



Developpez

Le Mag

Edition de Août - Septembre 2010.

Numéro 29.

Magazine en ligne gratuit.

Diffusion de copies conformes à l'original autorisée.

Réalisation : Alexandre Pottiez

Rédaction : la rédaction de Developpez

Contact : magazine@redaction-developpez.com

Sommaire

Java/Spring	Page 2
Android	Page 10
PHP	Page 11
AJAX	Page 25
(X)HTML/CSS	Page 27
SGBD	Page 31
Qt	Page 32
Sécurité	Page 35
Conception	Page 48
Liens	Page 50

Article Sécurité



Les pièges de l'Internet

« Arrêtez le mal avant qu'il n'existe ; calmez le désordre avant qu'il n'éclate. » Lao-tseu, VI^e siècle av. J.C.

par **Pierre Therrode**
Page 35

Article (X)HTML/CSS



Mise en page sans tableau

Ce tutoriel ne tente pas de réinventer la poudre, mais simplement de présenter les atouts d'une mise en page full CSS par rapport à une mise en page par tableaux.

par **Pascale Lambert**
Page 27

Éditorial

Avec ce nouveau numéro du magazine des professionnels de l'informatique, vous êtes sûr de bien vous préparer pour une bonne rentrée, grâce notamment à treize pages sur la sécurité pour éviter tous les pièges de la navigation Internet.

La rédaction

Tests transactionnels JUnit4 combiné avec Spring et SpringMVC

Ce billet a pour objectif d'illustrer, à l'aide d'un exemple assez complet et proche de cas réels, la mise en place des tests transactionnels pour l'ensemble des couches applicatives.

Il aborde JUnit4 enrichi avec les annotations de Spring 2.5+ et ses lanceurs pour exécuter facilement les tests. Des illustrations en mode transactionnel vous sont proposées à la fin de ce billet. Le framework JUnit est l'œuvre conjointe de Kent Beck (créateur de XP) et Erich Gamma (auteur des Design Patterns). Avec la version 4, JUnit tente de rattraper son retard sur TestNG tout en gardant la compatibilité avec JUnit3x ainsi qu'une parfaite intégration aux éditeurs Eclipse, Netbeans...

Avec les lanceurs de Spring, les tests deviennent plus attrayants et facilitent l'introduction des transactions par une simple annotation.

Spring encourage ainsi à adopter l'approche TDD "Test Driven Design".

L'utilisation de Spring avec JUnit4 n'exige, en aucun cas, de maîtriser Spring.

Dans l'exemple détaillé plus loin, les annotations de Spring 2.5 allègent considérablement la configuration XML.

La présentation de ce billet est divisée en trois axes :

- * créer une application Web JEE sous Eclipse/Maven2, SpringMVC et ses annotations, en particulier @Controller et @Autowired ;
- * mettre en place, via le framework JUnit4.x, des tests unitaires et des tests d'intégration couvrant l'ensemble des couches applicatives (DAO, Service et Web) ;
- * exécuter les tests en mode transactionnel.

Ces trois axes sont divisés ci-dessous en neuf étapes.

1. Introduction

La pratique des tests unitaires est l'un des principes des méthodes agiles et c'est avec JUnit4 et Spring2.5+ que les tests, en particulier ceux d'intégration, deviennent aisés. L'un des avantages des tests est d'avoir un retour (feedback) rapide et beaucoup moins cher sur les réglages à apporter au logiciel et ainsi d'anticiper les anomalies.

Le second avantage des tests (unitaires et d'intégration) est de limiter le nombre d'itérations (en phase recette/production) de mise en conformité du logiciel et, par conséquent, de réduire son coût total. Signalons qu'en phase de mise en recette/production les personnes impliquées sont de diverses compétences d'où le coût économique élevé d'une itération à ce stade !

Avant de rentrer dans le vif du sujet, rappelons un critère permettant de mesurer la suffisance des tests :

"L'investissement fait en tests doit être égal à celui passé sur le design. Et si le design répond facilement au changement alors les tests sont suffisants."

Liste des prérequis :

- connaissance des applications Web dans le monde JEE (servlet, jsp...);
- connaissance sommaire de Spring, Spring MVC

avec ses annotations ;

- connaissance sommaire de JUnit4.x avec ses annotations ;
- connaissance sommaire de la notion de transaction.

2. ETAPE 1. Création de l'application Web Spring MVC sous Maven2

Pour toute la suite, l'application Web exemple sera nommée «*spring-mvc-webapp*». C'est le nom de la servlet frontale dans le fichier *web.xml*.

Avant de revenir sur *web.xml*, nous allons créer ensemble, étape par étape, un projet sous Eclipse et Maven2 (sous Windows).

Avec la console Dos, se positionner dans le répertoire *workspace* d'Eclipse puis taper la commande :

```
projet maven
mvn archetype:create
-DarchetypeGroupId=org.apache.maven.archetypes
-DarchetypeArtifactId=maven-archetype-webapp
-DgroupId=com.netapsys.springmvc
-DartifactId=spring-mvc-webapp
```

Ce qui génère la structure du projet Maven dans le répertoire *spring-mvc-webapp*.

S'assurer enfin que le contenu du fichier généré *pom.xml* pointe sur la version java 1.5 ou plus et pour cela, vérifier dans le *pom.xml* les lignes suivantes :

```
<artifactid>maven-compiler-plugin</artifactid>
<configuration>
<source>1.5</source>
<target>1.5</target>
</configuration>
```

Il reste à faire de ce projet Maven un projet Eclipse à l'aide de la commande :

```
projet eclipse:classpath
mvn eclipse:eclipse
```

Pour finir, importer ce projet dans Eclipse. Passons à la configuration des dépendances Maven pour SpringMVC.

Nous ajoutons au fichier *pom.xml* les dépendances nécessaires à SpingMVC en insérant entre les deux tags :

```
<dependencies>
</dependencies>
```

les lignes suivantes :

```
<dependency>
<groupid>org.springframework</groupid>
<artifactid>spring</artifactid>
<version>2.5.6</version>
</dependency>
<dependency>
<groupid>org.springframework</groupid>
<artifactid>spring-web</artifactid>
<version>2.5.6</version>
</dependency>
<dependency>
<groupid>org.springframework</groupid>
<artifactid>spring-webmvc</artifactid>
<version>2.5.6</version>
</dependency>
<dependency>
<groupid>org.springframework</groupid>
<artifactid>spring-test</artifactid>
<version>2.5.6</version>
</dependency>
```

Nous ajoutons les dépendances pour JUnit et log4j en insérant également les lignes suivantes :

```
<dependency>
<groupid>junit</groupid>
<artifactid>junit</artifactid>
<version>4.4</version>
<scope>runtime</scope>
</dependency>
<dependency>
<groupid>commons-logging</groupid>
<artifactid>commons-logging</artifactid>
<version>1.1</version>
</dependency>
<dependency>
<groupid>log4j</groupid>
<artifactid>log4j</artifactid>
<version>1.2.14</version>
</dependency>
```

Revenons au *web.xml* dont voici le contenu (que le strict minimum utile à ce stade) :

```
<web-app>
<display-name>Spring MVC et JUnit4 </display-name>
<context-param>
<param-name>contextConfigLocation</param-name>
<param-value>
/WEB-INF/spring-mvc-webapp-servlet.xml,
classpath:/spring.xml
</param-value>
</context-param>
<listener>
<listener-class>
org.springframework.web.context.ContextLoad
erListener
</listener-class>
</listener>
<!-- déclare la servlet frontale centrale -->
<servlet>
<servlet-name>spring-mvc-webapp</servlet-name>
<servlet-
class>org.springframework.web.servlet.DispatcherS
ervlet</servlet-class>
<load-on-startup>1</load-on-startup>
</servlet>
<!-- Toutes les requêtes se terminant par
.html seront servies par la servlet principale
-->
<servlet-mapping>
<servlet-name>spring-mvc-webapp</servlet-
name>
<url-pattern>*.html</url-pattern>
</servlet-mapping>
</web-app>
```

Ci-après quelques brèves explications :

dans le fichier *web.xml* ci-dessus, la variable «*contextConfigLocation*» pointe, entre autres, vers le fichier "*spring.xml*" afin que le contexte de l'application Web charge aussi d'autres beans nécessaires.

Attention, en l'absence de cette indication, vous seriez en face d'exceptions difficiles à déchiffrer !

Nous poursuivons ci-dessous avec la classe *controller* de SpringMVC, qui elle est chargée de traiter les requêtes transmises par la servlet dispatcher, en réponse aux requêtes http du client de l'application Web.

Le bloc

```
<servlet>...</servlet>
```

permet d'identifier la servlet frontale de Spring MVC chargée de répondre à toutes les requêtes (*.html) d'un client de l'application Web.

La section suivante détaille le fichier «*spring-mvc-webapp-servlet.xml*», nommé ainsi conformément à la convention.

3. ETAPE 2. Configuration de Spring MVC

Le fichier «*spring-mvc-webapp-servlet.xml*» doit contenir ces lignes :

```

<?xml version="1.0" encoding="UTF-8"?>
<beans
  xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:p="http://www.springframework.org/schema/p"
  xmlns:context="http://www.springframework.org/schema/context"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-2.5.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-2.5.xsd"
  default-autowire="byName">
  <context:component-scan base-
package="com.netapsys.fr.springmvc.web"/>
  <context:annotation-config/>
  <!--
- Les controllers de cette application
fournissent une annotation @RequestMapping
- qui peut être déclarée de deux manières
différentes :
- Au niveau de la classe :
- par exemple
@RequestMapping("/addVisit.html")
- Pour ce type de controllers on peut annoter
les méthodes pour une requête Post ou Get,
- Au niveau de chaque méthodes, différents
exemples seront fournis.
-->
  <bean
class="org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter"/>
  <!--
Ceci est le view resolver, il permet de définir
la technologie de vue utilisée et comment
sélectionner une vue. Ici on prendra la solution
la plus simple elle permet de mapper le nom de la
vue retournée avec la sélection d'une jsp. Ex. si
le nom de la vue retournée est "hello" alors on
utilisera le fichier WEB-INF/jsp/hello.jsp pour
construire la vue.
-->
  <bean
class="org.springframework.web.servlet.view.InternalResourceViewResolver"
p:prefix="/WEB-INF/jsp/" p:suffix=".jsp"/>
</beans>

```

À noter la présence au début de ce fichier de l'instruction *default-autowire="byName"*. Celle-ci est systématiquement utilisée dans tous les fichiers de configuration de Spring, afin d'auto-injecter les beans de Spring en utilisant plutôt leurs noms et non leurs types (valeur par défaut).

Depuis la version 2.5, Spring a étendu l'annotation *@Resource* en introduisant l'annotation *@Autowired* pour augmenter le niveau de contrôle de l'auto-injection. L'annotation *@Autowired* devient active dès que le bean *AutowiredAnnotationBeanPostProcessor* est défini dans le fichier xml ou, encore plus simplement, dès que le namespace *'context'* est introduit avec l'instruction *<context:annotation-config/>* comme illustré ci-dessus.

Signalons que l'emploi des namespaces (par exemple *xmlns:context*) réduit énormément la verbosité de la configuration XML de Spring.

En dehors des commentaires et explications, ce fichier contient peu de lignes. Si le projet continue à grossir, ce

fichier de configuration n'évolue que très peu. Spring va scanner les packages dans *<context:component-scan>*, à la recherche des beans à utiliser.

La dernière ligne, très bien commentée, définit la vue retournée en réponse à une requête http.

Classe du Controller de spring MVC

Le code de la classe multi-controller SpringMVC nommée *ClientControllerSpringMVC.java* est comme suit :

ClientControllerSpringMVC.java

```

package com.netapsys.fr.springmvc.web;
//import
@Controller("clientControllerSpring")
public class ClientControllerSpringMVC {

  private MyService myService;

  @Autowired
  public void setMyService(MyService myService) {
    this.myService = myService;
  }

  @RequestMapping(value="/getClient.html",method
= RequestMethod.GET)
  public String getClient(
    @RequestParam(value=ATTRIBUTE_NAME,required=true) String nom,
    @RequestParam(value=ATTRIBUTE_LASTNAME,required=false) String prenom,
    ModelMap model) {

    Client client=myService.getClient(nom,
prenom);

    if(client!=null) {
      model.addAttribute("client",client);
      return SUCCESS;
    }else {
      model.addAttribute("errorMsg", "Client
"+"prenom+" "+nom+" ' inexistant");
      return ECHEC;
    }
  }

  @RequestMapping(value="/createClient.html",meth
od = RequestMethod.GET)
  public String createClient(
    @RequestParam(value=ATTRIBUTE_NAME,required=true) String nom,
    @RequestParam(value=ATTRIBUTE_LASTNAME,required=false) String prenom,
    ModelMap model) {

    Client client=myService.createClient(nom,
prenom);
    model.addAttribute("client",client);
    return SUCCESS;
  }

  @RequestMapping(value="/updateClient.html",meth
od = RequestMethod.POST)
  public String
updateClient(@ModelAttribute("client") Client
client,
  BindingResult result, SessionStatus status) {

    myService.updateClient(client.getCliId(),
client.getCliNom(), client.getCliPrenom());
    status.setComplete();
  }
}

```

```

return null;
}
}

```

Notez que la classe est annotée avec le stéréotype `@Controller`. Donc chacune de ses méthodes va être analysée par la servlet dispatcher pour traiter les requêtes (*.html) client selon la valeur de l'annotation `@RequestMapping`. Le reste des stéréotypes `@repository` ou `@Service` sont commentés dans le code.

Mises à part les annotations `@RequestMapping` et `@RequestParam`, le 'controller' ne fait qu'appeler les méthodes de la couche service détaillée à l'étape 4 ci-après.

Mais avant cela, l'étape qui suit revient sur le contenu du fichier `spring.xml`.

4. ETAPE 3. Configuration Spring des beans des couches DAO et Service

Le fichier de configuration Spring nommé `spring.xml` (pas de convention ici !) sert à déclarer les beans métier qui seront consommés par l'application Web. Il indique les packages Java à scanner afin d'auto-injecter ces beans. Il est important de noter qu'il déclare une `dataSource` pour la couche DAO.

```

<?xml version="1.0" encoding="UTF-8"?>
<beans
  xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:p="http://www.springframework.org/schema/p"
  xmlns:context="http://www.springframework.org/schema/context"
  xmlns:tx="http://www.springframework.org/schema/tx"
  xmlns:aop="http://www.springframework.org/schema/aop"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-2.5.xsd
    http://www.springframework.org/schema/tx
    http://www.springframework.org/schema/tx/spring-tx-2.5.xsd
    http://www.springframework.org/schema/aop
    http://www.springframework.org/schema/tx/spring-aop-2.5.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context-2.5.xsd"
  default-autowire="byName">
  <context:annotation-config/>
  <context:component-scan base-package="com.netapsys.fr.springmvc"/>
  <bean id="dataSource"
    class="org.apache.commons.dbcp.BasicDataSource">
    <property name="driverClassName"
      value="com.mysql.jdbc.Driver">
    </property>
    <property name="url"
      value="jdbc:mysql://localhost:3306/test">
    </property>
    <property name="username"
      value="root">
    </property>
    <property name="password"
      value="root">
    </property>
  </bean>
  <!-- Pensez à modifier

```

```

ici votre login mysql -->
</bean>
</beans>

```

Le fichier n'a pas besoin d'être commenté. La seule chose à adapter étant la définition du bean `dataSource`.

L'étape suivante présente les classes DAO qui font appel à la `dataSource`.

5. ETAPE 4. Classes DAO et Service

Couche DAO

Notez que cette couche utilise deux classes beans (POJO) d'entités (`Client.java` et `Personne.java`). Le code simple n'est pas montré ici.

