

QUINZE LEÇONS EN QUINZE MINUTES POUR ANIMER VOS PAGES HTML

par Yves Piguet, IA-DGM

Depuis un peu plus d'une année, quiconque se balade sur le Web a pu constater la prolifération d'animations de toutes sortes: logos, publicité, division de cellules, cartes météorologiques... Après l'apparition des images il y a quatre ans, les animations étaient la suite logique du «progrès» en matière de multimédia.

En ce qui concerne les images, deux formats se sont imposés: GIF, efficace pour celles comportant un nombre limité de couleurs, qui reste très répandu malgré certaines restrictions imposées par le propriétaire de son algorithme de compression LZW; et JPEG, qui autorise un fort taux de compression pour les photographies au prix d'une dégradation souvent imperceptible s'il est bien utilisé. Le format GIF possède un certain nombre d'options particulièrement intéressantes pour le Web. L'entrelacement consiste à changer l'ordre des lignes pour permettre l'affichage rapide d'une première ébauche suivie d'améliorations successives. La transparence de certains pixels donne l'apparence d'une

forme quelconque se détachant sur le fond de la page. Enfin, depuis que Netscape (suivi de Microsoft) a amélioré la compatibilité de son décodeur avec le standard GIF fin 1995, on peut créer des animations en ayant plusieurs images par fichier GIF. Le succès des animations GIF est considérable. Les avantages des GIF sont préservés: tous les serveurs et la plupart des browsers les supportent sans aucune configuration spéciale, et ils sont très faciles à créer et à placer dans un document HTML. Leurs inconvénients sont aussi conservés: pour des films vidéo, il vaut mieux utiliser MPEG ou QuickTime, dont le taux de compression est bien supérieur et qui permettent d'ajouter une piste sonore. Pour les GIF animés, les deux programmes les plus anciens sont encore les plus utilisés. Sur PC, il s'agit de GIF Construction Set (shareware), qui nécessite certaines connaissances de la structure interne du format GIF. Les utilisateurs de Mac utilisent pour la plupart

Suite en page 20

SOMMAIRE

- 1 Quinze leçons en quinze minutes pour animer vos pages HTML
- 2 Comment se procurer des logiciels pour Mac et PC à l'EPFL
- 3 LiveConnect
centre nerveux de la trilogie Java, JavaScript et Plug-ins
- 7 ASP, qu'est-ce que c'est? (Active Server Pages de Microsoft)
- 12 Réseaux informatiques et services Internet
- 16 L'outil Helpdesk de Sun Microsystems
- 17 Une carte PC dans votre Mac
- 23 CO4, une salle pour le téléenseignement
- 24 Publisher
le couteau suisse des logiciels de mise en page
- 26 Formation
- 29 Clavitudes
ou les acrobaties de PageMaker 6.5
- 32 Calendrier

PROCHAINES PARUTIONS

	parution FI	délai FI
SP	02.09.97	26.06.97
7	16.09.97	28.08.97
8	21.10.97	02.10.97
9	18.11.97	30.10.97
10	19.12.97	25.11.97

COMMENT SE PROCURER DES LOGICIELS POUR MAC ET PC À L'EPFL

par Fred-Ami Rougemont, SIC

RAPPEL

LOGICIELS D'INTÉRÊT GÉNÉRAL

Par logiciel d'intérêt général, nous entendons tout logiciel dont les commandes sont d'au moins une dizaine pour une période de 3 mois (si ce n'est pas le cas, il s'agit d'un produit spécifique!). Le SIC fournit ces logiciels.

Nous n'avons pas de licences de site (nombre de copies ne devant pas être dénombrées) pour les logiciels PC ou Mac à l'EPFL, à part quelques cas bien particuliers (Eudora, McAfee,...). Par contre, la plupart du temps, nous achetons des droits de copies. Mais pour cela, nous devons bien évidemment connaître le nombre de copies faites à l'EPFL afin d'effectuer les commandes (paiement à l'éditeur)... et ceci jusqu'à ce jour, ce qui n'est pas toujours facile (pour nous!).

Cas des Produits MICROSOFT

La situation des licences concernant les produits MICROSOFT à l'EPFL est la suivante:

- pour les produits OFFICE (WORD, EXCEL, POWERPOINT), nous avons une licence correspondant au nombre de machines (Mac et PC) sur le site de l'EPFL. Chacun (chaque personne employée à l'EPFL et utilisant du matériel fourni par son cher employeur!) peut donc installer une copie (ou se mettre à jour) sans aucune formalité en se connectant sur les serveurs Olympe ou Cyclope.
- pour le produit OFFICE PRO, c'est-à-dire avec ACCESS, il y a lieu là aussi d'acheter des droits de copie.
- pour le système Windows NT, nous avons un contrat de fourniture de licence, et périodiquement nous devons indiquer le nombre de licences utilisées. Au cas où vous auriez l'intention de vous procurer une telle licence, il y a lieu encore une fois de nous informer.
- pour Windows 95 (bientôt 97!), nous avons acheté une mise à jour pour les ordinateurs qui fonctionnaient sous DOS. La version de Windows 95 de l'été 95 peut donc être utilisée. Malheureusement, Microsoft ne nous a pas encore indiqué comment utiliser des distributions plus récentes qui demandent des numéros de série.

Pour le matériel neuf, la licence Windows 95 doit être fournie par le vendeur de l'ordinateur.

PRODUITS SPÉCIFIQUES

Dans le cas d'un logiciel spécifique à une unité, cette dernière le commande directement (sur ses crédits logiciels OCFIM) à l'aide d'une demande d'achat à l'OCFIM.

LA BONNE NOUVELLE

Les procédures ne paraissant pas toujours simples... nous allons tendre ces prochains mois à centraliser l'accès à tous les logiciels de bureautique.

En commençant par la plate-forme PC, ainsi que partiellement pour la plate-forme Mac, nous allons introduire un service de distribution fonctionnant entièrement par le réseau dès le début du mois de juin:

- tous les logiciels (d'intérêt général, enfin presque tous...) seront distribués via le réseau... plus de disquettes ou CD-ROM (ceci est une politique de l'éditeur afin de diminuer les coûts des logiciels à l'EPFL);
- pour le monde Mac, à partir de l'URL ci-dessous, vous accéderez ou serez informés, en fonction de l'évolution du service, sur la manière d'accéder aux logiciels (ou de nous indiquer le nombre de licences);
- l'information sur la commande de logiciels spécifiques sera disponible par ce même service;
- pour commander un logiciel (n'importe lequel et y compris les logiciels de Microsoft) un service de distribution de logiciels sous WWW se trouvera donc depuis le début juin à l'adresse suivante:

<http://pcline.epfl.ch/pc/log/distrilo.htm>

Ce service doit être utilisé au moyen d'un lecteur WWW (MS iExplorer ou Netscape) afin de nous permettre de connaître le nombre de copies utilisées à l'EPFL et d'automatiser les commandes ET les distributions.

Par ce service, d'un seul clic (ou presque), vous nous signalerez votre désir de faire une copie d'un logiciel ET vous vous le procurerez!

L'extension de ce service pour monde Mac sera signalée à l'URL ci-dessus.

Et pour les questions: rougemont@sic.epfl.ch ■

LIVECONNECT

CENTRE NERVEUX DE LA TRILOGIE JAVA, JAVASCRIPT ET PLUG-INS

par Francis Lapique, SIC

INTRODUCTION

Tout le monde connaît Netscape dans sa fonction de navigateur. On sait aussi qu'il est capable d'interpréter des *applets* Java. Mais, on connaît mal l'architecture LiveConnect qui met en scène d'autres acteurs: JavaScript et tout module extérieur sous sa forme *plug-in* pour générer des pages d'une nouvelle dimension. LiveConnect peut potentiellement transformer vos pages en véritables applications qui se caractériseront par un haut degré d'interactivité.

Le cœur de LiveConnect est JavaScript qui se présente comme un langage interprété très fortement couplé au *browser*. JavaScript peut faire appel aux services d'*applets* ou fonctions fournies par un module extérieur.

Des exemples concrets, après une brève introduction à JavaScript, vont nous montrer comment les choses fonctionnent. Avant toute chose il faut s'assurer que LiveConnect est actif: pour activer LiveConnect choisir *Network Preferences* sous le menu *Options* de Netscape, puis cochez sous *languages* les options *Java* et *JavaScript*.

JAVASCRIPT

Il n'y a pas de programme JavaScript en tant que tel. Un code JavaScript est intimement lié à la page HTML qui l'héberge. Ce n'est pas à proprement parler un langage orienté-objet. C'est un langage simple qui manipule des objets. On va appliquer sur ces objets des méthodes. A titre d'exemple, en appliquant la méthode *write()* sur l'objet *document* on va commander l'affichage d'une page. Ces objets ont des *properties* comme *bgColor*, *title*, *location*, etc. Ils sont également sensibles à un certain nombre d'événements comme *onLoad* qui sera émis au chargement d'une page.

Voici un des script les plus simples que vous pouvez écrire en JavaScript

```
<html>
<script language="JavaScript">
document.write ("Hello World");
</script>
</html>
```

Les tags `<script>` et `</script>` marquent respectivement le début et la fin de ce script. En invoquant cette page on va appliquer la méthode *write ("Hello World")* sur l'objet *document*. Le *browser* concrétise cette action

en affichant la chaîne de caractères "Hello World" dans la page HTML. C'est un mauvais exemple d'utilisation de JavaScript qui se substitue à HTML pour diffuser une page statique.

L'exemple qui suit est meilleur, il montre comment le contenu d'une page peut changer en fonction du jour de son chargement. On instancie un objet *Date* à travers l'opérateur *new*. En appliquant la méthode *getDay()* sur notre objet *today* on va différencier le contenu de la page.

```
<html>
<title> My Web Page </title>
<script>
//Get the current date
var today = new Date();
//Print an initial welcoming message for
this document
document.writeln ( "Welcome to "+
document.title + "." );
//Depending on the day of the week, print
// an additional message
if ( today.getDay() == 1 )
document.write ("Welcome back!");
if ( today.getDay() == 5 )
document.write ("Have a great weekend." );
</script>
</html>
```

UTILISATION D'UNE FONCTION

Dans ces deux premiers exemples on déroule le code du script au chargement de la page. Mais on peut très bien déclarer le code d'un script, sous la forme d'une fonction, dans l'en-tête de la page, à savoir entre les tags `<head>` et `</head>` puis invoquer cette fonction au moment voulu. Pour faire coïncider ce moment avec celui du chargement de la page on va indiquer dans le tag `<body>` que l'on associe l'événement `onLoad` à l'appel de la fonction *sayHello()* `<body onLoad="sayHello()"` Voici l'exemple complet:

```
<html>
<head>
<title> My Web Page </title>
<script language="JavaScript">
// Declare a function
<!-- hide
function sayHello()
{
// Get the date
var today = new Date();
var errWind =
window.open("", "errors", "HEIGHT=200,WIDTH=400")
// Print the initial message
```

```
errWind.document.write("<HTML><BODY
  BGCOLOR=RED>" + "<H1>Bienvenue sur mon
  serveur</H1>" + "</HTML>");
// Select the last part of the message based
// on the date.
if(today.getDay() == 1)
errWind.document.write("<HTML><BODY
  BGCOLOR=RED>" + "<h2>A bientôt!</h2>" +
  "</HTML>");
if(today.getDay() == 5)
errWind.document.write("<HTML><BODY
  BGCOLOR=RED>" + "<h2>Bon weekend</h2>" +
  "</HTML>");
}
// ->
</script>
</head>
<body onLoad="sayHello()">
</body>
</html>
```

Dans un dernier exemple nous allons faire appel au service d'un module extérieur que nous introduisons par le tag `<embed>`. Avant tout nous vérifions la présence de ce module au moyen de la méthode `mimeTypes` que l'on applique sur l'objet `navigator`:

```
<html>
<script language="JavaScript">
<!-- Hide from non-Javascript browsers
var plugin = navigator.mimeTypes ["video/
quicktime"];
if ( plugin != null )
  document.write ( "<embed src=movie.qt
width=200 height=200>" )
else
  document.write( "No QuickTime plugin
found!" )
//-->
</script>
</html>
```

En cas de problème on diffuse un message d'alerte pour signaler l'absence du *plug-in*.

Les tags `<applet>` et `<embed>` ayant un *attribut name*.

```
<applet code=Hello.class name=hello
width=200 height=50>
<embed src=data.dg name=population width=400
height=400>
```

on peut désigner leur nom depuis JavaScript au moyen de la syntaxe:

```
var hello = document.applets["hello"]
document.embeds["population"]
```

ACCÈS À DES APPLETS JAVA DEPUIS JAVASCRIPT

Pour illustrer ce premier mécanisme de communication entre Java et JavaScript, considérons le code Java suivant:

```
import java.applet.*;
import java.awt.Graphics;
public class Hello extends Applet {
  String str = "Hello, World";
  int x = 25;
  int y = 25;
  public void paint ( Graphics g ) {
```

```
    g.drawString ( str, x, y );
  }
  public void setString ( String newstr )
  {
    str = newstr;
    repaint();
  }
  public String getString() {
    return str;
  }
}
```

Nous y trouvons trois méthodes qui sont *paint*, *setString* et *getString* et qui vont respectivement afficher, charger/afficher et retourner une chaîne. Le champ protégé *str* compte une valeur initiale.

On intègre cet *applet* au document en lui donnant le nom `hello`:

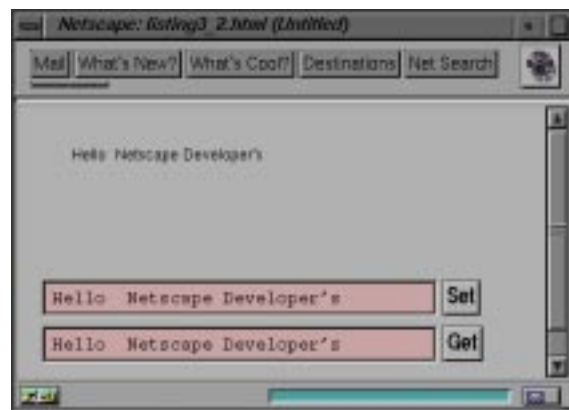
```
<applet name=hello
  code=Hello.class
  width=200 height=100>
</applet>
```

On va établir un formulaire qui va dialoguer avec notre *applet hello*. Ce formulaire compte deux champs texte *In* et *Out* qui vont respectivement dialoguer avec les méthodes *setString(String)* et *getString()*. Pour ce qui concerne le champ *In* on associe l'événement *onclick* à la méthode *setString(String)* de *hello* de la façon suivante:

```
<input name="In" type=text size=35>
<input type=button value="Set"
onclick="document.hello.setString(form.In.value)">
```

Pour le champ *Out* on affecte son contenu *form.Out.value* à la valeur de retour de la méthode *getString()*

```
<input name="Out" type=text size=35>
<input type=button value="Get"
onclick="form.Out.value=document.hello.getString()">
```



En activant le bouton *Set* vous demandez à l'*applet hello* de prendre en charge l'affichage du champ *In*, en activant *Get* vous affichez dans le champ *Out* la valeur courante de la chaîne de *str*.

ACCÈS À JAVASCRIPT DEPUIS JAVA

Pour communiquer, depuis Java, avec JavaScript il vous faut importer `netscape.javascript` fourni par Netscape. Ce produit vous donne accès à la classe `JObject` qui vous permet de manipuler la hiérarchie

d'objets JavaScript.

En appliquant la méthode *getWindow()*, au chargement de votre *applet*, sur l'objet *JSObject* vous allez récupérer l'objet *window* qui contient votre *applet* :

```
import java.applet.*;
import java.awt.Graphics;
import netscape.javascript.JSObject;

public class Hello extends Applet {
    JSObject win;

    public void paint ( Graphics g ){
        g.drawString ("Hello World", 25, 25)
    }
    public void init()
    {
        win = JSObject.getWindow ( this );
    }
}
```

Il est plus que conseillé de faire cette opération dans la méthode *init()*.

