



Developpez

Magazine

Edition de Septembre 2005.
Numéro 3.
Magazine en ligne gratuit.
Diffusion de copies conformes à l'original autorisée.
Directeur de la publication : Bobby Joe Lambert
Réalisation : Hugo Etiévant
Rédaction : la rédaction de Developpez
Contact : magazine@redaction-developpez.com

Developpez.com

Perl

Débutez avec
facilité et rapidité
page 4

Java.....	page 2
C/C++.....	page 7
Access.....	page 8
.NET.....	page 8
SGBD.....	page 11
UML.....	page 11
Web.....	page 11
Pascal.....	page 12
Delphi.....	page 12

Liens.....page 13

Editorial

La rentrée 2005 annonce
sont lot de nouveautés : une
nouvelle rubrique, de
nouvelles F.A.Qs... Et
comme toujours son
contingent de cours de
haute volée, les conférences
et autres critiques de livres.
Bonne rentrée à tous !

Cours .NET

ClickOnce : Le déploiement facile d'applications .NET

*ClickOnce est la nouvelle technologie de déploiement
d'applications. Cet article vous montrera les différentes
possibilités que cette technologie vous apporte, tant au
niveau du déploiement que de la mise à jour de vos
applications.*

par **LEBRUN Thomas** et **MORAND Louis-Guillaume**
page 9

Nouvelle FAQ !

F.A.Q PostgreSQL

*Découvrez la nouvelle FAQ PostgreSQL
avec déjà près de 50 Q/R. Elle a été
réalisée à partir des contributions des
membres du forum PostgreSQL de
Developpez.*

-> <http://dgiessinger.developpez.com/postgresql/faqpg/>



Nouvelle rubrique !

InterBase/Firebird

-> <http://interbase.developpez.com/>

Developpez Magazine est une publication
du site Developpez.com



Java

Les derniers tutoriels

Drag & Drop avec Style et l'API Swing

Comment pouvons nous nous débarrasser des curseurs de drag & drop simplistes offerts par Java ? La solution réside dans la démo Drag n' Ghost que je vais vous montrer.

Lire le cours de **Romain Guy (lien 1)**

Attendre avec style et l'API Swing

A Démonstration sur l'utilisation de composants Swing pour attendre lors de l'exécution de tâches longues.

Lire le cours de **Romain Guy (lien 2)**

Mixer Java3D et Swing

Tutoriel d'introduction sur l'utilisation de l'API Java3D avec la librairie graphique Swing.

Lire le cours de **Romain Guy (lien 3)**

Introduction à JSF avec NetBeans

Article d'introduction à JSF (Java Server Faces) avec l'EDI NetBeans.

Lire le cours de **Olivier SCHMITT (lien 4)**

Introduction à Log4J

Ce cours est une introduction au système de journalisation pour java de Jakarta : log4j. L'accent est mis sur la façon de le configurer.

Lire le cours de **Sébastien Le Ray (lien 5)**

Construction d'une application MVC distribuée avec Spring Remoting

Partie 1 : HttpInvoker, Hessian, Burlap, RMI

Lire le cours de **Serge Tahé (lien 6)**

Vu dans la F.A.Q Java

Qu'est-ce que l'auto-boxing / auto-unboxing ?

L'auto-boxing gère la transformation des types primitifs (boolean, byte, char, short, int, long, float, double) vers la classe 'wrapper' correspondante (Boolean, Byte, Character, Short, Integer, Long, Float, Double) et inversement pour l'auto-unboxing... Tout est transparent, il n'y a donc plus aucune

conversion explicite. Ainsi, le code suivant est tout à fait correct :

```
Integer integer = 0;
int j = integer;
Map map = new HashMap();
map.put ( 2.1, "Valeur" );
```

adiGuba

Vu dans le blog

Valider les saisies grâce aux annotations.

Lors de la rédaction de mon tutoriel sur les annotations, j'ai bien entendu consulté plusieurs articles sur le sujet. Parmi ceux-ci, le plus intéressant est sans aucun doute celui de Anders Holmgren, publié en mars 2005 sur le site de Sun, et traitant de la possibilité d'ajouter des contraintes de validation sur les propriétés d'un bean en utilisant seulement des annotations : *Using Annotations to add Validity Constraints to JavaBeans Properties* (en anglais bien sûr, **lien 7**).

Je vous conseille fortement d'y jeter un coup d'oeil...

Tout cela pour dire que je me suis mis en tête de développer une librairie basique pour la validation de formulaire web dans une application J2EE, à la manière du plugin Validator de Struts, mais qui utiliserait les Annotations de Java 5.0.

Pour le moment, je vois trois principaux modules dans cette librairie:

- Un module qui se chargera de valider les différentes propriétés en fonction de leurs contraintes.
- Un module utilisant APT afin d'effectuer des vérifications lors de la compilation des beans.
- Un module permettant de générer des formulaires HTML dans une page JSP (via des taglibs).

adiGuba

F.A.Q Général Java

Déjà 302 réponses à vos questions dont 22 nouvelles. Cette faq a été réalisée à partir des questions fréquemment posées sur les forums de Developpez

Lire la F.A.Q :

<http://java.developpez.com/faq/java/>



Conférences

Le symposium **JA00 2005** se déroulera sur six jours (du 25 au 30 septembre) à Aarhus (Danemark). Elle est la conférence européenne la plus importante autour de l'ingénierie logicielle, des méthodes de développement et "Best practices". Ses centres d'intérêts sont les technologies Java, l'orienté objet et .Net.

Inscriptions : <https://secure.eos.dk/registration/>

Javapolis 2005 du 12 au 16 décembre 2005, vous pourrez écouter, et même cotoyer, les personnes qui font vivre Java. Elle se tient dans le complexe cinématographique Metropolis d'Anvers, en Belgique.

Inscriptions : <http://www.javapolis.com/jp05reg/>



Retrouvez la rubrique Java sur : <http://java.developpez.com>

Java

Critique de livre

Au coeur de Java 2, Volume 2, Fonctions avancées JDK 5.0

Cay S Hortsman & Gary Cornell
Editions Campus Press Référence



Livraison gratuite sur Amazon



ISBN: 2-7440-1962-3
858 pages
juin 2005

Résumé éditeur

Plus de 250 000 programmeurs ont utilisé Au cœur de Java 2 Volume 1- Notions fondamentales pour consolider leur savoir sur la technologie Java. Ce second volume va plus loin encore et constitue un guide complet pour résoudre les problèmes quotidiens des développeurs. La cinquième édition du volume 2 entre en détail dans des sujets tels que la programmation de l'interface utilisateur et les fonctions professionnelles de la plate-forme java 2, édition standard (JSE™). Totalement mis à jour, cet ouvrage traite les multithreads, qui permettent d'exécuter des tâches en parallèle, l'API JDBC 2.0, CORBA, le glisser-déposer et le modèle de sécurité Java 2. Ce dernier a été revu pour le JDK 5.0, de façon à prendre en compte les algorithmes de cryptage AES et RSA. Les techniques de Swing avancées sont abordées, dont les composants GUI complexes d'arbres et de tables, les indicateurs de progression et l'implémentation Java d'une interface à "plusieurs documents". Un chapitre décrit les techniques de conception des cadres de collections et l'exploitation des données structurales fondamentales autorisées par la bibliothèque standard Java. Cette édition explique aussi comment internationaliser des applications Java, générer du XML et utiliser les transformations XSL. Le chapitre consacré au XML a été mis à jour pour

inclure l'API XPath. Enfin, un nouveau chapitre a été ajouté il traite des annotations et des métadonnées, des fonctionnalités ajoutées au JDK 5.0.

