expression rйguliиre en morceaux plus petit et plus lisible. Le caractиre # est aussi traitй comme un mйtacaractиre qui introduit les commentaires, par exemple:

*(*

*(abc) # commentaire 1*

*| # Vous ne pouvez pas utiliser les espaces pour formater l'e.r. - TRegExpr l'ignorera.*

*(efg) # commentaire 2*

*)*

Ceci signifie que si vous voulez avoir des espace ou des caractиres # dans le modиle (а l'extйrieur de la classe, oщ ils ne sont pas affectйs par /x), que vous aurez а mettre des йchappements ou les encoder avec des valeurs octal ou hexadйcimale. Pris ensemble, cette option va plus loin pour l'йcriture d'expressions rйguliиre pour les rendent plus lisible.

**Extensions Perl**

(?imsxr-imsxr)

Vous pouvez l'utiliser dans les e.r. pour les modifier sur le champ. Si la construction est encapsulй dans une sous expression, alors seulement la sous expression sera affectй.

**Exemples:**

(?i)Saint-Petersburg *Trouve 'Saint-petersburg' et 'Saint-Petersburg'.*

(?i)Saint-(?-i)Petersburg *Trouve 'Saint-Petersburg' mais pas 'Saint-petersburg'.*

(?i)(Saint-)?Petersburg *Trouve 'Saint-petersburg' et 'saint-petersburg'.*

((?i)Saint-)?Petersburg *Trouve 'saint-Petersburg', mais pas 'saint-petersburg' .*

(?#text)

Un commentaire, le texte est ignorй. Veuillez noter que TRegExpr ferme le commentaire aussitфt qu'il voit une parenthиse ")", Aussi il n'y a aucune faзon de placer une parenthиse dans le commentaire sans fermer celui-ci.

**Explication du mйcanisme interne**

Vous voulez connaоtre les secrets interne de TRegExpr?

Bien, cette section est en construction, svp veuillez attendre quelque temps..

N'oubliez pas de lire la [FAQ](#FAQ) (spйcifiquement la section 'non-vorace' ou cette Question).

# Interface de TRegExpr

**Mйthodes et propriйtйs publique de TRegExpr**:

**property Expression** : string

Fonction

Contient l'expression Rйguliиre. Pour L'optimisation, TRegExpr va automatiquement le compiler en 'P-code' (vous pouvez le voir avec l'aide de la mйthode Dump) et stockй dans sa structure interne. La vrai [re]compilation survient quand c'est rйellement le cas - en appellant Exec[Next], Substitute, Dump, etc et seulement si l'expression ou un autre P-code a affectй les propriйtйs qui ont йtй changйes aprиs la derniиre [re]compilation.

Erreur

Si une erreur survient durant la [re]compilation, une mйthode d'erreur est appelйe (par dйfaut une erreur d'exception est levйe - voir plus bas).

**property ModifierStr** : string

Fonction

Ajuste/rйcupиre les valeurs par dйfaut des modifications d'e.r.. Le format de chaоne est similaire а (?ismx-ismx). Par exemple si ModifierStr := 'i-x' va mettre а On le modifier /i, а Off /x et laisser les autres inchangйs.

Valeurs Possibles

-i-s-m-x ou ismx ou,...

Valeurs par Dйfaut

-i-s-m-x

Erreur

Si vous essayez des modifications non supportйes, une erreur sera appelйe (par dйfaut les erreurs lиve une exception dans ERegExpr).

**property ModifierI** : boolean

Fonction

Modifier /i - ("casse des caractиres ignorйe"), initialisй avec la valeur RegExprModifierI.

Valeurs Possibles

True = Ignore les majuscules/minuscules.

False = Fait une distinction entre les majuscules/minuscules.

Valeur par Dйfaut

False

Erreur

Aucune

**property ModifierR** : boolean

Fonction

Modifier /r - ("extension de syntaxe Russe"), initialisй avec la valeur RegExprModifierR.

Valeurs Possibles

True = Active l'extension de syntaxe Russe.

False = Dйsactive l'extension de syntaxe Russe.

Valeur par Dйfaut

True

Erreur

Aucune

**property ModifierS** : boolean

Fonction

Modifier /s - '.' veut dire n'importe quel caractиre (normalement il ne comprend pas les LinesSeparators et LinePairedSeparator), initialisй avec la valeur RegExprModifierS.