Par contre, le code de l'interface `IDao.java` contient ces lignes :

```

package com.netapsys.fr.springmvc.dao;
import com.netapsys.fr.springmvc.entites.Client;
public interface IDao {
    boolean isExistId(String id);
    boolean findByName(String nom);
    Client getClient(String nom);
    Client getClient(String nom,String prenom);
    Client getClient(long id);
    Client createClient(String nom, String prenom)
    throws SQLException;
    Client updateClient(long id,String nom, String prenom);
    void deleteClient(long id);
}

```

Et l'implémentation de cette interface est faite dans la classe `DaoImpl.java` qui contient les lignes suivantes :

```

package com.netapsys.fr.springmvc.dao;
import java.sql.*;
import org.apache.log4j.Logger;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.dao.DataAccessException;
import org.springframework.jdbc.core.simple.*;
import org.springframework.stereotype.Repository;
import com.netapsys.fr.springmvc.entites.Client;
import static com.netapsys.fr.springmvc.tb.constants.Constants;

@Repository /** marqueur Spring pour auto-translation d'exception**/
public class DaoImpl implements IDao{

    final private Logger logger =
    Logger.getLogger(getClass().getName());

    private SimpleJdbcTemplate jt=null;
    private DataSource dataSource;
    @Autowired
    public void setDataSource(DataSource dataSource) {
        //Init SimpleJdbcTemplate ici si une
        //datasource
        jt=new SimpleJdbcTemplate(dataSource);
    }

    public Client getClient(String nom) {

```



```

return getClient(nom,null);
}

public Client getClient(long id) {
return getClient( String.valueOf(id) );
}

public Client getClient(String nom, String
prenom) {
String sql=SQL_REQUETE_CLIENT + " WHERE
upper(CLINOM)='"+nom.toUpperCase()+"'";

if(prenom!=null && !"".equals(prenom) )
sql+=" AND
upper(CLIPRENOM)='"+prenom.toUpperCase()+"'";

if(!findByName(nom) ) {
return null; //Si nom n existe pas
renvoie null
}

ParameterizedRowMapper<Client> mapper=new
ParameterizedRowMapper<Client>(){
public Client mapRow(ResultSet rs,int
rowNm) throws SQLException{
return populateClient(rs);
}
};

Client client=jt.queryForObject(sql, mapper);

return client;
}

private Client populateClient(final ResultSet
rs) throws SQLException {
if(rs==null) return null;
Client client=new Client();
client.setCliId (rs.getLong("cliId"));
client.setNom (rs.getString("cliNom"));
client.setPrenom (rs.getString("cliPrenom"));
return client;
}

public Client createClient( String nom, String
prenom) {
Client client=new Client();
client.setNom(nom);
client.setPrenom(prenom);
try{
jt.update(SQL_REQUETE_INSERT_CLIENT+
""+nom+ " ", "+prenom+"+"+"");
long id=jt.queryForLong("select
LAST_INSERT_ID()");
client.setCliId(id);
return client;
}catch(DataAccessException e){
logger.error( e.getMessage());
return null;
}
}

public boolean isExistId(String id) {
final String sql = SQL_REQUETE_COUNT_CLIENT+"
WHERE cliId='"+ id + "'";
int count = jt.queryForInt(sql);
return count > 0 ? true : false;
}

public boolean findByName(String nom) {
return findByName(nom,null);
}

```

```

}

public boolean findByName(final String
nom,final String prenom) {

String sql = SQL_REQUETE_COUNT_CLIENT+ "
WHERE UPPER(cliNom)='"+nom.toUpperCase() + "'";

if(prenom!=null && !"".equals(prenom) )
sql+=" AND
UPPER(CLIPRENOM)='"+prenom.toUpperCase() + "'";

int count = jt.queryForInt(sql);

return count > 0 ? true : false;
}

public void deleteClient(long id) {

final String sql= SQL_DELETE_ALL_CLIENT+"
WHERE cliId='"+ id + "'";
jt.update(sql);
}

public Client updateClient(long id, String nom,
String prenom) {
Client client=new Client();
client.setCliId(id);
client.setCliNom(nom);
client.setCliPrenom(prenom);

final String sql="update Client set
clinom='"+nom+"', cliprenom='"+prenom+" WHERE
cliId='"+id+"'";

return client;
}
}

```

L'apport de Spring est manifeste. L'emploi de SimpleJdbcTemplate simplifie l'accès par jdbc. Le mappage d'un ResultSet vers l'entité Client est immédiat.

C'est du code court, propre et simple.

Couche service

La couche service contient, en dehors de son interface identique à celle de IDao, l'implémentation nommée *MyService.java* dont voici le code :

```

package com.netapsys.fr.springmvc.service;
import
org.springframework.beans.factory.annotation.Auto
wired;
import org.springframework.stereotype.Service;
import com.netapsys.fr.springmvc.entites.Client;
import com.netapsys.fr.springmvc.dao.DaoImpl;

@Service
public class MyService implements IService{

private DaoImpl daoImpl;

@Autowired
public void setDaoImpl(DaoImpl daoImpl) {
this.daoImpl = daoImpl;
}

public Client getClient(String nom, String

```

```

prenom) {
    return daoImpl.getClient(nom, prenom);
}

public boolean isExistId(String id) {
    return daoImpl.isExistId(id);
}

public boolean findByName(String nom) {
    return daoImpl.findByName(nom);
}

public Client getClient(String nom) {
    return daoImpl.getClient(nom);
}

public Client createClient( String nom, String
prenom) throws SQLException {
    return daoImpl.createClient( nom, prenom);
}

public void deleteClient(long id) {
    daoImpl.deleteClient(id);
}

public Client getClient(long id) {
    return daoImpl.getClient(id);
}

public Client updateClient(long id, String nom,
String prenom) {
    return daoImpl.updateClient(id, nom, prenom);
}
}

```

La couche service sert à appeler les méthodes de la couche DAO. Et c'est dans celle-ci que les aspects transactionnels, les logs et les mesures de temps d'exécution sont gérés. Les règles métier spécifiques doivent être également observées dans cette couche.

Observons enfin l'auto-injection de *daoImpl* via la méthode setter comme ailleurs dans tout le code.

6. ETAPE 5. Fichiers jsp

Le fichier *index.jsp* redirige la requête vers l'url */createClient.html*.

La requête */createClient.html* est interceptée par la servlet frontale de SpringMVC, qui à son tour décide de l'action (méthode) à appeler dans le "controller" nommé "ClientControllerSpringMVC" et qui décide ensuite de la vue à rendre en réponse à cette requête.

Dans notre cas, c'est la méthode "public String createClient" de la classe "ClientControllerSpringMVC.java" qui sera appelée. Celle-ci, en cas de succès renvoie la chaîne "success" stockée dans Constants.SUCCESS de la classe utilitaire Constants. C'est cette chaîne qui permet de traduire la vue gérant la présentation de la réponse, qui est dans ce cas *success.jsp*.

Voici donc les quelques lignes de *index.jsp* :

```

<html>
<!-- index.jsp -->
<head>
<title>Spring mvc sample</title>
</head>

```

```

<body>
<%
    final String
urlAction="/createClient.html?";
    final String nom =
request.getParameter("name");
    final String prenom =
request.getParameter("lastName");

    if (nom != null && !"".equals(nom))
    {
        response.sendRedirect(request.getContextP
ath() + urlAction+"name=" + nom + "&lastName" +
prenom);
    }
    else
    {
        response.sendRedirect( request.getContext
Path() + urlAction +
"name=nom007&lastName=prenom007");
    }
%>
</body>
</html>

```

Le fichier *success.jsp* :

```

<%@ page language="java" contentType="text/html;
charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<%@ page isELIgnored="false" %>
<%@ taglib prefix="form"
uri="http://www.springframework.org/tags/form" %>
<html><!-- success.jsp -->
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=ISO-8859-1">
<title>Success:Spring MVC sample</title>
</head>
<body>
<h2>Edition de la fiche client créée </h2><br
/>
<% final String url=request.getContextPath()
+"/updateClient.html"; %>
<form:form commandName="client" action="<
%=url %%" method="post">
<table>
<tr><td></td>
<td><form:hidden path="cliId" /></td>
</tr>
<tr><td>Nom:</td>
<td><form:input tabIndex="1"
autocomplete="true" path="cliNom" /></td>
</tr>
<tr><td>Prénom:</td>
<td><form:input tabIndex="2"
autocomplete="true" path="cliPrenom" /></td>
</tr>
<tr><td colspan="2">
<input type="submit" value="Valider" />
</td>
</tr>
</table>
</form:form>
</body>
</html>

```

Dans *success.jsp*, l'action du tag "<form:form" pointe sur */updateClient.html* avec la méthode http "POST". La méthode "public String updateClient" du "controller"

renvoie constamment "null". Ainsi, la vue utilisée dans ce cas est /WEB-INF/jsp/updateClient.jsp conformément aux déclarations du fichier *spring-mvc-webapp-servlet.xml*. Le fichier updateClient.jsp contient ces lignes :

```
<%@ page language="java" contentType="text/html;
charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<%@ page isELIgnored="false" %>
<%@ taglib prefix="form"
uri="http://www.springframework.org/tags/form" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html><!-- updateClient.jsp -->
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=ISO-8859-1">
<title>Spring MVC sample</title>
</head>
<body>

<h2>Page Modification client</h2><br/>
<%

final String path=request.getContextPath();
final String
urlAction=path+"/updateClient.html";

%>

<form:form commandName="client" action="<
%=urlAction %>" method="post">
<table border=0>
<tr>
<td colspan="2"><form:hidden
path="cliId" id="id"/></td>
</tr>
<tr>
<td>Nom:</td>
<td><form:input autocomplete="true"
path="cliNom" /></td>
</tr>
<tr>
<td>Prénom:</td>
<td><form:input autocomplete="true"
path="cliPrenom" /></td>
</tr>
<tr>
<td colspan="2">
<input type="submit"
value="Valider" />
</td>
</tr>
</table>
</form:form>
</body>
</html>
```

7. ETAPE 6. Tests de l'application Web

Pour tester l'application Web ainsi complétée, nous allons lancer une console Dos, puis se positionner dans le répertoire du projet et lancer la commande :

```
mvn jetty:run
```

S'assurer que la base Mysql nommée test (contenant une table client avec trois champs cliId, cliNom et cliPrenom) est en service.

Puis lancer le navigateur Web avec l'URL

"http://localhost:8080/spring-mvc-webapp/index.jsp".

8. ETAPE 7. Classe de test du controlleur SpringMVC

La classe JUnit4, nommée "ClientControllerSpringMVCTest.java", permet de tester le "controller" de SpringMVC. Elle est constituée des lignes suivantes :

```
package com.netapsys.tests.springmvc.web.tests;
import org.apache.log4j.Logger;
import org.junit.*;
import
org.springframework.beans.factory.annotation.Auto
wired;
import
org.springframework.test.context.ContextConfigura
tion;
import
org.springframework.test.context.junit4.SpringJUN
it4ClassRunner;
import org.springframework.ui.ModelMap;
import static
com.netapsys.fr.springmvc.tb.constants.Constants;
import
com.netapsys.fr.springmvc.web.ClientControllerSpr
ingMVC;

@ContextConfiguration(locations={"classpath:/conf
ig/spring-mvc-webapp-
tests.xml","classpath:/config/spring-test.xml"})
@RunWith( SpringJUnit4ClassRunner.class)
public class ClientControllerSpringMVCTest {

final String NAME2TEST="Agent007";

final String LASTNAME2TEST="007";

private static ModelMap model;

protected ClientControllerSpringMVC
clientControllerSpring;

@Autowired
public void setClientControllerSpring(
ClientControllerSpringMVC
clientControllerSpring) {
this.clientControllerSpring =
clientControllerSpring;
}

@BeforeClass()
public static void testAvantTout() {
model = new ModelMap();
}

@AfterClass()
public static void apresTousLesTests() {
model.clear();
}

@Before public void initAvant() {
Assert.assertTrue(clientControllerSpring!
=null)
}

@Test
public void testGetClient() {
final String
str=clientControllerSpring.getClient(NAME2TEST,
LASTNAME2TEST, model);
```



```

Assert.assertEquals(SUCCESS, str);
Assert.assertTrue(model.get("client") != null);
}
}

```

Les parties importantes pour la compréhension des lanceurs de spring sont bien documentées dans le code.

L'annotation `@RunWith` définit le lanceur spring qui enrichit considérablement les tests JUnit4 avec les fonctionnalités supplémentaires (ex. l'auto-injection) offertes par Spring.

Spring offre aussi des annotations inexistantes dans JUnit4 utiles à l'exécution de ces tests.

Signalons que `@ContextConfiguration` emploie deux fichiers de configuration de SpringMVC et de Spring spécifiques aux tests. Ces deux fichiers sont identiques, à une déclaration près, aux fichiers `spring-mvc-webapp-servlet.xml` et `spring.xml` explicités auparavant. La *seule différence* est l'ajout de la déclaration de la transaction dans `spring-test.xml` :

DataSourceTransactionManager

```

<!-- transaction -->
<bean id="transactionManager"
class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
<property name="dataSource" ref="dataSource"/>
</bean>

```

Ainsi, avec le bean "transactionManager" les tests en mode transactionnel ci-après deviennent possibles comme illustré à l'étape suivante.

9. ETAPE 8. Classe de test en mode transactionnel

Voici le code de la classe `MyServiceTest.java`. Elle comporte toutes les indications pour exécuter les méthodes en mode transactionnel :

```

package com.netapsys.tests.springmvc.web.tests;
import java.util.Random;
import org.apache.log4j.Logger;
import org.junit.*;
import org.springframework.test.context.*;
import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;
import org.springframework.transaction.annotation.*;
import com.netapsys.fr.springmvc.entites.*;
import com.netapsys.fr.springmvc.service.MyService;

@ContextConfiguration(locations={"classpath:/config/spring-mvc-webapp-

```

```

tests.xml","classpath:/config/spring-test.xml"})
@RunWith( value=SpringJUnit4ClassRunner.class ) /
**indispensable**/
@TransactionConfiguration(transactionManager="transactionManager",defaultRollback=true)

```

```

public class MyServiceTest {

    private MyService myService;

    @Autowired public void
    setMyService(MyService myService) {
        this.myService = myService;
    }

    @Test @Transactional
    (propagation=Propagation.REQUIRED)
    public void testCreateClient() {
        final String nom
        =giveRandomName("Agent007test");
        final String
        prenom=giveRandomName("007test");
        Client client=null;
        client=myService.createClient( nom, prenom);
        Assert.assertTrue(client!=null);
    }

    private String giveRandomName(final String
    prefix) {
        final Random random=new Random();
        return prefix+random.nextInt(100);
    }
}

```

Pour tester, lancer la commande "mvn test".

Vous pouvez modifier le paramètre `defaultRollback` à false afin d'insérer des clients dans la base.

10. ETAPE 9. Conclusion

Combiner Spring 2.5+ et JUnit4 permet d'avoir sous la main un framework de test puissant facilitant la mise en place des tests unitaires et d'intégration.

Bien que l'apprentissage exige un léger effort, une fois ces deux frameworks maîtrisés, l'efficacité et le gain économique sont énormes et la qualité du livrable ne sera que meilleure.

Observons, en particulier, dans les classes de tests en mode transactionnel, le confort qu'apporte Spring à JUnit4. Les annotations Spring `@TransactionConfiguration` et `@Transactional` rendent les transactions à la portée de tous.

Retrouvez l'article d'Abderrazek CHINE en ligne : [Lien1](#)



L'art du développement Android

À l'aide d'exemples simples et faciles à exécuter, apprenez à développer des applications pour terminaux Android.

Smartphones, PDA et autres terminaux mobiles connaissent aujourd'hui une véritable explosion. Dans ce contexte, Android, le système d'exploitation mobile créé par Google, présente le double avantage d'être gratuit et open-source. Libre donc à tout un chacun d'en exploiter l'énorme potentiel !

Dans cet ouvrage, Mark L. Murphy, développeur et membre actif de la communauté Android, vous explique tout ce que vous avez besoin de savoir pour programmer des applications - de la création des interfaces graphiques à l'utilisation de GPS, en passant par l'accès aux services Web et bien d'autres choses encore ! Vous y trouverez une mine d'astuces et de conseils pour réaliser vos premières applications Android mais aussi pour accéder facilement aux séquences de code qui vous intéressent.

À travers des dizaines d'exemples de projets, vous assimilerez les points techniques les plus délicats et apprendrez à créer rapidement des applications convaincantes.

Critique du livre par Davy Leggieri

Mark Murphy développeur didactique et impliqué dans la communauté Android n'est plus à présenter. Il ne s'agit pas de son premier ouvrage sur Android et le sujet est parfaitement maîtrisé. Cet ouvrage se focalise sur la manipulation des éléments de l'interface Homme-Machine sous Android. Il suffit juste de faire un tour du sommaire pour s'en rendre compte.

Les explications sont claires et faciles d'accès pour peu que vous ayez déjà touché un tel framework. Les chapitres et différentes explications sont toujours bien illustrés et un exemple permet d'assimiler le ou les concepts. Le code source des exemples est également disponible sur le site de l'éditeur pour les plus fainéants.

Face aux ouvrages concurrents, vous trouverez des thèmes qui sont traités de manière plus complète et avancée, tels que les ListView qui sont détaillées sous toutes les coutures. Cet avantage est aussi un inconvénient, car du coup, le périmètre du SDK couvert par ce livre est limité. Oublié l'envoi de SMS, l'utilisation de l'accéléromètre, etc. N'oublions pas qu'en 365 pages l'exercice est difficile et il est impossible de tout présenter.

Si vous êtes un grand débutant avec Android, je vous conseille un autre ouvrage couvrant plus le SDK, afin de vous rendre compte de toutes les possibilités offertes. En revanche, si vous avez déjà eu un aperçu du framework Android et que vous désirez rafraîchir et/ou approfondir

certains concepts, ce livre est pour vous.

Au final, ce livre se dédie aux débutants/intermédiaires et les quelques lacunes formulées dans cette critique ne sont pas majeures, d'autant plus que la qualité de l'ouvrage est très bonne.

Critique du livre par Yan Verdvaïne

Ce livre est une traduction de "The Busy Coder's Guide to Android Development" écrit par Mark Murphy ([Lien2](#)) et traduit par Eric Jacoboni avec la contribution d'Arnaud Farine.

Écrit par un développeur pour un développeur, ce livre est assez agréable à lire et très bien découpé. Il va, de manière progressive, vous expliquer un à un les points importants pour bien comprendre les fondations de l'architecture Android. Les chapitres sont très explicites et réussissent à faire passer leurs messages. Pour cela, chaque chapitre se base sur un exemple simple et concret qui facilite la compréhension. Pour faciliter le lien entre les explications et le code, l'auteur a décoré ses explications de code Java, XML et d'images résultat. La traduction ne change en rien la clarté et la qualité des explications.

Lorsque l'on s'aperçoit que le livre est basé sur Android 1.5, on peut penser qu'il est déjà obsolète. Ce qui est totalement faux car les fondations de l'architecture ont très peu changé depuis et resteront stables dans le futur. Bien sûr, rien ne vaut la documentation officielle ([Lien3](#)) pour vous permettre de rester à jour.