Critique

Ce second tome complète à merveille le premier (voir critique du 1er tome sur developpez.com)

Le 1er chapitre a été entièrement réécrit, pour aborder la nouvelle API concernant la Concurrency introduite dans la version 5 de Java. Cette nouvelle API met à votre disposition tout un ensemble de classes et interfaces vous permettant de construire plus facilement des applications multithreads performantes que ce qui était possible jusqu'à présent.

Le second chapitre, couvrant les Collections, a été étendu pour aborder la notion de Queue, la classe PriorityQueue, Ce chapitre explique également les tables de hachage, et pourquoi il est important de choisir un bon code de hachage. Il aborde également la notion de Weak Reference, bien que cela soit traité de façon plus superficielle que dans le livre *HardCore Java*, de chez O'Reilly.

J'ai une remarque négative à faire à l'encontre des traducteurs. Pourquoi ont-ils traduit les noms des classes et/ou interfaces en français dans les figures montrant les anciennes classes, comme Vector, Map et Properties. Mais cela ne concerne qu'une figure sur une page dans tout le livre.

Je n'ai pas eu l'impression qu'il y a beaucoup de modification concernant les chapitres 3 à 10 par rapport à l'ancienne version. Même si le chapitre sur la sécurité couvre maintenant AES et RSA, et que le chapitre XML couvre XPath. Mais cela est assez concis et rien de tel qu'un ouvrage de référence sur XML et XPath. Et que les exemples ont été revus. Par exemple, dans le chapitre 9, il explique comment appeler une méthode printf depuis Java.

Le chapitre 11 concernant les annotations est quant à lui tout à fait nouveau et vous explique en détails. Il aborde également BCEL, qui vous permet de manipuler directement du ByteCode.

Pour ceux qui possèdent une ancienne édition de ce deuxième tome, l'achat de celui-ci vous sera utile si vous voulez connaître les nouveautés introduites au niveau du Multithreading, des collections, des certificats RSA et AES, et connaître les annotations. Pour ceux qui ne possèdent pas d'ancienne édition de ce deuxième tome, je ne peux que vous conseiller ce livre de référence. En tout cas, si vous avez acheté récemment la dernière édition du Tome 1, n'hésitez pas à acheter son compagnon.

Vincent Brabant

Retrouvez la critique complète ([lien 7](#))

Perl

Apprendre Perl rapidement, 2ème partie

Dans la première partie, nous avons vu ensemble les 3 variables fondamentales de Perl. A présent, nous irons plus loin dans la compréhension de ce langage pour nous faciliter la vie dans nos développements. Ainsi, nous apprendrons à partager notre code pour ne plus le rendre "monolithique", à découvrir des nouvelles boucles et à faire prendre des décisions à nos programmes. **GLDavid**

I. Conditions en Perl

Il peut être bien utile, dans un programme, de prendre des décisions en fonction des variables et des valeurs que vous leur donnerez. Et comme dirait l'adage : "avec des si, on mettrait Paris en bouteille !". Alors, appliquons ce proverbe dans le code suivant :

```
condition1.pl
#!/usr/bin/perl
print "Rentres une ville :\n";
$ville = <stdin>;
chomp $ville;
if ($ville eq "Paris") {
    print $ville." rentre dans une bouteille
!\n";
}
```

Voyons ce code. La première ligne vous est bien connue, il s'agit de la ligne shebang où nous déclarons le chemin de l'interpréteur Perl. Puis, nous affichons avec print une chaîne de caractères. La ligne suivante demande à l'utilisateur de rentrer une valeur, ici, le nom d'une ville qui sera intégré dans la variable scalaire \$ville. La fonction chomp supprime le retour chariot contenu dans la variable \$ville.

Maintenant, viens notre propos. le programme se pose la question suivante : "Qu'est-ce que je fais en fonction de la valeur de \$ville ?" A cela, nous lui apportons la solution : "si" la valeur de \$ville est égale à Paris, alors, tu affiches la chaîne concaténée entre la valeur de \$ville et une phrase. Notez que nous mettons la solution à la condition entre une paire d'accolades ouvrantes-fermantes que l'on appelle un bloc.

Mais, j'entend votre question : et si l'utilisateur ne rentre pas Paris ? Et bien, nous allons voir un autre mot-clé bien utile si un "if" est insuffisant. Il s'agit du mot else soit le "sinon" anglais. Prenons l'exemple suivant :

```
condition2.pl
#!/usr/bin/perl
print "Rentres un chiffre pour connaître son
inverse :\n";
$nombre = <stdin>;
chomp $nombre;
if($nombre == 0){
    print "La division par 0 est impossible !";
} else {
$resultat = 1/$nombre;
    print "L'inverse de ".$nombre." est
".$resultat."\n";
}
```

Après avoir écrit notre ligne shebang, nous demandons à l'utilisateur de rentrer un nombre et nous enlevons le retour chariot. Comme nous divisons 1 par ce nombre pour obtenir l'inverse du nombre, il est évident qu'une division par 0 est impossible. Aussi, à l'aide de la paire if-else, nous allons analyser le nombre rentré. Dans notre "if", nous regardons si le nombre est égal à 0. Si c'est le cas, nous affichons une remontrance envers l'utilisateur (si, si, parfois, l'informaticien se montre sévère envers ses utilisateurs) et le programme se termine. Dans le cas contraire, le bloc else est alors invoqué : nous stockons le résultat dans la variable scalaire \$resultat de la division de 1 par notre nombre \$nombre. Enfin, nous affichons le résultat de cette opération dans une chaîne concaténée.

Mais, Perl va encore plus loin ! En effet, ce langage propose même un mot clé contraire à "if". Il s'agit du mot clé unless (à moins que en français). Considérons le code suivant qui reprend le précédent :

```
condition3.pl
#!/usr/bin/perl
print "Rentres un chiffre pour connaître son
inverse :\n";
$nombre = <stdin>;
chomp $nombre;
unless($nombre == 0){
    $resultat = 1/$nombre;
    print "L'inverse de ".$nombre." est
".$resultat."\n";
} else {
    print "La division par 0 est impossible !";
}
```

Notre condition se lit donc de la manière suivante : "à moins que \$nombre soit égal à 0, on procède à la division. Sinon, on affiche l'erreur. Vous aurez compris la subtilité : if se contente de vérifier si la condition est vraie alors qu'unless vérifie si la condition est fausse. Bien naturellement, vous devez à chaque appel d'une condition placer un bloc (paire d'accolades ouvrantes-fermantes) à la suite de votre condition. Enfin, une dernière remarque pour les tests de comparaison. Vous aurez vu que j'ai utilisé 2 opérateurs différents pour tester une variable scalaire qui contient une chaîne de caractère et une autre qui contient un nombre. Retenez ainsi que si vous voulez comparer une égalité entre chaînes de caractères, vous devez avoir recours à l'opérateur eq (pour équivalent). Son opposé est ne (pour non-équivalent), toujours pour les chaînes de caractères. Concernant les nombres, nous retrouvons des opérateurs connus : == (pour l'égalité, attention de ne pas utiliser un = seul !), <, > (pour les différences strictes), <=, >= (pour les différences strictes ou égales) parmi les plus usités.