Valeurs Possible

True = Sйparateur de ligne inclus pour le mйtacaractиre ".".

False = Sйparateur de ligne pas inclus pour le mйtacaractиre ".".

Valeur par Dйfaut

True

Erreur

Aucune

**property ModifierG** : boolean;

Fonction

Modifier /g. En le mettant а Off tous les opйrateurs fonctionne en mode non-vorace, donc si ModifierG = Faux, alors '\*' est comme '\*?', tous les '+' comme '+?' et ainsi de suite, initialisй avec la valeur RegExprModifierG.

Valeurs Possible

True = Mode vorace.

False = Mode non-vorace.

Valeur par Dйfaut

True

Erreur

Aucune

**property ModifierM** : boolean;

Fonction

Modifier /m Traite les chaоnes comme des lignes multiples . Ceci fait, changer `^' et `$' de correspondre au dйbut ou а la fin de la chaоne, а partir d'une nouvelle ligne ou а la fin d'une ligne, initialisй avec la valeur RegExprModifierM.

Valeurs Possible

Vrai = Considиre une nouvelle ligne ou la fin d'une ligne comme le dйbut d'une chaоne.

False = Doit absolument кtre au dйbut ou а la fin de la chaоne.

Valeur par Dйfaut

False

Erreur

Aucune

**property ModifierX** : boolean;

Fonction

Modifier /x - ("syntaxe йtendue"), initialisй avec la valeurRegExprModifierX.

Valeurs Possible

Vrai = Syntaxe йtendue activй (des commentaires peuvent кtre insйrйs dans l'e.r.).

False = Syntaxe йtendue dйsactivй (aucun commentaire possible).

Valeur par Dйfaut

False

Erreur

Aucune

**function Exec** (const AInputString : string) : boolean;

Fonction

Compare une recherche а la chaоne AInputString.

Note

La fonction Exec stocke AInputString dans la propriйtй InputString.

**function ExecNext** : boolean;

Fonction

Trouve l'occurrence suivante de Exec(AString);

Note

fonctionne comme

Exec (AString);

if MatchLen [0] = 0 then ExecPos(MatchPos [0] + 1)

else ExecPos (MatchPos [0] + MatchLen [0]);

mais est plus simple !

**function ExecPos** (AOffset: integer = 1) : boolean;

Fonction

Trouve une occurrence de recherche pour de dйpart de InputString а partir de la position Aoffset (AOffset=1 - premier caractиre de InputString).

**property InputString** : string;

Fonction

Retourne le chaоne d'entrйe courante (а partir du dernier appel de Exec ou de la derniиre dйsignation de cette propriйtй).

Note

Une modification а cette propriйtй efface les propriйtйs Match\* !

**function Substitute** (const ATemplate : string) : string;

Fonction

Retourne ATemplate avec '$&' ou '$0' remplacй par l'occurrence complиte de l'e.r. et '$n' remplacй par l'occurrence de la sous expression #n.

Valeur de Retour

Contient la chaоne avec les modification apportйes.

Note

Depuis la v.0.929 '$' utiliser plutфt '\' (pour les futures extensions et pour plus de compatibilitй avec Perl) pour accepter plus d'un caractиre numйrique.

Si vous voulez placer le gabarit dans le modиle '$' ou '\', utiliser le prйfixe '\'.

Exemple: '1\$ is $2\\rub\\' -> '1$ est <Match[2]>\rub\'

Si vous voulez placer un caractиre numйrique aprиs '$n' vous devez dйlimiter n avec des accolades '{}'.

Exemple: 'a$12bc' -> 'a<Match[12]>bc', 'a${1}2bc' -> 'a<Match[1]>2bc'.

**procedure Split** (AInputStr : string; APieces : TStrings);

Fonction

Divise AInputStr en piиces dans APieces par les occurrences de l'e.r.

Note

Appelй au niveau interne Exec[Next].

**function Replace** (AInputStr : string; const AReplaceStr : string) : string;

Fonction

Retourne AInputStr avec les occurrences de l'e.r remplacй par AReplaceStr

Note

Appelй au niveau interne Exec[Next].

**property SubExprMatchCount** : integer; // **LectureSeulement**

Fonction

Le nombre de sous expressions qui a йtй trouvй dans la derniиre exйcution de Exec\*.