Pour rendre tout ce mécanisme opérationnel il faut encore autoriser l'accès de l'*applet* aux objets JavaScript au moyen du paramètre *mayscript* qu'il faut mettre à vrai :

```
<html>
<title> My Web Page </title>

<applet code=Hello.class name=Hello
        width=200 height=100
        mayscript=true >
</applet>
</html>
```

Une fois cet objet *JSObject* connu vous pouvez récupérer par exemple le titre du document, ce qui se fait en deux étapes :

```
public boolean mouseUp ( Event e, int x, int
y )
{
    JSObject document;
    //Retrieve the window's document member

    document=(JSObject)win.getMember (" document" );
    //Get the title of the current document

    String title =
    (String) document.getMember ("title");
}
```

La méthode *getMember* retournant un *Object* notez son cast en *JSObject*.

Vous pouvez également appliquer des méthodes JavaScript. Imaginez que vous vouliez appliquer la méthode JavaScript *alert* sur un objet *JSObject*. Pour ce faire vous faites appel au service de la méthode *call* que vous appliquez de la manière suivante :

```
Object args[] = new Object[1];
args[ 0] ="Thanks for visiting";
win.call ( "alert", args);
```

Le premier argument de *call()* est le nom de la méthode JavaScript et le second un vecteur de type *java.lang.Object* pour passer les arguments.

LIVECONNECT CENTRE NERVEUX

Une page HTML peut se voir comme une collection d'objets capables d'émettre et de diffuser des messages. L'acheminement du message d'un objet à l'autre est pris en charge par LiveConnect d'où cette analogie de centre nerveux.

Dans le FI du 20 mai 97, Olivier Michel donne l'exemple d'une scène VRML, ensemble de nœuds qui envoient et reçoivent des événements, communiquant avec trois *applets*. La proposition de paquetage *vrml.external* qui est mis en jeu dans cet exemple est suffisamment intéressante pour que l'on en donne quelques détails. Imaginez une scène VRML avec une sphère dont vous souhaitez contrôler la couleur au moyen d'une *applet*. Pour faciliter l'explication voici le code de la scène VRML :

```
#VRML V2.0 utf8
Group {
  children [
    Shape {
      appearance Appearance {
        material DEF MAT Material {
          diffuseColor 0.8 0.2 0.2}
        geometry Sphere {}},
    ]
}
```

L'*applet* doit, dans un premier temps, prendre contact avec la scène VRML de la page HTML. Cette opération se fait en quatre temps :

```
JSObject win = JSObject.getWindow(this);
JSObject doc = (JSObject)
win.getMember("document");
JSObject embeds = (JSObject)
doc.getMember("embeds");
Browser browser = (Browser)
embeds.getSlot(0);
```

Une fois l'objet *embeds* atteint, on applique la méthode *getSlot* pour trouver le tag *<embed>* et notre scène VRML avec laquelle nous allons travailler. Notez le *casting* vers le type *Browser* qui est un objet du paquetage *vrml.external.Browser*.

Pour changer la couleur de la sphère il faut envoyer un message *set_diffuseColor* sur le nœud MAT. Ici l'opération se fait en trois temps :

- premier temps on obtient le nœud :
Node material = browser.getNode("MAT");
- deuxième temps on fait l'association message-nœud :
diffuseColor =
(EventInSFColor)
material.getEventIn("set_diffuseColor");
- troisième temps on applique la méthode *setValue* pour diffuser l'action sur le nœud :
val[0] = 0.2f;
val[1] = 0.2f;
val[2] = 0.8f;
diffuseColor.setValue(val)

Si on avait voulu recevoir un message venant de VRML, il aurait fallu définir un message de sortie, tout comme nous venons de le faire pour l'entrée, et implémenter la méthode *callback()* de l'interface *EventOutObserver*.

Les possibilités de visualisation et d'interaction de ces nouvelles technologies sont immenses. Tout est prêt pour coupler ces technologies avec celles des bases de données. Pour vous donner un avant goût vous pouvez charger l'exemple suivant:

<http://sgline.epfl.ch/LiveConnect/ex1.html>

En touchant ou en s'approchant d'objets que vous avez introduits dans une scène VRML vous déclenchez des interrogations dans une base de données qui vont produire un son, un fichier HTML ou les deux.



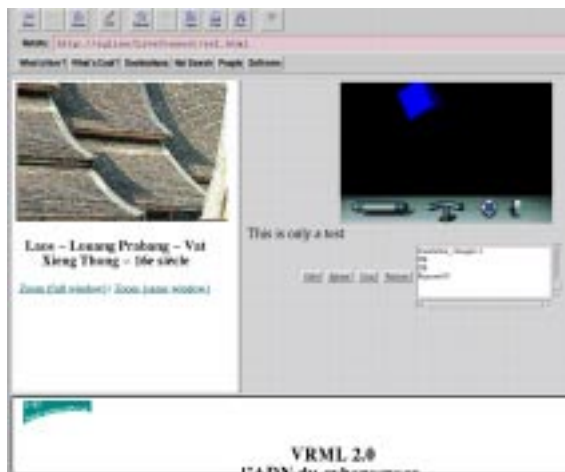
Si votre *browser* est correctement configuré (voir plus loin) la fenêtre du *plug-in* VRML et de *applet* doivent apparaître dans la partie supérieure droite de votre écran. L'*applet* se présente sous la forme de quatre boutons *Cube*, *Sphere*, *Cone* et *Remove* et d'une zone texte dans laquelle vont s'afficher des messages. La scène VRML ne comptant aucun nœud, nous avons un rectangle noir.



En activant le bouton *Cone* vous chargez dynamiquement un objet qui manifeste sa présence sous la forme d'un cône de couleur verte. En activant le bouton *Remove* vous faites disparaître cet objet. Vous pouvez tourner ce cône sur lui-même en le pointant et en déplaçant la souris. En cliquant sur le cône vous déclenchez une requête dans une Web DataBlade d'Illustra qui se manifeste par l'affichage d'une page HTML dans la fenêtre de gauche;

En cliquant le bouton *Sphere* vous faites venir une sphère rouge que vous pouvez déplacer horizontalement

et verticalement. Déplacez cette sphère sur la droite, relâchez la souris, vous déclenchez une requête Illustra et une autre en la déplaçant sur la gauche.



Le dernier nœud *Cube* est le plus intéressant. En vous approchant de lui suffisamment près vous déclenchez deux actions: l'affichage d'une URL et la diffusion d'un son. En vous éloignant de lui, vous arrêtez la diffusion du son et vous affichez une autre URL.

Vous avez un compte-rendu du dialogue *applet/plug-in* dans le champ texte de l'*applet*.

Pour visualiser l'exemple mettant en scène des objets VRML, il vous faut utiliser Netscape 3.0 ou plus et installer le *plug-in* CosmoPlayer 1.0.2 Beta3 (pour SGI) ou CosmoPlayer 1.0 Beta3 (pour Windows). En cas de problème, vérifiez que la variable d'environnement *CLASSPATH* est correctement définie:

sur SGI: `setenv CLASSPATH /usr/CosmoPlayer/classes`
 sur Windows95 et NT, il est parfois nécessaire de déplacer
`C:\java\classes\vrml\vrml vers`
`C:\Program`
`Files\Netscape\Navigator\Program\Java\classes\`

CONCLUSION

Netscape apparaît de plus en plus comme un espace de communication entre des objets de tout type. Cette collection d'objets communiquant à l'intérieur d'une page HTML fait que celle-ci s'apparente de plus en plus à une application ayant un haut degré d'interactivité. Le temps où nous aurons sur Internet autant d'applications que de pages HTML n'est pas très éloigné.

Biblio et internau -graphie

- pour débuter en JavaScript:
<http://rummelplatz.uni-mannheim.de/~skoch/js/index.htm>
- JavaScript Authoring Guide, la référence par Netscape
<http://home.netscape.com/eng/mozilla/2.0/handbook/javascript/index.html>
- JavaScript Bible, *Danny Goodman*, IDG Books
- Netscape Developer's: Guide to plug-ins, *D.A. Young*, Prentice Hall ■

ASP, QU'EST-CE QUE C'EST?

(ACTIVE SERVER PAGES DE MICROSOFT)

par Christian Zufferey, SIC

Dans l'excellent article de *Stéfane Bernel*, *Illustra en test* de l'automne passé, nous avons été *ébahis* par la facilité avec laquelle on pouvait fabriquer des pages HTML depuis une base de données comme par exemple:

```
<HTML>
<HEAD>
<TITLE>Test Illustra</TITLE>
</HEAD>
<BODY>
<H1>Index des Flash Informatique</H1>
<TABLE BORDER>
<TR><TH>No<TH>Auteur<TH>Titre
<?MISQL SQL=" select
ID, Titre, NomMail (Auteur), Ref from FI order
by Ref asc;">
<TR><TH>$1<TH>$(REPLACE, $3, NULL, )
<TH> <A
HREF=" Webdriver?MITab=FI&MIval=$4">$2</A>
<? /MISQL>
</TABLE>
<P>
</BODY>
</HTML>
```

En lisant les autres articles concernant ce sujet et en voyant à quelle vitesse les pages HTML statiques vieillissent, on comprend rapidement que l'avenir (sérieux) du WEB passera par des pages HTML *pondues au vol* via des bases de données. On a aussi rapidement *déchanté* quand on s'est aperçu du prix de la solution Illustra et de sa *relative* complexité à mettre en œuvre... bref, c'était un beau jouet, mais pas pour tout le monde. Mais l'idée était lancée.

Depuis pas mal de temps déjà, il y avait bien, Microsoft avec son *Internet Database Connector* comme solution boiteuse, mais ce n'était pas la panacée. Les vendeurs de bases de données (Oracle, Ingres et même Filemaker) avaient aussi une solution database/WEB, mais ce n'était toujours pas *le pied*... et enfin Billy est arrivé avec ses gros pieds et ASP!

Petite parenthèse, tout le monde croit que je suis pro Microsoft... hum... hum... en fait, je ne suis pas fondamentalement pro Microsoft, je suis pro solution simple, puissante, accessible à tous et bon marché. Et en l'occurrence, celui qui a assez d'argent pour *racheter* des solutions simples et puissantes et ensuite noyer le marché à des prix défiant toute concurrence, l'a fait... Voilà qui est dit.

Bref, fin de l'année 1996 Microsoft sort ASP.

ASP, qu'est-ce au juste? très peu connu (bizarre pour un produit Microsoft) même chez Microsoft France qui, ce printemps encore nous disaient *qu'ils ne connaissaient pas ce genre de produit* (véridique). J'écris donc cet article pour vous faire découvrir ce *merveilleux* produit si peu

connu. Avant de l'écrire, j'avais déjà rédigé un petit *cours flash* sur ASP, alors, vu la place limitée dans ces pages je ne peux que vous conseiller de jeter un petit coup d'œil sur <http://pcline.epfl.ch/pc/net/asp/coursasp.htm>.

ASP, QU'EST-CE?

C'est l'acronyme de **Active Server Pages**, qui à lui seul résume l'ensemble. C'est un serveur WEB de pages HTML actives.

ASP est un module supplémentaire ajouté au serveur IIS 2 (*Internet Information Server*) de Microsoft (serveur HTTPD). En fait il devient IIS 3 et fait partie intégrante de Windows NT 4 Server (d'où son relativement faible coût...). On peut aussi installer ASP sur le *Personal WEB Server* de Microsoft qui tourne sous Windows 95.

ASP ne peut pas tourner sans un serveur, souvenez-vous: **Active Server Pages**; donc il est exclu de lire des pages *.asp* depuis un CD-ROM par exemple.

LA PHILOSOPHIE ASP

ASP possède un interpréteur de commandes VBScript ou JScript et un module interface ODBC (Open Data Base Connectivity), qui permet de rendre actives des pages HTML. C'est-à-dire: les pages HTML ne sont plus statiques mais dynamiques, elles évoluent en fonction de certaines actions ou de certaines données. Et c'est tout, je répète pour les personnes qui n'auraient pas très bien entendu... un interpréteur de commande et une interface ODBC, mais **pas** de base de données. Oh, oh, on avait justement dit précédemment que c'était important d'avoir une base de données. C'est là que se trouve l'astuce d'ASP. ASP ne nous fige pas à une base de données soit *disant propriétaire*, il utilise toute base de donnée qui possède un connecteur ODBC. Et cela va d'un fichier ASCII ou Excel à une base de données de puissance modulable, de MS Access à Oracle en passant par MS SQL Server, on pourrait même envisager une base relationnelle-objet quelconque. Tout va se jouer avec l'ODBC qui est en fait un traducteur «base de données / application SQL».

De manière conventionnelle (âge de la pierre taillée), on utilisait une *moulinette* (généralement script PERL) générant la page HTML. Le squelette de la page HTML est alors le script, c'est le script qui produit les commandes HTML. Généralement le nom de l'URL se termine

par un **.pl**, ce qui permet au serveur (sous Windows NT) de traiter le fichier avant de l'envoyer.

Voici le script en PERL produisant une page HTML qui *convertit une monnaie dans une autre selon un certain taux*:

```
#!/usr/local/bin/perl
print "Content-type: text/html\n\n";
print "<title>essai conversion</title>\n\n";

$query = $ENV{ QUERY_STRING };
@fields = split (/&/, $query);
for $field (@fields) {
    next unless ($field =~ /=/);
    ($name, $value) = split (/=/, $field, 2);
    $CONTENTS{$name} .= ' ' if ($CONTENTS{$name});
    $CONTENTS{$name} .= "$value";
}
print "<h1>résultat</h1>\n\n";
print "la quantité $CONTENTS{ quant} au taux
de $CONTENTS{ taux} est " . $CONTENTS{ quant}
* $CONTENTS{ taux} . "<br> Merci.\n\n";
exit;
```

On se trouve dans un mode d'interprétation de commandes PERL qui produit de l'HTML au *vol...* et il est exclu d'ouvrir ce fichier avec un éditeur HTML pour en modifier la mise en page... tout doit être donc fait *à la main*.

On voit la *relative* difficulté de récupération des paramètres d'entrée de l'URL, ceux qui rendent justement la page HTML dynamique. (Mais attention, le PERL est un langage très puissant pour la gestion des chaînes de caractères, mais il n'est pas forcément accessible à tous en 5 minutes).

Voyons maintenant la même chose, mais avec un serveur ASP:

```
<html>
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<meta name="GENERATOR" content="Microsoft
FrontPage 2.0">
<title>Résultat de la conversion...</title>
</head>
<h1 align="center">Résultat de la conver-
sion...</h1>
<p>Alors, <%= Request.Form("t2") %> converti
au taux de change <%= Request.Form("t1") %>
font : <%= Request.Form("t1") *
Request.Form("t2") %>
</p><p>Merci.</p>
</html>
```

Ici, par contre, le squelette est un fichier en HTML et il y a des intrusions en commandes ASP. Ce fichier peut être traité sans autre avec un éditeur HTML pour sa mise en page.

On voit bien la *relative* difficulté de récupération des paramètres d'entrée de l'URL, ceux qui rendent justement la page HTML dynamique.

Le nom de l'URL se termine toujours par un **.asp**, ce qui permet au serveur de traiter le fichier avant de l'envoyer.

Le langage utilisé par l'ASP peut être actuellement (on peut très bien envisager d'autres langages) soit le VBScript ou le Jscript. Le VBScript sera préféré par les utilisateurs PC qui connaissent le Visual Basic (voire pour certains le Basic). Le Jscript lui sera préféré par les utilisateurs de C. On voit donc qu'il y en a pour tout le monde.