Pour moi, le public visé est plutôt le niveau débutant/intermédiaire :

- débutant : il faut les bases de Java. Le bouquin est sur Android. Avoir pratiqué un peu Android est un plus ;
- intermédiaire : remet bien les idées en place. Un très bon compromis quand on a commencé, comme moi, "en live" ;
- Avancé : tu seras sûrement déçu. Le livre reste exclusivement sur les fondamentaux.

Pour conclure, c'est un très bon livre sur Android. Si vous voulez débiter ou revoir les fondamentaux, ce livre est pour vous. Après lecture, le cycle de vie d'une activity ou d'un service, les ressources, les layouts, les content providers... n'auront plus de secret pour vous (enfin presque).

Les codes sources du livre sont disponibles sur le site de l'éditeur ([Lien4](#)). Vous y trouverez aussi quelques extraits du livre qui vous permettront de vous faire une idée.

Retrouvez cette critique de livre sur la page livres Java : [Lien5](#)

Les liaisons UML implémentées avec PHP

UML définit plusieurs liens remarquables entre les classes. La totalité de ces liens pondérés permet de mesurer ce que l'on appelle le couplage. Savoir lire un diagramme de classes est aujourd'hui indispensable pour la conception d'une application Web. PHP ne fait pas exception à cela, son modèle objet étant très mûr et tout à fait capable. Nous allons ici présenter les différentes liaisons, leurs caractéristiques ainsi que leurs avantages/inconvénients et quand les utiliser. Association, héritage, agrégation, composition : dans cet article, nous allons voir comment utiliser ces liaisons avec PHP.

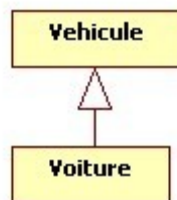
1. Héritage

Une notion que tout le monde connaît certainement, nous passerons vite sur sa définition. L'héritage se traduit par un lien fort entre deux classes dont l'enfant "est un" - "est une sorte de" vis à vis de son parent.

Si l'on ne peut pas dire cela, alors il ne s'agit pas d'un héritage. L'héritage vient rapidement à l'esprit pour résoudre tous les problèmes de conception mais il est rarement la solution et il faut l'utiliser avec modération (une profondeur d'héritage de 10 par exemple, est beaucoup trop et traduit un problème de conception).

Nous allons voir que l'héritage possède quelques astuces.

En UML, l'héritage se représente par une flèche blanche généralement tournée vers le haut.



Héritage

Une classe véhicule

```
<?php
class Vehicule { }
```

Un héritage envers véhicule

```
<?php
require_once 'path/to/Vehicule.php';
class Voiture extends Vehicule { }
```

Première notion : l'héritage est un lien fort. Une partie du code du parent va être portée dans l'enfant, celui-ci a donc besoin de la définition de la classe de son parent pour exister d'où l'instruction *require*.

1.1. Privé - protégé

Lorsqu'on parle "d'une partie du code qui va être portée", il s'agit du code non privé. La différence de visibilité entre privé et protégé n'a de sens que pour l'héritage. Tout ce qui est protégé (et donc public) va être porté du parent vers l'enfant automatiquement alors que ce qui est

privé n'est pas porté.

L'héritage et le portage de la visibilité privée en PHP

```
<?php
class A
{
    private $foo;
}
class B extends A { }
var_dump(new B);

/* affiche
object(B)#1 (1) {
    ["foo":"A":private]=>
    NULL
}
*/
```

Comme on le voit lorsque PHP dump un objet il utilise une notation "*nom-attribut*":"*nom-classe*":*private* et donc le nom de la classe qui possède l'attribut privé est bien visible. Ce mécanisme permet à PHP de sérialiser/désérialiser des objets (entre autres choses).

Preuve du non-portage de la visibilité privée

```
<?php
class A
{
    private $foo = 'A';
    protected $bar = 'A';

    public function dumpPrivate()
    {
        return $this->foo;
    }

    public function dumpProtected()
    {
        return $this->bar;
    }
}
class B extends A
{
    private $foo = 'B';
    protected $bar = 'B';
}

$b = new B;
echo $b->dumpPrivate();
echo $b->dumpProtected();
```

```

/* Affiche
A
B
*/

```

Le code ci-dessus démontre que la visibilité privée n'est pas portée dans l'héritage, ce qui est le cas de la visibilité protégée.

Bien sûr nous ne parlons pas ici de la visibilité publique accessible partout et qui est portée dans l'héritage comme la visibilité protégée.

1.2. Héritage multiple - arbre d'héritage

En PHP l'héritage multiple n'existe pas : une classe ne peut hériter que d'une et une seule classe. Ainsi l'héritage doit se réfléchir car il représente un point de blocage (attention, je crée un héritage, si dans le futur je veux hériter d'une autre classe : je ne pourrai plus).

En revanche on peut allonger l'arbre d'héritage - tant qu'on respecte la relation "est un", souvent vue comme "généralisation - spécialisation" - et tant que l'on ne bloque pas l'arbre au moyen du mot-clé *final*.

Non respect de la relation d'héritage : fatal error

```

<?php
class A
{
    public function foo() { }
}

class B extends A
{
    private function foo() { }
}

/* Affiche
Fatal error: Access level to B::foo() must be
public (as in class A)
*/

```

L'exemple ci-dessus brise la relation d'héritage car B hérite de A mais B n'est pas un A car il ne peut pas faire ce que A sait faire (on ne peut appeler la méthode *foo()* sur lui mais on le peut sur son père). Idem si l'on utilise *protected* dans le parent et *private* dans l'enfant.

La règle d'héritage est simple : l'enfant doit savoir faire tout ce que sait faire le père et éventuellement plus, mais en aucun cas moins ("est-un").

Si vous voulez bloquer l'héritage d'une classe (et donc terminer l'arbre d'héritage), PHP vous le permet au moyen du mot-clé *final*.

Utilisation de final pour bloquer l'héritage

```

<?php
class A { }
class B extends A { }
final class C extends B { }

class D extends C { } // Fatal error: Class D may
not inherit from final class (C)

```

Dans l'exemple ci-dessus, nous sommes en présence d'un arbre d'héritage d'une profondeur de trois niveaux dont le dernier niveau est final : toute tentative de prolongement de l'arbre via sa branche C se soldera par une erreur fatale (la classe D dans l'exemple), il reste possible par contre de prolonger l'arbre en recréant une branche à partir de B. L'utilisation de *final* devant une classe est rare mais peut s'avérer pratique.

final peut être utilisé devant une méthode empêchant ainsi sa surcharge dans une classe fille.

1.3. Substitution de Liskov

Le principe de substitution est clair : un objet utilisateur de A doit pouvoir manipuler un fils de A sans s'en rendre compte. Si vous brisez ce principe, PHP vous avertira mais contrairement à l'héritage, il n'enverra qu'une erreur de niveau Strict si vous avez activé ce rapport d'erreur (qui ne l'est pas par défaut en PHP 5.3 ni inférieur).

Cassure du principe de Substitution de Liskov

```

<?php
/* indispensable sinon l'erreur n'apparaît pas,
en général
E_STRICT est activé avec E_ALL dans php.ini pour
le développement */
error_reporting(E_STRICT);

class A
{
    public function foo($a, $b) { }
}

class B extends A
{
    public function foo($a, $b, $c) { }
}

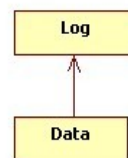
/* Affiche
Strict standards: Declaration of B::foo() should
be compatible with that of A::foo()
*/

```

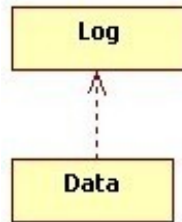
On voit clairement ici qu'un objet (C par exemple) qui utilisait A ne peut pas utiliser B à sa place car il devrait alors passer un paramètre en plus lors de l'appel de la méthode *foo()* sur celui-ci. Le principe de substitution rejoint celui de l'arbre d'héritage mais le pousse un peu plus loin.

2. Association

L'association est une utilisation ponctuelle ou permanente d'une méthode d'un objet au sein d'un autre.



Association



Association ponctuelle

Si l'utilisation est permanente, l'inclusion de la classe utilisée est obligatoire. Dans le cas contraire elle pourra être chargée à l'utilisation et le trait UML devient alors pointillé.

Une association permanente

```
<?php
require_once 'path/to/Log.php';

class Data
{
    public function __construct()
    {
        Log::write('text');
        // ...
    }
    // ...
}
```

Ici, dès la création de l'objet **Data** la classe **Log** va être utilisée, elle est donc absolument nécessaire. On aurait pu imaginer d'autres cas identiques fonctionnellement : toutes les méthodes de **Data** utilisent **Log** par exemple.

Une association ponctuelle

```
<?php
class Data
{
    public function foo(Log $l)
    {
        $l->method();
        // ...
    }
    public function bar() { }
}
```

Dans le cas ci-dessus, on peut utiliser l'objet **Data** sans jamais avoir besoin de **Log** (on peut par exemple n'appeler que la méthode *bar()* et jamais *foo()*).

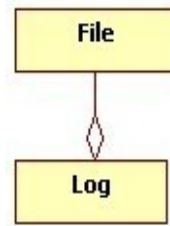
Il s'agit donc d'une association mais ponctuelle. On aurait pu imaginer d'autres exemples où une méthode de **Data** utilise la classe **Log** en l'incluant plutôt que d'en recevoir un objet via une méthode.

Rappel : PHP n'a pas besoin de connaître la définition de la classe lorsqu'on type un paramètre de méthode sur cette classe-là.

3. Agrégation

L'agrégation est une association particulière dans laquelle un objet est encapsulé dans un autre avec possibilité d'entrée-sortie. C'est une association permanente mais plus précise qui se traduit par la présence de getters/setters. On l'utilise pour effectuer de la délégation de responsabilités et de l'étude de variabilités/communalités.

En UML, la relation est caractérisée par un losange blanc qui va de l'objet agrégeant vers l'objet agrégé. On traduit l'agrégation par une relation "a un", "est composé d'un", "utilise les services d'un".



Agrégation

Une agrégation simple en PHP

```
<?php
class Log
{
    protected $file;

    public function setFile(File $file)
    {
        $this->file = $file;
    }

    public function getFile()
    {
        return $this->file;
    }

    public function event($priority, $message)
    {
        if ($this->file) {
            $this->file->write(sprintf("Priority:
%d, message: %s", $priority, $message));
        }
    }
}
class File
{
    public function write($message) { }
```

Il est très important de noter que, bien que les deux objets soient liés, ils ne le sont que si on le désire et si l'on prend la peine de les construire et les assembler à la main (principe de l'inversion de contrôle).

C'est LA différence avec la relation de composition (chapitre suivant). Dans notre exemple, l'objet agrégé **File** est partageable et les accesseurs get/set de **Log** permettent de partager ou de récupérer l'objet.

Ici il s'agit de set/get, la cardinalité est donc de 1 : l'objet **Log** ne peut posséder en lui qu'un et un seul objet **File**. En général on matérialise cela sur le schéma UML directement en indiquant la cardinalité près de la relation. Personnellement je préfère utiliser des conventions : si la cardinalité avait été de plusieurs (plusieurs objets **File** peuvent être ajoutés/retirés à **Log**), alors j'aurais écrit des méthodes *addFile()* et *removeFile()* et non get/set qui laissent suggérer "un et un seul".

3.1. Agrégateur automatisé

Créer toutes les méthodes get/set des agrégats dans une classe peut devenir pénible. À titre d'exemple voici une

utilisation des méthodes magiques et de la Reflection en PHP pour créer un agrégateur automatisé en interceptant les appels de méthodes set/get.

La classe agrégateur

```
class Agregator
{
    public function __call($meth, $args)
    {
        if (preg_match("#(g|s)et(\\w)+#", $meth,
        $matches)) {
            list(,$getOrSet,$class) = $matches;
            try {
                $param = new ReflectionProperty($this,
                $paramName = strtolower(substr($meth, 3));
            } catch(ReflectionException $e) {
                $this->fail("Unknown attribute", $e);
            }
            if (!$param->isPublic()) {
                switch ($getOrSet) {
                    case 'g':
                        return $this->{$paramName};
                    break;
                    case 's':
                        if (array_key_exists(0, $args) &&
                        $args[0] instanceof $paramName) {
                            $this->{$paramName} = $args[0];
                        }
                        return $this;
                    break;
                }
            }
            $this->fail("Aggregate attribute should not
            be public", $e);
        }
    }

    private function fail($message, Exception $e =
    null)
    {
        throw new Exception($message, null, $e);
    }
}
```

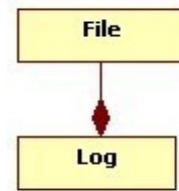
Dans cet exemple, une méthode `__call()` est utilisée pour intercepter les appels aux méthodes `get*()` ou `set*()`.

On analyse ensuite le paramètre passé à ces méthodes et on regarde si dans la classe il existe un attribut non public ayant le même nom (par exemple `setA(new A)`). Si c'est le cas on s'exécute, sinon on renvoie une erreur.

Ce type d'artefact "magique" est utilisé sous une forme déformée dans certains frameworks comme Zend Framework ou Symfony.

4. Composition

La composition est une agrégation non-partageable (non-isolable). Concrètement, cela signifie que, comme pour l'agrégation, un objet va être contenu dans un autre. La différence est qu'il n'y a aucun moyen de créer l'objet contenu, ni de le récupérer.



Composition

Exemple de composition

```
<?php
require_once 'path/to/File.php';

class Log
{
    protected $file;

    public function __construct($path)
    {
        $this->file = new File($path);
    }

    public function event($priority, $message)
    {
        $this->file->write(sprintf("Priority: %d,
        message: %s", $priority, $message));
    }
}

class File
{
    public function write($message) { }
}
```

La composition se reconnaît par rapport à l'agrégation par l'absence de `get/set` (non-partageable) et par la présence d'un "new" au sein d'une classe (ou d'une fabrique).

Dans l'exemple ci-dessus, la création d'un objet **Log** entraîne la création d'un objet **File** (celle-ci peut cependant être différée) et surtout la destruction de l'objet **Log** entraîne la destruction de l'objet **File** le composant.

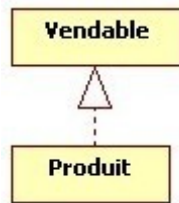
La composition est peu recommandée car elle rend le couplage trop fort, elle est l'inverse du principe d'inversion de contrôle et rend ainsi les programmes difficilement testables.

5. Implémentation

L'implémentation est une liaison qui fait intervenir une classe et une interface. Une classe implémente une interface. La relation se décrit comme "permet", "admet", "a la capacité de".

Contrairement à l'héritage, une classe peut implémenter autant d'interfaces qu'elle le souhaite. Une implémentation n'est pas un héritage, même si ça y ressemble dans la forme.

C'est un lien fort qui nécessite l'inclusion du code de l'interface dans le code de la classe l'implémentant (présence d'une instruction *require*). En UML, cela se traduit par la même flèche qu'un héritage, mais le trait est discontinu.



Implémentation

Une interface quelconque

```
<?php
interface Vendable
{
    function vendre($nb);
}
```

Implémentation

```
<?php
require_once 'path/to/Vendable.php';

class Produit implements Vendable
{
    public function vendre($nb) { }
}
```

Si une classe implémente deux interfaces définissant une même méthode, il y a alors conflit, PHP renverra une erreur fatale et vous demandera de vous arranger (renommer une méthode par exemple).

Conflit d'interfaces

```
interface Foo
{
    function baz();
}
interface Bar
{
    function baz();
}

class A implements Foo, Bar
{
    public function baz() { }
}

/* Affiche
Fatal error: Can't inherit abstract function
Bar::baz() (previously declared abstract in Foo)
*/
```

6. Lecture d'un schéma UML

Entraînons-nous à la lecture d'un diagramme de classes UML simplifié dans lequel n'apparaissent que des relations.

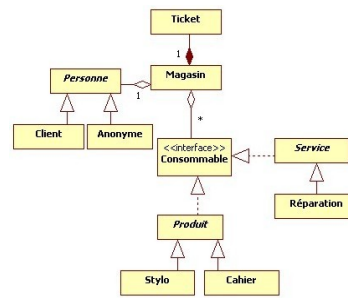


Schéma général

À la lecture de ce schéma simplifié nous pouvons affirmer (chaque mot des phrases ci-après est réfléchi) :

1. La classe "principale" est **Magasin**, on peut lui passer une **Personne** et des **Consommables** ;
2. Il existe deux types concrets de **Personne** : **Client** et **Anonyme** ;
3. Il existe deux types abstraits de **Consommable**, **Service** et **Produit** chacun ayant des sous-types concrets ;
4. Le **Magasin** gère un **Ticket**.

Attention à "un" et "des". Si c'est "un" et qu'il s'agit d'une agrégation, des méthodes get/set seront présentes. Si c'est "des", pour une agrégation toujours, alors des méthodes add/remove seront présentes. "On peut lui passer" définit l'agrégation, en revanche "Foo gère un Bar" laisse suggérer une composition de Bar dans Foo.

7. Conclusion

Les liaisons UML dans les diagrammes de classes sont très importantes. La compréhension d'un système d'information orienté objet se décompose en deux parties : la compréhension de la responsabilité de chaque objet ET la manière dont cet objet est lié au système général. Exactement comme en mécanique, ou de manière plus générale en "étude de systèmes" (dans notre cas : informatiques).

UML est un langage de modélisation permettant de dessiner des schémas représentant une partie précise d'un système informatique. Il existe des tonnes de schémas, pas que le diagramme de classes même si celui-ci est le plus utilisé.

