

PHP et Paradox

I : Qu'est-ce que PHP ?

- Un langage de script exécuté côté server via un interpréteur
- Un langage cross-plateformes (windows, unix...)
- Un langage open source, freeware, extensible librement (en langage C)
- Des accès natifs vers de nombreux SGBD (Oracle,Sybase,Informix,Interbase,MySql,PostgreSql...)

=> Un succès extraordinaire et mérité car fiabilité , puissance et simplicité sont au rendez-vous

Le trio Apache/Php/MySql est ainsi devenu très rapidement l'une des solutions les plus populaires pour mettre en place un site web dynamique.

- Autre point notable, une communauté de développeurs très importante et de très nombreuses bibliothèques de code accessibles librement.

II: Quel rapport avec Paradox ?

- Php sait attaquer nativement les fichiers DBase et , via cette fois ODBC, les tables Paradox
- Php est donc éligible pour construire des pages web dynamiques liées à des données stockées dans des tables Paradox.

III : Concrètement ?

Il faut une machine Windows avec :

- Un logiciel serveur Web , exemple Apache (freeware)
- L'interpréteur Php (freeware)
- Le BDE , Borland Database Engine (livré avec Paradox)
- Les pilotes ODBC pour Paradox (livré avec Paradox)

Une solution simple pour installer Apache , Php (et MySql) en un clic est d'utiliser le package EasyPhp , disponible sur : www.easyphp.org . Une fois le package installé, le lancement du programme (peut-être ouvert automatiquement avec Windows) démarre le serveur Apache et l'interpréteur Php (ainsi par défaut que le serveur MySql). A ce stade tout peut fonctionner.

Pourquoi faut-il le BDE ? Parce que les pilotes ODBC pour Paradox nécessitent aujourd'hui le BDE afin de permettre le fonctionnement multi-utilisateurs , via le fichier .Net partagé entre toutes les sessions BDE accédant aux tables Paradox.

Méfiance concernant l'accès natif DBase (donc sans BDE) de Php. Il ne supporte pas les index, ni le mode multi-utilisateurs. Il est en fait dédié aux fonctionnalités d'import / export.

IV : L'exemple minimal :

Soit une table Test.db dans un dossier c:\test avec les champs suivants :

CodeClient : autoincrémenté (+) , Nom : A35

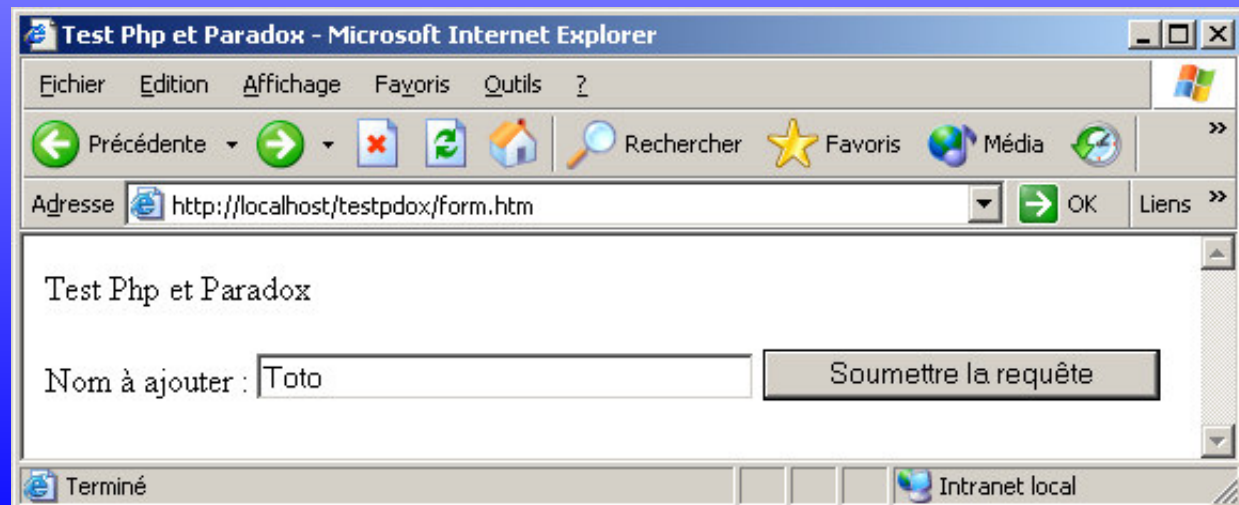
a) Créer une entrée DNS système pour le pilote ODBC fourni avec Paradox