ATTENTION:

Il ne faut pas confondre le VBScript ou Jscript exécuté par le navigateur avec le VBScript ou Jscript exécuté par le serveur ASP. Dans le premier cas, il faut un navigateur qui soit capable d'exécuter lesdites commandes; dans le second cas, comme les commandes sont exécutées par le serveur, n'importe quel navigateur fera l'affaire.

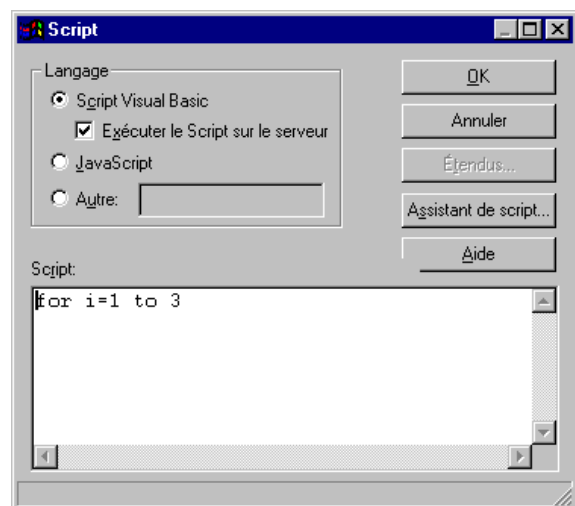
UTILISATION DE VBSCRIPT AVEC FP97

Comme on a pu s'en apercevoir dans le petit source ASP donné plus haut, toutes les commandes ASP doivent débiter par `<%` et se terminer par `%>`. Si l'on édite les pages HTML avec un éditeur autre que FP97 (*FrontPage97*), il faudra chaque fois les ajouter. FP97 lui, par contre le fera automatiquement.

La représentation des groupes de commandes ASP sous FP97 a la forme suivante:



Chaque groupe est représenté par une petite icône, par un double-clic dessus il est possible de modifier les commandes:



Il ne faut pas oublier de spécifier le type de langage utilisé et surtout de forcer l'exécution sur le serveur.

Nous allons voir maintenant comment *afficher 3 fois le mot hello et la valeur du compteur de la boucle*, le résultat va nous donner:

```
hello 1
hello 2
hello 3
```


le source aura la forme suivante:

```
<%for i=1 to 3%>
hello <%= i%><p>
<%next%>
```

et l'aspect de ce source sous FP97:



Dans la première icône nous avons: <%for i=1 to 3%>, c'est l'initialisation de la boucle avec comme variable de compteur i.

Dans la seconde icône nous avons: <%= i%>, c'est l'affichage de la variable i, le *pseudo print* du basic est remplacé ici par un =. Attention à ne pas l'oublier et aussi à ne pas le mettre quand on utilise la variable dans un calcul par exemple.

Et dans la troisième icône nous avons: <%next%>, c'est la fin de la boucle.

Nous allons voir maintenant, le même exemple, mais avec l'affichage dans un tableau:

hello	1
hello	2
hello	3

et l'aspect de ce source sous FP97:



Il faut faire un tableau de 3 lignes et 2 colonnes, les deux premières cellules et les deux dernières doivent être fusionnées et contenir le début de la boucle et la fin de la boucle. Les deux cellules du milieu du tableau contiennent l'information à afficher. C'est ainsi que l'on peut faire un tableau dynamique sous FP97, avec rupture automatique des lignes du tableau.

CRÉATION DE FORMES AVEC FP97

Voyons maintenant, ce dont nous avons toujours rêvé, la création de formes ET l'utilisation de formes sous FP97, donc avec un petit script qui va s'occuper du résultat de la forme.

Nous allons développer une petite application pour convertir une monnaie dans une autre monnaie selon un taux donné. Afin de simplifier la compréhension, nous l'avons coupée en deux: un fichier statique .htm et un fichier dynamique .asp.

La petite application se présente dans un *browser* comme suit:

Conversion de monnaie...

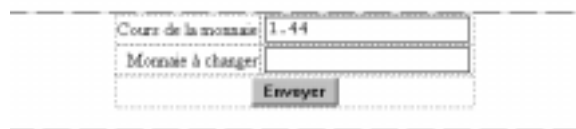
ce qui donnera dans un *browser*:

Résultat de la conversion...

Alors, 12.50 converti au taux de change 1.44 font : 18
Merci.

Elle sera représentée dans le fichier .htm dans FP97 ainsi:

Conversion de monnaie...



et dans le fichier .asp dans FP97:

Résultat de la conversion...

Alors, 12.50 converti au taux de change 1.44 font : 18
Merci.

Avec le source ASP:

```
<h1 align="center">Résultat de la conversion...</h1>
<p>Alors, <%= Request.Form("t2")%> converti au taux de change <%= Request.Form("t1")%> font : <%= Request.Form("t1") * Request.Form("t2") %></p>
<p>Merci.</p>
```

On travaille, ici, avec des variables de type *VARIANT*, c'est le VBScript qui se *débrouille* pour que les types jouent; on mélange agréablement des variables de type numérique avec du texte. Bien entendu, on peut spécifier le type de variables quand on a des grands codes afin de d'éviter des erreurs de mauvais type de variables.

On peut, c'est très agréable, voire indispensable, utiliser un éditeur HTML pour la mise en page des pages .asp, parce que, justement, le principe de l'ASP est d'avoir un squelette HTML avec des *tags* ASP.

UTILISATION DU MODULE ODBC

On se rend compte tout de suite que lorsqu'on fait des pages HTML dynamiques, il faut généralement utiliser une grande quantité de données ou d'information. C'est là que rentre en jeu l'utilisation d'une base de données, avec tous les outils propres à sa gestion, pour fabriquer les pages HTML. Toute la puissance de l'ASP est révélée alors ici.

Le grand avantage d'un serveur ASP (voir ci-dessus), c'est qu'il utilise comme gateway entre le WEB et la base

de donnée le standard ODBC (Open Data Base Connectivity). Cela veut dire qu'autant une base de données MS Access, qu'un fichier MS Excel, texte, ou DBASE ou encore qu'une base de données SQL, enfin tout ce qui peut traiter des données et qui possède un driver ODBC, peut être utilisé par le serveur ASP.

Dans un premier temps, pour faire votre maquette ou pour de petits serveurs, vous pouvez utiliser la base de données MS Access, puis, si votre application est très exigeante au niveau des performances ou des tailles de données, vous pouvez passer par la suite sur un SQL Server.

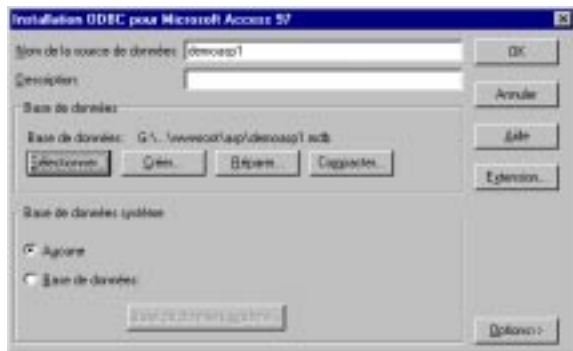
Pour information, le moteur de base de données MS Access est fourni avec IIS 3 (le moteur, pas l'application), et c'est amplement suffisant pour faire tourner un serveur ASP.

Pour m'en convaincre j'ai fait une petite base de données de 4'200'000 enregistrements (oui, plus de quatre millions, taille du fichier > 500MB) que j'ai fait tourner via un serveur ASP sur un bon vieux 486-33MHz-24MBytes RAM avec Windows NT Server, eh bien vous ne le croirez pas, mais cela fonctionne de manière tout-à-fait correcte... incroyable non ?

Encore une chose, un connecteur ODBC sur un fichier texte n'est pas une idée stupide en soi, pensez aux nombreux fichiers à plat produits par certaines administrations !!

Pour utiliser le module ADO (ActiveX Data Objects) avec un serveur ASP, vous devez:

1. créer votre base de données, par exemple avec MS Access;
2. la *connecter* sur un port ODBC DSN System, via le ODBC du Control Panel;



3. puis enfin, créer les pages ASP.

Nous allons voir un petit exemple pour *Affichage sous forme de table la base de données*:

Affichage sous forme de table la base de donnée

Nom	Prénom	Email	NumTel
Cionca	Ion	cionca@nic.epfl.ch	4586
Roulet	François	roulet@nic.epfl.ch	4590
zuzu	toto	zuz	zuzu

Voici l'aspect de .asp sous FP97:

Affichage sous forme de table la base de donnée



et son code source:

```
<p><%Set Conn = Server.CreateObject("ADODB.
Connection")
Conn.Open "DemoASP1"
Set RS = Conn.Execute("SELECT * FROM Re-
quête1") %></p>
<div align="center"><center>
<table border="1">
<tr>
<td><% For i = 0 to RS.Fields.Count - 1
%></td>
<td><b><% = RS(i).Name %></b></td>
<td><% Next %></td>
</tr>
<tr>
<td colspan="3"><% Do While Not RS.EOF
%></td>
</tr>
<tr>
<td><% For i = 0 to RS.Fields.Count - 1
%></td>
<td><% = RS(i) %></td>
<td><% Next %></td>
</tr>
<tr>
<td colspan="3"><% RS.MoveNext
Loop
RS.Close
Conn.Close %>
</td>
</tr>
</table>
</center></div>
```

Ici tout est dynamique, c'est à dire que l'on *dump* toute la base dans une table.

Voici maintenant un exemple d'affichage d'un enregistrement à la fois avec une barre d'outils pour se déplacer dans la base:

Index:

Nom:

Prénom:

Num Tel:

eMail:



Voici l'aspect de .asp sous FP97:



Nombre de records dans le recordset: ↴

Index:	<%= RS("#Index") %>
Nom:	<%= RS("Nom") %>
Prénom:	<%= RS("Prénom") %>
Num Tel:	<%= RS("NumTel") %>
eMail:	<%= RS("eMail") %>

et son code source (extrait):

```
<form action="db2.asp" method="GET">
  <div align="center"><center><table border="0">

    <tr>
      <td align="right">Index:</td>
      <td><input type="text" size="20"
        name="T0" value="<%= RS("#Index")
        %>"></td>
    </tr>
    <tr>
      <td align="right">Nom: </td>
      <td><input type="text" size="20"
        name="T1" value="<%= RS("Nom")
        %>"></td>
    </tr>
    <tr>
      <td align="right">PrÈnom: </td>
      <td><input type="text" size="20"
        name="T2" value="<%= RS("PrÈnom")
        %>"></td>
    </tr>
    <tr>
      <td align="right">Num Tel: </td>
      <td><input type="text" size="20"
        name="T3" value="<%= RS("NumTel")
        %>"></td>
    </tr>
    <tr>
      <td align="right">eMail: </td>
      <td><input type="text" size="20"
        name="T4" value="<%= RS("eMail")
        %>"></td>
    </tr>
  </table>
</center></div><div
  align="center"><center><table border="0" cellpadding="5">
  <tr>
    <td align="center"><input
      type="submit" name="B1"
      value="|<"><input type="submit"
      name="B2" value="<"><input
      type="submit" name="B3"
      value=">"><input type="submit"
      name="B4" value=">|"></td>
  </tr>
</table>
</center></div>
</form>
```

```
value=">"><input type="submit"
name="B4" value=">|"></td>
</tr>
</table>
</center></div>
</form>
```

EN CONCLUSION...

Et nous voici à la conclusion, la partie la plus importante d'un article, car c'est celle qu'on lit quand on n'a pas le temps...

On a tous compris que l'avenir du WEB sera aux pages dynamiques et si possible avec une base de données. Pour mémoire les solutions pages WEB dynamiques proposées par Illustra ou Oracle coûtent quelques dizaines de milliers de francs, donc sont inaccessibles à des PME ou des petites gens comme nous.

La solution ASP est de l'ordre de grandeur 5 kF pour la machine et 1kF pour Windows NT4 Server (ASP, le moteur MS Access et FrontPage 97 sont compris dans NT).

Avec une configuration de ce type vous pourrez développer une application du genre RegArt (l'agenda culturel de Suisse romande, <http://www.regart.ch>) sans faire sauter vos crédits. ■

RÉSEAUX INFORMATIQUES ET SERVICES INTERNET

par Jean-Jacques Dumont, SIC

L'accès à Internet, le plus célèbre des réseaux informatiques, fait désormais partie intégrante de la vie quotidienne de la plupart d'entre nous, les branchés, au même titre que le téléphone ou la télévision. D'ailleurs, ces trois utilitaires aujourd'hui encore très différents seront amenés probablement, dans un avenir plus ou moins proche, à converger en un seul outil de communication universelle, fixe ou mobile.

L'évolution désormais galopante des services disponibles sur Internet n'a pourtant pu être suivie et comprise depuis ses origines que par une population limitée: celle des informaticiens opérant dans les universités ou grandes entreprises, ces lieux privilégiés où réseaux informatiques et réseaux de communication ont pu commencer leur processus de convergence, il y a une quinzaine d'années.

UN PEU D'HISTOIRE

Le concept d'Internet est né de cette fusion de deux types de réseaux bien distincts à l'origine. Comme son nom le suggère, il s'agit en effet de l'immense toile progressivement tissée par l'interconnexion anarchique d'une multitude de réseaux informatiques plus ou moins indépendants à l'aide de lignes louées aux opérateurs de télécommunication, même si celles-ci étaient à l'origine prévues pour la téléphonie analogique.

MAIS REPRENONS L'HISTOIRE PAR LE DÉBUT

Donc, au premier jour, il existait deux mondes disjoints:

- celui de l'informatique, par définition le domaine du traitement de l'information, constitué d'ordinateurs de divers types et marques, utilisant des logiciels propriétaires pour le traitement de données codées dans des formats propres aux constructeurs;
- celui des télécommunications, régi par l'organisation mondiale répondant au doux nom de CCITT, dont la problématique était de gérer les communications téléphoniques locales ou internationales à l'aide d'un système de commutation (switching) normalisé.

Durant les années 70, le besoin d'échanger des données entre ordinateurs se fit de plus en plus pressant: certes, le stockage de ces données sur bande magnétique sur un ordinateur, suivi d'une relecture sur un autre était

une façon de résoudre un problème, mais la connexion directe à travers un câble de couplage apparut de plus en plus comme la voie d'avenir, surtout lorsque les données devaient être distribuées rapidement entre plusieurs ordinateurs. Ainsi naquit la notion de réseau informatique local (LAN: Local Area Network).

Très longtemps et jusqu'à récemment, chaque constructeur ne fournissait qu'une solution propriétaire fermée pour ces besoins de communication: deux ordinateurs de marque Digital pouvaient communiquer par DECnet pendant que deux ordinateurs IBM échangeaient des données via SNA, mais le passage entre le monde DEC et le monde IBM ne pouvait se faire qu'à travers des passerelles très lourdes et peu fiables.

D'autre part, l'arrivée des machines individuelles PC et Apple fut à l'origine durant les années 80 d'une nouvelle génération de réseaux informatiques locaux propres à ces technologies: LanManager ou Novell pour les PC, PhoneNet puis AppleTalk pour les Mac... A nouveau, la connexion avec les machines hôtes, que l'on n'appelait pas encore serveur, ou entre réseaux différents de machines individuelles ne pouvait se faire qu'à travers des passerelles bien peu pratiques et n'autorisant que des fonctions simples: échange de fichiers de données, puis de texte. Ainsi apparurent les premiers protocoles de courrier électronique, soit l'échange direct d'informations textuelles entre utilisateurs de divers systèmes informatiques.