Notre rubrique générale UML ([Lien6](#))

Les patterns de tests avancés avec PHPUnit ([Lien7](#))

Retrouvez l'article de Julien Pauli en ligne : [Lien8](#)

Maîtrise de la gestion des variables en PHP

Tout le monde sait manipuler des variables en PHP. Peut-on en dire autant quand les concepts de "copie" et de "références" entrent en jeu ? Nous allons détailler comment fonctionnent les variables au niveau interne à PHP. Cela nous permettra de comprendre très nettement les mécanismes mis en œuvre lors de l'utilisation d'objets, de références, de fonctions ou de simples variables globales.

Vous devez déjà être habitué aux variables PHP pour lire cet article et vous devez déjà avoir manipulé celles-ci, notamment les références (signe &).

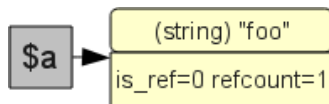
1. Héritage

PHP est faiblement typé, ce qui signifie qu'une variable peut changer de type dynamiquement durant son exécution. En interne, PHP stocke ses variables dans des structures C appelées "Zval".

Le comportement qui en ressort est une haute flexibilité des variables en PHP. La structure zval contient entre autres choses, la donnée elle-même, son type, ainsi que des informations relatives aux références. En comprenant comment PHP stocke précisément ces informations, on comprend comment, pourquoi et dans quels cas utiliser ou non des références.

Schématisons le tout :

```
<?php
$a = "foo";
```



Ici, nous voyons la variable \$a et nous voyons la zval qui représente son contenu. \$a est liée à cette zval (la flèche représente cette liaison).

Les deux indications **is_ref** et **refcount** sont très importantes. **refcount** compte le nombre de variables PHP (que l'on appelle aussi symboles) qui pointent vers la zval en question.

is_ref vaut 0 si aucune affectation par référence n'a été faite vers un symbole, 1 sinon.

Vocabulaire : un "symbole" est une variable PHP (\$a par exemple).

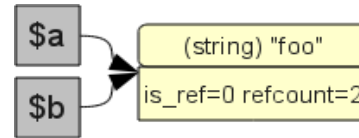
"zval" est l'unité de stockage interne élémentaire d'une donnée utilisateur. C'est une valeur en mémoire sur laquelle pointe un ou plusieurs symboles.

2. COW : Copy On Write

PHP est suffisamment intelligent pour ne pas dupliquer une variable mémoire (zval) lorsque cela n'est pas absolument nécessaire. Voyez ce qu'il se passe avec deux symboles cette fois-ci :

La mémoire n'est pas dupliquée

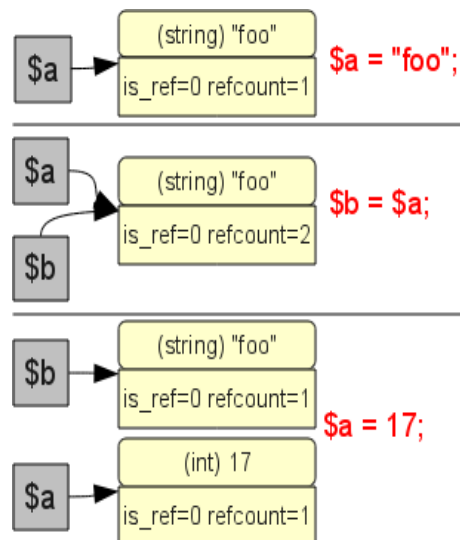
```
<?php
$a = "foo";
$b = $a;
```



La zval n'est pas dupliquée et la mémoire est donc sauvegardée. En réalité, la mémoire va être dupliquée lorsqu'un symbole va changer de valeur :

Le système CopyOnWrite

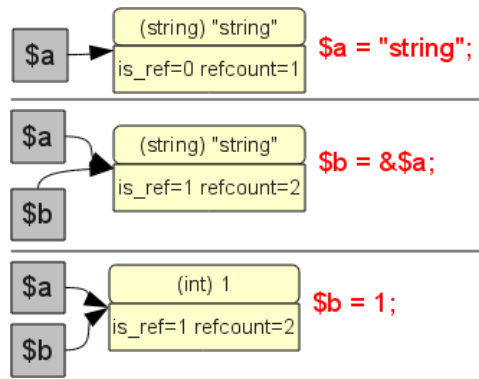
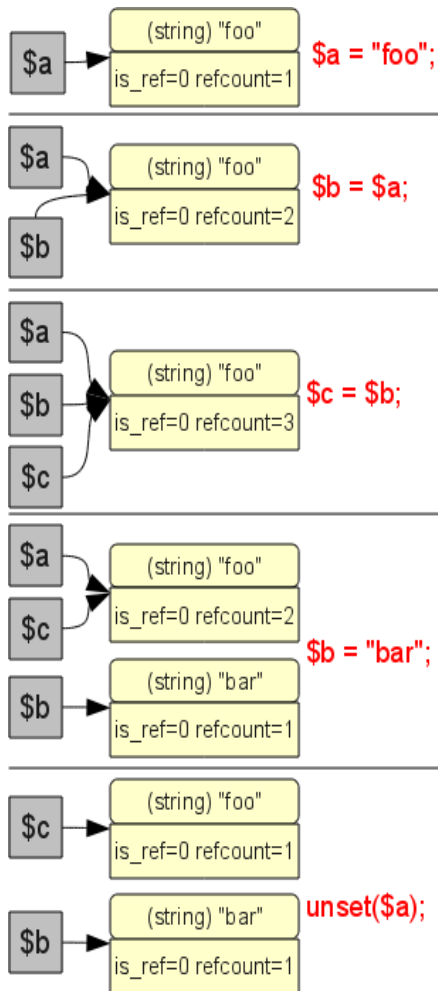
```
<?php
$a = "foo";
$b = $a;
$a = 17;
```



C'est un système appelé "Copy on Write"(COW) , littéralement "copie à l'écriture". Le **refcount** représente donc le nombre de symboles PHP pointant vers une zval mémoire. À chaque nouveau symbole pointant, le **refcount** est incrémenté et inversement. Lorsque le **refcount** atteint 0, la zval est détruite et la mémoire est libérée.

Voyons un exemple plus concret :

```
<?php
$a = "foo";
$b = $a;
$c = $b;
$b = "bar";
unset($a);
```



Voyez sur ce schéma ce qu'il s'est passé. Le **is_ref** est un booléen qui vaut 1 dès qu'un symbole PHP a été affecté par référence. **is_ref** va servir pour le COW : si un des symboles affectés par référence change sa valeur par la suite, le COW ne va pas intervenir et c'est bien la zval elle-même qui va être changée : on tombe bien sur le comportement que l'on connaît tous des références en PHP. Dès qu'un symbole est lié par référence (signe "&"), **is_ref** passe à 1. Lorsque dans le futur un des symboles change de valeur, comme **is_ref** est à 1, PHP ne duplique pas la zval mais change directement sa valeur.

Quand **unset()** est appelé sur un symbole affecté par référence, il ne se passe rien de plus que si l'affectation avait été faite par copie : la valeur de **refcount** est décrémentée de 1 et le symbole est détruit. Si son **is_ref** valait 1 et que son **refcount** tombe à 1, alors **is_ref** est passé à 0 car un seul symbole ne peut pas être une référence à lui seul (cela n'a pas de sens).

Un autre exemple intéressant, voyez plutôt :

Vous voyez le principe ? Simple et efficace n'est-il pas ?

Les valeurs de **is_ref** et **refcount** peuvent être obtenues grâce à l'extension PHP Xdebug ([Lien9](#)) et sa fonction **xdebug_debug_zval**.

Bien entendu un débogueur C comme gdb permet aussi de prendre connaissance de ces valeurs.

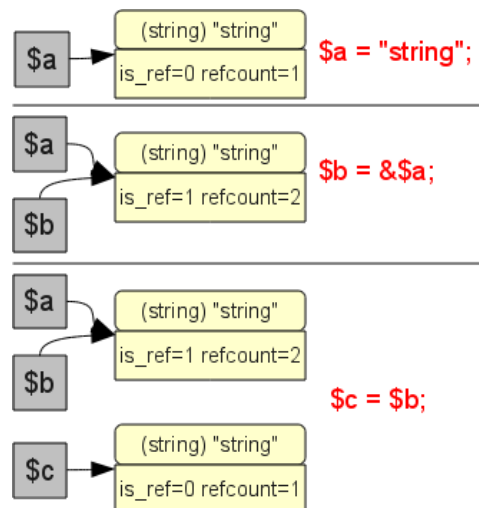
Comme on le voit, "copier" un symbole ($\$b = \a) ne va pas forcément à tous les coups copier la structure mémoire de ce symbole, donc utiliser une référence en pensant "économiser de la mémoire" est absolument faux. Cela peut même produire l'effet inverse parfois, nous verrons cela un peu plus loin.

3. Jouons avec les références

Les références PHP ne sont pas des pointeurs C et de manière générale **PHP n'est pas C**, ainsi il ne faut pas "penser mémoire" comme avec du C, lorsqu'on fait du PHP (à moins de très clairement maîtriser le fonctionnement interne de PHP concernant les variables, ce que tente d'expliquer cet article). Voyons ce qu'il se passe lorsque l'on utilise des références.

```
<?php
$a = "string";
$b = &$a;
$b = 1;
```

```
<?php
$a = "string";
$b = &$a;
$c = $b;
```



On voit clairement qu'ici l'affectation par copie $\$c = \b ; a copié la valeur de la zval, $\$c$ est donc une copie de la valeur sur laquelle pointent $\$a$ et $\$b$ (étant des références). Ainsi modifier la valeur de $\$c$ ne modifiera pas $\$a$ ni $\$b$ (qui ne pointent pas vers la même zval), mais modifier la valeur de $\$a$ modifiera la valeur de $\$b$.

Nous acceptons l'égalité vocabulaire "affectation par copie" = "affectation par valeur". On voit souvent les deux. En revanche c'est tout à fait différent de

"l'affectation par référence", comme nous venons de le démontrer.

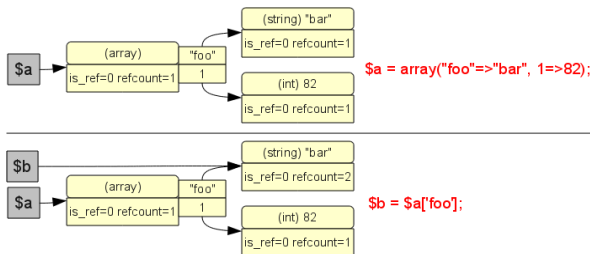
Utiliser des références en pensant économiser de la mémoire est donc faux.

La fonction **unset()** ne libère pas forcément de la mémoire : elle ne fait que détruire un symbole PHP. Si et seulement si le **refcount** de la zval sur laquelle pointait le symbole tombe à zéro, alors la mémoire est libérée, en revanche si un autre symbole pointe encore sur la zval, la mémoire ne sera pas libérée. C'est pour cela que si \$a et \$b sont deux références vers une zval, détruire \$a ne détruira en aucun cas \$b.

4. Le cas des tableaux

Le cas des tableaux est très simple et suit la logique vue jusqu'ici. La seule différence est qu'un tableau est une structure contenant des zvals.

```
<?php
$a = array("foo"=>"bar", 1 => 82);
$b = $a["foo"];
```

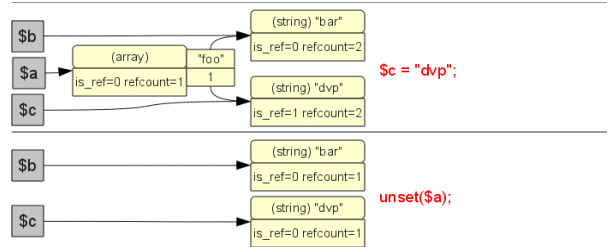
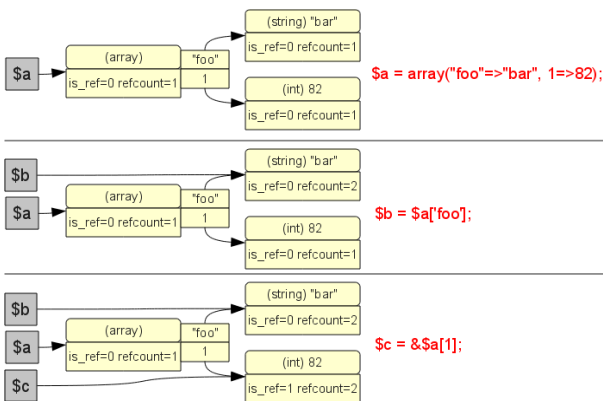


On notera que les clés d'un tableau ne sont pas des zvals, elles ne sont que des indices permettant d'accéder aux zvals et consomment ainsi très peu de mémoire.

Rappel : les clés d'un tableau ne peuvent être que des chaînes ou des entiers, si ce sont des chaînes représentant des entiers ("12"), alors PHP les transtypera en entiers.

Un autre exemple plus complexe avec des références sur un tableau :

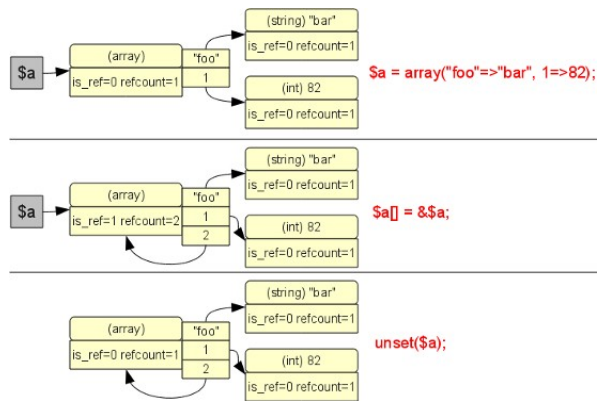
```
<?php
$a = array("foo"=>"bar", 1 => 82);
$b = $a["foo"];
$c = &$a[1];
$c = 'dvp';
unset($a);
```



4.1. Les fuites mémoire avec les tableaux

À partir de là, créer une fuite mémoire n'est pas difficile, il suffit de créer une référence circulaire : un des symboles du tableau est le tableau d'origine lui-même. Voyez plutôt:

```
<?php
$a = array("foo"=>"bar", 1 => 82);
$a[] = &$a;
unset($a);
```



Ici PHP n'a aucun moyen de nettoyer la référence circulaire : il n'existe plus aucun symbole PHP dans l'espace courant pointant vers la zval, mais comme son refcount est à 1, PHP ne la nettoie pas. Il en résulte une fuite mémoire, qui peut être gênante dans le cas de grosses variables ou dans le cas où le script PHP tourne comme démon.

Il se passe la même chose avec les objets s'auto-référençant eux-mêmes. Depuis PHP5.3, un mécanisme de ramasse-miettes a été mis en place et explique clairement le problème et sa solution. Toutes les infos se trouvent sur le manuel ([Lien10](#)).

5. Le cas des objets

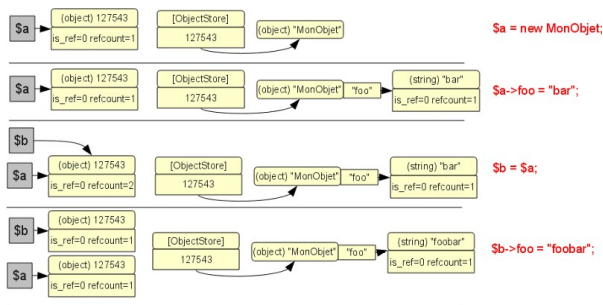
On entend souvent dire : "en PHP5, les objets sont toujours des références..." Hum, c'est faux. Il n'y a une référence que s'il y a un signe "&".

Si en PHP5 nous ne sommes plus obligés de mettre un caractère '&' devant nos variables lorsque ce sont des objets (pour passer l'objet 'lui-même'), c'est tout simplement parce qu'en interne PHP5 gère les objets de manière totalement différente de feu PHP4. En PHP4, un objet était représenté dans un conteneur zval tout à fait classique. En PHP5, ce n'est plus l'objet qui est dans la zval, mais un numéro unique servant à retrouver cet objet dans une mémoire globale dans le ZendEngine (Object store).

Ainsi, en PHP5, les affectations sur des symboles représentant des objets se font bel et bien par copie, mais on ne fait que copier un numéro d'identification qui au bout du compte, représente bien le même objet.

Comportement des objets en PHP5

```
<?php
$a = new MonObjet();
$a->foo = "bar";
$b = $a;
$b->foo = "foobar";
```



Comme on le voit sur l'image ci-dessus, le signe '=' a bien réagi comme il se doit : **is_ref** est toujours à 0, **refcount** a été décrémenté à la modification de \$b et sa zval a été dupliquée ; cependant les deux zval de \$a et \$b portent en elles le même numéro d'objet, et pointent donc sur le même objet en mémoire. Certes, il peut y avoir à un moment donné beaucoup de zval qui représentent le même objet, mais elles sont très légères en mémoire car la valeur qu'elles encapsulent n'est qu'un petit identifiant de rien du tout.

Notez aussi que PHP vous enverra une erreur E_DEPRECATED si vous affectez par référence un objet sur un symbole.

Ceci produit une erreur

```
$o =& new Stdclass;
// Deprecated: Assigning the return value of new
by reference is deprecated
```

Ce changement dans la gestion des objets, ainsi que l'introduction des Exceptions ont été les deux plus gros changements dans le code source de PHP à cette époque. Le moteur de PHP ZendEngine est passé de la version 1 (PHP4) à la version 2 (PHP5) à cette date (2004 officiellement).