- Panneau de configuration / ODBC Data Sources (32 bit)
(Sous XP le gestionnaire ODBC se trouve dans les outils d'administration)
- Clic sur l'onglet System DNS
- Clic sur Add et choisir le pilote INTERSOLV 3.11 32 Bit Paradox Files (*.db)
- Onglet général :
 - Data Source Name : TestPdox
 - Database Directory : c:\test
- Onglet Advanced :
 - Network Directory : Emplacement du répertoire NetDir utilisé par Paradox
 - Cocher la case International Sort

Valider tout cela

b) Un formulaire html classique :

```
<html>
<head>
<title>Test Php et Paradox</title>
</head>
<body>
Test Php et Paradox
<form method="post" action="test.php">
Nom à ajouter :<input type="text" name="fnom" size="35" maxlength="35">
<input type="submit">
</form>
</body>
</html>
```



c) Quelques lignes de Php pour insérer puis afficher les données:

```
<?php
$base="TestPdox";
$login="";
$password="";

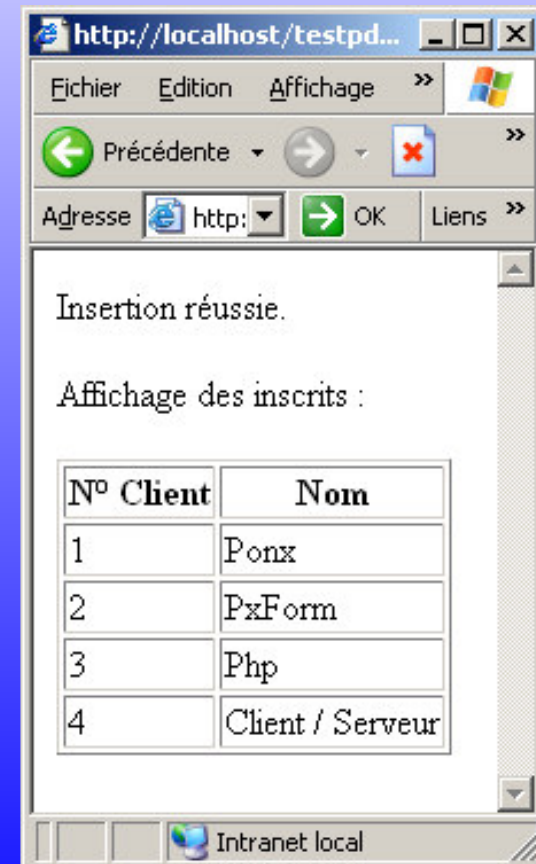
$connexion=odbc_connect($base,$login,$password) or die("Erreur de connexion");

$query="INSERT INTO Test (Nom) VALUES ('$fnom')";
odbc_do($connexion,$query) or die("Requête d'insertion incorrecte");
echo("Insertion réussie.\n");

$query="SELECT * FROM Test";
$result=odbc_do($connexion,$query) or die("Requête d'affichage incorrecte");

echo("<br><br>Affichage des inscrits :<br><br>\n");
echo("<table border='1'\>\n");
echo("<tr><th>N° Client</th><th>Nom</th></tr>\n");

while (odbc_fetch_row($result))
{
$codeclient=odbc_result($result,1);
// ou $codeclient=odbc_result($result,"CodeClient");
$nom=odbc_result($result,2);
echo("<tr><td>$codeclient</td><td>$nom</td></tr>\n");
}
echo("</table>\n");
?>
```



V : Les problèmes à résoudre

a) Insertion du caractère apostrophe :

Oui cela provoque une erreur car la chaîne SQL est terminée trop tôt. Php , par son paramétrage des Magic quotes (voir le paramètre magic_quote_gpc du fichier c:\windows\php.ini) peut escaper automatiquement pour nous les caractères à problèmes dans les données entrantes (en Get, POST ou Cookie) mais malheureusement le pilote ODBC de Paradox ne reconnaît pas \', il attend " (Double quote).