Le besoin suivant fut de permettre une communication directe entre universités ou entre branches d'entreprises géographiquement dispersées. Au départ, une seule solution pour cela: l'utilisation de lignes téléphoniques à l'aide de modems. Ceux-ci permettent en effet le transfert par signaux analogiques d'informations numériques émises d'un côté de la ligne par un ordinateur et reçues de l'autre côté par un autre ordinateur, avec évidemment des vitesses de transmission assez réduites.

Cette situation allait s'améliorer considérablement avec la généralisation durant les années 90 des techniques numériques en téléphonie, en particulier celle de la commutation par paquets (X.25). Celle-ci permet en effet d'optimiser l'utilisation de la bande passante, puisqu'elle permet de partager la même ligne entre plusieurs communications individuelles (multiplexage). D'autres progrès suivirent alors rapidement: les lignes purement numériques de type RNIS (Réseau Numérique à Intégration de Service), le remplacement des câbles coaxiaux en cuivre par la fibre optique, qui permet un très haut débit

de transmission avec une grande fiabilité, et l'utilisation de plus en plus intensive des satellites pour les communications intercontinentales.

A ce point de notre histoire, réseaux informatiques et réseaux de communications, désormais numériques eux aussi, ont pratiquement fusionné. La preuve: on ne parle plus désormais que des services sur LE réseau, qui constituent la télématique d'aujourd'hui. Mais, si vous avez bien suivi l'exposé, ceci n'est a priori vrai qu'au niveau physique du transport d'information numérique, sous forme de bits et d'octets. Par quel miracle les informations macroscopiques peuvent-elles être interprétées correctement par virtuellement tous les ordinateurs de la planète bleue, et restituées sous forme de texte, images, son ou vidéo de façon compréhensible pour leurs utilisateurs?

L'INTERNET PROTOCOL

Pour comprendre cela, il nous faut à nouveau remonter dans le temps, à notre point de départ des années 70. A cette époque, dans les universités américaines, des chercheurs mandatés par le Département de la Défense mettaient au point des méthodes permettant de maintenir des communications numériques en permanence sur toute la surface de la planète, même en cas de conflit nucléaire ou de cataclysme naturel. Unix étant le système en vogue à cette époque dans les milieux de la recherche, c'est sur ce type de système que fut donc d'abord développé le protocole IP (Internet Protocol), qui est fondamentalement une méthode d'adressage et de "routage" de paquets d'information numérique: comme à la poste, l'information proprement dite est enrobée d'une enveloppe où sont codées les adresses d'expédition et de destination, ainsi que le type de contenu; l'enveloppe est envoyée vers un premier bureau de poste (un router en l'occurrence) qui la réexpédie vers un second bureau choisi sur la base de l'adresse; et ainsi de suite de proche en proche jusqu'au destinataire final. Avantage de la méthode: si un des bureaux de poste intermédiaires est fermé ou détruit, une autre route peut être choisie. Il faudra sans doute plus longtemps pour que l'enveloppe arrive, mais elle arrivera.

Le protocole IP suppose que chaque machine potentiellement partenaire dans les échanges de paquets, y compris les routers, est identifiée univoquement par une adresse, composée de quatre champs numériques. Par exemple, l'adresse IP du PC sur lequel cet article a été écrit est: 128.178.1.225.

La première partie (128.178) indique que ce PC est connecté au *réseau* appartenant à l'EPFL. La coordination d'attribution d'adresses de réseau se fait sur un plan mondial. La seconde partie précise qu'il appartient au sous-réseau 1, avec le numéro d'identification 225 dans ce sous-réseau. La coordination de l'attribution des adresses à l'intérieur d'un réseau se fait localement (par le SIC, pour le domaine de l'EPFL). A chaque réseau ou sous-

réseau est associé un router chargé de diriger le trafic entrant et sortant.

Le cerveau humain n'étant pas très adapté à la manipulation et mémorisation de chiffres, il existe des tables dénommées Name Server permettant d'associer des noms logiques à chacune de ces adresses. Ainsi, le nom du réseau de l'EPFL est-il répertorié dans ces tables comme appartenant au domaine "epfl.ch", tandis que le PC de l'auteur est connu sous le nom de "slpc3". En accolant ces deux particules, on obtient comme nom unique sur la planète et probablement dans notre univers: "slpc3.epfl.ch". Cependant, il est aussi possible d'associer plus d'un nom à une machine ou parfois d'attribuer plusieurs identificateurs à une même machine.

Même si les noms de machines sont seulement connus localement à l'intérieur d'un domaine, les tables de noms de domaines ont une tendance certaine à l'embonpoint. Il est donc de bon ton de concentrer ces informations sous forme d'un service local sur une machine du domaine, que l'on appelle le DNS (Domain Name Server).

Telle est donc l'idée maîtresse qui allait progressivement s'imposer comme standard d'adressage universel, sans qu'aucun organisme officiel extérieur ou qu'aucune entreprise commerciale ne soit jamais intervenue dans le processus. Pourquoi et comment?

- le succès d'IP fut dans les premiers temps lié à celui d'Unix, ce système, créé par Bell Lab, développé ensuite dans les universités pour les universités. C'est lors de l'avènement des stations de travail individuelles durant la deuxième moitié des années 80 que cet environnement allait se propager: pour les petites entreprises telles que Sun Microsystems démarrant sur un marché alors dominé par le précurseur Apollo, il était économiquement et techniquement très raisonnable d'utiliser un système déjà existant et surtout gratuit au lieu de re-développer à grands frais un nouveau système propriétaire qui n'apporterait rien de neuf.
- nous avons vu à quel point il était pénible d'établir une connexion fiable entre ordinateurs de marques différentes. Or, ces difficultés disparaissent d'un coup si l'on installe Unix, donc le protocole de communication IP, sur ces machines. Une très grosse pression allait ainsi être exercée par les utilisateurs sur les constructeurs pour qu'ils fournissent sur leurs machines soit un système Unix, soit au moins IP comme couche de communication pour leurs systèmes propriétaires.
- l'évolution du protocole, notamment de façon à le rendre plus facilement intégrable dans les environnements existants, s'effectue selon un processus très efficace: toute amélioration est suggérée sous forme d'un RFC (Request for Comment) distribué à la communauté des utilisateurs, qui peuvent en évaluer les bénéfices. Si un consensus se dégage, la proposition peut être entérinée par l'IETF (Internet Engineering Task Force), l'organisme en principe indépendant de toute influence politique ou commerciale qui s'ef-

force de garantir la qualité technique des protocoles constituant l'armature d'Internet.

Mais bien sûr, le travail de l'IETF ne se limite pas à l'évolution de IP. Une fois la couche de communication physique (câbles et signaux) et la couche d'adressage logique (IP et DNS) assurée, il reste encore à établir le plus important du point de vue de l'utilisateur: les services.

LES SERVICES D'INTERNET

PROTOCOLES

Comme en bon français, un protocole est un ensemble de règles arbitraires sur lesquelles les participants aux activités d'échanges d'information se sont mis d'accord. Dans le chapitre précédent, nous avons vu comment les règles d'adressage IP permettaient à tous les ordinateurs de la planète de s'identifier entre eux. Ci-après, nous allons voir comment d'autres règles plus complexes, éditées elles aussi par l'IETF, permettent de construire les services de communication conviviaux dont l'utilisateur final peut jouir aujourd'hui.

Dans le modèle en couche cher à l'ISO, entre la couche d'adressage (IP) et la couche applicative proprement dite telle que perçue par l'utilisateur (transfert de fichiers, e-mail, www...), il existe une couche intermédiaire définissant le type d'interaction entre les partenaires de la fonction de communication. Si l'on excepte NFS (Network File System), qui utilise pour cette couche intermédiaire le protocole asynchrone UDP (User Datagram Protocol) et NNTP (les *news*), qui utilise le très ancien protocole UUCP (Unix to Unix Copy Program) selon la logique de *store-and-forward*, les principaux services disponibles aujourd'hui sur Internet sont basés sur le protocole TCP (Transmission Control Protocol). Celui-ci assure la synchronisation des paquets: la série de paquets composant un message est reconstituée à leur arrivée chez le destinataire dans le même ordre que lors de leur émission. TCP permet en outre d'établir un circuit virtuel entre deux adresses d'Internet, ce qui assure la bidirectionnalité en temps réel du service. L'omniprésence de TCP explique pourquoi l'on dit souvent que TCP/IP est LE protocole d'Internet.

Avant de passer aux protocoles applicatifs, mentionnons encore les protocoles de ligne point-à-point SLIP (Serial Line IP) et PPP (Point to Point Protocol) qui autorisent l'attribution dynamique d'une adresse IP à une machine qui se connecte au réseau durant un temps limité, en général par une ligne téléphonique. C'est la technique utilisée par les prestataires de service Internet pour gérer un nombre de clients qui dépasse largement le nombre d'adresses IP dont ils disposent.

CLIENT/SERVEUR

La plupart des applications Internet obéissent à la topologie dite *client/serveur*: l'utilisateur dispose sur sa machine d'un logiciel dit *client* qui peut envoyer des re-

quêtes codées vers une autre machine où est installé le logiciel *serveur* correspondant. Celui-ci tourne en permanence, et est à l'écoute de toute requête qui lui est adressée. C'est ce qu'on appelle un *daemon* dans le jargon emprunté au monde Unix. Le daemon utilise le protocole spécifique de l'application comme langage de dialogue avec le client. Il est attaché à un canal logique de communication dénommé *port*, identifié par un numéro spécifique. En tant que *serveur*, le daemon doit pouvoir gérer plusieurs connexions simultanées.

CONNEXIONS À DISTANCE: TELNET

Dans le modèle informatique du *mainframe*, les utilisateurs utilisent un terminal pour se connecter à l'ordinateur central de l'entreprise via une ligne série. Telnet, une des premières applications TCP/IP à voir le jour, réalise l'émulation de ce modèle en utilisant un second ordinateur comme terminal. Au contraire d'un terminal simple et *bête* qui est en permanence rattaché à son *mainframe*, l'application Telnet *client* permet de se connecter en principe à toute machine multi-utilisateur dotée d'une application Telnet *serveur* (dénommée *telnetd* dans le jargon) atteignable par Internet. Comme à partir d'un terminal, la seule condition est de disposer d'un compte utilisateur sur la machine cible, et de connaître le mot de passe associé.

TRANSFERT DE FICHIERS: FTP

Comme son nom l'indique clairement, FTP (File Transfer Protocol) permet de transférer un fichier quelconque (données, texte, programmes binaires, graphique...) d'un ordinateur à un autre. Le mécanisme est semblable à celui de Telnet, puisqu'il faut d'abord établir une connexion entre la machine cliente et la machine cible. Comme pour Telnet, il faut donc disposer d'un compte utilisateur sur cette dernière. Mais il faut en plus s'assurer que cet utilisateur dispose des droits d'accès aux fichiers soit en lecture pour les fichiers que l'on veut importer (commande GET), soit en écriture pour les fichiers que l'on veut exporter (commande PUT).

L'E-MAIL (OU MÉL SELON LA COMMISSION JUPPÉ!): SMTP

Comme son nom ne l'indique pas, SMTP (Simple Mail Transfer Protocol) est le squelette d'un système de messagerie relativement complexe. Dans sa version originale sur Unix, son architecture est fort semblable à celle de FTP: une application dénommée *sendmail* envoie ou reçoit des fichiers de type *message*, avec ou sans annexes. Cette application peut desservir plusieurs utilisateurs dans sa version *serveur*.

Outre les fonctions de base *envoyer* ou *recevoir*, une multitude d'autres fonctions complémentaires se sont ajoutées au cours du temps. Outre les adjonctions au protocole SMTP lui-même, d'autres enrichissements ont été introduits par le protocole MIME (Multi-purpose Internet Mail Extensions), qui représente une tentative

de codifier ces extensions sans les limiter. On conçoit que ce foisonnement d'idées n'a pas vraiment permis de simplifier les interfaces proposées aux utilisateurs.

L'approche POP (Post Office Protocol) a néanmoins permis aux utilisateurs de Mac et de PC de bénéficier de ces progrès de façon élégante. L'idée est de revenir au modèle client-serveur simple, selon le modèle des cases postales gérées par nos braves vieilles PTT: au lieu d'être expédié vers la boîte aux lettres de l'utilisateur, le courrier électronique est adressé au bureau de poste le plus proche (le serveur POP) qui le classe dans des cases étiquetées. L'utilisateur peut venir vider sa boîte quand il le désire pour autant qu'il en possède la clé (le mot de passe, en l'occurrence). C'est l'application *cliente* (Eudora par exemple à l'EPFL) qui établit la connexion avec le serveur POP et effectue toutes les démarches pour ramener le courrier correctement traité sur le PC ou le Mac de l'utilisateur. De ce fait le poste client ne doit pas être connecté en permanence sur Internet, ce qui est particulièrement avantageux lorsqu'il s'agit d'un PC portable emmené lors d'un voyage en Australie!

A noter que SMTP n'est pas le seul protocole *Mé* en vigueur: d'un côté, les Telecom PTT d'obédiences nationales ont longtemps essayé d'imposer leur protocole X.400; de l'autre, il existe une multitude d'autres systèmes de messageries locales qui ont eu le temps de s'implanter bien avant l'avènement d'Internet. Il faut aussi mentionner FidoNet, qui est un protocole de messagerie conçu pour des lignes de mauvaise qualité, comme on peut en rencontrer dans les pays du tiers monde. Le résultat est un assemblage de passerelles qu'il n'est pas toujours évident de maîtriser, surtout si le site ne dispose pas d'une gestion centralisée. D'où aussi la frustration de certains usagers, lorsque le message n'est pas transmis comme espéré.

Signalons également qu'il est désormais possible sur Internet, grâce au système PGP (Pretty Good Privacy), de crypter les messages confidentiels, voire de garantir leur authenticité grâce à une signature électronique.

LE SALON OÙ L'ON CAUSE: IRC

Internet Relay Chat est une application très populaire dans les CyberCafés: elle permet en effet de participer en direct à une multitude de conversations sur des sujets divers, des plus sérieux aux plus anodins. Sa mauvaise réputation provient du fait que les pirates du câble utilisent préférentiellement ce moyen pour communiquer rapidement entre eux, et de façon parfaitement occulte, la dernière recette miracle qui permet de trander un système. Mais de la même façon, ce moyen pourrait être utilisé pour organiser les secours lors d'un tremblement de terre ou autres cataclysmes moins naturels.

Topologiquement, un serveur IRC gère plusieurs canaux, attribués chacun à un sujet de conversation différent (*chat*). Un utilisateur quelconque peut se déclarer membre du groupe de personnes participant au *chat* à un moment donné. Ses interventions tapées au clavier sont alors répercutées vers tous les autres membres con-

nectés. Pour chaque *chat*, il existe en principe un modérateur qui a le pouvoir d'exclure tout membre qui se livrerait à des excès.

DISTRIBUTION DES NEWS: NNTP

Comme suggéré dès le début de ce chapitre, le protocole très populaire utilisé encore actuellement pour la distribution d'informations à une grande population est en fait très ancien et peu adapté à l'évolution d'Internet. Il n'est pourtant plus possible de le modifier sans provoquer l'effondrement de ce vénérable édifice.

Comme pour IRC, le principe pour l'utilisateur des news est de *s'abonner* à un certain nombre de sujets (*newsgroups*). Mais cette fois les réactions des lecteurs par rapport à la publication d'un article d'information ne peuvent avoir lieu qu'en temps différé: l'article publié par un abonné dans un certain *newsgroup* sera mis à disposition des autres abonnés via un *serveur de news* et c'est ce dernier qui est responsable de collecter les articles selon un mécanisme qui lui est propre. Notamment, il peut n'effectuer cette collecte que s'il existe effectivement des utilisateurs abonnés, comme le serveur Dnews de l'EPFL. Ou il peut ne le faire qu'à des intervalles éloignés, pour éviter des temps de connexions inutiles. Le résultat est un outil de dissémination d'informations très économique, mais parfois assez lent.