Valeur de Retour

S'il n'y a aucune sous expression mais que l'expression complиte а йtй trouvй (Exec\* а retournй vrai), alors SubExprMatchCount=0, si aucune sous expression et aucune e.r. complиte a йtй trouvй (Exec\* retourne Faux) alors SubExprMatchCount=-1.

Noter que quelques sous expressions peuvent ne pas кtre trouvйes et pour de telles sous expressions, MathPos=MatchLen=-1 and Match=''.

Par exemple: L'Expression := '(1)?2(3)?';

Exec ('123'): SubExprMatchCount=2, Match[0]='123', [1]='1', [2]='3'

Exec ('12'): SubExprMatchCount=1, Match[0]='12', [1]='1'

Exec ('23'): SubExprMatchCount=2, Match[0]='23', [1]='', [2]='3'

Exec ('2'): SubExprMatchCount=0, Match[0]='2'

Exec ('7') - return False: SubExprMatchCount=-1

**property MatchPos** [Idx : integer] : integer; // **LectureSeulement**

Fonction

La position d'entrйe de la sous expression #Idx en test а la derniиre exйcution de Exec\*.

Paramиtre

La premiиre sous expression a une valeur de Idx=1, derniиre - MatchCount, l'e.r. a une valeur de Idx=0.

Valeur de Retour

Retourne -1 si dans l'e.r. il n'y a pas de sous expression trouvйe dans la chaоne.

**property MatchLen** [Idx : integer] : integer; // **Lecture Seulement**

Fonction

La longueur d'entrйe de la sous expression #Idx e.r. en test а la derniиre exйcution de Exec\*. La premiиre sous expression a la valeur Idx=1, derniиre - MatchCount, l'e.r. entiиre a une valeur de Idx=0.

Valeur de Retour

Retourne -1 si dans l'e.r. il n'y a pas de sous expression ou que cette expression n'as pas йtй trouvй dans la chaоne.

**property Match** [Idx : integer] : string; // **Lecture Seulement**

Fonction

== copy (InputString, MatchPos [Idx], MatchLen [Idx])

Valeur de Retour

Retourne '' si dans l'e.r. il n'y a pas de sous expression ou que la sous expression n'as pas йtй trouvй dans la chaоne.

**function LastError** : integer;

Fonction

Retourne l'ID de la derniиre erreur, 0 s'il y a aucune erreur (inutilisable si l'erreur a gйnйrйe une erreur d'exception) et efface la valeur interne а 0 (pas d'erreur).

**function ErrorMsg** (AErrorID : integer) : string; virtual;

Fonction

Retourne un message d'erreur pour l'erreur avec ID = AErrorID.

**property CompilerErrorPos** : integer; // Lecture Seulement

Fonction

Retourne la position dans l'e.r. ou le compilateur a stoppй. Trиs pratique pour diagnostiquer les erreurs.

**property SpaceChars** : RegExprString

Fonction

Contient les caractиres traitйs comme \s (initialement remplit avec les valeurs de la variable globale RegExprSpaceChars).

**property WordChars** : RegExprString;

Fonction

Contient les caractиres traitйs comme \w (initialement remplit avec les valeurs de la variable globale RegExprWordChars).

**property LineSeparators** : RegExprString

Fonction

Les sйparateurs de ligne (comme Unix \n), initialement remplit avec les valeurs de la variable globale RegExprLineSeparators). Voir aussi a propos des séparateurs de ligne.

**property LinePairedSeparator** : RegExprString

Fonction

Paire de sйparateur de ligne (pour le Dos et Windows \r\n). Doit contenir exactement deux caractиres ou pas de caractиres du tout, initialement remplit avec les valeurs de la variable globale RegExprLinePairedSeparato). Voir aussi a propos des séparateurs de ligne.

Note

Par exemple, si vous avez besoin du style Unix, assigner а LineSeparators := #$a (caractиre de nouvelle ligne) et LinePairedSeparator := '' (chaоne vide), si par contre vous voulez accepter les sйparateurs "\x0D\x0A" mais pas "\x0D" ou "\x0A" seul, alors assigner LineSeparators := '' (chaоne vide) et LinePairedSeparator := #$d#$a.