II. Répéter une opération, les boucles

Dans le cours précédent, nous avons commencé à voir quelques boucles mais sans rentrer dans les détails. Maintenant, rentrons dans les détails! Mais, à quoi sert une boucle ? Une boucle va vous aider à répéter un bloc d'instructions jusqu'à ce qu'elle satisfasse une condition. Toutefois, il existe aussi des cas où des boucles seront infinies ! C'est le cas des IHM (Interfaces Homme-Machine).

Mais reprenons notre propos. Comme toujours, nous allons nous aider des traductions anglais => français pour comprendre une boucle. Débutons avec la boucle while. La boucle while se réalise tant que la condition n'est pas atteinte. Un petit exemple :

```
boucle1.pl
#!/usr/bin/perl
print "Table de multiplication de 2\n";
$i = 0;
while($i<=10){
    print "$i*2 = ".$($i*2)."\n"; $i++;
}
```

Vous aurez compris que nous allons afficher la table de multiplication de 2. Nous initialisons une variable scalaire \$i à 0. Cette variable va être utilisée dans notre boucle. Notre boucle while nous dit ceci : "tant que \$i est inférieur ou égal à 10, j'effectue ce que l'on me demande". la

boucle while est suivie alors d'une paire d'accolades ouvrantes-fermantes contenant les instructions de la boucle (ce qu'elle doit faire). Ainsi, dans le bloc ou corps de boucle, nous demandons à afficher la valeur de \$i actuelle et de la multiplier par 2. Enfin, nous incrémentons \$i. L'incrémentement consiste à ajouter 1 à une valeur donnée. Il existe 2 types d'incrémentations :

- La pré-incrémentation : ++\$i; où la valeur sera d'abord incrémentée.
- La post-incrémentation : \$i++; où la valeur sera après incrémentée.

Enfin, il existe l'opérateur de décrémentation (--), qui se base sur les mêmes règles que l'incrémentation. Notre boucle se termine lorsque sa valeur atteint 10.

En fait, comme tout langage, Perl vous autorise à écrire une boucle selon votre préférence. Par exemple, la boucle for permet de rassembler en 1 ligne l'initialisation d'une variable, sa condition et son incrémentement. La boucle for signifiant "pour", reprenons notre exemple de tout à l'heure :

```
boucle2.pl
#!/usr/bin/perl
print "Table de multiplication de 2\n";
for ($i=0; $i<=10; $i++){
    print "$i*2 = ".$($i*2)."\n";
}
```

Un code plus court, n'est-ce pas ? L'utilisation d'un for n'est guère différente de la boucle while. La boucle for nous dit : "Pour \$i égal à 0 et tant que \$i est inférieur ou égal à 10, j'incrémenterais de 1 \$i et je ferais ce que l'on me dit de faire".

Dans notre partie consacrée aux conditions, nous avons vu qu'il existait un opposé à if nommé unless. Comme dans la nature, il existe un opposé pour chaque élément, la boucle while a aussi son opposé : until. Les boucles while et for se réalisent tant que la condition est vraie, la boucle until se réalise tant que sa condition est fautive. En anglais, until se traduit par "jusqu'à ce que". Voyons comment utiliser until avec notre exemple de table de multiplication :

```
boucle3.pl
#!/usr/bin/perl
print "Table de multiplication de 2\n";
$i = 0;
until($i>10){
    print "$i*2 = ".$($i*2)."\n";
    $i++;
}
```

Le code n'est pas si différent de notre exemple avec while. Mais la pensée est différente. Comme dans notre exemple de boucle1.pl, nous initialisons une scalaire \$i à 0. mais notre boucle nous indique cette fois: "Jusqu'à ce que \$i soit supérieur strictement à 10, je fais ce que l'on me dit". Le corps de la boucle ne change pas. Seule la condition a été modifiée.

Mais certaines boucles sont bien utiles pour des types de variables plus évoluées comme les listes ou tables de hachages. Ainsi, la boucle foreach est très pratique pour parcourir les éléments d'une liste. Cependant elle peut aussi manipuler aisément une table de hachage :

```
boucle4.pl
#!/usr/bin/perl
@liste = (0,1,2,3,4,5,6,7,8,9,10);
print "Table de multiplication de 2\n";
foreach $i (@liste){
    print "$i*2 = ".$($i*2)."\n";
}
```

Nous avons ainsi initialisé une liste avec une série de chiffres. Puis la boucle indique "Pour chaque élément noté \$i de la liste @liste (écrite entre parenthèse), je fais ce que l'on me dit". Le corps de la boucle multiplie ainsi l'élément en cours par 2 et affiche le résultat.

III. Variables spéciales de Perl

Perl contient quelques variables réservées qui sont bien utiles notamment pour manipuler les paramètres d'appels d'un script, rapporter une erreur ou connaître son environnement. Loin de nous est l'objectif de dresser une liste exhaustive de ces différentes variables, nous nous contenterons de vous citer les plus utiles.

Commençons par quelques variables scalaires spéciales. Savez-vous qu'il existe une variable scalaire par défaut ? Celle-ci est notée \$_. Elle contient tout ce qui est en cours d'une boucle ou elle est la variable par défaut de nombreuses fonctions.

```
special1.pl
#!/usr/bin/perl
@liste = (0,1,2,3,4,5,6,7,8,9,10);
print "Table de multiplication de 2\n";
foreach (@liste){
    print "$_*2 = ".$($_*2)."\n";
}
```

Vous connaissez ce code, il s'agit de l'affichage de la table de multiplication de 2 à l'aide d'une boucle foreach. Mais vous me direz qu'il manque une variable. Pas du tout ! Dans notre ligne où nous décrivons notre boucle foreach, la variable spéciale \$_ peut être écrite explicitement sinon elle est appelée implicitement par le code. Notre code aurait pu s'écrire de la manière suivante avec le même résultat :

```
special2.pl
#!/usr/bin/perl
@liste = (0,1,2,3,4,5,6,7,8,9,10);
print "Table de multiplication de 2\n";
foreach $_ (@liste){
    print "$_*2 = ".$($_*2)."\n";
}
```

Il est aussi parfois bien utile de rapporter une erreur. La variable \$! sert ainsi à enregistrer l'état de votre erreur :

```
special3.pl
#!/usr/bin/perl
print "Rentrez un nom de fichier :\n";
$fichier = <stdin>;
chomp $fichier;
if (-f $fichier){
    print "$fichier existe.\n";
} else {
    print "$!\n";
}
```