LE WORLD WIDE WEB: HTTP

A tout seigneur tout honneur: nous terminerons par WWW, l'application d'Internet la plus en vogue actuellement, celle qui a provoqué la prise de conscience par le grand public du potentiel infini offert par les nouvelles technologies de l'information.

L'HyperText Transfer Protocol qui est à la base de WWW n'est pas tout neuf non plus. Il fut développé et utilisé dès le début des années 90 au CERN pour la dissémination de documentation sur le réseau entre tous les partenaires de cette organisation scientifique mondiale. Mais au début, les liens hypertextes permettant de passer facilement d'un document rébarbatif à l'autre faisaient plutôt penser à un immense plat de spaghettis relativement informe.

L'application Gopher, utilisant le même principe, présentait déjà de grands progrès au niveau de l'interface utilisateur, mais c'est finalement l'interface graphique universelle *mosaic* qui allait définitivement affirmer le succès de l'approche hypertexte.

La suite, vous la connaissez: Netscape Navigator et MS Internet Explorer, deux descendants directs de *mosaic*, se disputent actuellement le marché des *browsers*. D'autre part HTML, le langage de description des pages affichées à l'écran par ces butineurs, s'est considérablement développé (alourdi?) et permet maintenant les plus grandes fantaisies graphiques, voire l'utilisation d'illustrations sonores ou vidéo. Mais le principe est le même, en restant fidèle au modèle client/serveur: la partie cliente de l'application (le *browser*) envoie des requêtes vers un serveur dont le nom est indiqué dans l'URL (Universal

Resource Locator), selon les instructions de l'utilisateur (normalement un simple clic sur une zone de l'écran à laquelle est associée un lien hypertexte). Le serveur HTTP (httpd en jargon Unix) cherche localement le fichier mentionné dans la requête URL et l'expédie vers le client, s'il en a l'autorisation. Car il est bien sûr possible de

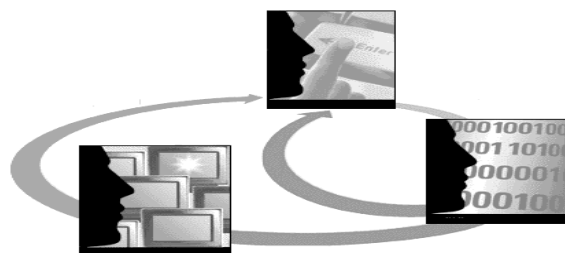
n'autoriser l'accès à l'information qu'à des machines ou personnes dûment validées...

Nous arrêterons là l'exploration de l'univers WWW: la littérature ne manque pas sur ce sujet, et poursuivre au delà de cette brève introduction nous ferait sortir du sujet de cet essai. ■

L'OUTIL HELPDESK DE SUN MICROSYSTEMS

Présentation le 24 juin 1997 à 15h00 à la salle de conférences du SIC

par Sun Microsystems



INTRODUCTION

Sun Microsystems offre en plus des stations et serveurs toute une gamme de solutions. Dans ce cadre Sun propose de vous présenter un nouveau système HelpDesk.

Cette application développée par la société Remedy est la plus largement employée sur le marché pour créer des solutions d'assistance et pour automatiser des processus internes comme la gestion de parcs, la gestion des changements, celle des incidents, etc.

Le but étant de gérer un appel d'un utilisateur à son support.

ARCHITECTURE CLIENT/SERVEUR EN TROIS ÉTAGES

L'architecture unique de AR System combine un moteur de workflow largement personnalisable et une interface utilisateur ouverte et évolutive permettant une intégration rapide. Cela permet d'assurer la compatibilité avec une large gamme d'environnements informatiques, de produits et de technologies.

L'accès depuis le poste client s'effectue via le logiciel client (multiplateforme) de AR System, un navigateur Web ou la messagerie électronique.

Les deux composants côté serveur sont, d'une part, le serveur AR System qui gère les requêtes des clients, le contrôle d'accès et le traitement du workflow, et, d'autre part, le serveur de la base de données, optimisant la gestion des données pour les bases relationnelles les plus connues.

ETAPE D'UN APPEL

- Le problème ou la question est soumis via le helpdesk.
- L'utilisateur peut déjà trouver une solution dans une base de données avec un outil de recherche.
- Le support de premier niveau réceptionne l'appel. Cette personne peut être un administrateur système de département comme une secrétaire pour un problème administratif.
- Si la solution est connue elle peut être transmise soit par le logiciel soit par E-Mail. De même plus d'information peut être demandée.
- Le problème peut aussi être envoyé à un support de deuxième niveau qui peut être un service de l'EPFL ou une entreprise externe telle que Sun Service.

AVANTAGES D'UN HELPDESK

- Permettre aux administrateurs système de gérer efficacement les appels utilisateurs, depuis leur prise en compte jusqu'à leur résolution.
- Amélioration du niveau de service aux utilisateurs.
- Meilleur suivi des incidents et meilleures mesure et analyse de l'activité.

PRÉSENTATION À L'EPFL

Cette présentation s'adresse à celles et ceux qui posent des questions ou qui y répondent. ■

UNE CARTE PC DANS VOTRE MAC

par *Christophe Salzmann, DGM-IA*

Comme promis le mois dernier (ça m'apprendra à être aussi précis dans mes délais de rédaction), je vais vous parler des cartes PC pour les Macintosh. Il en existe au moins une dizaine. Apple en propose trois, toutes pour bus PCI, et en proposait une pour NuBus. Différents constructeurs dont Orange Micro (<http://www.orangemicro.inter.net>), Reply (<http://www.reply.com>) – racheté récemment par Radius (<http://www.radius.com>) – proposent également des cartes PC soit pour le bus PCI soit pour NuBus. Elles vont du 486 au MMX à 200 Mhz. Toutes ces cartes sont en fait des PC qui se logent dans votre Mac. Elles *émulent* entièrement ou partiellement un PC. La plupart des cartes interagissent avec le hardware de votre mac pour accéder au réseau et aux différents périphériques (Floppy, HD, CD,...).

La carte PCI Pentium 100 MHz d'Apple (<http://product.info.apple.com/productinfo/datasheets/dt/pccompcards4ppc.html>) a retenu mon attention pour des raisons de coût et de disponibilité. Elle coûtait environ 1400 Sfr. Elle a été remplacée par une nouvelle carte contenant un Pentium à 166 Mhz (<http://product.info.apple.com/productinfo/datasheets/dt/pccompcard5ppc.html>) pour un prix équivalent. Cette carte est d'ailleurs proposée en *bundle* avec un Powermac 7300/180 (http://powermacintosh.apple.com/products/7300_180PC.html). Cette configuration n'est pas encore proposée en Suisse.

L'installation de la carte se fait sans problème pour les gens qui sont habitués à installer de la RAM dans des PowerMac. Une fois la carte installée dans son slot PCI, il vous reste à donner de la voix à votre PC en connectant la sortie son de la carte entre la sortie son du CD et la carte mère. L'installation diffère quelque peu d'une machine à l'autre. A l'extérieur de la carte, vous avez un connecteur pour Joystick et une sortie écran. Vous avez la possibilité de brancher un écran dédié à votre PC ou d'employer l'écran de votre Mac. La carte possède un emplacement pour une seule barrette de mémoire (DIMM). Si vous ne mettez pas de mémoire sur la carte, une partie de la mémoire de votre Mac sera employée par la carte PC. Il est vivement recommandé de mettre au minimum 8 Mb de RAM, idéalement 16 MB. Il faut faire **très** attention au contrôleur de votre barrette DIMM si vous voulez pouvoir accéder à la totalité de votre mémoire. Je tiens à remercier Jean Metzener de la maison EPPS qui a patiemment testé les 20 barrettes de son stock avant d'en trouver une qui fonctionne avec ma carte. Vous pouvez également augmenter la *vidéo Ram* et la porter à 4 MB.

La carte est livrée avec DOS 6.22. La licence Windows 95 ou 3.11 est à acheter en sus. Windows NT ne fonctionne pas sur cette carte, je n'en connais pas la

raison exacte, car la plupart des cartes des autres constructeurs supportent Windows NT. Apple fournit un CD ainsi qu'un set de disquettes DOS et quelques disquettes d'utilitaires. Vous devez commencer par mettre à jour le système de votre Mac (7.5.3) si cela n'est pas déjà fait. Ensuite, vous créez un disque dur PC qui est vu comme un fichier par le finder. Vous pouvez monter ce fichier sur le finder et accéder directement aux fichiers PC depuis le finder sans que votre carte PC soit active (Fig. 1) Vous installez DOS 6.22 puis Windows95. Il faut veiller à installer les bons drivers (réseau, etc...); ceux-ci se trouvent sur le CD fourni par Apple. Une fois la configuration de votre PC terminée, il est judicieux de faire une copie de sauvegarde de disque «C». En cas de problème il suffira de remplacer le fichier défectueux par la copie.

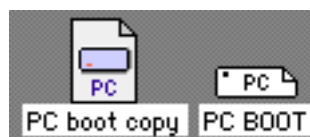


Fig 1: le disque dur «C» vu du côté Mac (unmounted and mounted)

Le manuel décrit précisément la démarche à suivre. Pour le reste de la configuration vous vous retrouvez devant un vrai PC; le niveau de connaissance de l'utilisateur fera la différence... Comme pour l'émulateur Soft Windows, vous avez un numéro IP spécifique, distinct de celui de votre Mac, pour votre carte PC. Le Mac et la carte peuvent accéder au réseau en même temps (Fig 2).

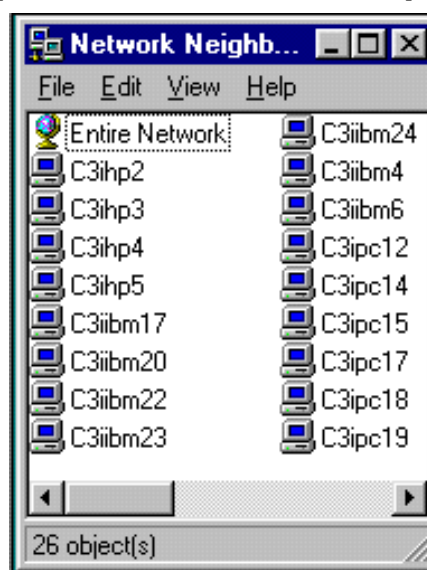


Fig 2: connexion au réseau PC



Fig. 8: Benchmark

Il n'est pas besoin de faire de *benchmarks* pour se rendre compte que les performances sont là, on se trouve vraiment sur une machine avec un Pentium à 100 Mhz. Pour des raisons de temps je n'ai pas fait un test aussi complet que pour l'émulateur SoftWindows. Mais l'intégration d'un PC dans un Mac semble sans faille.

Le Mac n'ayant pas de port parallèle, on retrouve les mêmes limitations potentielles que pour SoftWindows. Vous avez uniquement la possibilité de connecter un joystick. Le PC serait-il uniquement une machine de jeux?

Les cartes proposées par les autres constructeurs comprennent, suivant les modèles, des ports parallèles, des ports série supplémentaires, des port joystick ainsi que des ports audio in et out. La nouvelle carte d'Apple (166 MHz) supporte le port parallèle via une extension.

EMULATION OU CARTE PC?

Il y a quelques temps j'aurais répondu sans aucun doute la carte PC si vous en faite un emploi quotidien. Mais les récents progrès des émulateurs en font une solution à envisager. Un émulateur prendra les temps CPU et de la mémoire de votre Mac. Au contraire, la carte PC peut-être complètement autonome et fonctionner en parallèle avec votre Mac sans que vous ne vous en rendiez compte. L'installation de Softwindows s'est faite en moins d'une heure, alors que l'installation de la carte PC a pris plus de temps. Il a fallu installer DOS puis Windows 95 et ensuite configurer le tout pour pouvoir travailler. Le mode d'emploi est bien fait; il explique clairement les étapes à suivre, mais n'ayant qu'une connaissance limitée de Windows j'ai quand même dû faire appel à mon gourou PC (Merci PYR). La carte Apple testée a quelques limitations. La première est l'absence de port parallèle, très utile pour les clés hardware, et la deuxième est l'impossibilité de faire fonctionner Windows NT. Alors qu'il est possible d'ajouter une extension à la nouvelle carte d'Apple (166 MHz) pour avoir un port parallèle, il n'est toujours pas possible de faire fonctionner Windows NT. Certaines cartes de Reply/

Radius et d'Orange Micro sont compatibles Windows NT.

CONCLUSION

Une carte PC dans votre Mac est un solution très intéressante pour un emploi quotidien, les performances sont là, l'intégration avec le monde Mac est très bien faite. Cette solution semble idéale, elle a l'avantage par rapport à un PC de ne pas prendre de la place supplémentaire sur votre bureau.

Si vous avez besoin d'un port parallèle ou si vous envisagez d'employer Windows NT vous choisirez un carte non-Apple, pour une configuration standard avec Windows 95 la carte Apple est tout à fait adaptée. Pour le moment une solution hardware est plus adaptée à un emploi quotidien qu'une solution émulée. Les différences pourraient s'aplanir avec l'arrivée de nouveaux émulateurs PC.

QUELQUES RÉFÉRENCES

- Data sheet de la nouvelle carte Apple (166MHz)
<http://product.info.apple.com/productinfo/datasheets/dt/pccompcard5ppc.html>
- Data sheet des cartes OrangePC
<http://www.orangemicro.com/page2a.html>
- Data sheet des cartes Reply - Radius
<http://www.reply.com/dompci12.htm>
- un article dans MacWeek qui compare les deux cartes PCI d'Apple
http://www8.zdnet.com/macweek/mw_1027/rev_dos_boards.html
- un autre article de MacWeek évaluant l'alternative donnée par les cartes PC
http://www8.zdnet.com/macweek/mw_1043/sol_dos_boards.html
- une liste des cartes PC
<http://www.paradata.com/~dslk/MacPCI/PCICards/ProcessorCards.html>
- une compilation des différents cartes
<http://home.earthlink.net/~jawagner/>
- OrangePC a un représentant en Suisse Romande. Ce fait est assez rare pour que je vous indique ses coordonnées: ElecSup Distribution SA, 1110 Morges, Tél. 021/803.28.82 ■

Suite de la première page

GifBuilder¹ (freeware). Le but de cet article est de montrer à quel point ce dernier est facile à utiliser².

Pour résumer, tout ce que vous avez à faire est de créer un fichier par image avec votre éditeur graphique préféré, puis faire glisser les icônes correspondantes du Finder à la fenêtre de GifBuilder, éventuellement changer quelques options, prévisualiser l'animation pour vérifier qu'elle correspond à vos désirs, et enregistrer le tout. Le résultat peut être utilisé comme n'importe quel autre fichier GIF. Voyons cela en détails avec la dernière version en français, GifBuilder 0.5 F.

CRÉATION DE L'ANIMATION

1

Utilisez n'importe quel logiciel graphique pour dessiner les images. Enregistrez chacune d'entre elles au format PICT (ou picture), GIF, TIFF ou Photoshop dans le même dossier pour faciliter leur utilisation.

2

Lancez GifBuilder. Pour partir des mêmes options que nous, choisissez Options/Options d'usine.

3

Vérifiez que les fenêtres **Images** et **Sans-titre.gif** sont ouvertes. Si tel n'est pas le cas, choisissez Fenêtres/Images et Fenêtres/Prévisualisation. Si vous avez déjà ouvert une animation, choisissez Fichier/Nouveau.

Dans GifBuilder, une seule animation peut être ouverte à la fois. Les fenêtres peuvent être ouvertes ou fermées; contrairement aux logiciels basés sur des documents, le fait de fermer une fenêtre ne fait que la cacher et ne perd jamais d'information.