Solution : Une petite fonction PrepareDonneeToParadox

```
function PrepareDonneeToParadox($data)
{
    if (get_magic_quotes_gpc()) $data = stripslashes($data);
    $result="";
    for ($i=0;$i<strlen($data);$i++)
    {
        $l=substr($data,$i,1);
        if ($l<>'') $result .= $l; else $result .= ''.$l;
    }
    $result = trim($result);
    return $result;
}
```

b) Insertion de dates :

Les dates doivent être passées au pilote ODBC sous le format **{d'aaaa-mm-jj}**

```
function FormatDateToParadox($date)
{
  if ((!isset($date)) or ($date==""))
    return "";
  else
  {
    $dateCut = explode('/', $date);
    $newDate = "{d'". $dateCut[2]."-". $dateCut[1]."-". $dateCut[0]."'"}";
    return $newDate;
  }
}
```

```
function FormatDateFromParadox($date)
{
  if ((!isset($date)) or ($date==""))
    return "";
  else
  {
    $dateCut = explode('-', $date);
    $newDate = $dateCut[2]."/". $dateCut[1]."/". $dateCut[0];
    return $newDate;
  }
}
```


c) Insertion automatique Date + Heure :

On peut le faire également côté Paradox en créant un champ DateTime avec pour défaut : **NOW**

d) Utiliser Date et DateTime dans les requêtes SQL :

Equivalent des EXTRACT :

SELECT {fn month(MaDate)} FROM... ou plus simple : SELECT month(MaDate) FROM...
Importance des fonctions scalaires disponibles (voir documentation ODBC)

Pour une comparaison sur une colonne date :

```
WHERE MaDate = {d '2003-10-24'}
```

Pour une comparaison sur une colonne datetime :

```
WHERE MaDate = {dt '2003-10-24'}
```

e) Utilisation de caractères accentués :

Attention ! Il faut des tables Paradox en ANSI INTL ou ANSI ASCII

f) Autres points :

Les champs alpha renvoyés conservent les blancs. Penser à les traiter par trim() en SQL ou Php

Les requêtes LIKE sont casse sensitive (contrairement à MySQL) , utiliser UPPER() des 2 côtés (Attention UPPER ne fonctionnera pas sur un champ mémo...)

Les champs mémos sont bien reconnus mais pas les mémos formatés (driver 3.11)
Selon la documentation ils sont reconnus par les versions plus récentes 4.x et 5.x (non livrés avec Pdox)

Le driver supporterait le mot de passe principal des tables Paradox mais pas les secondaires via une chaîne de connexion. Pas testé.

La fonction odbc_num_rows retourne -1 pour les SELECT avec Paradox (Ok pour INSERT UPDATE et DELETE) , utiliser SELECT COUNT(*) ...

g) Compteur autoincrémenté

Il n'est semble-t-il pas possible de verrouiller une table comme en MySQL (LOCK TABLE.) mais on peut verrouiller un enregistrement avec un SELECT ...FOR UPDATE

Une petite table compteur NoPiece.db avec (Champ A10 * , NextID I)

```
function GetNextId($champ)
{
  global $connexion;
  $query="SELECT Champ,NextID FROM NoPiece WHERE Champ='$champ' FOR UPDATE OF Champ,NextID";
  $result=odbc_do($connexion,$query) or die("Requête 1 incorrecte ".odbc_errormsg());
  odbc_fetch_row($result);
  $id=odbc_result($result,"NextID");
  if ($id=="") return 0;
  $query="UPDATE NoPiece SET NextID = NextID +1 WHERE Champ='$champ'";
  $result=odbc_do($connexion,$query) or die("Requête 2 incorrecte ".odbc_errormsg());
  $id=$id+1;
  return $id;
}
```

VI : Liens utiles (en plus de msdn sur ODBC) :

a) DataDirect documentation :

<http://www.datadirect-technologies.com/download/docs/odbc/odbcref/title.htm>

b) Pear :

A connaître , au sein du projet PEAR (extensions unifiées autour de PHP) un accès direct, en lecture aux tables Paradox via le package PECL qui s'appuie sur la librairie pxlib (écrite en C):

<http://pear.php.net/manual/fr/pecl.paradox.php> et <http://pxlib.sourceforge.net>