Par dйfaut le mode 'mixe' est utilisй (dйfinit par dйfaut dans les constantes globales RegExprLine[Paired]Separator[s]): LineSeparators := #$d#$a; LinePairedSeparator := #$d#$a. Le comportement de ce mode est dйcris dans la section syntaxe.

**class function InvertCaseFunction** (const Ch : REChar) : REChar;

Fonction

Convertit Ch en majuscule si c'est minuscule et vice-versa (en utilisant les ajustement du systиme local).

**property InvertCase** : TRegExprInvertCaseFunction;

Fonction

Ajuster cette propriйtй si vous voulez йviter la fonctionnalitй de l'ignorance des minuscules/majuscules.

Note

Crйe une interdiction а la fonction RegExprInvertCaseFunction (InvertCaseFunction par dйfaut).

**procedure Compile**;

Fonction

[Re]compile l'e.r. Trиs pratique pour les applications qui utilise les йditeurs graphique pour vйrifier la validitй des propriйtйs.

**function Dump** : string;

Fonction

Crйe pour le visionnement une e.r. compilйe en une forme plus comprйhensive.

**Constantes Globales**

Valeurs par dйfaut des Modifiers:

**RegExprModifierI** : boolean = False; // TRegExpr.ModifierI

**RegExprModifierR** : boolean = True; // TRegExpr.ModifierR

**RegExprModifierS** : boolean = True; // TRegExpr.ModifierS

**RegExprModifierG** : boolean = True; // TRegExpr.ModifierG

**RegExprModifierM** : boolean = False; // TRegExpr.ModifierM

**RegExprModifierX** : boolean = False; // TRegExpr.ModifierX

**RegExprSpaceChars** : RegExprString = ' '#$9#$A#$D#$C;

// Valeur par dйfaut pour la propriйtй SpaceChars

**RegExprWordChars** : RegExprString =

'0123456789'

+ 'abcdefghijklmnopqrstuvwxyz'

+ 'ABCDEFGHIJKLMNOPQRSTUVWXYZ\_';

// Valeur par dйfaut pour la propriйtй WordChars

**RegExprLineSeparators** : RegExprString =

#$d#$a{$IFDEF UniCode}#$b#$c#$2028#$2029#$85{$ENDIF};

// Valeur par dйfaut pour la propriйtй LineSeparators

**RegExprLinePairedSeparator** : RegExprString =

#$d#$a;

// Valeur par dйfaut pour la propriйtй LinePairedSeparator

**RegExprInvertCaseFunction** : TRegExprInvertCaseFunction = TRegExpr.InvertCaseFunction;

// Valeur par dйfaut pour la propriйtй InvertCase

**Fonctions globales pratiques**

**function ExecRegExpr** (const ARegExpr, AInputStr : string) : boolean;

Fonction

Retourne vrai si la chaоne AInputString concorde а l'expression ARegExpr.

Note

!Va lever une exception s'il y a une erreur de syntaxe dans ARegExpr.

**procedure SplitRegExpr** (const ARegExpr, AInputStr : string; APieces : TStrings);

Fonction

Sйpare AInputStr en piиces dans APieces par les occurrences de l'e.r. ARegExpr.

**function ReplaceRegExpr** (const ARegExpr, AInputStr, AReplaceStr : string) : string;

Fonction

Retourne AInputStr avec l'occurrence de l'e.r. remplacй par AReplaceStr.

**function QuoteRegExprMetaChars** (const AStr : string) : string;

Fonction

Remplace tous les mйtacaractиres avec une reprйsentation simple, par exemple 'abc$cd.(' est converti en 'abc\$cd\.\('.

Note

Cette fonction est trиs pratique pour l'autogйnйration d'e.r. а partir d'entrйe utilisateur.

**function RegExprSubExpressions** (const ARegExpr : string; ASubExprs : TStrings; AExtendedSyntax : boolean = False) : integer;

Fonction

Fabrique une liste de sous expression trouvй dans l'e.r. ARegExpr.

Note

Dans ASubExps chaque item reprйsente une sous expression, а partir de la premiиre jusqu'а la derniиre, dans le format:

Chaоne - texte de sous expression (sans les '()').

bas mot de l'objet - position de dйpart dans ARegExpr, incluant '(' s'il existe ! (la premiиre position est 1).

haut mot de l'objet - La longueur, incluant le dйpart '(' et la fin ')' s'il existent!