4

Faites en sorte que la fenêtre **Images** de GifBuilder et le dossier contenant vos images dans le Finder soient visibles. Que GifBuilder ou le Finder soit actif n'a pas d'importance.

5

Sélectionnez vos images dans le Finder et faites-les glisser dans la fenêtre **Images**. Le nom des fichiers est affiché et la première image apparaît dans la fenêtre de prévisualisation (figure 1).

Votre Mac peut être configuré pour cacher le Finder lorsqu'une autre application est active. Cela rend le transfert des images du Finder à GifBuilder plus difficile. Vous pouvez afficher le Finder en choisissant l'option correspondante dans le tableau de bord Général.

1 GifBuilder est disponible sur Cyclope dans Applications:Imagerie.

2 Pour éviter tout malentendu, il est utile de préciser que l'auteur de GifBuilder et celui de cet article ne font qu'un.

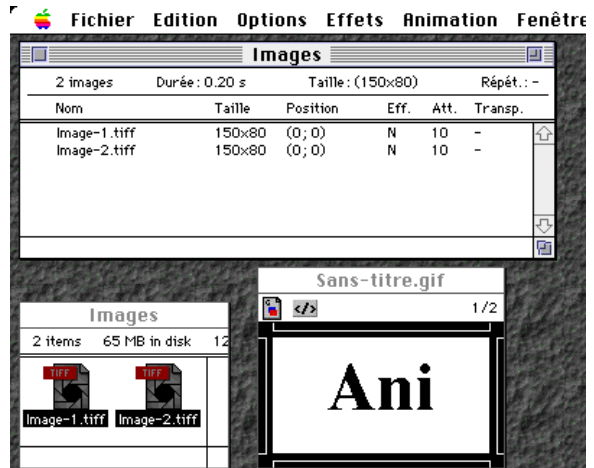


Fig. 1: les images ont été créées avec Photoshop et enregistrées en TIFF. Pour les importer dans GifBuilder, on les a fait glisser du Finder à la fenêtre «Images».

6

Suivant comment les fichiers sont triés dans le Finder, les images peuvent être dans le désordre. Il y a plusieurs moyens de les remettre dans l'ordre. Tout d'abord, vous pouvez déplacer les images dans la fenêtre **Images** en les faisant glisser. Pour sélectionner plusieurs images, enfoncez la touche Majuscule (pour une sélection continue) ou Pomme (pour une sélection quelconque). Vous pouvez aussi trier les images sélectionnées par ordre alphabétique ou inverser leur ordre avec Edition/Trier la sélection et Edition/Inverser la sélection.

A ce stade, vous pourriez choisir Fichier/Enregistrer pour avoir une animation prête à l'emploi. Mais avant de le faire, changeons quelques options.

7

Dans la fenêtre **Images**, la première colonne indique normalement le nom des images. Pour afficher une vue réduite des images, choisissez Fenêtres/Montrer les images. Les colonnes suivantes montrent les options associées à chaque image. Elles correspondent au deuxième groupe du menu Options.

Pour changer d'option, sélectionnez une ou plusieurs images, puis changez un paramètre dans le menu Options (figure 2).

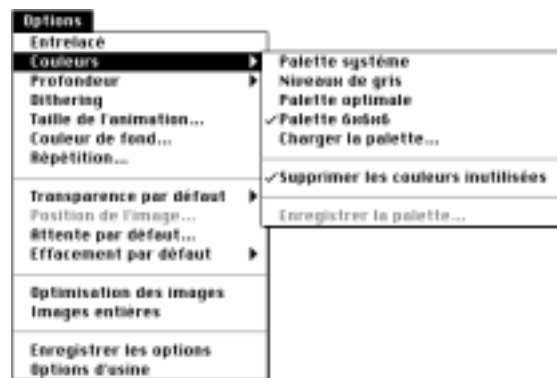


Fig. 2: menu Options

Commençons par l'attente entre chaque image. Choisissez Edition/Tout sélectionner, puis Options/Attente inter-image. Une boîte de dialogue apparaît. Évitez une attente *aussi courte que possible*, car cela risque d'aller vraiment trop vite sur un ordinateur moderne.

8

Quand l'animation est téléchargée sur le Web, sa vitesse est déterminée principalement par la vitesse du réseau. De plus, elle va commencer à s'animer avant que la page ne soit totalement affichée. Pour permettre au cybernaute de jouir de votre animation dans des conditions optimales, il vaut mieux répéter l'animation quelques fois ou indéfiniment. Choisissons cette dernière option avec Options/Répétitions.

Vous pouvez maintenant prévisualiser l'animation en choisissant Animation/Démarrer.

9

Suivant la façon dont vous avez créé les images, il se peut que les dessins ne soient pas parfaitement alignés, ce qui provoque un sautillerment. Arrêtez l'animation avec Animation/Stop. Pour aligner la seconde image sur la première, sélectionnez-la dans la fenêtre «Images», puis faites glisser l'image dans la fenêtre de prévisualisation. Si vous maintenez la touche Option enfoncée, vous verrez l'image précédente par transparence.

10

Lorsque vous créez une nouvelle animation, sa taille est déterminée par la taille et la position des images. Ce mode automatique est indiqué par des parenthèses autour de la taille de l'animation dans la fenêtre **Images**. Vous pourriez la changer avec Options/Taille de l'animation, mais il y a plus pratique. Maintenez la touche Control enfoncée, et sélectionnez dans la fenêtre de prévisualisation la zone que vous voulez conserver. Si vous n'êtes pas satisfait, cliquez à l'extérieur du rectangle pour annuler, ou déplacez-le avec Control-touche de curseur. Vous pouvez lancer l'animation pour vérifier que tout est parfait. Cliquez à l'intérieur du rectangle pour accepter la nouvelle taille. Vous pouvez toujours utiliser Edition/Annuler si vous vous ravisez...

11

Pour enregistrer l'animation, choisissez Fichier/Enregistrer. La plupart des browsers préfèrent les noms de fichier sans espace ni caractère bizarre et se terminant par «.gif». Mettez le fichier dans le même dossier que le document HTML qui l'utilisera pour faciliter la suite.

TRANSITIONS

En dessinant chaque image à la main, vous pouvez obtenir exactement la séquence que vous voulez, mais au prix de beaucoup de travail. GifBuilder peut créer lui-même les images intermédiaires pour assurer une transition fluide entre deux images.

12

Sélectionnez une ou plusieurs images. Les images de transitions se basent sur chaque image sélectionnée et l'image précédente. Ne sélectionnez la première image que si l'animation se répète.

13

Choisissez Effets/Transitions/Effeuille, l'une des animations les plus riches en options (figure 3).

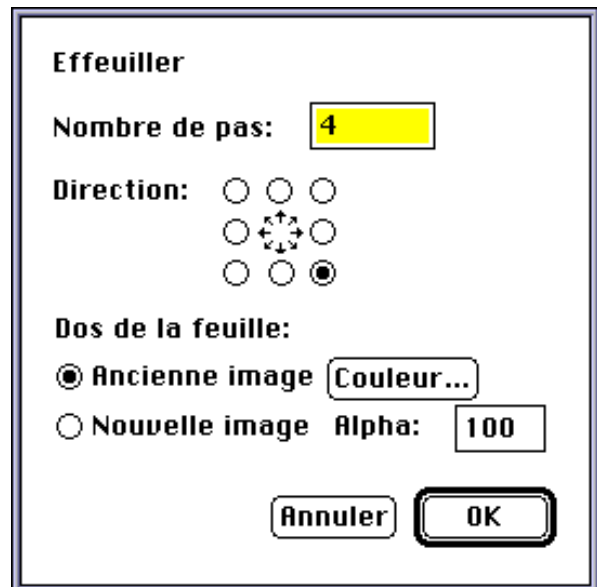


Fig. 3: options pour l'effet «Transition/Effeuille»

Le nombre de pas vaut un de plus que le nombre d'images intermédiaires. Au dos de l'image qui est enlevée, vous pouvez conserver des traces de l'ancienne image (qui serait visible à travers la feuille) ou de la nouvelle image (qui aurait déteint). Ces traces sont atténuées et combinées avec une couleur uniforme que vous pouvez choisir. L'intensité de cette couleur est spécifiée par la valeur alpha, qui peut prendre n'importe quelle valeur entre 0 (ancienne ou nouvelle image) et 255 (couleur uniforme).



Fig. 4: image intermédiaire ajoutée par la transition «Effeuille»

14

Cliquez sur OK, et les images intermédiaires sont ajoutées avec l'attente inter-image par défaut (figure 4).

Comme seules les images intermédiaires sont sélectionnées, vous pouvez facilement changer cette attente si vous le désirez.

Remarquez que les nouvelles images sont en italique. Cela signifie qu'elles n'existent qu'en mémoire vive, contrairement aux images en romain correspondant à des fichiers sur disque. Vous pouvez toujours éditer une image sur disque en double-cliquant son nom, ce qui l'ouvre dans le logiciel qui l'a créée; pour les autres, vous devez les copier-coller pour les retoucher.

15

Enregistrez le résultat en choisissant Fichier/Enregistrer.

OPTIONS AVANCÉES

Avant de voir comment insérer les animations dans un document HTML, il est utile de donner quelques conseils sur l'utilisation d'options qui ont une grande importance sur le résultat. L'entrelacement est à éviter pour les animations, car il détruit totalement leur fluidité. Les GIF, animés ou non, utilisent des palettes de 2 à 256 couleurs. Comme de nombreux ordinateurs sont aussi limités dans le nombre de couleurs qu'ils peuvent afficher simultanément, il est conseillé de se limiter à la palette standard utilisée par la plupart des browsers pour éviter des changements aux résultats souvent catastrophiques. Cette palette est nommée «Palette 6x6x6» dans GifBuilder, en référence aux six valeurs que prend chacun des canaux rouge, vert et bleu. Pour réduire le nombre d'octets transférés sur le réseau, on peut charger GifBuilder de trouver la plus petite palette qui contient le sous-ensemble des couleurs utilisées en activant Options/Couleurs/Supprimer les couleurs inutilisées. Enfin, la méthode d'effacement spécifie ce que devient chaque image lorsque la suivante est affichée. Pour des images sans transparence, Ne pas effacer est recommandé; conjointement à l'option Optimisation des images, elle permet de réduire considérablement la taille de l'animation en n'enregistrant que les parties qui ont changé d'une image à l'autre. Réafficher le fond est utile pour les animations à fond transparent. Dans ce cas, vous pouvez choisir la couleur des pixels qui sont remplacés par le fond de la page HTML.

DOCUMENT HTML

Si vous avez déjà inséré des images dans vos documents HTML, vous savez comment utiliser une animation GIF. Ajoutez le code suivant à l'endroit où vous voulez faire apparaître l'animation:

```

```

en remplaçant évidemment "animation.gif" par le nom de votre animation! Toutes les options d'img sont disponibles, telles que *alt*, *width*, *height* et *align*. Si votre éditeur supporte le glisser-déposer, glissez simplement l'icône </> à partir de la fenêtre de prévisualisation dans celle de votre éditeur. Les animations ne peuvent pas être utilisées comme fond de document, malheureusement ou heureusement selon les points de vue. Enfin, pour diffuser l'animation sur le Web, faites comme pour n'importe quel GIF: placez-le dans la hiérarchie des documents servis par votre serveur HTTP, que ce soit par une copie sur votre Mac ou par un transfert FTP en mode binaire (et surtout pas MacBinary, cause fréquente d'erreur) sur un serveur Unix.

Pour conclure, il faut encore préciser que GifBuilder peut convertir directement des animations PICS ou QuickTime créées avec un logiciel d'animation ou de dessin en trois dimensions tel qu'Infini-D (le logo de l'Institut d'automatique, visible à iawww.epfl.ch, en est l'illustration), ou être scripté pour générer automatiquement des animations complexes, par exemple avec Matlab. Pour vous mettre en appétit, voici un script AppleScript pour animer une courbe de Lissajou:

```
set s to
"x=pi*(0:0.01:2);plot(cos(2*x+phi),sin(3*x));axis
off"
repeat with i from 0 to 39
  tell application "MATLAB"
    activate
    DoScript "phi=0.05*pi*" & i & ";" & s
    Copy
  end tell
  tell application "GifBuilder"
    activate
    paste
  end tell
end repeat
```

Enfin, un dernier conseil: n'abusez pas des animations et limitez leur taille à des valeurs raisonnables. Les surfeurs australiens apprécieront. ■

NB: pour les personnes hors EPFL qui désirent se procurer GifBuilder, il est sur un serveur ftp anonyme:

```
ftp iaftp.epfl.ch
...
cd Software
```

PRÉSENTATION DES POSSIBILITÉS DE TÉLÉENSEIGNEMENT À L'EPFL

INAUGURATION DE LA SALLE CO4

par Franck Perrot, SIC

En collaboration avec des partenaires européens, les EPF ont dès 1995 expérimenté un environnement de téléenseignement multi-sites qui s'appuie sur une plateforme de communication tout ATM. Les cours de téléinformatique du Prof. Le Boudec et plusieurs conférences ont ainsi été retransmis à Zurich.



Depuis le début de cette année, le SIC a mis en place un dispositif opérationnel engageant plus encore l'EPFL sur la voie du téléenseignement.

C'est ainsi qu'en salle CO4, des événements réguliers sont dispensés entre notre école et l'ETHZ:

- cours du Prof. Kieffer (invite ETHZ) pour l'école doctorale SSC
- séminaires SSC

Des vidéoconférences sont réalisées simultanément à l'EPFL, l'ETHZ et le CSCS/SCSC de Manno. D'autres ont été organisées avec des institutions européennes. Des présentations de thèses ont déjà été expérimentées avec succès au travers de notre infrastructure. Plusieurs contacts européens nous confortent sur la justesse de notre engagement et sur le potentiel du concept utilisé.

L'EPFL rejoint ainsi la majorité des grands centres de formation d'Europe, d'Amérique du Nord, du Japon et d'Australie. Tout semble indiquer qu'une pratique nouvelle de l'enseignement verra le jour dès le siècle prochain, laquelle modifiera nos habitudes, sans que nulle part ne soit remise en question la nécessité de l'enseignement traditionnel.

Le téléenseignement conduit à l'abstraction de l'espace. Pour certaines formations, les étudiants auront (et ont déjà dans quelques pays) la possibilité de choisir un centre de formation sur un plan national et international, et ce, en restant physiquement près de leurs lieux de résidence. Les seuls critères de choix seront alors l'excellence de l'enseignement dispensé au travers de ces nouveaux outils, la renommée internationale du centre de formation et la singularité des experts invités.

L'audace et l'originalité du concept utilisé se fondent sur la qualité des retransmissions. Sur ce point, la Suisse a, quoiqu'il arrive, une avance conséquente.

Pour toutes ces raisons, nous vous invitons le **23 juin 1997 à partir de 17h00** en salle CO4 (Bât. La Coupole, 1er étage) à une démonstration multi-sites et interactive pendant laquelle vous pourrez mieux appréhender les possibilités de téléenseignement et de vidéoconférence qui vous sont dès aujourd'hui offertes.

PROGRAMME DU 23 JUIN 1997 (SIMULTANÉMENT À L'EPFL, L'ETHZ ET LE CSCS DE MANNO)

17h15 – EPFL

- Allocution du professeur et Vice-Président de l'EPFL de Werra
- Allocution du professeur J.-P. Hubaux
- Témoignage du professeur J.-Y. Le Boudec à propos de son expérience du téléenseignement

17h30 – Manno

- Allocution sur le thème "TelePoly, a chance and a challenge for CSCS/SCSC" du directeur du CSCS/SCSC J.-P. Therre

17h40 – ETHZ

- Allocution du professeur et Vice-Recteur H. Sticher

17h50

- Discussion ouverte entre les sites

18h15

- Verrée.

