Envoie d'un Mail en ASP

Dans cet article, nous verrons comment envoyer un mail en ASP par le serveur IIS de Microsoft. En effet, lors de vos développements de sites Internet (personnel ou professionnel), vous verrez que l'envoi de mail est un passage obligé de cet apprentissage.

En ASP, l'envoi de mail passe forcément par un composant à installer sur le serveur WEB, par enregistrement d'une DLL sur celui-ci ou par installation d'un programme.

Nous montrerons donc l'envoi du même mail par 3 composants différents et tous gratuits, il faudra donc vous renseigner auprès de votre hébergeur pour connaître la version installée chez celui-ci.

Les composants seront les suivants :

- **CDONTS** (Composant d'origine dans le Serveur Windows 2000 ayant le service SMTP installé)
- **ASPeMail** (<http://www.aspemail.com>)
- **Jmail** (<http://dimac.net>)

Le mail envoyé aura les informations suivantes stockées dans des variables (Destinataire, Origine, Message_Mail, Sujet_Mail) du type Texte, les valeurs de ces variables seront les suivantes :

Dans notre source nous initialiseront nos variables avec les valeurs de la manière suivante :

```
<%  
Dim Destinataire, Origine, Message_Mail, Sujet_Mail
```

```
Destinataire = "votre@mail.com"  
Origine = "destinataire@mail.com"  
Message_Mail = "Bonjour "& vbCrLf & " Ceci est un test d envoi de Mail en ASP"  
Message_Mail = Message_Mail & vbCrLf & "Bon Coding"& vbCrLf & "F____"  
Sujet_Mail = "Test d envoi de Mail par ASP"  
%>
```

```
De : destinataire@mail.com  
A : votre@mail.com  
Sujet : Test d envoi de Mail par ASP
```

```
Message :  
Bonjour  
Ceci est un test d envoi de Mail en ASP  
Bon Coding  
F____
```

Cela donnera le message suivant chez le destinataire :

Nous allons donc développer maintenant les 3 fonctions avec les 3 composants différents cités plus haut et l'appel pour cette fonction avec les variables définies.

1- Le Composant Cdonts :

Commençons par le composant fourni en natif avec le serveur IIS 5 ou supérieur (Windows 2000 et XP) et utilisable quand le service SMTP est installé sur ce même serveur.

```
<%  
Fonction SendmailCdon(LeDest, Expéditeur, MonSujet, MonMsg )  
Dim MonSendMail  
Set MonSendMail = Server.CreateObject("CDONTS.NewMail")  
MonSendMail.To = LeDest  
MonSendMail.From = Expéditeur  
MonSendMail.Subject = MonSujet  
MonSendMail.Body = MonMsg  
MonSendMail.Send  
Set MonSendMail = Nothing  
End Function  
%>
```

On appellera la fonction de la façon suivante :

```
<%  
SendmailCdon(Destinataire,, Origine, Sujet_Mail, Message_Mail)  
%>
```

2- Le Composant ASPeMail :

Voyons maintenant l'exemple avec le composant ASPeMail. Il faut bien sur avoir installé le produit sur le serveur WEB.

```
<%  
Fonction SendASPeMail (LeDest, Expéditeur, MonSujet, MonMsg )  
Dim MonSendMail  
Set MonSendMail = Server.CreateObject("Persits.MailSender")  
MonSendMail.From = Expéditeur  
MonSendMail.AddAddress = LeDest  
MonSendMail Subject = MonSujet  
MonSendMail.Body = MonMsg  
MonSendMail.Send  
Set MonSendMail = Nothing  
End Function  
%>
```

On appellera la fonction de la façon suivante :

```
<%  
SendASPeMail(Destinataire,, Origine, Sujet_Mail, Message_Mail)  
%>
```

On voit déjà à ce stade que les 2 fonctions sont quasiment les mêmes, mais il faut bien voir que dans cet exemple nous n'avons que le minimum du mail, c'est-à-dire, l'expéditeur le destinataire, le sujet et le message. Il existe beaucoup d'autres fonctions possibles suivant le composant choisi (Pièce jointe, copie, copie cachée, ...). Il faut voir les détails sur les sites des éditeurs.

3- Le Composant JMail :

Voyons enfin le cas du composant Jmail, l'installation du composant est aussi indispensable.

```
<%  
Fonction SendJMail (LeDest, Expediteur, MonSujet, MonMsg)  
Dim MonSendMail  
Set MonSendMail = Server.CreateObject("JMail.SMTPMail")  
    MonSendMail.ServerAddress = "localhost:25"  
    MonSendMail.Sender = Expediteur  
    MonSendMail.AddRecipient = LeDest  
    MonSendMail.Subject = MonSujet  
    MonSendMail.Body = MonMsg  
    MonSendMail.Execute  
Set MonSendMail = Nothing  
End Function  
>%
```

On appellera la fonction de la façon suivante :

```
<%  
    SendJMail(Destinataire,, Origine, Sujet_Mail, Message_Mail)  
>%
```

Conclusion :

On voit donc que les objets mail présentés ont tous la même base, il est donc très facile de passer d'une version à une autre (suivant le cas des hébergeurs et de leurs configurations).

Romelard Fabrice (Alias F____)