A l'aide de ce code, nous demandons au script de vérifier si le fichier dont le nom est rentré par l'utilisateur existe. Si c'est le cas (avec l'option -f d'un if, nous testons l'existence d'un fichier dont le nom est contenu dans une variable scalaire), le script affiche que le fichier existe. Dans le cas contraire, nous affichons le contenu de notre erreur. Si vous rentrez un nom de fichier qui n'existe pas, \$! contiendra alors la valeur:

```
No such file or directory
```

Il existe aussi des listes spéciales à Perl. Aussi, vous savez que de nombreux programmes ont besoin de paramètres pour pouvoir s'exécuter. Un exemple sur les machines Unix est que certaines commandes vous affichent de l'aide lorsqu'elles n'ont pas d'arguments ou que ceux-ci sont inexacts. Aussi, comment pouvons-nous tester l'existence de paramètres pour un programme Perl. Tous les paramètres invoqués par un script Perl sont ainsi contenus dans un tableau spécial, le tableau @ARGV. Sa manipulation est identique à tous les tableaux que vous créez en Perl :

```
special4.pl
#!/usr/bin/perl
foreach (@ARGV){
    print $_;
}
```

Developpez Magazine

Dans ce simple code, nous voulons afficher la liste des paramètres passés à notre programme. Comment passe t'on un argument ? Quand vous appelez votre script, il suffit d'écrire à sa suite les arguments :

```
perl special4.pl toto tata tutu
```

Ainsi, notre programme special4.pl vous affichera toto, tata puis tutu. Ceci est particulièrement utile quand vous passez à votre programme un nom de fichier :

```
special5.pl
#!/usr/bin/perl
foreach $file (@ARGV){
    if (-f $file){
        print "OK, $file existe.\n";
    } else {
        print "$file : $!\n";
    }
}
```

Enfin, il existe des listes associatives ou tables de hachages spéciales. Nous citerons ainsi %ENV qui contient ainsi tous les renseignements sur votre système :

```
special6.pl
#!/usr/bin/perl
print "Liste des informations de votre système
:\n";
while (($cle, $valeur) = each (%ENV)){
    print "$cle a comme valeur $valeur.\n";
}
```

Naturellement, cette table de hachage se manipule comme toute bonne table de hachage :

```
special7.pl
#!/usr/bin/perl
print "Votre processeur est un
$ENV{'PROCESSOR_IDENTIFIANT'} :\n";
```

IV. Ecrire des fonctions

Il est souvent bien utile de compartimenter notre code en des blocs qui auront des tâches bien spécifiques à remplir. Ces blocs sont appelés des fonctions. Ces fonctions peuvent prendre ainsi des arguments, traiter ensuite ces éventuels arguments et peuvent aussi rendre un résultat. Moralité, les informaticiens répondent à l'adage "Diviser pour mieux régner".

L'écriture d'une fonction n'est guère compliquée. Vous devez ainsi déclarer votre fonction précédée du mot clé sub (pour sub-fonction, en anglais sous-fonction), suivi ensuite d'une paire ouvrante-fermante d'accolades. Celles-ci définiront le corps de la fonction soit les instructions que votre fonction devra traiter. Prenons un exemple simple:

```
fonction1.pl
#!/usr/bin/perl
sub fonction{
    print "La fonction est appelée.\n";
}
#Corps principal de notre script
print "Début de notre script.\n";
fonction();
print "C'est tout pour aujourd'hui.\n";
```

Ainsi, notre script débutera toujours par son corps principal. En l'occurrence, notre code affichera d'abord la chaîne de caractères "Début de notre script.\n". Puis, nous appelons notre fonction par son nom suivi d'une paire de parenthèses. Ces parenthèses contiendront les arguments passés à notre fonction. Ici, nous avons rien à transmettre donc, nous laissons notre paire de parenthèses vide. La fonction ainsi appelée va donc faire ce pourquoi elle a été créée : afficher la chaîne de caractères "La fonction est appelée.\n". Enfin, le corps principal de notre fonction reprend la main et affiche la dernière chaîne "C'est tout pour aujourd'hui.\n".

Nous avons vu que nous pouvons transmettre des arguments à une fonction via la paire de parenthèses invoquées lors de l'appel. Ainsi, prenons l'exemple suivant :

```
fonction2.pl
#!/usr/bin/perl
sub inverse{
    if ($_[0] == 0){
        print "Division par 0 impossible !\n";
    } else {
        $resultat = 1/$_[0];
        print "L'inverse de $_[0] est $resultat.\n";
    }
}
print "De quel nombre voulez-vous avoir l'inverse
?\n";
$nombre = <stdin>;
chomp $nombre;
inverse($nombre);
```

Dans ce code, nous demandons à l'utilisateur de rentrer un nombre. Celui-ci est stocké dans la variable scalaire \$nombre. Puis, cette scalaire \$nombre sera transmise à la fonction inverse dans la paire de parenthèses. venons-en à notre fonction inverse. Notre nombre est contenu dans une variable nommée \$_[0] qui est testé dans un if pour vérifier si, bien entendu, ce nombre n'est pas égal à 0. Et oui, nous avons une nouvelle variable spéciale ! Cette variable est en réalité le tableau @_ qui contient tous les paramètres transmis à une fonction. Le tableau spécial @_ peut aussi bien contenir des variables scalaires, des nombres, des chaînes de caractères et même des tableaux et des tables de hachages. Vous pouvez tout transmettre à une fonction. Dans le cas où notre valeur est égale à 0, la fonction répond par un message d'erreur. Dans le cas contraire, elle procède au calcul de l'inverse et renvoie le résultat.

Une fonction peut très bien renvoyer un résultat qui soit une variable scalaire, une liste ou une table de hachage. Reprenons notre calcul d'inverse :

```
fonction3.pl
#!/usr/bin/perl
sub inverse{
    if ($_[0] == 0){
        print "Division par 0 impossible !\n";
    } else {
        $resultat = 1/$_[0];
        return $resultat;
    }
}
print "De quel nombre voulez-vous avoir l'inverse
?\n";
$nombre = <stdin>;
chomp $nombre;
$reponse = inverse($nombre);
print "L'inverse de $nombre est $reponse.\n";
```

Dans le corps principal de notre script, vous remarquerez que nous demandons à ce que le résultat de la fonction soit contenu dans la variable scalaire \$reponse. Notre fonction va donc tester la valeur de son argument (est-elle égale à 0 ou non) et dans le cas où celui-ci n'est pas égal à 0, la fonction procède au calcul de l'inverse. Ce calcul est contenu dans la variable \$resultat. Puis, la fonction va retourner la valeur de \$resultat au corps principal de notre script par l'intermédiaire du mot clé return suivi de la variable à retourner (return se traduit par retourne en anglais). Notre script reprend la main avec la valeur \$reponse égale au calcul de l'inverse du nombre demandé à l'utilisateur et nous affichons le résultat.