Si cet événement vous intéresse, prenez contact avec:

Franck Perrot, tél. 021 693 22 55

e-mail: perrot@sic.adm.epfl.ch ■

PUBLISHER

LE COUTEAU SUISSE DES LOGICIELS DE MISE EN PAGE

par Jacqueline Frey, arobasque

Cet épatant logiciel (un de mes préférés) arrive à se rendre rapidement indispensable lors d'une utilisation professionnelle ou d'une utilisation privée sur PC uniquement. Il offre de plus, par le biais de son interface intuitive et de ces nombreux assistants, l'avantage d'être plus facile d'approche que les autres logiciels de PAO. Ceci dit, Microsoft adresse ce produit aux utilisateurs comme vous et moi (en tout cas comme moi) et non pas aux professionnels de l'édition qui s'orienteront naturellement vers des programmes plus pointus (et moins farfelus) comme par exemple Xpress ou PageMaker.

Que vous ayez une brochure, une banderole pour une fête, des cartes de visite, un calendrier, un bon de commande, du papier à en-tête à réaliser, Publisher peut vous aider à le faire.

Vous trouverez ce logiciel sur Olympe/olymp. L'installation prend quelques minutes et environ 35 Mo sur le disque si vous optez pour toutes les options (polices, images, démonstrations, assistants, convertisseurs...).

Conseils:

Publisher travaillant avec le principe des assistants, n'hésitez pas à modifier vos compositions car les images et polices choisies par lui ne sont pas toujours du plus bel effet, voire même parfois moches... mais ceci n'engage que moi.

Les résultats obtenus sont beaucoup plus satisfaisants avec une imprimante couleur. Si vous n'en avez pas, utilisez des feuilles A4 de couleur soit de même grammage (80 gr /m²) que le papier blanc soit de grammage plus fort sans toutefois dépasser 120 gr/m² (risque de bourrage papier).

Vous pourrez également embellir vos compositions à l'aide d'objets WordArt, de tableaux, de formes prédéfinies (étoile, bulle...), d'objets spéciaux provenant de la bibliothèque ainsi qu'à l'aide de bordures spéciales. Mais n'en faites pas trop non plus... au risque de rapidement verser dans le *kitch*.

Pour vous donner une idée des possibilités de Publisher et pour vous mettre l'eau à la bouche, voici quelques idées de réalisations possibles. J'ai commencé tout d'abord par les créations dites professionnelles afin que cet article ait l'air sérieux... nous passerons ensuite à des sujets plus ludiques.

Bien que ce logiciel soit relativement facile à utiliser, des explications et des exercices sont parfois indispensables pour bien saisir le fonctionnement de ce logiciel. Pour cela, j'ai le plaisir de vous informer que le SIC a

mis sur pied un cours de 3 demi-journées destiné aux utilisateurs débutant sur Publisher. Les prochains cours ont été fixés après la rentrée (voir page des cours).

BULLETINS

Vous êtes chargé de la réalisation d'un bulletin interne de plusieurs pages pour votre département. Publisher vous propose différentes mises en page avec des zones réservées pour du texte, des images, une petite table des matières. Vous pourrez opter pour une impression recto-verso et même rajouter une étiquette au dos de la dernière page pour inscrire l'adresse du destinataire.

PROSPECTUS

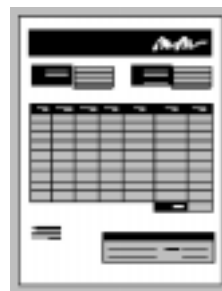


Permet de réaliser rapidement une page A4 présentant une annonce simple avec textes et logos.

BROCHURES

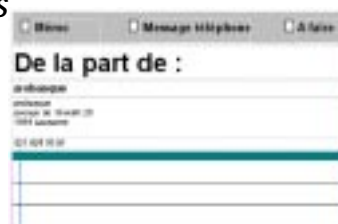
Permet de réaliser et d'imprimer une composition sur une page A4 laquelle sera ensuite pliée en 2 ou 3.

FORMULAIRES



Permet de réaliser toutes sortes de formulaires tels que factures, notes de frais, bulletins de commande, relevés d'heures, etc.

EN-TÊTES



Pour réaliser des en-têtes de lettre, de fax, et des notes internes.

PANNEAUX

Aide à la réalisation d'affiches style panneau de porte (horaire d'ouverture d'un magasin, parcage interdit, etc.)

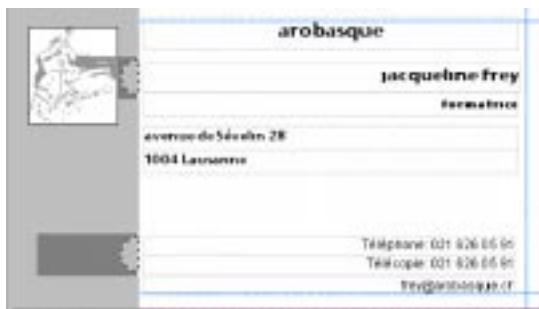
CALENDRIERS



Très utile. Construit automatiquement des calendriers par mois ou par année en plus de 10 langues.

Voilà pour les utilisations dites professionnelles. Pour se décontracter un peu, les sujets qui suivent s'adressent plutôt à tout ceux qui passent leurs dimanches de pluie à pianoter sur leur ordinateur.

CARTES DE VISITE



Pour réaliser des cartes de visite. Publisher imprime plusieurs cartes sur une page A4. A vous ensuite de couper le papier à l'aide d'un massicot ou d'un cutter. C'est d'ailleurs le plus gros du boulot !

Conseil : il existe dans le commerce des feuilles A4 déjà prédécoupées au format carte de visite.

ENVELOPPES

Permet d'imprimer des enveloppes avec texte et logo.

CARTES DE VŒUX ET INVITATIONS



Publisher vous propose des cartes de vœux pour différentes occasions avec en prime des propositions de tex-

tes pour ceux qui sont à cours d'idées. Idem pour les cartons d'invitation qui seront remplis des quatre côtés (côtés extérieurs et intérieurs) et ensuite pliés...

ÉTIQUETTES



Pour réaliser des étiquettes de bocaux, de disquettes, de CD, de vidéo et des étiquettes d'expédition.

BANDEROLES

Très chouette pour réaliser de longues banderoles pour un anniversaire ou toute autre occasion. S'imprime sur plusieurs pages A4 que l'on assemble ensuite avec du scotch.

CURRICULUM VITAE

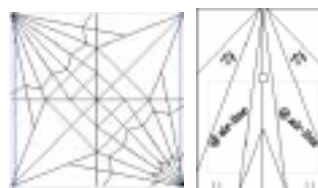
Propose deux types de CV : l'un chronologique et l'autre thématique avec ensuite plusieurs mises en page possibles.

COMPOSITIONS SPÉCIALES



Sous cette rubrique, vous trouverez différentes réalisations telles que carte de menu, étiquette pour cadeaux, certificat, fiche de recettes, etc.

AVIONS ET ORIGAMIS



Permet de réaliser quelques avions en papier avec logos et plis imprimés et instructions de pliage. Pour les origamis même idée que les avions mais nécessite plus d'habileté.

Sympa pour les enfants, surtout les dimanches de pluie.

Voilà j'espère que cet article vous aura donné quelques informations utiles sur Publisher et peut être même aura-t-il titillé votre fibre créatrice. Toute l'équipe d'arobasque vous souhaite à tous d'excellentes vacances et un bel été ensoleillé. ■

FORMATION

Les cours ci-dessous sont ouverts à tous, membres ou non de l'EPFL. Pour le personnel de l'EPFL, le SIC se charge des frais de cours.

Inscriptions et renseignements (matin uniquement):

Josiane Scalfo, SIC-EPFL, CP 121, 1015 Lausanne

☎ 021 693 2244 - Fax: 021 693 2220

E-mail: scalfo@sic.adm.epfl.ch

Pour tout changement, consultez aussi les News, ou le serveur:

<http://sawwww.epfl.ch/SIC/SA/cours/cours.html>

COURS SUR MACINTOSH

Cycle de base complet A + B (13 demi-jours)

Cycle de base A «logiciels standards»

Introduction au Macintosh, à ClarisDraw 1.0, Internet (présentation d'Internet, Intranet • Netscape et navigation sur le Web • Recherche et moteurs de recherche • Bookmarks), Word 6.0, Excel 5.0, FileMaker Pro 3.0.

N° 4151 A (7 demi-jours)
14 & 15.07.97 08h15 - 17h00
& 16, 17 & 18.07.97 08h15 - 12h00

N° 4154 A (7 demi-jours)
11 & 12.08.97 08h15 - 17h00
& 13, 14 & 15.08.97 08h15 - 12h00

N° 4157 A (7 demi-jours)
09, 11, 16, 18, 23, 25 & 30.09.97 13h30 - 17h15

Cycle de base B «communication»

Introduction à l'utilisation des réseaux, Internet (Présentation d'Internet et d'Intranet plus poussée • Configuration du produit • FTP - transferts de fichiers • Netscape et navigation sur le Web • Présentation d'HTML), Messagerie & Astuces pratiques du système.

N° 4151 B (6 demi-jours)
21 & 22.07.97 08h15 - 17h00
& 23 & 25.07.97 08h15 - 12h00

N° 4154 B (6 demi-jours)
18.08.97 08h15 - 17h00
& 19, 20, 21 & 22.08.97 08h15 - 12h00

N° 4157 B (6 demi-jours)
07, 09, 14, 16, 21 & 23.10.97 13h30 - 17h15

BASES DE DONNÉES

FileMaker Pro 3.0 avancé (5 demi-jours)
N° 4153 04, 05, 06, 07 & 08.08.97 08h15 - 12h00
N° 4160 30.09, 02, 07, 09 & 14.10.97 08h15 - 12h00

METTEUR EN PAGE

PageMaker 6.5 (3 demi-jours)
N° 4177 10.09.97 08h15 - 17h15
& 12.09.97 08h15 - 12h00

Prérequis: connaissances approfondies de Word et d'un logiciel de dessin!

PRÉSENTATION

PowerPoint 4.0 avancé, niv. 1 (2 demi-jours)
N° 4175 04 & 05.08.97 13h30 - 17h15
N° 4178 15.09.97 08h15 - 17h15

PowerPoint 4.0 avancé, niv. 2 (1 jour)
N° 4140 23.06.97 08h15 - 17h15
N° 4181 29.09.97 08h15 - 17h15
Prérequis: connaissances de PowerPoint !

TABLEUR

Excel 5.0 avancé, niv. 1 (3 demi-jours)
N° 4172 28.07.97 08h15 - 17h15
& 29.07.97 08h15 - 12h00
N° 4176 04.09.97 08h15 - 12h00
& 08.09.97 08h15 - 17h15

Excel 5.0 avancé, niv. 2 (2 demi-jours)
N° 4173 29.07.97 13h30 - 17h15
& 30.07.97 08h15 - 12h00
N° 4179 17.09.97 08h15 - 17h15

Excel 5.0 macros (2 demi-jours)
N° 4174 30.07.97 13h30 - 17h15
& 31.07.97 08h15 - 12h00
N° 4180 23 & 26.09.97 08h15 - 12h00

TRAITEMENT DE TEXTE

FrameMaker 5.1 introduction (3 demi-jours)
N° 4161 28.08, 02 & 04.09.97 13h30 - 17h15

Mailing (Word - FileMaker) (1 demi-jour)
N° 4134 03.07.97 08h15 - 12h00
N° 4165 24.09.97 08h15 - 12h00
Prérequis: connaissances de base de Word et FileMaker Pro!

Word 6.0 avancé (5 demi-jours)
N° 4150 07, 08, 09, 10 & 11.07.97 08h15 - 12h00
N° 4159 27, 29.08, 01, 03 & 05.09.97 08h15 - 12h00

Word 6.0 formulaires (1 demi-jour)
N° 4155 21.08.97 13h30 - 17h15
N° 4164 25.09.97 08h15 - 12h00

Word 6.0 longs documents (2 demi-jours)
N° 4152 19 & 20.08.97 13h30 - 17h15
N° 4163 16 & 18.09.97 08h15 - 12h00

Word 6.0 trucs + astuces (1 demi-jour)
N° 4166 01.10.97 08h15 - 12h00

TRAITEMENT D'IMAGE

PhotoShop 3.05 (3 demi-jours)
N° 4182 07, 08 & 09.07.97 13h30 - 17h15
N° 4183 09, 11 & 19.09.97 08h15 - 12h00

WEB

FrontPage (3 demi-jours)
Ce cours est destiné aux personnes qui devront mettre de l'information sur les serveurs de l'École.
• les principes de base de (modèle client-serveur, Internet, hypertexte, URL) • les commandes HTML les plus importantes • les différentes méthodes pour créer/récupérer des textes pour un serveur
N° 4156 26, 28.08 & 02.09.97 08h15 - 12h00
Prérequis: avoir déjà utilisé le logiciel Mosaic ou Netscape

COURS SUR PC - WINDOWS'95

Cycle de base complet A + B (13 demi-jours)

Cycle de base A «logiciels standards»
Introduction à Windows'95, PowerPoint 97, Internet (présentation d'Internet, Intranet • Internet Explorer et navigation sur le Web • Recherche et moteurs de recherche • Bookmarks), Word 97, Excel 97, FileMaker Pro 3.0.

N° 2634 A (7 demi-jours)
07 & 08.07.97 08h15 - 12h00
& 09 & 10.07.97 08h15 - 17h00
& 11.07.97 08h15 - 12h00

N° 2636 A (7 demi-jours)
04 & 05.08.97 08h15 - 17h00
& 06, 07 & 08.08.97 08h15 - 12h00

N° 2637 A (7 demi-jours)
26, 28.08, 02, 04, 09, 11
& 16.09.97 08h15 - 12h00

Cycle de base B «communication»
Introduction à l'utilisation des réseaux, Internet (Présentation d'Internet et d'Intranet plus poussée • Configuration du produit • FTP - transferts de fichiers • Internet Explorer et navigation sur le Web • Présentation d'HTML), Messagerie & Astuces pratiques de Windows '95.

N° 2634 B (6 demi-jours)
14 & 15.07.97 08h15 - 12h00
& 16.07.97 08h15 - 17h00
& 17 & 18.07.97 08h15 - 12h00

N° 2637 B (6 demi-jours)
18, 23, 25, 30.09, 02 & 07.10.97 08h15 - 12h00

Transition Macintosh - Windows 95 (1 demi-jour)
No 2664 29.07.97 13h30 - 17h15
No 2665 28.08.97 08h15 - 12h00

BASES DE DONNÉES

Access '97 introduction (2 demi-jours)
N° 2653 18.08.97 08h15 - 17h15
N° 2657 01 & 03.09.97 13h30 - 17h15

Access '97 avancé (2 jours)
N° 2646 01 & 03.07.97 08h15 - 17h15
N° 2654 19 & 20.08.97 08h15 - 17h15

Access Basic (2 demi-jours)
N° 2655 21.08.97 08h15 - 17h15

DESSIN

Designer 6.0 (2 demi-jours)
N° 2612 23 & 25.06.97 08h15 - 12h00

LANGAGE DE PROGRAMMATION

VisualBasic 5.0 intro niv. 1 (2 demi-jours)
N° 2650 11.08.97 08h15 - 17h15
N° 2659 10 & 17.09.97 13h30 - 17h15

VisualBasic 5.0 intro niv. 2 (2 jours)
N° 2651 12 & 13.08.97 08h15 - 17h15

METTEUR EN PAGE

MS Publisher (3 demi-jours)
N° 2643 06, 08 & 10.10.97 08h15 - 12h00

PRÉSENTATION

PowerPoint '97 avancé, niv. 1 (2 demi-jours)
N° 2652 14.08.97 08h15 - 17h15
N° 2658 08.09.97 08h15 - 17h15

TABLEUR

Excel '97 avancé, niv. 1 (3 demi-jours)
N° 2647 21.07.97 08h15 - 17h15
& 22.07.97 08h15 - 12h00
N° 2656 01, 03 & 10.09.97 08h15 - 12h00

Excel '97 avancé, niv. 2 (1 jour)
N° 2648 23.07.97 08h15 - 17h15
N° 2660 24.09.97 08h15 - 17h15

Excel '97 macros (2 demi-jours)
N° 2649 24.07.97 08h15 - 17h15
N° 2661 29.09 & 03.10.97 08h15 - 12h00

TRAITEMENT DE TEXTE

Word '97 avancé (5 demi-jours)
N° 2635 28.07.97 08h15 - 17h00
& 29, 30, 31.07.97 08h15 - 12h00
N° 2638 15, 18, 25, 29.09 & 02.10.97 13h30 - 17h15

Word '97 formulaires (1 demi-jour)
N° 2640 06.10.97 13h30 - 17h15

Word '97 longs documents (2 demi-jours)
N° 2639 07 & 09.10.97 13h30 - 17h15

Word'97 mailing (1 demi-jour)
N° 2614 02.07.97 08h15 - 12h00
N° 2641 13.10.97 13h30 - 17h15

Prérequis: connaissances de base de Word.