AExtendedSyntax - Doit кtre Vrai si le modifier /x est а On durant l'utilisation de l'e.r.

Utile pour les йditeurs avec interface graphique (Vous pouvez trouver un exemple d'utilisation dans le projet [TestRExp.dpr](#tregexpr_testrexp)).

**Code Résultat** **Sens**

**0** Succčs. Pas de parenthčse non balancées trouvé.

**-1** Il n'a pas assez de parenthčse de fermeture.

**-(n+1)** Ŕ la position **n** était trouvé '[' sans fermeture ']'.

**n** Ŕ la position **n** était trouvé ')' sans ouverture '('.

Si le rйsultat <> 0, alors ASubExprs peut contenir des items vide ou de items illйgaux.

**Type d'exception**

Routine par Dйfaut des erreurs d'exception pour TRegExpr:

**ERegExpr** = class (Exception)

public

ErrorCode : integer; // code d'erreur. Les erreurs de compilation du code sont avant 1000.

CompilerErrorPos : integer; // Position dans l'e.r. oщ l'erreur est survenue.

end;

**Comment utiliser les Unicode**

TRegExpr supporte maintenant les UniCode, mais il travaille **trиs lentement**.

Qui veut se risquer а l'optimiser ?

L'utiliser seulement si vous avez vraiment besoin du support des Unicode !

Pour utiliser les WideString, enlever le '.' dans {.$DEFINE UniCode} dans le fichier regexpr.pas.

# Projet Demo (TestRExp)

Simple programme pour explorer et tester les e.r., distribuй en fichiers sources (projet TestRExp.dpr) et avec l'exйcutable TestRExp.exe.

Noter qu'il utilise les propriйtйs de plusieurs VCL qui existent seulement dans Delphi 4 ou plus rйcent. En compilant dans Delphi 3 ou Delphi 2 vous recevrez quelques message d'erreurs a propos de **propriйtйs inconnues**. Vous pouvez les ignorer, ces propriйtйs sont seulement nйcessaire lorsque vous redimmensionner la fenкtre du programme.

Avec l'aide de ce programme, vous pourrez aisйment dйterminer le nombre de sous expression que vous кtes en train de modifier, aller а n'importe quelle sous expression dйfinie (dans le code compilй de l'e.r. autant que dans le rйsultat des chaоnes d'entrйe), jouer avec les substituts, remplacer, sйparer et mкme plus.

Et de plus, le projet inclus une bonne quantitй d'exemple - utiliser les pour apprendre la syntaxe des e.r. ou pour apprendre rapidement les avantages des fonctionnalitйs de TRegExpr.

# Exemple: Décorateur HyperLinks

Fonctions pour dйcorer les liens hyperlinks en convertissant le texte standard en format HTML.

Par exemple, remplace '*http://anso.da.ru*' avec '*<a href="http://anso.da.ru">anso.da.ru</a>*' ou '*anso@mail.ru*' avec '*<a href="mailto:anso@mail.ru">anso@mail.ru</a>*'.

**fonction Dйcorer les URLs**

Trouve et remplace les liens comme 'http://...' ou 'ftp://..' aussi bien que les liens sans protocol, mais qui commence avec 'www.' Si vous voulez dйcorer les e-mails, vous pouvez utiliser la fonction DecorateEMails.

**function** DecorateURLs (**const** AText : **string**; AFlags : TDecorateURLsFlagSet = [durlAddr, durlPath]) : **string**;

**Description**

retourne le texte d'entrйe AText avec les liens hyperliens dйcorйs.

AFlags dйcris quelle partie de l'hyperlien lien doit кtre inclus dans la partie VISIBLE du lien:

Par exemple, si [durlAddr] alors hyperlien 'http://anso.da.ru/index.htm' sera dйcorй comme '<a href="http://anso.da.ru/index.htm">anso.da.ru</a>'.

**type**

TDecorateURLsFlags = (durlProto, durlAddr, durlPort, durlPath, durlBMark, durlParam);

TDecorateURLsFlagSet = jeux de TDecorateURLsFlags;

**Description**

Voici les valeurs possibles pour TDecorateURLsFlagSet:

**Value** **Meaning**

durlProto Protocole (comme 'ftp://' ou 'http://').

durlAddr Adresse TCP ou nom de domaine (comme 'anso.da.ru').

durlPort Numйro de port si spйcifiй (comme ':8080').

durlPath Chemin au document (comme 'index.htm').

durlBMark Bookmark (comme '#mark').

durlParam Paramиtres URL (comme '?ID=2&User=13').