6. Les fonctions et les variables

Lorsqu'on introduit le concept de fonctions, il faut veiller à deux points :

1. Le passage des variables comme arguments de fonctions ;
2. La valeur de retour de la fonction.

Les principes sont à peu de chose près les mêmes qu'en langage C avec la notion de "pile d'arguments". Une fonction en PHP possède son propre espace de variables, isolé de l'espace extérieur (généralement l'espace global). Aussi lorsqu'on passe des variables PHP en arguments à une fonction, celles-ci sont copiées sur la pile et les copies entrent dans l'espace de visibilité de la fonction (par défaut) ce qui fait que la fonction ne travaille pas sur les mêmes zval : il y a bien isolation des contextes.

Bien entendu si l'on veut qu'une fonction PHP puisse modifier le contenu de la variable qu'elle reçoit en paramètre, il faudra alors lui passer une référence (là où en

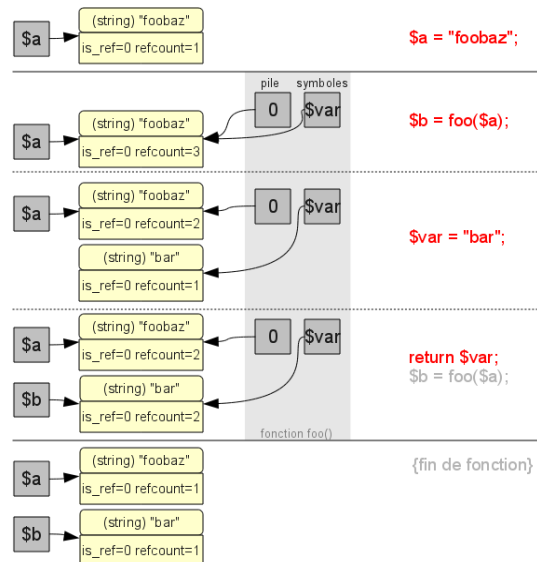
C on passe un pointeur). Ce n'est pas du tout une histoire d'économie de mémoire, comme en C (où l'on évite de passer une variable trop lourde à une fonction, de peur de déborder la pile ou dupliquer beaucoup de mémoire). Là encore, les références PHP ne sont pas exactement l'équivalent des pointeurs C, même si elles en empruntent certains comportements.

Dernier point : le retour d'une fonction est aussi par défaut copié, si l'on veut récupérer une référence, on devra le préciser.

exemple simple d'appel de fonction en PHP

```
<?php
function foo($var)
{
    $var = "bar";
    return $var;
}

$a = "foobaz";
$b = foo($a);
```



Comme on peut le noter sur ce schéma, une fonction possède une pile d'arguments et une table de symboles isolée de la table des symboles globaux. À part ça, aucun mécanisme particulier n'entre en jeu lors de l'utilisation de fonctions.

On pourra noter que la pile "consomme" la zval : elle incrémente bien son **refcount** de 1 car elle est liée à la zval. Ceci est correct car PHP nous permet de récupérer les valeurs de la pile des arguments d'une fonction au moyen des fonctions `func_get_args()` et `func_num_args()` notamment. La pile a bien une durée de vie égale à la durée de vie de la fonction, ce qui est aussi le cas pour le contexte des symboles de la fonction, sauf si le mot-clé `static` est utilisé (auquel cas les symboles de la fonction ne sont pas détruits à sa sortie, mais sa pile l'est).

Si l'on veut passer une référence à une fonction, on doit alors déclarer celle-ci dans la signature de la fonction et non à l'appel.

Un passage d'argument de fonction par référence correct

```
<?php
function foo(&$bar)
```



```
{
    // du code ici
}

$val = "value";
foo($value);
```

Un passage d'argument de fonction par référence incorrect

```
<?php
function foo($bar)
{
    // du code ici
}

$val = "value";
// Warning: Call-time pass-by-reference has been deprecated; If you would like to pass it by reference, modify the declaration of foo() [...]
foo(&$value);
```

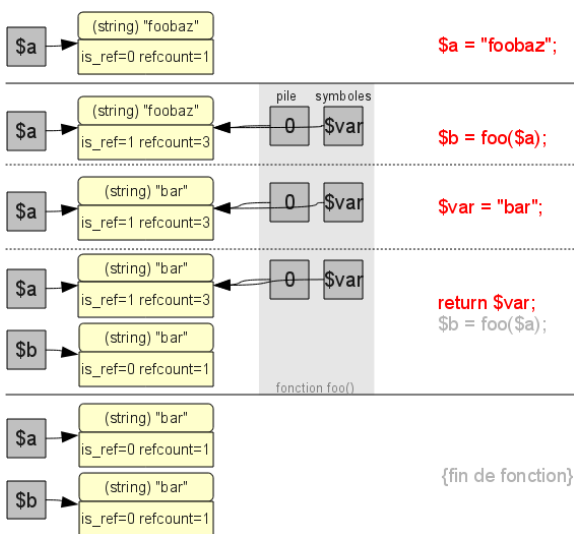
Même si l'on peut tout de même utiliser le passage par référence à l'appel de fonction (cas numéro 2) en jouant avec `allow_call_time_pass_reference=1` dans `php.ini`, c'est très fortement déconseillé car ce comportement est faux, date de PHP4, risque de disparaître dans le futur et fait beaucoup travailler le ZendEngine pour rien. Adoptez impérativement le cas numéro 1.

Que se passe-t-il lors du passage d'un argument par référence à une fonction ? Voyons cela :

Exemple de passage d'argument de fonction par référence

```
<?php
function foo(&$var)
{
    $var = "bar";
    return $var;
}

$a = "foobaz";
$b = foo($a);
```



Un comportement déjà vu : lors du passage par référence de l'argument à la fonction, `is_ref` est passé à 1. Puis la fonction change la valeur de son argument (étape 3 du schéma ci-dessus) ce qui a pour effet de changer directement la `zval` car son `is_ref` est bien à 1. Enfin, le

retour des fonctions se faisant par défaut par copie en PHP, l'étape 4 duplique bien la `zval`, phénomène que nous avons déjà constaté dans le chapitre consacré aux références.

Regardez l'exemple ci-dessus : on pourrait croire que le fait de passer par référence va nous "faire économiser de la mémoire". Regardez le résultat : il n'en est rien, bien au contraire, on se retrouve avec la même valeur utile mais dans deux conteneurs `zval` distincts.

Il reste un cas non étudié : le retour par référence. Par défaut toute valeur retournée par une fonction l'est "par copie". Si vous voulez qu'une fonction retourne une référence, il faut la déclarer comme cela :

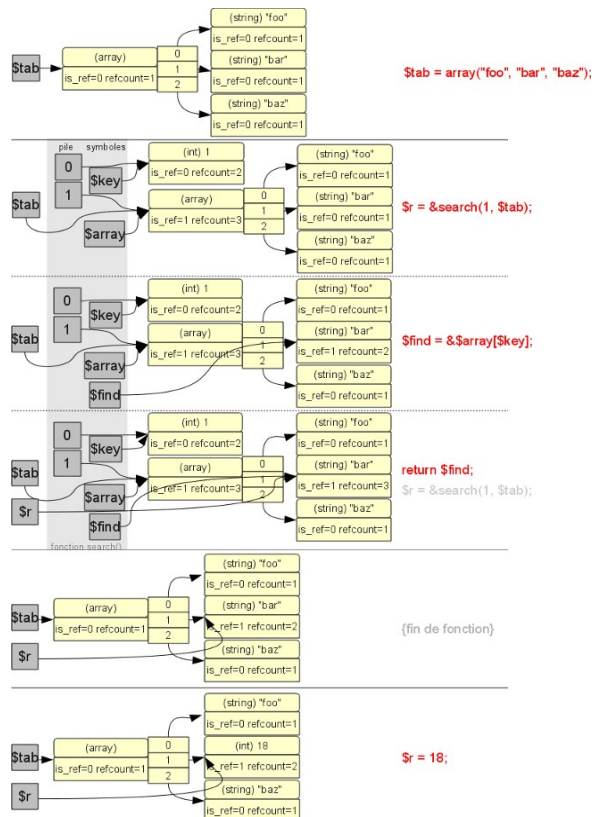
Exemple de retour d'une fonction par référence

```
<?php
function &search($key, &$array)
{
    $find = &$array[$key];
    return $find;
}

$tab = array("foo", "bar", "baz");
$r = &search(1, $tab);
$r = 18;
```

Cette fonction cherche une valeur dans un tableau et la retourne (par référence). Bien sûr, elle prend ce tableau-là comme argument par référence, sinon il sera copié lors de l'entrée dans la fonction.

Voici ce qu'il se passe en mémoire :



7. Astuces et pièges concernant les références

Le plus bel exemple que l'on donne souvent est celui-ci :

Les pièges pas faciles des références en PHP

```
<?php
$a = array(1, 2, 3);
foreach ($a as &$v) { }
foreach ($a as $v) { }

// $a vaut array(1, 2, 2) ici
```

Que s'est-il passé dans l'exemple ci-dessus ?

C'est en fait assez simple : lorsque le premier foreach utilisant une référence sur les valeurs du tableau termine, en mémoire, \$v est une référence sur la dernière case du tableau \$a (la boucle foreach ayant terminé l'itération).

Autrement dit : entre les deux foreach, \$v est liée à la dernière case du tableau (son index 2) et toute modification future de la variable \$v changera la valeur de la dernière case du tableau. Eh bien, cette modification a lieu ! C'est le deuxième foreach qui itère sur le tableau depuis le début et remplit à chaque itération \$v (donc aussi la dernière case du tableau) respectivement des valeurs 1, 2... puis 2.

Comment corriger ce code pour qu'il fonctionne correctement ? Ne pas utiliser de référence, oui certes, mais aussi utiliser unset(\$v) entre les deux foreach, ou encore utiliser autre chose que la variable \$v dans le deuxième foreach.

Comme exercice, vous pourriez tracer les schémas mémoire des zvals et leurs valeurs de is_ref et refcount sur cet exemple.

Une fonction créatrice de référence

```
<?php
function &make_ref(&$var)
{
    return $var;
}

$a = 8;
$b = &make_ref($a); // $a et $b sont des
références vers la même zval
```

Le code ci-dessus dessine une fonction qui crée une référence d'une variable sur une autre. Le piège ici est de bien retourner par référence ET affecter ce retour par référence.

Voyons un autre exemple :

```
<?php
function &foo(&$param) {
    $param = 99;
    return $param;
}

$a = 1;
// $a vaut 1
$b = &foo($a);
// $a vaut 99, $b vaut 99
$b = 2;
// $b vaut 2, $a vaut 2
```

Même exemple que ci-dessus mais en oubliant l'affectation par référence

```
<?php
<?php
function &foo(&$param) {
```

```
    $param = 99;
    return $param;
}

$a = 1;
// $a vaut 1
$b = foo($a); // il manque l'affectation par
référence &
// $a vaut 99, $b vaut 99
$b = 2;
// $b vaut 2, $a vaut toujours 99
```

Ce piège est classique, la fonction retourne bien une référence (il y a un '&' devant son nom à la déclaration), mais l'affectation de son retour se fait par valeur et non par référence. La référence est donc perdue.

Attention aussi au passage par référence, il ne se fait bien entendu que sur des variables PHP :

seule une variable peut être passée par référence

```
<?php
function take_by_ref(&$ref)
{
    // du code ici
}

take_by_ref("some string"); // Fatal Error: Only
variables can be passed by reference
```

Dernière recommandation : n'utilisez des références qu'en cas de besoin. Par exemple, la majorité des fonctions PHP prend en paramètre des copies, ceci signifie que si vous avez utilisé précédemment une référence, vous allez forcer PHP à en créer une copie.

Utilisation d'une référence qui force PHP à dupliquer la zval mémoire

```
<?php
function ma_fonction(&$var)
{
    if (strlen($var) == 20) {
        $var .= "foo";
    }
}

$a = "une chaîne de caractères";
ma_fonction($a);
```

Dans le code ci-dessus, l'appel à la fonction *strlen()* va obliger PHP à dupliquer la zval de \$var sur la pile de *strlen()*, car il ne peut utiliser la même zval puisqu'elle a été passée par référence. Ici, la référence produit un effet inverse : elle va obliger PHP à dupliquer la zval de \$var lors de l'appel à *strlen()*, alors que si \$var avait été passée par valeur à *ma_fonction()*, *strlen()* aurait simplement utilisé la même zval et ne l'aurait dupliqué que si elle avait effectué une opération d'écriture dessus (le CopyOnWrite) or ce n'est pas le cas de la fonction *strlen()*.

8. Différences entre références PHP et pointeurs C

Rappel : les références PHP ne sont pas les pointeurs C.

Le seul point commun que l'on peut noter existe lors du passage d'arguments à une fonction. Passer un pointeur C à une fonction revient à passer l'adresse mémoire de l'objet (la variable) considéré et donc fera en sorte que la

fonction travaillera sur l'objet lui-même tel qu'il est défini hors de cette fonction.

Ici le principe est semblable en PHP lorsqu'une fonction travaille sur une référence : elle travaille sur la variable réelle.

En C on utilise beaucoup plus les pointeurs dans les arguments de fonctions, qu'on n'utilise les références en PHP, car un pointeur a une taille fixe (généralement de la taille d'un int, soit en général quatre octets sur une machine 32 bits). Passer une variable lourde à une fonction en C revient à dupliquer l'espace mémoire de cette variable sur la pile de la fonction, ce qui peut mener à des débordements de pile ou à du gâchis pur et simple des ressources.

Ce n'est pas le cas en PHP, passer une référence à une fonction ne sera pas plus économe en mémoire, c'est d'ailleurs quelques fois le contraire (à cause du COW, voir exemple chapitre précédent).

C'est à peu près tout. En C, on peut effectuer des opérations sur les pointeurs, qui représentent réellement une adresse mémoire. Ce n'est pas du tout le cas en PHP où une référence ne représente absolument pas une adresse mémoire mais une structure C appelée `zval` : PHP est un niveau au-dessus.

Le signe `&` signifie "adresse de" en C et sert à obtenir un pointeur sur la variable en question ; en PHP il signifie une affectation ou un passage par référence (là où C utilisera plutôt le caractère étoile `*`) : c'est différent.

C peut effectuer des calculs (assez particuliers) sur des pointeurs (arithmétique des pointeurs) et les pointeurs sont typés (ils ont le type qu'ils représentent), là encore, rien à voir avec PHP où une référence n'est absolument pas un type mais représente "une liaison" entre deux variables PHP.

Les développeurs C qui utilisent PHP - et en ont analysé le fonctionnement - connaissent les différences énormes entre les références PHP et les pointeurs de C, tout simplement car PHP est écrit en C.

Les développeurs PHP qui ne connaissent pas le langage C entendent parler d'équivalence entre les pointeurs C et les références PHP ainsi que d'économie de mémoire : il n'en est rien, et je conseille fortement à ce type de population de regarder par eux-mêmes, d'apprendre le langage C et de lire la source de PHP.

Exemple simple d'une fonction C utilisant un pointeur

```
#include <stdlib.h>

typedef struct _mastruct{
    char* string,
    int entier
} mastruct;

char* lire_struct(mastruct* s)
{
```

```
    return s->string;
}

int main(int argc, char** argv)
{
    mastruct* s = (mastruct *)
    malloc(sizeof(mastruct));
    (*s).string = "foo";
    printf("La structure contient la chaîne
%s", lire_struct(s));
    free(s);
    return 1;
}
```

Code PHP à peu près équivalent

```
<?php
function lire_struct(array $a)
{
    return $a['string'];
}

$mastruct = array("string" => "foo", "entier" =>
0);
printf("La structure contient la chaîne %s",
lire_struct($mastruct));
```

Dans l'exemple (très simple) ci-dessus, la fonction `lire_struct()` en C prend en paramètre un pointeur afin d'éviter au programme de dupliquer l'argument sur la pile de la fonction et donc, de consommer plus de mémoire. En PHP, on a vu que le langage ne dupliquait une structure mémoire (`zval`) qu'à l'écriture sur celle-ci et non à la lecture. Dans l'exemple PHP, la fonction `lire_struct()` n'a donc aucun besoin d'utiliser une référence dans son argument : PHP ne copiera pas la variable mais incrémentera son **refcount**

9. Conclusion

Comprendre les références en PHP est essentiel, car elles font partie du langage et peuvent s'avérer utiles lorsqu'on les maîtrise bien. PHP a été construit pour être économe au maximum en mémoire et ne duplique la mémoire réservée à une variable (un symbole) que dans des cas bien précis qu'il convient de maîtriser pour tirer parti de toute la puissance du langage, et surtout pour ne pas le rendre contre-performant.

Le parallèle avec le langage C et ses pointeurs peut être fait, mais à la seule condition qu'il soit clairement compris du développeur, c'est-à-dire que le développeur maîtrise les concepts du C (ce qui malheureusement aujourd'hui est encore trop rare). PHP propose des outils pour analyser ses variables d'un point de vue interne et prendre connaissance des valeurs de **is_ref** et **refcount** : l'extension `Xdebug`. Les plus techniques pourront se pencher sur des solutions plus "ardues" comme `valgrind/calgrind` ou encore `gdb`.