Cette manière d'écrire son code rend plus simple la lecture et permet aussi au programmeur de réutiliser des fonctions suffisamment génériques dans d'autres codes. D'où un intérêt dans la simplification de résolution d'un problème et une portabilité des solutions apportées.

Retrouvez la rubrique Perl sur : <http://perl.developpez.com>
Developpez.com

C/C++

Les derniers tutoriels

Créez votre propre moteur 3D en C++.

6ème partie : la gestion du texte

L'affichage de texte à l'écran est une fonctionnalité qui peut se révéler très utile pour la suite du développement : affichage d'informations diverses et de debugging, console, interface graphique, ... Nous verrons donc dans cette partie une gestion efficace du texte via les polices graphiques.

Lire le cours de **Laurent Gomila** ([lien 21](#))

Les structures de données en C,

quatrième partie : les files

Dernière structure de donnée basée sur les listes (enfin !), il s'agit des files. Pour nous faciliter la vie, et ne pas réinventer la roue, nous allons utiliser la bibliothèque de listes doublement chaînées pour créer celle des files.

Lire le cours de **Nicolas Joseph** ([lien 22](#))

Les incompatibilités entre le C et le C++

Le C et le C++ sont deux langages proches au niveau de leur syntaxe et de leur grammaire (si l'on met de côté les propriétés objet du C++), cependant il existe un certain nombre d'incompatibilités qui fait qu'un programme écrit en C peut ne pas compiler avec un compilateur C++.

Lire le cours de **Nicolas Joseph** ([lien 23](#))

Critique de livre

C en action - Solutions et exemples pour les program-meurs en C

Yves Mettier
Editions O'Reilly



Livraison gratuite sur Amazon



ISBN: 2841772896
540 pages
juin 2005

Résumé éditeur

C en action rassemble une sélection de solutions efficaces aux problèmes les plus souvent rencontrés par les programmeurs en C. À la fois source d'idées, moyen d'apprendre C par l'exemple ou de programmer plus vite et plus facilement, cet ouvrage est construit sur le modèle : un problème, une solution, une discussion.

Critique

L'auteur du livre annonce la couleur dès les premières pages : ce livre est écrit pour des lecteurs ayant des connaissances en C : aucune initiation n'est proposée, aucun résumé des fonctions de la bibliothèque standard n'est inclu en annexe. Le lecteur est censé connaître et

surtout maîtriser le langage C. C en action vous propose de passer en revue différents points techniques qui reviennent régulièrement lors de développement en C : gestion de la mémoire, manipulations des chaînes de caractères, les structures de données... Mais aussi des rubriques plus inattendues telles que la compression ou l'utilisation de la libxml2. Chaque sous chapitre est décomposé en trois parties : Problème/Solution/Discussion de telle sorte que le livre se lit aussi bien de la première à la dernière page que comme un recueil de sources prêtes à l'emploi. On appréciera aussi la présence d'une section "Voir aussi" à la fin des chapitres qui renvoie à un site internet ou à des pages man.

L'auteur ne s'arrête pas au C-ANSI, la plupart des chapitres contient un exemple en C-ANSI, un basé sur la glib et un dernier avec les fonctions de la norme POSIX (il y a des chapitres, celui des signaux par exemple, valable uniquement pour des systèmes d'exploitation conformes POSIX)

On retiendra de ce livre la qualité des sources proposées, qui démontre une maîtrise certaine du C par l'auteur, ainsi que leurs abondances, qui rythme de façon agréable chaque chapitre.

Nicolas Joseph

Retrouvez la critique complète ([lien 24](#))

Les F.A.Q

- F.A.Q C : 121 Q/R
- F.A.Q C++ : 238 Q/R
- F.A.Q Visual C++ : 300 Q/R
- F.A.Q C++ builder : 400 Q/R ([lien 25](#))

Les sources

67 codes sources libres pour C++ Builder à télécharger ([lien 26](#))

Les compilateurs

Les meilleurs compilateurs gratuits pour Windows, DOS et Unix à télécharger. ([lien 27](#))

Retrouvez la rubrique C/C++ sur : <http://c.developpez.com>

Access

Les derniers tutoriels

Communication entre Access et Excel

Microsoft Excel est le tableur le plus utilisé du marché, il ne requiert pas de compétences de programmation et est d'une souplesse incomparable en matière de graphiques et de présentations.

C'est pourquoi, les applications Access les plus intelligemment conçues ont parfois recours à des échanges avec Excel.

Dans ce tutorial, nous allons aborder les différentes méthodes de communications

entre les deux produits et étudier à quels cas de figures elles s'appliquent le mieux..

Lire le cours de **Cafeine** ([lien 28](#))

Repousser les limites des formulaires Access en mode continu

Challenge de programmation : réaliser un formulaire continu affichant des images externes.

Lire le cours de **Cafeine** ([lien 29](#))

Une petite fonction de cryptage en VBA

La recherche de la sécurité, toujours plus de sécurité. Vous connaissez certainement César et son cryptage tout simple. Mais Vigenère, vous en avez déjà entendu parler ? Je vous propose ci-après de découvrir un cryptage très simple à réaliser.

Lire le cours de **Maxence HUBICHE** ([lien 30](#))

Retrouvez la rubrique Access sur : <http://access.developpez.com>

.NET

Les derniers tutoriels

VB.Net 2005 : le nouveau namespace nommé My

Visual Basic, dans sa version 2005, introduit un nouveau namespace nommé My. Ce namespace a pour caractéristique d'être un point d'accès rapide à tout un tas de classes du Framework.

Lire le cours de **Ronald VASSEUR** ([lien 8](#))

A la découverte d'Avalon

Le développement avec Avalon et XAML, comment cela marche ?

Lire le cours de **Jean-Alain Baeyens** ([lien 9](#))

Créer une application web .NET avec XMLRAD

XMLRAD est une alternative à ASP.NET pour créer des applications web avec .NET. La technologie est basée sur XML/XSL pour générer les pages HTML.

Lire le cours de **Jean-Philippe Bempel** ([lien 10](#))

A la découverte de Visual Studio 2005

Cet article présente les nouvelles fonctionnalités de Visual Studio 2005

Lire l'article de **MORAND Louis-Guillaume** ([lien 11](#))

Le déploiement facile d'applications .NET avec ClickOnce

ClickOnce est la nouvelle technologie de déploiement d'applications. Cet article vous montrera les différentes possibilités que cette technologie vous apporte, tant au niveau du déploiement que de la mise à jour de vos applications..

Lire le cours de **LEBRUN Thomas & MORAND Louis-Guillaume** ([lien 12](#))

SQL Server 2005 : procédures stockées en .Net

Au cours de cet article, nous allons voir les apports d'une innovation majeure de SQL Server 2005, à savoir l'intégration de la CLR 2.0. Nous verrons comment créer et déployer du code .Net directement à l'intérieur du serveur de base de données.