**Fonction Dйcorer les EMails**

Remplace les e-mails avec '<a href="mailto:ADDR">ADDR</a>'. Par exemple, remplace 'anso@mail.ru' avec '<a href="mailto:anso@mail.ru">anso@mail.ru</a>'.

**function** DecorateEMails (**const** AText : **string**) : **string**;

**Description**

Retourne le texte d'entrйe avec les dйcoration e-mails dans Atext.

# FAQ

Q.

**Comment utiliser TRegExpr avec Borland C++ Builder?**

J'ai un problиme depuis qu'il n'y a plus aucun fichier d'en-tкte (.h or .hpp) n'est pas disponible.

R.

· Ajouter RegExpr.pas au projet bcb.

· Compiler le projet. Ceci gйnиre le fichier RegExpr.hpp.

· Maintenant vous pouvez utiliser l'unitй RegExpr.

· Ne pas oublier d'inclure #include "RegExpr.hpp" au dйbut de votre programme.

Q.

**Pourquoi TRegExpr retourne plus d'une ligne?**

Par exemple l'e.r. <font .\*> retourne le premier <font, et ensuite **le reste du fichier** incluant le dernier </html>...

R.

Pour la compatibilitй prйcйdente le modifier /s est а 'On' par dйfaut.

Le mettre а off et ensuite le '.' concordera а tout exceptй les séparateurs de ligne - comme voulu.

Mais je suggиre aussi l'e.r. suivante '<font ([^\n>]\*)>', dans Match[1] sera l'url.

Q.

**Pourquoi TRegExpr retourne plus que prйvu?**

Par exemple l'e.r. '<p>(.+)</p>' appliquй а la chaоne '<p>a</p><p>b</p>' retourne 'a</p><p>b' mais pas 'a' comme prйvu.

R.

Par dйfaut tous les opйrateurs fonctionne en mode "vorace", aussi il correspond а compare le plus possible. Si vous voulez le mode "non-vorace" vous pouvez utiliser les opйrateurs '+?' et ainsi de suite (nouveau dans la v. 0.940) ou changer tous les opйrateurs en mode "non-vorace" avec l'aide du modifier "g" (utiliser les propriйtйs convenablement dans TRegExpr ou avec une inscription dans l'e.r. comme '?(-g)').

Q.

**Comment analyser des sources comme du HTML avec l'aide deTRegExpr**

R.

Dйsolй les gars, mais c'est pratiquement impossible!

Bien sur que vous pouvez utiliser TRegExpr pour extraire des informations comme dйmontrй dans mes exemples, mais si vous voulez faire une analyse prйcise, vous devez utiliser un vrai analyseur, pas l'e.r.!

Vous pouvez lire les explications de Tom Christiansen et Nathan Torkington dans le document 'Perl Cookbook', par exemple. Pour faire une histoire courte, il y a plusieurs expressions qui peuvent кtre analyser facilement avec un vrai analyseur mais pas toutes par e.r., et les vrai analyseurs sont PLUS rapide pour faire l'analyse, parce que l'e.r. ne scanne pas l'entrйe avant, il fait plutфt une optimisation de recherche qui peut prendre beaucoup de temps.

Q.

**Est-ce qu'il y a une faзon d'avoir une correspondance multiple d'un modиle sur TRegExpr?**

R.

Vous pouvez faire une loupe et procйder comparaison par comparaison avec la mйthode ExecNext.

Зa ne peut кtre plus fait plus facilement parce que delphi n'est pas un interprйteur comme Perl (et les interprйteurs fonctionnent gйnйralement trиs lentement!).

Si vous voulez quelques exemples, svp visionner la mйthode TRegExpr.Replace. ou aux exemples dans [HyperLinksDecorator.pas](#HyperLinksDecorator).

Q.

**Je vйrifie l'entrйe d'utilisateur. Pourquoi TRegExpr retourne 'Vrai' pour une mauvaise chaоne d'entrйe?**

R.