Retrouvez l'article de Julien Pauli en ligne : [Lien11](#)

Les dernières news

Les points forts d'un framework PHP tel que Symfony

Modèle MVC

Le modèle « Modèle Vue Contrôleur » est très répandu dans le développement d'applications et occupe également une place importante dans le développement Web. Il

permet de structurer une application en distinguant la partie présentation d'une part et le code applicatif d'autre part, ce qui facilite le développement en équipe, la relecture et la maintenance. Dans le contexte d'une application Web, on obtient les éléments suivants :

- **Modèle** : ce sont les données manipulées par le site (les données stockées en base – correspondent au mapping ORM, que l'on abordera plus loin) ;
- **Vue** : ce sont les différentes pages du site, qui affichent les informations. Le rôle de « vue » est généralement rempli par des templates ;
- **Contrôleur** : le traitement des actions utilisateurs. Dans un contexte web, on utilise généralement un contrôleur frontal (Front-Controller) qui reçoit directement les requêtes utilisateur (URL et paramètres) et qui se charge d'exécuter le code adapté : les actions. Les correspondances entre les URL et les différentes actions sont spécifiées par le développeur et sont souvent appelées « routes » dans les frameworks PHP.

```
// Exemple de route
http://www.monsite.fr/news/list/start/10 ->
Contrôleur : news -> Action : list -> Paramètre :
start=10
```

Le modèle MVC a fait depuis longtemps ses preuves dans le monde Java avec Struts, Spring, WebWork...

Événements

La programmation événementielle est un modèle différent du modèle MVC qui est surtout pratiqué dans le développement d'applications de type « client-lourd » qui s'exécutent sur le poste de travail de l'utilisateur.

Chaque élément de la page (boutons, liens, champs texte...) dispose d'événements correspondant par exemple au clic, à la modification ou à la sélection de l'objet auxquels il est possible d'associer différentes fonctions.

Ce modèle encourage l'utilisation de composants indépendants tels que des formulaires d'authentification, ou des listes d'éléments qui intègrent les fonctions de tri et de pagination.

Templates

Les templates (gabarits) rentrent dans le processus de la séparation du code applicatif et de la présentation.

Un template est un fichier contenant essentiellement du code HTML complété de quelques commandes pour l'affichage d'informations issues du code métier. Il est également possible d'utiliser des templates différents pour l'interface générale du site et pour la mise en page des contenus. Un template peut inclure d'autres templates, ce qui permet de mutualiser certaines parties des pages.

L'intérêt des templates :

- La présentation est séparée du code applicatif : il est possible de la modifier sans se préoccuper du traitement des données et inversement ;
- La création des templates peut être confiée à des

graphistes, car elle ne nécessite que très peu de connaissances techniques PHP ;

- Il est possible de combiner des templates simples pour former un template plus complexe et ainsi réutiliser le code au maximum.

Cache

Un système de cache permet de stocker le résultat de l'affichage de certaines pages ou d'actions utilisateur afin de le réutiliser directement lors du prochain accès. La montée en charge des applications s'en trouve améliorée, et les temps d'exécution se rapprochent des pages statiques.

Il est souvent possible de déclarer un cache pour des pages entières ou uniquement pour certaines zones.

Accès aux données

Les applications Web accèdent pratiquement à chaque clic à une base de données. L'accès aux données est donc une fonctionnalité essentielle, mais également critique et gourmande en temps de développement lorsqu'on utilise uniquement les fonctions standard de PHP.

Type de base de données

Un premier problème est l'indépendance vis à vis de la base de données. Même si MySQL est la base la plus utilisée dans les applications PHP, on souhaite généralement pouvoir remplacer la base de données lorsque cela est nécessaire. Malheureusement, PHP utilise des API différentes pour chaque base et on constate que l'implémentations du langage SQL change suffisamment pour empêcher le fonctionnement de l'application avec une base différente de celle avec laquelle elle a été développée.

Un framework peut donc proposer une API unique et des fonctions de création de requêtes SQL, qui génèrent du code SQL adapté à la base de données utilisée. La simple modification d'un fichier de configuration permet alors de changer le type de base utilisé.

Mapping de relationnal object

Le framework peut également proposer une fonctionnalité d'ORM (Object Relationnal Mapping), c'est-à-dire qu'il permet de masquer la complexité du langage SQL et d'effectuer la plupart des opérations par l'intermédiaire d'objets très simples, ce qui allège significativement le travail du développeur.

Traditionnellement, un ORM demande un peu de configuration pour créer et enregistrer dans l'application des objets très simples appelés DAO (Data Access Object) qui représentent un élément d'une table donnée. La plupart des implémentations permettent de générer la configuration et les objets directement à partir de la structure de la base elle-même.

L'ORM tel que l'implémente Ruby on Rails sous le nom d'ActiveRecord ne nécessite aucune configuration et inspecte la base de données pour créer les différents objets. Le développement et la maintenance s'en trouvent

grandement simplifiés au prix d'une légère perte de performances.

```
// Insertion d'une entrée dans une table
commentaire = new Commentaire();
commentaire->setTitre( « Mon titre »);
commentaire->setCorps( « Mon commentaire »);
commentaire->save(); // Obtenir toutes
les entrées d'une table
commentaires = Commentaire.findAll();
```

L'objectif de l'ORM n'est pas de remplacer ou de masquer complètement le langage SQL, mais de prendre en charge la majorité des requêtes utilisées dans l'application. Pour les requêtes les plus complexes ou les plus critiques en vitesse d'exécution, le développeur peut choisir d'utiliser directement le langage SQL si cela est nécessaire.

Conventions

Pendant le développement d'une application, un ensemble de règles sont généralement définies concernant le nom des fichiers ou leur emplacement.

L'utilisation de conventions au niveau du framework permet d'utiliser ces règles pour configurer et lier implicitement les différents modules et classes de l'application, ce qui a pour effet de diminuer le nombre et le volume des fichiers de configuration dans l'application.

Des étapes de configuration sont seulement nécessaires lorsqu'on souhaite s'écarter des réglages par défaut.

La mise en place de conventions prend tout son intérêt lorsqu'elle est couplée à de **la génération de code**.

Génération de code

La mise en place d'un nouveau projet demande généralement la mise en place d'une structure globale et la création de nombreux fichiers.

La génération de code est utilisée pour gagner du temps grâce à l'initialisation automatique de la structure d'une application et à la création et déclaration de nouveaux éléments ou plugins ([Lien12](#)) via une simple ligne de commande.

Échafaudage

« L'échafaudage » (ou scaffolding) ajoute sans aucun développement une interface d'administration qui permet l'affichage, l'ajout, la suppression et l'édition d'éléments contenus dans une table de la base de données.

Fortement lié à la fonctionnalité d'ORM, il permet d'obtenir instantanément une interface temporaire qui pourra plus tard être remplacée par une interface plus évoluée.

En général cette interface est générée à la volée par le framework sans qu'aucun code ne soit présent dans l'application. Il est cependant possible de demander la création du code correspondant et de s'en servir de base pour le développement de la véritable interface.

Gestion des droits

Le framework peut offrir des méthodes pour définir les rôles des utilisateurs ainsi que les droits nécessaires pour exécuter chaque opération. Il se charge ensuite de vérifier les autorisations à chaque appel d'action et de bloquer l'exécution si nécessaire.

URL conviviales

Les applications Web dynamiques utilisent généralement des paramètres dans les adresses (URL) pour transmettre des informations de page en page.

Les URL obtenues sont difficilement lisibles et ne sont pas très adaptées à l'indexation effectuée par les moteurs de recherche.

Les URL conviviales donnent l'impression à l'utilisateur (ou au moteur d'indexation) d'avoir affaire à un site statique :

<http://www.monsite.fr/index.php?modu...ms&action=list>

devient

<http://www.monsite.fr/index.php/items/list>

et même , si l'on a la possibilité de modifier la configuration du serveur Web :

<http://www.monsite.fr/items/list>

Ces types d'URL sont souhaitables à la fois pour un bon référencement et pour le confort des visiteurs.

Ajax

Ajax est le nom donné à l'utilisation de Javascript dans un site afin de mettre à jour de façon asynchrone le contenu de certaines parties d'une page.

Un framework prend en charge les fonctionnalités de base de l'application Web, ce qui rend difficile de concevoir des requêtes Ajax si le framework n'en prévoit pas l'utilisation.

Le support d'Ajax peut être complété par l'intégration d'une ou plusieurs bibliothèques (prototype, script.aculo.us, YUI...), ce qui facilite leur utilisation.

Commentez cette news de symfolve en ligne : [Lien13](#)

Les derniers tutoriels et articles

AJAX cross-domain avec fXHR

L'objet de ce court article est de montrer comment faire des requêtes AJAX entre deux domaines distincts. Par défaut et pour des raisons de sécurité ce type d'appel AJAX est interdit.

Cette interdiction oblige donc à passer par un intermédiaire (proxy) qui exécute directement la requête (en réalité, il invoque l'URL) pour :

- * en récupérer la réponse ;
- * la transmettre au JavaScript.

Ensuite, au développeur de faire ce qu'il souhaite de la réponse.

Le plugin Flash Player ne supportant pas les communications synchrones, la librairie fXHR ne permet donc pas les appels synchrones.

Cette librairie vous est utile si vous n'avez besoin que de faire des appels asynchrones.

1. Prérequis

- La librairie JavaScript fXHR : [Lien14](#).
- Téléchargement sur fXHR-1.0.5 ([Lien15](#)) ou [Lien16](#).
- Un navigateur ou browser supportant la technologie Flash.

2. Sur le serveur cible des requêtes AJAX

- Il est nécessaire d'y installer un fichier de règles d'autorisation interdomaine **crossdomain.xml** (spécifications Flash sur [@adobe.com](#) ([Lien17](#))).
- Le fichier de règles **crossdomain.xml** placé sur un domaine distant A, autorise ou non l'accès aux fichiers de ce domaine A depuis le domaine B où est installée la librairie fXHR.
- Un exemple de fichier **crossdomain.xml** autorisant des sites externes à accéder à des ressources internes via AJAX.

```
<?xml version="1.0"?>
<!DOCTYPE cross-domain-policy SYSTEM
"http://www.adobe.com/xml/dtds/crossdomain-
policy.dtd">
<cross-domain-policy>
  <site-control permitted-cross-domain-
policies="master-only"/>
  <allow-access-from domain="*" />
  <allow-http-request-headers-from domain="*"
headers="*" />
</cross-domain-policy>
```

- Installer le fichier **crossdomain.xml** à la racine du serveur cible de la requête AJAX.

3. Déploiement des librairies sur votre site Web

- Décompresser le fichier fXHR-1.0.5.zip téléchargé préalablement (voir [Lien15](#)).
- Installer le contenu du dossier flensed-1.0/deplo/

dans le dossier intégrant vos scripts JavaScript.

4. Requêteur AJAX

La première chose à faire est d'inclure dans votre page Web le fichier **fXHR.js** :

ce fichier (cette librairie) a des dépendances qu'il vous faut installer aussi (voir chapitre 3 ci dessus).

```
<script type="text/JavaScript"
src="./js/fXHR.js"></script>
```

Ensuite, nous pouvons travailler sur notre fonction AJAX que nous appellerons callAJAX.

Notre méthode AJAX va gérer deux événements distincts :

- le premier pour la gestion des erreurs en cas d'échec de la requête appellera la fonction `handleError(errObj)` ;
- le second pour le traitement à effectuer au retour de la requête AJAX appellera la fonction `handleLoading(XHRobj)`.

Nous intégrerons directement la définition de ces deux fonctions dans la définition de la fonction `callAjax`.

Dans notre exemple, nous n'avons paramétré notre fonction qu'avec l'URL cible de notre requête AJAX, libre à vous de la paramétrer selon vos besoins.

Le corps de la fonction `callAjax` :

```
function callAJAX(url) {
  var flproxy = new flensed.fXHR( {
    xmlResponseText : false,
    onerror : function
handleError(errObj) {
  // ici on indique ce que
l'on fait en cas d'erreur
  // toutes les
informations relatives à l'erreur
```



```
l'objet errObj // sont stockées dans
},
onreadystatechange : function
handleLoading(XHRobj) {
4) {
// ici on indique
ce que l'on fait // avec la
réponse à la requête // AJAX stockée
dans XHRobj.responseText;
}
```

```
});
// ici on a le choix entre la
méthode GET ou POST pour appeler l'URL
flproxy.open("GET", url);
flproxy.send(null);
}
```

Il suffit d'appeler la méthode callAJAX avec pour paramètre l'URL cible pour effectuer la requête AJAX.

Retrouvez l'article de TheBigJim en ligne : [Lien18](#)

Mise en page sans tableau

Faire une mise en page avec deux ou trois colonnes peut se faire aisément sans tableau. Diverses techniques sont possibles. On trouve sur le net une multitude de tutoriels de webdesign sur la mise en page fluide sans tableau, à commencer par celui du W3C qui propose en outre une série de gabarits fort utiles.

Ce tutoriel ne tente pas donc de réinventer la poudre, mais simplement de présenter les atouts d'une mise en page full CSS par rapport à une mise en page par tableau.

1. Mise en page en colonnes sans tableau

Fonctionne avec :

- Firefox ;
- IE 5.x et + ;
- NS 6/7 ;
- Mozilla 1.x / SeaMonkey ;
- Opéra ;
- Safari / Chrome.

Attributs utilisés :

- clear ;
- float ;
- margin ;
- text-align.

1.1. Pourquoi éviter la mise en page par tableaux ?

Il ne faut pas systématiquement diaboliser la mise en page par tableaux, mais disons qu'elle a "vétu". C'était une bonne méthode lorsque les navigateurs implémentaient mal les normes CSS 2.1 (aux temps préhistoriques -à l'échelle du Web - de Netscape 4 et d'Explorer 5), mais maintenant...

Faire deux ou trois colonnes en CSS est relativement facile et le langage CSS est fait pour cela. Le code est plus facile à maintenir, cela offre plus de choix de mise en page sans modification du code HTML, et surtout c'est plus accessible.

Certes, on m'objectera qu'au delà de trois colonnes, les choses se compliquent et qu'il y a quelques heures d'arrachage de cheveux pour arriver à faire quelque chose d'uniforme pour tous les navigateurs. Mais sauf cas très particulier, il n'y a pas besoin de faire plus de trois colonnes. Alors en attendant que *column* (norme CSS3 ([Lien19](#))) soit implémenté par tous les navigateurs, faisons simple.

1.2. Comparaison de mises en page avec tableaux et avec les CSS

1.2.1. Avec tableaux

En règle générale, le webmestre qui réalise une mise en page avec tableau utilise aussi un logiciel WYSIWYG en lui faisant confiance. J'ai donc fait ce petit test avec l'interface WYSIWYG d'un logiciel très connu et voilà ce qu'il m'a donné pour une mise en page deux colonnes

(dont un menu à gauche de largeur de 150 pixels) encadrées d'un en-tête et d'un pied de page :

```
<table width="100%" border="0">
  <tr align="center" valign="top">
    <td colspan="2">En tête</td>
  </tr>
  <tr>
    <td width="150" valign="top">Menu</td>
    <td valign="top">Contenu</td>
  </tr>
  <tr align="center" valign="top">
    <td colspan="2">Pied de Page</td>
  </tr>
</table>
```

Mouais... Le code généré est bourré d'attributs dépréciés, il est lourd, et tout ça sans aucune couleur, image de fond ou quoi que ce soit. Que ce soit directement en HTML ou par le biais du CSS, il faudra rajouter des attributs dans ce code pour coloriser tout ça, ce qui alourdira encore la lisibilité du code et le poids de la page.

Voir le tableau (agrémenté de quelques fioritures supplémentaires pour une meilleure visualisation) : [Lien20](#).

1.2.2. Avec les CSS

Le principe est très simple : on divise chaque élément de la page par un *div*. La page se divise en quatre sections, chacune correspondant à une cellule de tableau précédemment illustré :

- Un en-tête : **div id="entete"** ;
- Un menu : **div id="menu"** ;
- Le contenu : **div id="contenu"** ;
- Un pied de page : **div id="footer"**.

Et si vous avez loupé le tutoriel sur les sélecteurs *class & id* ([Lien21](#)), il vaut mieux aller le voir de plus près avant de poursuivre...

```
<div id="entete">En tête</div>
<div id="menu">Menu</div>
<div id="contenu">Contenu</div>
<div id="footer">Pied de page</div>
```

Le code est plus clair et sa maintenance plus aisée qu'avec les balises d'un tableau. Moins chargé, il est donc aussi plus léger. Et surtout, l'accessibilité est améliorée.

Codes CSS

Pour positionner différents **div** dans une page, on peut utiliser le positionnement absolu, relatif ou flottant. En matière de choix, chaque méthode a ses avantages et ses inconvénients, c'est surtout une histoire de goût et parfois de contraintes.

Ce dont il faut surtout se souvenir, c'est que la position **absolute** sort complètement du flux (c'est-à-dire de la lecture naturelle de haut en bas du code), ce qui offre un risque de chevauchement avec d'autres éléments mais lui confère une bonne « autonomie » de placement dans la page. Ce problème de chevauchement (de dépassement de flux) est aussi présent avec *float*, mais il est plus facilement maîtrisable, ou du moins contournable.

Quelques rappels de principe sont disponibles sur le diaporama ([Lien22](#)) que je vous ai présenté lors de l'atelier CSS ([Lien23](#)) de ParisWeb 2007 ([Lien24](#)).

Ce code CSS est toujours, bien entendu, à enregistrer ([Lien25](#)) de préférence dans une feuille de style externe.