Lire le cours de **Ronald Vasseur** ([lien 13](#))

Les outils

Convertisseur de code C# <-> VB.NET disponible gratuitement en ligne. ([lien 14](#))

Retrouvez la rubrique .NET sur : <http://dotnet.developpez.com>

Actualité

- Résultats de l'enquête de satisfaction de MSDN France ([lien 16](#))
- Framework 2.0 et VS2005 final dispo en novembre ([lien 17](#))
- 101 exemples de code .Net 2.0 sur le site de Microsoft ([lien 18](#))
- Testez Visual Studio 2005 gratuitement ! ([lien 19](#))
- Principes des webservices selon Microsoft ([lien 20](#))

.NET

ClickOnce: Le déploiement facile d'applications .NET

ClickOnce est la nouvelle technologie de déploiement d'applications.

Cet article vous montrera les différentes possibilités que cette technologie vous apporte, tant au niveau du déploiement que de la mise à jour de vos applications. **LEBRUN Thomas & MORAND Louis-Guillaume**

Introduction

Le déploiement est aujourd'hui au milieu de toutes les stratégies dans une entreprise. Que cela soit le déploiement d'une mise à jour, le déploiement d'un logiciel (antivirus, ou autre) ou n'importe, le déploiement est une problématique et l'objectif principal de l'administrateur réseau est de le rendre le plus simple possible.

La plupart d'entre-vous doit connaître Windows Update qui permet à partir d'une page web de "deviner" les mises à jour qu'il est nécessaire d'installer, et peut-être connaissez-vous SUS (Software Update Service) qui permet de déployer automatiquement des mises à jour sur les postes du réseau.

Aujourd'hui, Microsoft nous propose une nouvelle merveille, qui combine les avantages des deux technologies citées juste avant. Avec ClickOnce, vous pourrez redistribuer vos applications .Net chez les clients qui en auront le besoin.

1. Présentation de ClickOnce

Disponible depuis la version 2.0 du framework .NET, ClickOnce est la nouvelle technologie de déploiement (et de mise à jour) d'applications. Elle se base sur le protocole HTTP pour effectuer les installations ou les mises à jour.

Le déploiement d'applications, au moyen d'HTTP, est disponible depuis la version 1.0 du framework .NET, mais il a subi de nombreuses modifications qui l'ont rendu plus performant. Pour mettre en place ClickOnce, il vous suffit de posséder:

- Visual Studio 2005 (actuellement en BETA 2) pour développer votre application
- un serveur Web IIS, pour publier votre application

Avec les précédentes versions du framework .NET, exécuter une application pouvait se faire via un simple lien dans une page Web:

```

Cliquer <a href="MonProjet.exe">ici</a> pour
télécharger et exécuter le logiciel
<b>MonProjet.exe</b>
    
```

Bien qu'efficace, cette technique présentait tout de même des inconvénients:

- le framework .NET devait être installé sur les machines cibles
- l'application est disponible "Offline" seulement si l'utilisateur choisit "Travailler en mode déconnecté" dans son navigateur Web
- le fichier .config de l'application n'est pas disponible par défaut
- l'application ne possède pas de raccourci sur le bureau ou dans le menu "Démarrer"

Si vous aviez besoin de plus de contrôle sur le déploiement ou la mise à jour de votre application, vous pouviez utiliser l'Updater Application Block-Version 2.0": il s'agit d'un composant, développé par Microsoft, qui peut détecter, télécharger, et mettre à jour votre application.

Cette technique, très efficace mais possédant malgré tout quelques "failles" (par exemple, le code exécuté possède les droits "Full Trust", ce qui est potentiellement dangereux, d'un point de vue sécurité, etc.) existe toujours mais laisse sa place, dans la version 2.0 du Framework .NET, à ClickOnce.

Pour faire simple, on peut dire que ClickOnce possède tous les

avantages de l'UAB (plus quelques autres), sans les inconvénients. Voici une liste des caractéristiques (non exhaustive) de ClickOnce:

- les mises à jour sont transactionnelles (c'est-à-dire qu'elles sont faites entièrement ou pas du tout)
- bonne intégration avec Visual Studio
- il est possible de proposer le téléchargement de pré-requis, nécessaires au bon fonctionnement de l'application
- il est possible d'avoir un raccourci dans le menu "Démarrer"

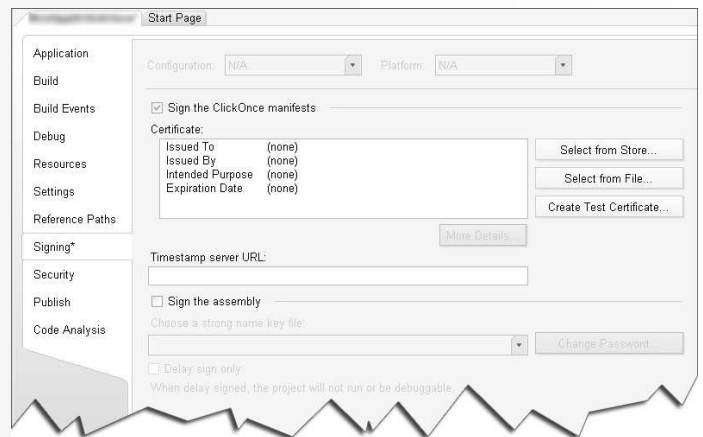
Maintenant que nous avons vu la théorie, passons un peu à la pratique: rien de tel pour apprendre ;)

2. Déploiement d'une application

2.1. Premier déploiement

Au cours de cette démonstration, et pour que l'explication soit la plus claire possible, nous prendrons un cas pratique réel, plus précisément, nous déploierons une application et ses différentes mises à jour.

Créer une application Winform basique. Compilez la, pour que nous puissions la déployer. Pour cela, veuillez ouvrir le panneau des propriétés de votre projet (menu Projet > nom_du_projet propriétés). Cochez la case et signez le manifeste ClickOnce.



Nous devons maintenant définir les autorisations de sécurité. Pour le moment, cochez la case qui indique que notre application est "full trust" (on peut lui faire totalement confiance). Par mesure de sécurité, il faudra mieux éviter de choisir cette option; nous verrons donc un peu plus tard dans ce tutoriel comment définir au mieux ces options de sécurité.

Passons maintenant à la partie la plus importante: la publication. Passons donc dans l'onglet publication, et définissez l'endroit où sera publié votre application ainsi que la page d'installation de votre application.