Word transition 7.0 à '97 (1 demi-jour)
N° 2642 04.09.97 13h30 - 17h15

Word'97 trucs + astuces (1 demi-jour)
N° 2615 30.06.97 08h15 - 12h00
Prérequis: connaissances de base de Word.

WEB

FrontPage (3 demi-jours)
N° 2644 15, 17 & 19.09.97 08h15 - 12h00

Labview avancé (2 jours)
N° 6009 16 & 17.10.97 08h15 - 17h15
N° 6011 13 & 14.11.97 08h15 - 17h15
N° 6013 11 & 12.12.97 08h15 - 17h15

**COURS SUR PC -
WINDOWS NT ET LABVIEW**

Les descriptifs des cours Windows NT et LabView peuvent être obtenus auprès de Jeremy Moinat, SIC-EPFL, 1015 Lausanne ou par e-mail : Jeremy.Moinat@sic.adm.epfl.ch.

Windows NT 4.0 Administration (3 jours)
N° 6151 du 21 au 23.10.97 08h15 - 17h15
N° 6152 du 02 au 04.12.97 08h15 - 17h15

Windows NT 4.0 Core Technologies (4 jours)
N° 6132 du 07 au 10.10.97 08h15 - 17h15
N° 6133 du 04 au 07.11.97 08h15 - 17h15

**Windows NT 4.0 Server 4.0
Dépannage avancé** (1 jour)
N° 6160 30.10.97 08h15 - 17h15
N° 6161 19.11.97 08h15 - 17h15

Introduction à LabView (3 jours)
N° 6008 du 13 au 15.10.97 08h15 - 17h15
N° 6010 du 10 au 12.11.97 08h15 - 17h15
N° 6012 du 08 au 10.12.97 08h15 - 17h15

COURS SUR STATIONS DE TRAVAIL

Unix introduction (1 demi-jour)
N° 3159 07.07.97 08h15-12h00
Prérequis: débutant = aucune connaissance de Unix.

CONDITIONS D'INSCRIPTION

En cas d'empêchement à suivre le(s) cours, l'élève avertira le Service informatique central au minimum une semaine à l'avance (sauf cas exceptionnel), faute de quoi le SIC se réserve le droit de facturer à son unité les frais occasionnés pour le cours.
Une confirmation parviendra à l'élève environ deux semaines avant le cours. S'il est déjà complet, l'élève sera informé de suite et son nom placé en liste d'attente. Dès qu'un cours identique sera fixé, il recevra un nouveau formulaire d'inscription.
Le SIC se réserve le droit d'annuler un cours si le nombre minimum de 4 participants n'est pas atteint ou pour des raisons indépendantes de sa volonté. Aucune compensation ne sera due par le SIC.

INSCRIPTION POUR LES COURS ORGANISÉS PAR LE SIC

Remplir une inscription par type de cours (Mac, PC, Unix, ...) et retourner à Josiane Scalfó, SIC-EPFL, 1015 Lausanne

Je, soussigné(e) Nom: Prénom:

Tél.: E-Mail: Fonction:

Institut: Dépt: Adresse:

m'engage à suivre le(s) cours dans son (leur) intégralité et à respecter l'horaire selon les conditions d'inscription:

N° du cours	Nom du cours	N° cours de remplacement	Date du cours
.....
.....

Date: Signature:

Autorisation du chef hiérarchique (nom lisible et signature):

INTÉRÊT ET SOUHAIT POUR D'AUTRES COURS

Description ou titre des cours que je souhaite voir organiser par le SIC:

.....
.....

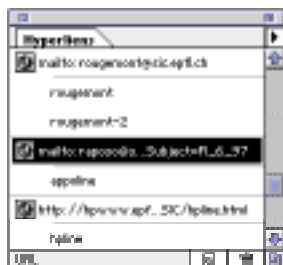
CLAVITUDES OU LES ACROBATIES DE PAGEMAKER 6.5

par Appoline Raposo de Barbosa, SIC

La nouvelle version 6.5 d'Adobe PageMaker entre véritablement dans le monde du Web. Oui, nous allons pouvoir convertir notre composition PageMaker en fichiers HTML ou PDF directement.

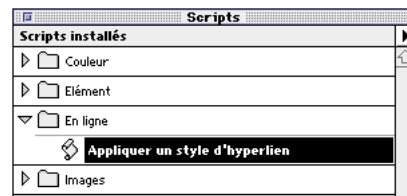
PDF

Tout d'abord préparons notre composition dans le but de la transformer en PDF en conservant les liens hypertextes internes et externes. Prenons, par exemple, cet article: lorsque je le lirai, je voudrai envoyer un message à l'auteur pour lui dire le bien que j'en pense. Par un simple clic sur son nom, une fenêtre e-mail avec son adresse s'ouvrira, rien de nouveau à cela me direz-vous, mais comment faire pour que ce lien s'insère automatiquement dans le PDF?



Dans ma composition, j'ai ouvert Fenêtres – Hyperliens et j'ai choisi Nouvelle source...; là, j'ai écrit mon adresse: mailto:raposo@sic..., cette adresse étant noircie, j'ai sélectionné dans l'article la chaîne de caractères qui sera le lien cliquable et dans cette fenêtre Hyperliens j'ai choisi Nouvelle source... que j'ai nommée appoline. La source doit apparaître sous le lien dans la fenêtre. Il est à noter que je peux avoir plusieurs sources sous le même lien.

Dans Fenêtres – Palette des Plug-ins j'ai choisi Afficher les Scripts et dans le sous-menu En ligne j'ai demandé Appliquer un style d'hyperlien que je désire noir et souligné; ce script est naturellement facultatif, à vous de choisir si vous désirez un lien visible dans le fichier PDF.



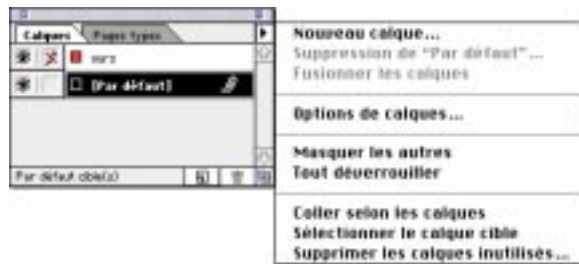
Puis, je suis revenue dans mon article et avec la main que j'ai prise dans la palette d'outils j'ai cliqué sur mon nom, oh! merveille, une fenêtre e-mail, m'étant destinée, s'est ouverte. Une première chose est faite.

Après avoir sauvé ma composition, j'ai demandé Fichier – Exporter – Adobe PDF... Là sous le bouton options PDF... j'ai choisi Conserver les liens externes.

Je peux maintenant exporter ma composition en un fichier PDF que j'ouvrirai avec Acrobat Reader 3.0 pour contrôler que tout marche et... ça marche.



tre d'isoler des éléments, de les afficher ou non, de les imprimer ou non et de les déplacer aisément sur des plans différents; le premier sur la liste étant au premier plan et le dernier en arrière plan. Il suffit de les faire glisser dans la fenêtre **Calques** pour les remonter d'un plan ou les faire descendre d'un autre. L'icône représentant un œil affichée à gauche du nom d'un calque signale que celui-ci est visible; vous pouvez aussi le masquer pour accélérer la vitesse d'affichage par un clic sur l'œil qui disparaîtra alors. Vous pouvez aussi verrouiller le calque en cliquant dans l'icône à droite de l'œil. Le crayon à droite du nom indique le calque actif.



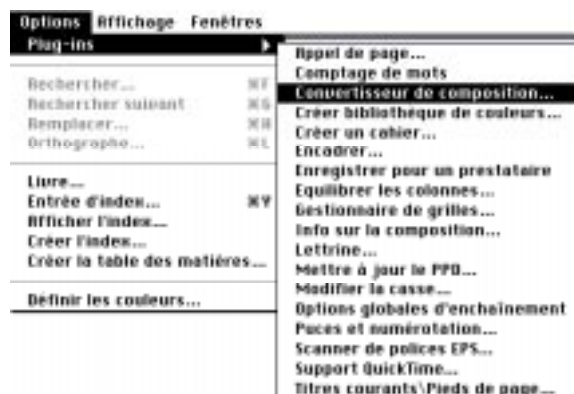
Vous pouvez naturellement utiliser un calque sur lequel vous noterez des instructions à votre imprimeur qui devra le masquer après l'avoir lu.

MODIFICATION AUTOMATIQUE DE LA DISPOSITION

Votre composition est en cours ou est terminée et vous aimeriez changer ses marges. **Fichier – Configurer le document... – modifier la mise en page** vous permet de le faire. Bien entendu, vous devrez réadapter certains éléments, mais les éléments placés contre les marges s'aligneront contre les nouvelles marges.

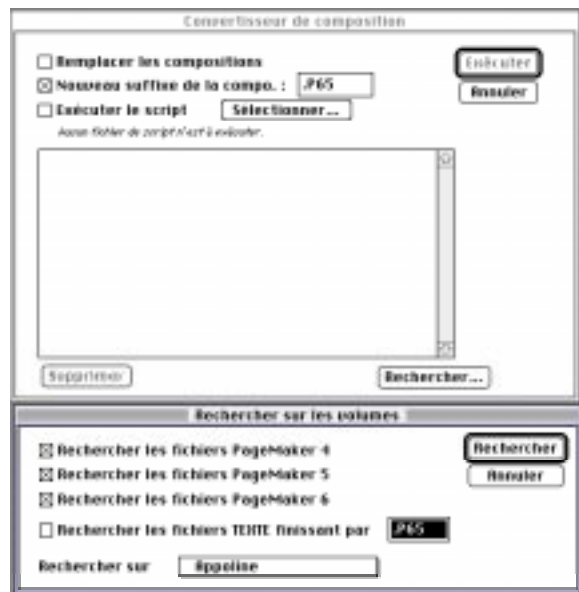
NOUVEAU PLUG-IN

CONVERTISSEUR DE COMPOSITION



Un nouveau plug-in est apparu sous le menu **Affichage – Plugs-in – Convertisseur de composition...** Ce plug-in comme son nom l'indique vous transforme vos compositions PageMaker 4, 5 ou 6 en compositions PageMaker 6.5. Vous pouvez avec cet outil vous permettre le luxe d'aller chercher automatiquement sur votre

disque toutes les anciennes versions qui s'y trouvent, de les convertir et de les sauver en leur attribuant un suffixe .P65 par exemple.



POUR CONCLURE

Adobe a su mettre à profit son expérience dans les domaines du graphisme et de la diffusion sur le Web pour nous concocter cette version 6.5 de PageMaker. Ne nous en plaignons pas! De plus, Adobe se met à concurrencer encore plus sérieusement Quark X-Press avec ses blocs de texte qui nous permettent enfin d'écrire aisément dans une forme. Je dirais que ce sont là les deux points forts de cette nouvelle version. ■

FLASH INFORMATIQUE

Les articles de ce journal ne reflètent que l'opinion de leurs auteurs. Toute reproduction, même partielle, n'est autorisée qu'avec l'accord de la rédaction et des auteurs.

Rédacteur en chef: Jacqueline Dousson, fi@sic.adm.epfl.ch
 Comité de rédaction: Jean-Daniel Bonjour, Jean-Michel Chenais, Milan Crvcinan, Laurent Desimone, Jean-Jacques Dumont, Pierre-André Haldy, Hervé Le Pezennec, François Roulet, Christophe Salzmänn, Christian Simm & Jacques Virchaux

Mise en page et graphisme: Appoline Raposo de Barbosa
 Impression: Atelier de Reprographie EPFL
 Tirage: 4000 exemplaires
 Adresse Web: <http://sawwww.epfl.ch/SIC/SA/publications/>
 Adresse: SIC-SA EPFL 1015 - Lausanne
 Téléphone: 021/693 22 46 & 22 47



ISSN 1420-7192

9 771420 719001

CALENDRIER

JUIN 97

Mardi 17	14h30	Salle Conférences SIC	MacLine — Groupe des utilisateurs Mactintosh François Roulet, ☎ 693.4590, ✉ roulet@sic.adm.epfl.ch
Jeudi 19	11h00	Cave sud du SIC	Vente de vieux matériels informatiques (<i>voir page 2 du FI5/97</i>)
	14h15	Salle Conférences SIC	PolyPC — Groupe des utilisateurs d'IBM PC et compatibles Ch. Zufferey, ☎ 693.4598, ✉ zufferey@sic.adm.epfl.ch Info sur: http://pcline.epfl.ch/pc/grp/home.htm
Lundi 23	17h00	Salle CO4	Inauguration de la salle de téléenseignement CO4 — (lire en page 23) Franck Perrot ☎ 693.2255, ✉ perrot@sic.adm.epfl.ch
Mardi 24	15h00	Salle Conférences SIC	Présentation Sun Microsystems — (lire en page 16) Krassimir Todorov, ☎ 693.2241, ✉ todorov@sic.adm.epfl.ch

JUILLET 97

Mardi 1er	14h15	Salle Conférences SIC	CTI — Commission Technique Informatique M. Reymond, ☎ 693.2210, ✉ reymond@sic.adm.epfl.ch
-----------	-------	-----------------------	--

AOÛT 97

mercredi 13	10h00	Salle Conférences SIC	HPLine — Groupe des utilisateurs de stations HP Ion Cionca, ☎ 693.4586, ✉ cionca@sic.adm.epfl.ch Info sur: http://hpwww.epfl.ch/SIC/hpline.html
-------------	-------	-----------------------	---

SEPTEMBRE 97

Mardi 10	08h30	Salle polyvalente du SIC	Comité de rédaction du FI J. Dousson, ☎ 693.2246, ✉ dousson@sic.adm.epfl.ch
Jeudi 18	14h15	Salle Conférence SIC	PolyPC — Groupe des utilisateurs d'IBM PC et compatibles Ch. Zufferey, ☎ 693.4598, ✉ zufferey@sic.adm.epfl.ch Info sur: http://pcline.epfl.ch/pc/grp/home.htm

HORAIRE D'ÉTÉ RÉCEPTION DU SIC

Nous avisons nos utilisateurs que pour la période

du 7 juillet au 22 août 1997 (y.c.)

l'horaire de la réception du SIC sera modifié comme suit:

du lundi au vendredi de 8h30 à 12h00 et de 14h00 à 16h30

Merci de votre compréhension.

Christiane Dubrit, SIC