Menu à gauche flottant

En positionnant le menu à gauche, on suit la logique du flux, ce qui est préconisé pour l'accessibilité.

```
#entete, #footer {
  text-align:center;
}
#menu {
  float:left;
}
#contenu {
  margin-left:155px;
}
```

Menu à droite flottant

Mais on peut aussi positionner le menu à droite sans toucher au code HTML ! Ce qui est bien sûr impossible avec une mise en page par tableaux. Cela dit, la logique du flux n'est plus respectée (ce qui est placé en premier dans le code se trouve après un élément positionné à gauche de l'écran), ce qui est déconseillé par les normes accessiweb.

```
#entete, #footer {
  text-align:center;
}
#menu {
  float:right;
}
```

```
}
#contenu {
  margin-right:155px;
}
```

Voir les résultats : ([Lien26](#))

Astuce : pour éviter que l'élément flottant (ici le menu) ne chevauche le pied de page en cas de hauteur supérieure de l'élément non flottant (ici le contenu), il suffit de rajouter *clear:both* à l'élément à ne pas chevaucher (ici le footer).

```
#footer {
  clear:both;
}
```

Attention au DocType : les déboires de rendu entre les différents navigateurs, notamment avec Internet Explorer, sont souvent dus à une mauvaise déclaration de *DocType*, ou à la présence du prologue XML qui fait basculer IE en mode "Quirks" (ce qui lui fait purement et simplement oublier les bonnes normes).

Si vous ne comprenez absolument rien à ce qui vient d'être dit, une lecture s'impose : les Types de Documents ([Lien27](#)).

1.3. Lectures complémentaires

- Problèmes liés à une mise en page par tableaux (OpenWeb) : [Lien28](#) ;
- Initiation au flux (OpenWeb) : [Lien29](#) ;
- Design CSS fluide encadré d'images (CSS4 design) : [Lien30](#).

2. Articles de la série

- Principe des feuilles de style CSS : [Lien31](#) ;
- Balises (X)HTML de base : [Lien32](#) ;
- Feuille de style CSS de base : [Lien33](#) ;
- Mettre en forme un texte en CSS : [Lien34](#) ;
- Gérer les marges en CSS : [Lien35](#) ;
- Résoudre les problèmes de marges en CSS : [Lien36](#) ;
- Sélecteurs class et id : [Lien21](#) ;
- Bordures en CSS : [Lien37](#) ;
- Image de fond en CSS : [Lien38](#) ;
- Effets Rollover à l'aide de :hover : [Lien39](#) ;
- Aligner une image et du texte à l'aide de la propriété float : [Lien40](#) ;
- Listes à puces en image : [Lien41](#) ;
- Citations et retraits : [Lien42](#).

Retrouvez l'article de Pascale Lambert en ligne : [Lien43](#)

Boutons graphiques en CSS

Le bouton en image qui change d'aspect au passage de la souris est très utilisé sur les pages Web. Longtemps, la majorité de ces boutons graphiques étaient animés par un JavaScript ou pire (car plus lourd) par un applet java. Avec les CSS, par effet "rollover", la légèreté et la simplicité est de mise pour créer de beaux boutons.

1. Compatibilité

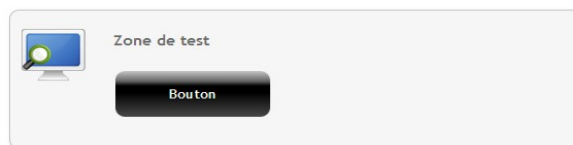
Fonctionne avec :

- Tous les navigateurs graphiques.

Attributs utilisés :

- background ;
- color ;
- display ;
- float ;
- line-height ;
- margin ;
- padding ;
- text-align ; text-decoration ;
- vertical-align ;
- width.

```
text-align:center;
vertical-align:middle;
background:url(images/bouton-noir.png) no-repeat;
color:white;
text-decoration:none;
}
.bouton a:hover {
background:url(images/bouton-noir-2.png) no-repeat;
}
```



2. Bouton CSS simple

2.1. Code (X)HTML

Un bouton étant en général utilisé pour faire un lien vers une autre page, les sélecteurs exploités dans le code CSS seront **a** et **a:hover** pour le changement d'aspect au survol du bouton (si changement souhaité bien sûr...).

Pour un seul bouton, le code HTML peut être le suivant :

```
<div class="bouton">
  <p>
    <a href="#">Bouton</a>
  </p>
</div>
```

Prenons maintenant deux images dont l'une servira pour le bouton au repos et l'autre pour le survol.



2.2. Code CSS

Ces deux images ayant une taille de 150 pixels sur 50, il va falloir donner au sélecteur **a** la même taille. Petit problème, **a** est de type "inline", on ne peut lui affecter de taille. Il va donc falloir lui changer son statut par défaut pour lui conférer un comportement de type "block". On utilisera pour cela l'attribut **display**.

Autre astuce, pour pouvoir centrer le texte verticalement, au lieu de donner la hauteur avec un **height**, on va définir une hauteur de ligne (**line-height**), et ainsi on pourra appliquer un **vertical-align:middle** qui alignera le texte parfaitement.

```
.bouton a {
display:block;
width:150px;
line-height:50px;
```

Voir le résultat ([Lien44](#)) !

N'est-ce pas superbe ? Non ? Oui bon, d'accord, côté design je suis une bique. Vous ferez bien entendu de magnifiques images de boutons pour un rendu bien plus sexy.

3. Plusieurs boutons CSS

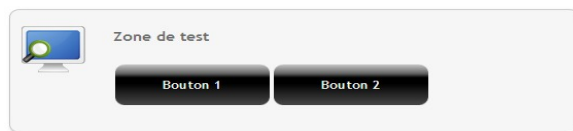
Étant donné que les liens ont été déclarés de type "block", à chaque balise **a** il y aura un saut de ligne. Si on veut faire plusieurs boutons les uns en dessous des autres, ce n'est pas un problème. En revanche, si l'on veut qu'ils soient alignés horizontalement, il va falloir rajouter quelques petites choses dans le code CSS.

On va tout simplement les rendre flottants. Attention cependant à bien rajouter un **clear:both** dans l'élément HTML qui suivra pour éviter des choses désagréables...

Il faudra donc rajouter ceci au code du sélecteur **a**.

```
float:left;
margin:2px;
```

Les marges ne sont pas obligatoires, c'est juste pour que les images ne se touchent pas.



Voir le résultat ([Lien45](#)) !

4. Lectures complémentaires

- Bouton et boutonnière verticale, réalisé par la même, sur OpenWeb : [Lien46](#).

Retrouvez l'article de Pascale Lambert en ligne : [Lien47](#)

Plusieurs images de fond grâce aux CSS

Grâce à la propriété `background-image`, il est très facile de mettre une image de fond dans le corps (body) d'une page Web. Mais on ne peut qu'en mettre une seule...

Pour pouvoir mettre plusieurs images de fond, il faut définir des cadres (des `div`) et attribuer une image à chacun.

1. Compatibilité

Fonctionne avec :

- tous les navigateurs graphiques.

Attributs utilisés :

- `background-position` ; `background-repeat` ; `background-image` ;
- `margin` ;
- `text-align`.

2. L'astuce

Prenons l'exemple d'une image (*houx2.gif*) placée aux quatre coins de la page. On va définir quatre cadres imbriqués (*hg*, *hd*, *bg*, *bd*), puis leur attribuer un **background-position** adéquat pour que les images se placent aux quatre coins. Les marges définies dans le cadre **contenu** sont précisées en fonction de la taille de l'image pour ne pas que le texte empiète dessus.

Code CSS

```
#hg, #hd, #bg, #bd {
  background-repeat:no-repeat;
  background-image:url(houx2.gif);
}
#hg {
  background-position:top left;
}
#hd {
  background-position:top right;
}
#bg {
```

```
  background-position:bottom left;
}
#bd {
  background-position:bottom right;
}
#contenu {
  margin-left:100px;
  margin-right:100px;
}
#contenu p {
  text-align:justify;
}
```

Code (X)HTML

```
<div id="hg">
  <div id="hd">
    <div id="bg">
      <div id="bd">
        <div id="contenu">
          <h1>Les quatre coins</h1>
          <p> [bla bla]</p>
        </div> <!-- /contenu -->
      </div> <!-- /bd -->
    </div> <!-- /bg -->
  </div> <!-- /hd -->
</div> <!-- /hg -->
```

Voir le résultat ([Lien48](#)) !

Attention tout de même à ne pas trop abuser de la méthode, trop de cadres tuent les cadres...

Retrouvez l'article de Pascale Lambert en ligne : [Lien49](#)

Optimisation des bases de données

Critique du livre par Antonio Gomes Rodrigues

Ce livre s'adresse aux développeurs voulant se sensibiliser à l'optimisation des bases de données (en particulier Oracle mais il y a une comparaison systématique pour MySQL et SQL Server).

Le 1er chapitre explique de manière concise tout ce qu'il faut savoir (tablespace, cache mémoire...) sur un SGBDR à des fins d'optimisation.

Puis on continue avec de la théorie sur les SGBDR en expliquant les bons réflexes sur le typage des données et la normalisation/dénormalisation. On restera un peu sur sa faim pour cette dernière partie qui aurait mérité plus d'exemples et de comparaisons avant/après afin d'affiner l'utilisation de la dénormalisation.

Après cette partie sur l'étude et l'optimisation du modèle de données, on passe à l'étude de l'optimisation des requêtes et comme pour toute optimisation, il faut des outils pour mesurer l'impact des changements qu'on fera. Pour cela l'auteur nous présente le plan d'exécution d'une requête et un certain nombre d'outils.

Un chapitre sur les statistiques, les index et le travail autour des tables (vue, cluster...) vient ensuite. Cette partie est bien expliquée mais malheureusement l'auteur semble avoir oublié ce qu'il a dit au chapitre précédent sur la mesure de la performance. Par exemple il parle de l'importance de l'ordre des éléments dans un index composite mais ne donne pas de chiffre pour affirmer ce qu'il dit. Il parle même de "très efficace" et de "moyennement efficace". De même, il parle du coût d'un index et de compression d'index mais sans les chiffrer.

Suit un chapitre sur les techniques d'optimisation standards des requêtes assez intéressant qui est une succession d'astuces d'optimisation avec des comparaisons

avant/après. Malheureusement le moteur d'optimisation d'Oracle réalise ces astuces la plupart du temps (l'auteur utilise des hints pour désactiver ces optimisations), ce qui limite l'intérêt de ce chapitre avec les dernières versions d'Oracle, mais sera utile pour une version antérieure d'Oracle et/ou une requête trop compliquée pour le moteur d'optimisation.

Puis un chapitre sur les optimisations avancées (hints, exécution parallèle, SQL avancé...) est traité. Chapitre intéressant mais à manier avec précaution si un DBA ne participe pas au projet.

Enfin une dernière partie sur l'optimisation applicative (binding, cache local à l'application...) et l'optimisation de l'infrastructure clos le livre.

À la lecture de la conclusion, on se dit que c'est vraiment dommage qu'il n'y ait pas eu un véritable exemple (traité dans chaque partie du livre), plus de comparaisons chiffrées et d'explications visuelles (schémas...), car on sent que l'auteur a de la bouteille.

On se retrouve avec un bon livre pour les développeurs débutants et/ou les développeurs peu familiers avec la performance et voulant un livre en français. Bien sûr comme pour tous les livres, il faudra mettre en application toutes les informations de celui-ci pour se les approprier. Ce que je conseille pour tous les livres et en particulier pour ceux traitant d'optimisation car chaque astuce dépend beaucoup du contexte et donc il ne faut pas les appliquer aveuglément.

Pour les développeurs ayant une bonne connaissance de l'optimisation des bases de données, il y a quand même des choses à apprendre mais ils ne sont pas la cible de ce livre.

Retrouvez cette critique de livre sur la page livres SGBD : [Lien50](#)



Qextend

Nous sommes heureux de vous présenter la version gamma de Qextend. Le but de cette version est de vous montrer à quoi va ressembler QExtend et avoir un maximum de retours de votre part.

Ajouté par yan ([Lien51](#)) il y a 2 mois ([Lien52](#))

Oyé, oyé, braves gens, damoiseaux et damoiselles !

En ces jours ensoleillés, nous sommes heureux de vous présenter la version gamma de Qextend ([Lien53](#)).

Le but de cette version est de vous montrer à quoi va ressembler QExtend et avoir un maximum de retours de votre part. Voici ce que vous allez y trouver :

- Emitter/Receiver : nouvelle méthode pour créer vos connexions ;
- ExtendPtr et ObjectPtr : pointeur intelligent un peu particulier ;
- Parser JSON ;
- système de log ;
- liste de QObject ;
- un tableau en deux dimensions ;
- un plugin DevIL pour la lecture d'image ;
- mesure de temps ultra précise ;
- gestion d'application avancée ;
- widget de visualisation d'image (attention ! dans la release, ça va dépoter !) ;

- un agrégateur de signaux pour en créer un nouveau (SignalCheckPoint) ;
- un planificateur d'action (ActionScheduler), qui permet de ne pas lancer un action si la précédente n'est pas finie et de la planifier juste après. Pour éviter les appels intempestifs, une seule planification est gardée.

Vous y trouverez également des démos pour illustrer l'utilité de ces composants, ainsi que leur méthode de fonctionnement. La doc en français et en anglais se trouve ici : [Lien54](#), [Lien55](#).

Vous trouverez un 7z des sources sur cette page : [Lien56](#).

Pour compiler, c'est très simple :

- pour Qt creator : ouvrir le fichier qextend.pro ;
- avec QMake : aller dans le répertoire racine et faire
 1. qmake -r
 2. nmake ou make suivant votre environnement de développement

N'hésitez pas à nous faire vos remarques ou à poser des questions à la suite de ce thread. Si vous avez des idées, n'hésitez pas à ajouter une demande sur Redmine ([Lien57](#))

Retrouvez ce projet en ligne : [Lien53](#)

Les dernières news

La documentation de Qt 4.6 traduite en français

Bonjour,

Aujourd'hui, l'équipe Qt de Developpez.com est heureuse de vous présenter... la traduction en français de la documentation officielle de Qt 4.6 !

Ce fut un travail de longue haleine mais en voici le résultat : plus d'une cinquantaine de pages traduites, classes et concepts confondus ! Notamment, tout QWidget ([Lien58](#)) a été traduit, ainsi que l'espace de noms Qt ([Lien59](#)), ce qui représente déjà un beau morceau. Aussi, toute la partie concernant les threads ([Lien60](#)) a subi le même sort. Vous pouvez consulter aisément la liste des pages traduites ([Lien61](#)).

Comment utiliser cette version de la documentation ?

C'est très simple : elle reprend la même architecture que celle de Nokia. Il vous suffit donc de remplacer la partie `doc.qt.nokia.com` par `qt.developpez.com/doc` pour obtenir automatiquement la version traduite, si elle est disponible. Les URL ont été traduites au maximum pour vous permettre de mieux vous y retrouver ; aussi, les URL anglaises sont automatiquement redirigées vers leurs

traductions.

Enfin, la documentation hébergée chez Nokia souffre de quelques défauts. Notamment, vous ne pouvez pas simplement entrer le nom de la classe après le radical pour obtenir sa documentation ; ici, ce genre d'URL fonctionne à merveille : [Lien62](#)

Quelques exemples

Pour obtenir l'index (réduit) de la documentation, [Lien63](#).

Pour obtenir l'index de la dernière version traduite de la doc, [Lien64](#).

Pour obtenir la traduction (si disponible) de la classe QWidget en version 4.6, [Lien65](#).

Si vous tombez sur une page qui n'est pas encore traduite, vous ne serez pas trop choqués : elle garde une unité de style avec les pages traduites. C'est la seule modification, elle reste malheureusement en anglais.

L'histoire du projet

Dans le but de fournir aux codeurs avides d'entraînement, leur met préféré, des sujets sur lesquels travailler : le site Exercice Qt vit le jour. Ce site, composé d'une fort jeune équipe, vécut une longue période difficile avec pas plus de quarante visites par jour, celles des habitués. Afin de combler cela et suite aux problèmes de compréhension

d'un bon nombre de débutants, l'un des administrateurs lança le projet de la traduction de la documentation de Qt. Un travail conséquent pouvant rapporter un grand nombre de visites et pouvant aider de nombreuses personnes n'ayant, par exemple, jamais eu la chance d'apprendre l'anglais. Ainsi, dans le plus grand secret, une équipe constituée de moins de personnes que les dix doigts d'une main se lancèrent dans la traduction.

Cependant, la tension monta, petit à petit, dans la petite équipe : de petits malentendus sur l'objectif du projet, sur sa réalisation technique. Au final, plus de ligne directrice. Jusqu'en août 2009, les membres disparaurent, les uns après les autres. Le projet de traduction, à l'eau.

Très rapidement contacté par l'équipe Qt de Developpez.com, cet administrateur accepta de continuer le projet avec eux et obtint les autorisations de la part des anciens traducteurs de remettre également leurs participations. C'est ainsi que le projet reprit son cours sous l'encadrement d'une équipe compétente et plus soudée.

Après des mois de travail de la part de l'équipe de traduction et de celle de nombreux relecteurs – que nous remercions encore et toujours pour leur excellent travail – la documentation a atteint un seuil de pages traduites suffisamment important pour que le projet soit dévoilé et pour que tous puissent profiter du projet.

Et aujourd'hui, nous avons l'honneur et la joie de vous présenter cet énorme projet !