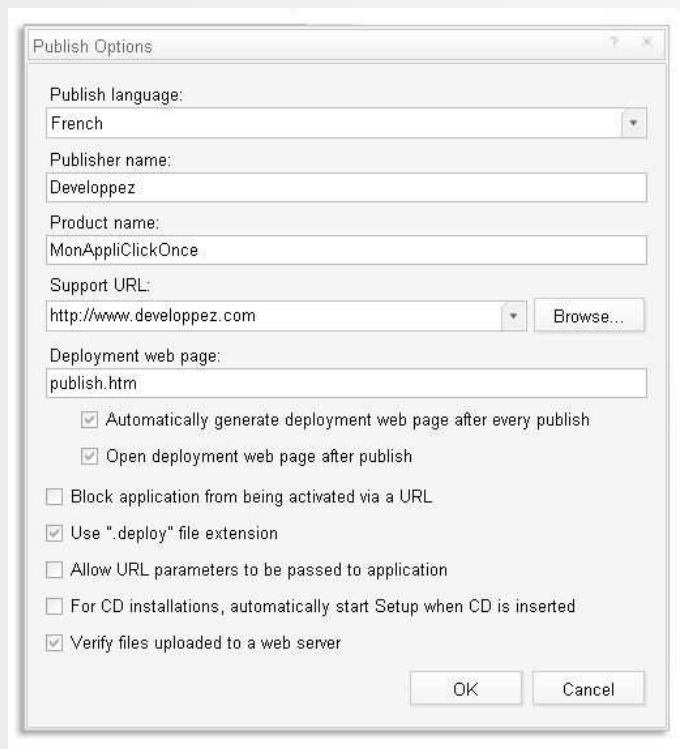
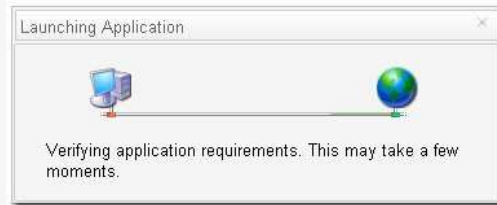
Sur cette page, vous allez définir si votre application sera publiée sur un ftp, un partage réseau ou un serveur IIS et vous pouvez également définir si l'application ne sera accessible que de façon online (retéléchargée à chaque fois) ou de façon offline, téléchargée, installée et accessible via le menu Démarrer.

- Publishing Location indique le partage Web depuis lequel votre application sera déployée/mise à jour
- Install Mode and Settings permet de spécifier si votre application

Developpez Magazine

doit être disponible "OnLine" et "OffLine"

- Prerequisites vous permet de spécifier les programmes pré-requis pour installer votre application
- Updates vous permet d'indiquer que votre application doit (ou non) vérifier la présence de mise à jour sur le serveur. Vous indiquez également à quel moment vous souhaitez faire cette vérification (avant ou après le démarrage de l'application)
- Grâce au panneau Options, vous pouvez renseigner différentes informations



2.3. Mise à jour automatique

Pour faire une mise à jour de votre application, rien de plus simple: il vous suffit faire vos modifications et de refaire un "Publish", dans Visual Studio.

Ensuite, exécuter votre application et dès le démarrage, celle-ci détectera la présence d'une nouvelle version, la téléchargera et relancera l'application pour prendre en compte les modifications.



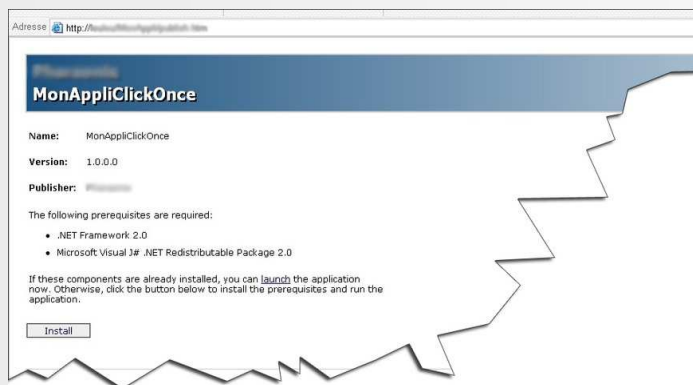
3. Utilisation avancée

Retrouver cette dernière partie et toutes les illustrations sur notre site Developpez.com ([lien 15](#)).

Une fois ces informations renseignées, un clic sur le bouton "Publish Wizard" (également disponible depuis le menu "Build" ou bien, dans l'explorateur de solution, via un clic droit) lance l'assistant de configuration de l'installateur ClickOnce.

Le "Publish Wizard" vous permet de configurer toutes les options nécessaires au déploiement de votre application.

Un clic sur le bouton "Finish" vous permet de déployer votre application sur le serveur IIS et vous ouvre la page de publication, qu'il vous suffit d'envoyer à vos utilisateurs pour qu'ils installent votre application:



2.2. Installation

Il ne nous reste plus qu'à installer l'application sur le poste client. Cette installation se fait manuellement grâce à la page publiée précédemment.

Un clic sur le bouton "Install" vous permet de lancer l'installation de l'application sur le poste et on aperçoit bien la vérification de la présence de mise à jour, juste avant le lancement:

Conclusion

ClickOnce est la nouvelle technologie de déploiement et de mise à jour qui possède de beaux jours devant elle: son utilisation simple et très agréable nous permet d'affirmer que ce sera sans doute la technologie qui remplacera, à terme, la technologie des MSI.

LEBRUN Thomas & MORAND Louis-Guillaume

Les auteurs de cet article sont respectivement *Microsoft CERTIFIED Application Developer* et *Microsoft CERTIFIED Professional*.



SGBD

Les derniers tutoriels

Correspondance des fonctions des divers SGBDR

Comparatif qui vous permettra traduire plus aisément une fonction d'un SGBD à l'autre..

Lire le comparatif de **Fabien Celaia** ([lien 31](#))

Les fonctions sur chaîne de caractères en Transact-SQL

Fonctions de traitement évoluées en Transact-SQL.

Lire le cours de **Igor Nikiforov** ([lien 32](#))

La gestion des clusters dans les divers SGBDR

Comparatif des solutions de clustering de divers éditeurs.

Lire le cours de **Fabien Celaia** ([lien 33](#))

Cache Buffer Chain

Comprendre les mécanismes internes de verrouillage d'Oracle

Lire le cours de **Jaouad** ([lien 34](#))

TKPROF et Explain Plan

Comment tracer une session ou une instance et interpréter le résultat avec l'outil TKPROF sous Oracle.

Lire le cours de **Jaouad** ([lien 35](#))

Des blocs multi-lignes au look "Java Swing JTable"

Plugin Oracle Forms

Lire le cours de **SheikYerbouti** ([lien 36](#))

Retrouvez la rubrique SGBD sur : <http://sgbd.developpez.com>

Nouvelle rubrique InterBase/Firebird (SGBD)

- Accueil : <http://interbase.developpez.com>
- La F.A.Q (55 Q/R) : <http://interbase.developpez.com/faq/>
- Les tutoriels : <http://interbase.developpez.com/cours/>
- Les outils : <http://interbase.developpez.com/outils/>

UML & méthodes

Le dernier tutoriel

La programmation complexe

La complexité d'un programme apparaît à un développeur dès que sa perception lui pose des problèmes de compréhension. La taille du projet est certainement un facteur important mais la complexité perçue dépend aussi de l'expérience et du savoir acquis par le programmeur.

La maîtrise de cette complexité a un caractère général et ne concerne pas uniquement la programmation. Le but de ce document est de s'intéresser à la manière dont on peut la gérer. Pour cela, on se propose de définir un modèle de comportement basé sur une Analyse Systémique que l'on définit à partir de quelques règles. Une section est consacrée au domaine de la programmation et une autre conclut ce document en présentant le concept d'informatique complexe.