Dans plusieurs cas de TRegExpr les utilisateurs oublient qu'une expression rйguliиre est pour chercher dans les chaоnes d'entrйes. Aussi, si vous voulez faire que l'utilisateur entre seulement 4 caractиres numйrique et que vous utiliser l'expression '\d{4,4}', vous pouvez ignorer les mauvaises entrйes comme '12345' ou 'n'importe quel caractиre 1234 et n'importe quoi'. Vous devez ajouter une vйrification pour le dйbut et la fin de la ligne et vous assurer qu'il n'y a rien d'autre comme dans l'expression suivante: '^\d{4,4}$'.

Q.

**Pourquoi que le mode non-vorace quelquefois fonctionne comme le mode vorace?**

Par exemple, l'e.r. 'a+?,b+?' appliquй а 'aaa,bbb' retourne 'aaa,b', mais normalement ne devrait-il pas retourner 'a,b' а cause de la nature non-vorace du premier itйrateur ?

R.

C'est une limite d'utilisation par la mathйmatique de TRegExpr (et plusieurs autre comme Perl et Unix). E.r. effectue seulement une 'simple' optimisation de recherche, et ne tente pas d'obtenir la **meilleure optimisation**. Dans plusieurs cas ce n'est pas bon, mais en gйnйral cette limite est plutфt avantageuse, а cause des performances et des prйvisions de raison.

La rиgle gйnйrale est que premiиrement e.r. essaie de trouver une correspondance а partir de sa position actuelle et seulement si c'est complиtement impossible de trouver une correspondance alors il avance d'un caractиre et rйessaie de nouveau а partir de cet emplacement. Aussi si vous utiliser 'a,b+?' il correspondra avec 'a,b', mais dans le cas de 'a+?,b+?' ce '**n'est pas recommandй**' (а cause du mode non-vorace) **mais possible** de correspondre а plus d'un 'a', aussi TRegExpr le fait mais le rйsultat obtenu ne sera pas une correspondance optimum. TRegExpr comme Perl ou les e.r. de Unix ne tente pas de bouger en avant et vйrifier qu'est-ce qu'il serait la meilleure correspondance. De plus, il ne peut comparer en terme 'plus ou moins bon'.

SVP, lire le section 'Syntaxe' du fichier d'aide pour plus d'explication.

# Auteur

Andrey V. Sorokin

Saint Petersburg, Russie

[anso@mail.ru](`mailto:anso@mail.ru',`',1)

[http://RegExpStudio.com](http://RegExpStudio.com/)

**Svp, si vous pensez avoir trouvй une erreur ou avez des questions а propos de TRegExpr, tйlйcharger la version la plus rйcente de TRegExpr а partir de ma page web et lire la** [**FAQ**](#FAQ) **avant de m'envoyez un e-mail!**

Cette librairie est dйrivй des fichiers sources de Henry Spencer.

J'ai transfйrй les fichiers source en objet pascal, et y ai ajoutй d'autres fonctions.

Plusieurs de ces nouvelles fonctions ont йtй faites а partir de suggestion ou а partir de modification partielle d'autre utilisateurs de TRegExpr (voir la liste de gratitude plus bas).

---------------------------------------------------------------

Gratitudes

---------------------------------------------------------------

· Guido Muehlwitz - Trouver et rйparer une erreur avec le traitement des grandes chaоnes.

· Stephan Klimek - Test en mode CPPB et suggestion/implйmentation de plusieurs fonctions.

· Steve Mudford - Implйmentation du paramиtre Offset.

· Martin Baur ([www.mindpower.com](`http://www.mindpower.com',`',1)) - Aide Allemande, suggestions utiles.

· Yury Finkel - Implйmentation du support UniCode, trouvй et fixй quelques erreurs.

· Ralf Junker - Implйmentation de plusieurs fonctions, plusieurs suggestions d'optimisation.

· Simeon Lilov - Aide Bulgare

· Filip Jirsбk et Matthew Winter (wintermi@yahoo.com) - Aide de l'implйmentation en mode non-vorace.

· Kit Eason - Plusieurs exemples pour l'introduction de l'aide.

· Juergen Schroth - Recherche d'erreurs et suggestions pratiques.

· Martin Ledoux - Pour la traduction en franзais.

Et plusieurs autres - pour leur gros travail de recherche des erreurs !

Keyword index

**No index entries found.**