Lire le cours de **Christophe Garnier** ([lien 37](#))

Retrouvez la rubrique UML sur : <http://uml.developpez.com>

Web

Les derniers tutoriels

Détection des collisions en Flash

Ce court article va vous expliquer ce qu'est une collision en flash, comment les détecter et dans quel cadre les utiliser. Vous devez disposer du flash player 7 pour pouvoir visualiser les animations. Il est disponible sur le site de Macromedia.

Lire le comparatif de **Stephane Eyskens** ([lien 38](#))

Débuter avec PHP et MySQL

Divers cours et astuces de PHP, MySQL.

Lire les cours de **Marian Muller** ([lien 39](#))

Retrouvez la rubrique Web sur : <http://web.developpez.com>

Pascal

Les ressources

- Cours et tutoriels : 9 cours
- Outils : 9 outils gratuits
- Exercices corrigés : 49 exercices
- Meilleurs livres : 4 critiques
- F.A.Q Pascal : 114 Q/R
- Codes sources : 16 sources
- Compilateurs : 14 outils testés et commentés



Retrouvez la rubrique Pascal sur : <http://pascal.developpez.com>

Delphi

Les derniers tutoriels

Strings et PChars en Delphi

Lorsqu'un ami qui écrivait des DLL renvoyant des chaînes de caractères passées sous la forme de PChar commença à trouver normal que Delphi lève des violations d'accès où renvoie parfois des valeurs incohérentes (mettant cela sur le dos d'une gestion des chaînes approximatives de Windows, bien sûr !), j'ai finalement décidé de terminer ce petit document destiné à éclaircir les idées de certains Delphistes. Il n'a pas vocation d'expliquer en totalité les mécanismes utilisés pour gérer les différents types de chaînes que vous pourrez rencontrer en tant que programmeur Delphi, mais

pourra vous éclaircir les idées et vous permettre d'éviter les erreurs courantes lors des conversions de chaînes Delphi en chaînes C.

Lire le cours de **Adrien Reboisson** ([lien 40](#))

Création et utilisation de DLL avec Delphi

Apprenez à créer simplement des DLL avec Delphi !.

Lire le cours de **Olivier Lance** ([lien 41](#))

Améliorer un composant par héritage, partie 2

Cette seconde partie va vous apprendre à améliorer un composant existant en héritant de celui-ci. En particulier, vous améliorerez le composant TImage pour lui ajouter une fonctionnalité de glisser-déposer.

Lire le cours de **Sébastien Doeraene** ([lien 42](#))

Utilisation du correcteur orthographique de Word

Utilisation du correcteur orthographique de Microsoft Word dans une application Delphi. Les exemples présentés ici sont testés avec Delphi 2005 et Word 2003.

Lire le cours de **Nono40** ([lien 43](#))

La F.A.Q Delphi

- 611 Q/R dont 64 nouvelles
- Téléchargeable au format HTML
- Moteur de recherche intégré

Vu dans la F.A.Q

Comment envoyer du texte au debugger ?

Il existe une fonction de l'API Win32, OutputDebugString, permettant d'envoyer du texte vers le debugger de Delphi, ces messages étant visibles dans le "Journal d'événements" (menu "Voir -> Fenêtres de débogage -> Journal d'événements" ou Ctrl+Alt+V). Cependant, cette fonction n'est pas très pratique à utiliser, car elle n'autorise que l'envoi d'une chaîne PAnsiChar "telle quelle". Une petite fonction de formatage s'avère rapidement nécessaire :

```
//La syntaxe de la fonction est strictement identique
// à celle de la procédure "Format".
Procédure DebugPrintFmt (
    Const FmtString : String ;
    Const Args : Array Of Const ) ;
Begin
    OutputDebugString (PAnsiChar (Format (FmtString, Args))) ;
End;
```

Retrouvez la rubrique Delphi sur : <http://delphi.developpez.com>



Liens

Vous trouverez ici la liste des URL des articles cités dans ce magazine. Pour des raisons pratiques de mise en page des URL longues, ces dernières sont fournies en fin de ce magazine. Ces URL sont actives : cliquez dessus pour accéder au document référencé.

1. <http://gfx.developpez.com/tutoriel/java/swing/drag/>
2. <http://gfx.developpez.com/tutoriel/java/swing/wait/>
3. <http://gfx.developpez.com/tutoriel/java/swing/java3d/>
4. <http://schmitt.developpez.com/tutoriel/java/jsf/introduction/>
5. <http://beuss.developpez.com/tutoriels/java/jakarta/log4j/>
6. <http://tahe.developpez.com/java/remote3tier-part1/>
7. <http://java.developpez.com/livres/#L2744019623>
8. <http://webman.developpez.com/articles/vbnet/2005/my/>
9. <http://jab.developpez.com/tutoriels/dotnet/avalon/>
10. <http://xmlrad.developpez.com/Articles/Application.NET/>
11. <http://lgmorand.developpez.com/dotnet/vs2005/>
12. <http://morpheus.developpez.com/clickonce/>
13. <http://webman.developpez.com/articles/sqlserver/sqlclr/>
14. <http://www.dotnet-developpez.com/Convertisseur/>
15. <http://morpheus.developpez.com/clickonce/>
16. <http://www.microsoft.com/france/msdn/enquete/satisfaction.mspx>
17. <http://www.microsoft.com/presspass/press/2005/jun05/TechEd2005Day2PR.mspx>
18. http://blog.developpez.com/index.php?blog=52&title=101_exemples_de_code_net_2_0_sur_le_site
19. http://blog.developpez.com/index.php?blog=50&title=testez_visual_studio_2005_gratuitement
20. http://blog.developpez.com/index.php?blog=20&title=principes_des_webservices_selon_microsof
21. <http://loulou.developpez.com/tutoriels/moteur3d/partie6/>
22. <http://nicolasj.developpez.com/articles/file/>
23. <http://nicolasj.developpez.com/articles/incompatibilites/>
24. <http://c.developpez.com/livres/#L2841772896>
25. <http://c.developpez.com/faq/>
26. <http://c.developpez.com/sources/bcb/>
27. <http://c.developpez.com/compilateurs/>
28. <http://cafeine.developpez.com/Excel/>
29. <http://cafeine.developpez.com/PseudoContinu/>
30. <http://mhubiche.developpez.com/Access/cryptage/vigenere/>
31. <http://fadace.developpez.com/sgbdcmp/fonctions/>
32. <http://nikiforov.developpez.com/>
33. <http://fadace.developpez.com/sgbdcmp/cluster/>
34. <http://oracle.developpez.com/guide/tuning/latch/>
35. <http://oracle.developpez.com/guide/tuning/tkprof/>
36. <http://sheikyerbouti.developpez.com/fjtable-v1/>
37. <http://garnier.developpez.com/cours/design/programmationComplexe/>
38. <http://stephaneey.developpez.com/tutoriel/flashmx2004/collision/>
39. <http://mmuller.developpez.com/?p=1>
40. <http://reisubar.developpez.com/tutoriel/delphi/strings-pchars/>
41. <http://olance.developpez.com/articles/delphi/creation-dll/>
42. <http://sjrd.developpez.com/delphi/delphitutoriels/composants/heritage/>
43. <http://nono40.developpez.com/tutoriel/delphi/word/correction/>