Quelques remerciements particuliers

Notamment à Jacques Thery ([Lien66](#)) pour ses nombreuses relectures ! En effet, presque la totalité des pages traduites sont passées entre ses mains pour des corrections très approfondies et d'une précision incroyable ! À ses côtés furent également présents Philippe Beaucart ([Lien67](#)), Wachter ([Lien68](#)) et Maxime Sannier ([Lien69](#)). Nous les remercions tous les quatre pour leur relecture, leur patience surtout pour nous aider à vous présenter cette traduction !

Aussi, nous fûmes quelques traducteurs, listés ici par ordre alphabétique : Jonathan Courtois ([Lien70](#)), Thibaut Cuvelier ([Lien71](#)), Abdelhafid Hafidi ([Lien72](#)), Florent Renault ([Lien73](#)), Mikael Sans ([Lien74](#)) et Louis du Verdier ([Lien75](#)).

Et vous ?

Quelques pages restent encore en anglais... mais plus pour longtemps ! Vous aussi, vous pouvez participer à ce projet de traduction ! Nous avons besoin principalement de relecteurs et de traducteurs. Si vous êtes intéressé, n'hésitez pas à nous envoyer un mail ([Lien76](#)), un MP ([Lien77](#)) ou à répondre dans ce fil.

La documentation Qt 4.6 en français : [Lien78](#).

Commentez cette news de Thibaut Cuvelier en ligne : [Lien79](#)

Sortie de Qt 4.7 en beta 2 et d'une préversion de Qt Creator 2.1

La deuxième bêta de Qt 4.7 est disponible sur la page de téléchargement de Qt ([Lien80](#)). Les sources sont toujours disponibles ainsi que des binaires pour Mac OS X (Carbon et Cocoa), MinGW 4.4.0 et Visual Studio 2008.

La convention de nommage a évolué comme pour Qt 4.6.x au sujet des paquets à destination des utilisateurs de Mac OS X, pour refléter le fait que Cocoa est maintenant préféré pour Qt 4.7. Les paquets pour Carbon n'existeront plus à partir de Qt 4.8 mais seront toujours disponibles pour toutes les versions de Qt 4.7.x.

Le but de cette seconde version bêta ? Modulariser un peu plus Qt (voir à ce sujet l'article d'Henry Haverinen ([Lien81](#)) : au final, Qt 4.7 ne devrait plus contenir le module Qt Multimedia, celui-ci étant intégré aux Qt Solutions) mais ainsi de fournir une base plus adaptée pour les tests et d'offrir une version finale d'encre meilleure qualité.

Des snapshots de Qt Creator 2.1 sont aussi disponibles aujourd'hui. En combinaison avec Qt 4.7 bêta 2, ils fournissent une preview de Qt Quick. Cette version de Qt Creator contient une première version du Qt Quick Designer et peut travailler avec des projets Qt Quick, avec la possibilité d'éditer et de déboguer des fichiers QML. Pour ceux qui utilisent le repository Git, un tag v4.7.0-beta2 devrait apparaître bientôt.

Sources : les Qt Labs (Qt 4.7 Beta2 and Qt Creator 2.1 Snapshots Available ([Lien82](#)) et Qt 4.7 scope change regarding Qt Multimedia ([Lien81](#))).

Et vous ?

Que pensez-vous du support de Qt Quick par Qt Creator ? Cette nouveauté va-t-elle révolutionner votre manière de coder ?

À propos de la modularisation de Qt, quels pourraient être les avantages, tant pour les développeurs et mainteneurs du framework que pour ses utilisateurs ?

Commentez cette news de Thibaut Cuvelier en ligne : [Lien83](#)

Qt 4.7 Release Candidate 1 et Qt Quick

Comme certains d'entre vous le savent déjà, quelques temps plus tôt a été mis à disposition une Release Candidate (RC) de Qt 4.7. Qt 4.7 est un grand pas en avant pour les performances pour l'introduction de Qt Quick. Avec Qt 4.7 nous allons voir les applications Qt tourner plus vite et être plus fluides qu'auparavant.

Avec la sortie de Qt 4.7 incluant Qt Quick, vous allez immédiatement avoir accès à des avancées pour écrire des applications et des interfaces fluides et animées. Qt Quick est conçu en trois parties : (1) le langage QML, (2) le

module QtDeclarative dans le framework Qt et (3) un composant dans Qt Creator.

La RC d'aujourd'hui et la release finale de Qt 4.7 vont former les deux premières pièces de Qt Quick. Avec la sortie de Qt Creator 2.1 dans le quatrième quart de cette année, vous allez avoir accès à un ensemble d'outils pour faire de votre expérience avec Qt Quick, la plus riche et la plus simple. Le dernier snapshot de Qt Creator 2.1 sera disponible prochainement.

Commentez cette news de Francis Genet en ligne : [Lien84](#)

Les virus informatiques : théorie, pratique et applications

Cet ouvrage est la deuxième édition du livre désormais incontournable d'Eric Filiol consacré aux virus informatiques (fondements théoriques, aspects algorithmiques et pratiques et applications des virus). Il comporte trois chapitres entièrement nouveaux qui permettront au lecteur de découvrir les évolutions récentes de la virologie et de prendre connaissance de codes malveillants qui n'étaient pas décrits dans la précédente édition (virus de documents, botnets, propagation des vers et générateurs de virus).

Le lecteur trouvera dans cet ouvrage un tableau clair et détaillé des différents types d'infections informatiques – fondé sur une formalisation théorique au moyen de machines de Turing, d'automates auto-reproducteurs et de fonctions récursives – et pourra apprécier les principaux enjeux de la protection et de la lutte antivirale. L'étude détaillée du code source de diverses familles représentatives de l'algorithmique virale permettra aussi de comprendre les mécanismes fondamentaux mis en œuvre par un virus ou un ver. Le langage C a été systématiquement utilisé dans les exemples afin de faciliter la compréhension des programmes étudiés.

Eric Filiol est expert en virologie et en cryptologie au ministère de la Défense et directeur du Laboratoire de virologie et de cryptologie opérationnelles de l'École supérieure en informatique, électricité et automatismes (ESIEA). Il est aussi l'auteur de Techniques virales avancées publié chez Springer, coll. « IRIS », et rédacteur en chef du Journal in Computer Virology.

Critique du livre par Pierre Therrode

Guidé par un regard expert, Eric Filiol nous ouvre les yeux sur une partie de l'informatique, farouchement gardée par

les éditeurs d'antivirus.

En plus d'être l'un des seuls livres français traitant du sujet des virus informatiques, l'auteur nous guide à travers les méandres de cet univers grâce à des exemples commentés et détaillés. Extrêmement bien présenté, ce livre fait partie des rares ouvrages à être aussi clairs et rigoureux dans les propos, tout en ayant la sagesse de se mettre à la place du débutant qui souhaite apprendre le fonctionnement des virus.

Ainsi, comme énoncé dans le titre du livre, celui-ci est composé de trois grandes parties dont chaque chapitre se termine par une série d'exercices.

La première s'étend de la machine de Turing en passant par la formalisation de F.Cohen et L.Adeleman, jusqu'à la présentation des différents types de virus. Celle-ci, néanmoins demande un effort de compréhension, plus encore si le lecteur n'est pas issu d'un cursus scientifique.

Puis une fois cette étape passée, le plaisir de la lecture n'en est que meilleur avec la seconde qui présente de façon concrète les différents virus (virus interprétés, compagnons, les vers, les botnets et les virus de documents), le tout allié à la puissance du langage C.

Enfin, la troisième décrit les tous premiers vers informatique issus du monde scientifique, l'utilisation de ces programmes faits par les entités gouvernementales (exemple : Magic Lanterne), pour se terminer sur une étude des virus BIOS ainsi que sur la cryptanalyse d'un malware.

Conclusion : ce livre est tout simplement un ovni, qui est à mettre entre les mains de tous ceux qui souhaitent en savoir plus sur le monde des virus informatiques.

Retrouvez cette critique de livre sur la page livres Sécurité : [Lien85](#)

Les derniers tutoriels et articles

Les pièges de l'Internet

« Arrêtez le mal avant qu'il n'existe ; calmez le désordre avant qu'il n'éclate. » Lao-tseu, VI siècle av. J.C.

1. Introduction

À l'heure actuelle, un nombre incalculable de données transitent tous les jours sur la toile de l'Internet, données qui sont d'ordre personnel, financier, culturel, audio et visuel. Nous autres, utilisateurs du World Wide Web ou plus communément appelé Internet, nous utilisons ce réseau afin de communiquer avec nos proches, effectuer

des recherches, consulter ses finances, c'est-à-dire pouvoir naviguer simplement et tranquillement sur le net. Certains individus possèdent des connaissances et un savoir-faire exceptionnel dans ce milieu. Ils sont définis comme des Hackers.

Certains Hackers de philosophie White-hat (c'est-à-dire ayant un certain sens de l'éthique) participent à

l'élaboration et au perfectionnement de l'utilisation d'Internet, d'autres, les Black-Hat, utilisent leurs connaissances afin d'en tirer un bénéfice financier avec la création de logiciels malveillants (malware en anglais) tels que les virus, les trojans, les vers, etc. afin de compromettre une ou des entreprises, ou de simples internautes. Dès lors, et dans l'optique de pouvoir répondre à ces attaques, les éditeurs d'antivirus n'hésitent pas à livrer bataille aux créateurs de malwares en proposant des solutions de sécurité toujours plus innovantes, à ceux qui veulent se prémunir contre ce type de désagréments. En échange d'une protection optimale et face à la recrudescence de la cybercriminalité, les éditeurs n'hésitent pas à proposer des solutions payantes (compter environ 40 euros pour une solution complète), mais rassurez-vous, l'article proposera des outils entièrement gratuits afin de répondre au problème sécuritaire auquel se heurtent bon nombre d'utilisateurs de chez Windows. Outre cela, l'article tentera par la même occasion de montrer certains vecteurs d'attaques, qui ne sont pas forcément liés aux faiblesses du système, mais à l'utilisateur lui-même.

2. Les risques

2.1. La motivation des pirates

Comme énoncé dans l'introduction, les pirates ont pour motivation deux vecteurs.

-Le premier est indiscutablement celui de l'argent. En effet, il n'est pas rare de trouver des "kits de piratage" afin de pirater des utilisateurs d'un système bien particulier qui sont liés à une vulnérabilité du moment. À cela, on peut citer "le couteau suisse" Mpack, créé par des pirates de l'est en 2007 et qui était alors vendu aux alentours de 80 à 500 euros suivant la spécification du produit (Attaque DDOS, Spam, Trojan, Phishing etc.). Depuis quelques temps, la vente de malwares est devenu pratique courante sur le net, au même titre que celle d'antivirus ou de pare-feux, car bon nombre de black-hat ont compris les bénéfices financiers qu'ils pourraient en tirer en créant des logiciels malveillants afin de satisfaire leur compte en banque ou tout simplement la commande d'un "client". En effet, cette pratique tend à se populariser, avec des notices d'utilisation accompagnant la version bêta du malware. Se rajoutent aussi les chantages numériques.

L'exemple suivant est un aperçu de la notice d'utilisation du programme malveillant Agobot (Agobot est un virus réticulaire (botnet) qui permet à un pirate d'avoir une prise de contrôle totale sur le système infecté. Il permet en outre de pouvoir relayer des envois de spams, récupérer des informations personnelles...), contenant un descriptif des différents prix suivant l'obtention de la version:

[5.3] What's the price for the private version ?
Minimum:

- * No updates, but bugfixes if they are requested
- * Price: 50\$
- * Update price: 10\$

Standard:

- * All updates and new scanners
- * Price: 100\$

Premium:

- * Linux scanners, all updates and new scanners
- * Price: 250\$

I accept payment through paypal.

I might also be willing to sell/rent/share bots but I don't have a price set, contact me and we can negotiate one.

-Le second, est celui de la notoriété. Cela est une particularité bien spécifique dans le milieu du piratage, car chaque "Team" cherche à égaler le ou les concurrents en s'attaquant à des sites bien spécifiques tels que les Casinos en ligne, les sites d'antivirus, etc, de telle sorte que cela touche le plus de cibles possibles - ici la notion de cible représente les utilisateurs d'un système ou d'une application. Vous l'aurez donc compris, leur but est de faire la une des médias par tous les moyens mis en place pour arriver à leur fin.



[ENABLE FILTERS]

Total attacks: 1856 of which 1856 single ip and 0 mass defacements

Legend:

- H - Homepage defacement
- M - Mass defacement (click to view all defacements of this IP)
- R - Radefacement (click to view all defacements of this site)
- ★ - Special defacement (special defacements are important websites)

Time	Attacker	H	M	R	★	Domain	OS	View
2009/06/28	NobodyCoder	H				www.ashley-t.com	Win 2003	mirror
2009/06/28	1923Turk	H	R			cheapcarsinamano.com	Win 2003	mirror
2009/06/28	1923Turk	H	R			cheapcarsinabile.com	Win 2003	mirror
2009/06/28	1923Turk	H	R			cheapcarsinaustin.com	Win 2003	mirror
2009/06/28	1923Turk	H	R			cheapcarsinbroonsville.com	Win 2003	mirror
2009/06/28	1923Turk	H	R			cheapcarsincopus.com	Win 2003	mirror
2009/06/28	1923Turk	H	R			cheapcarsindallas.com	Win 2003	mirror
2009/06/28	1923Turk	H	R			cheapcarsinlapaso.com	Win 2003	mirror
2009/06/28	Balal	H	R			www.tahibat.com	Win 2003	mirror
2009/06/28	Parisian Boys Hacking Team	H				www.jis8.com	Win 2003	mirror
2009/06/28	Parisian Boys Hacking Team	H				www.hhaosf.com	Win 2003	mirror
2009/06/28	CrazyHacker16					ordledmetalart.com/home.asp	Win 2003	mirror
2009/06/28	YaRaLi_1985		R			ironworkers712.com/portals/0/d...	Win 2003	mirror
2009/06/28	YaRaLi_1985					huntingvorkshop.com/portals/0/...	Win 2003	mirror
2009/06/28	YaRaLi_1985					hathemendler.com/portals/0/d...	Win 2003	mirror
2009/06/28	YaRaLi_1985					hbbaffa.com/portals/0/db:txt	Win 2003	mirror
2009/06/28	YaRaLi_1985					huntingutor.com/portals/0/db:...	Win 2003	mirror
2009/06/28	YaRaLi_1985					hamsionakos.com/portals/0/db:...	Win 2003	mirror

zone-h, sorte de panthéon numérique où les pirates affichent leurs exploits

Le deuxième vecteur d'attaque ne concerne pas cet article. En revanche, rien ne vous interdit de visiter le lien suivant, qui traite des méthodes de protection des sites Web. ([Lien86](#))

Afin de réaliser le premier vecteur, les pirates n'hésitent pas à utiliser divers moyens d'attaques tels que les :

- Rogues
- Faux codec
- Vers informatiques
- DDOS
- Ransomware
- Phishing

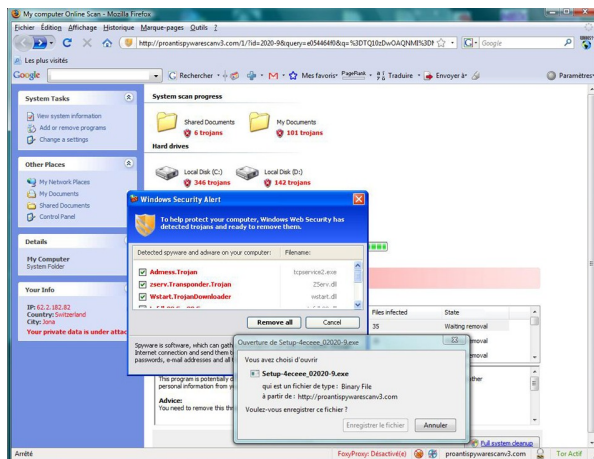
Ces termes peuvent paraître barbares pour celui qui ne s'y connaît pas trop, mais ne vous inquiétez pas ils seront détaillés plus loin. Enfin il est bon de souligner que ce genre d'infection est généralement discret et utilise dans la majorité des cas les derniers exploits, c'est-à-dire les dernières vulnérabilités liées à un système ou une application afin de rester le plus longtemps possible sur le système corrompu.

2.2. En énoncer le principe pour mieux s'en protéger

Neuf fois sur dix, une infection est due à la négligence de l'utilisateur, à sa crédulité et/ou à sa cupidité. En effet, la plus grande faiblesse de tout système reste l'utilisateur lui-même...

2.2.1. Les rogues

Lorsqu'on navigue sur Internet, il nous arrive parfois de tomber par hasard sur de faux sites d'antivirus faisant l'éloge d'un nouveau système de protection ultra performant. Le rogue est un terme anglais désignant un escroc ou un voyou. En informatique, cela désigne un faux logiciel de protection mis en place par des pirates afin d'infecter les utilisateurs d'un système bien spécifique. Les rogues qu'on trouve sur le net ont pour but de simuler une recherche de menaces potentielles présentes sur l'ordinateur et d'alerter l'utilisateur en lui annonçant que son système d'exploitation est -faussement- infesté de virus. Parfois, les pirates n'hésitent pas à vanter les mérites de leur pseudo antivirus en citant des noms d'antivirus connus, en reprenant l'ergonomie de certains éditeurs d'antivirus afin d'accentuer la crédibilité du produit et ainsi bernier un visiteur qui n'y connaît rien. D'autres, comme le montre l'exemple, font preuve d'imagination en recréant l'espace de travail de Windows.



Aperçu d'un rogue simulant la recherche de fichiers malveillants sur notre disque dur

De plus, il n'est pas rare que le rogue force le visiteur à télécharger le logiciel en question (à l'aide de scripts) et à l'acheter par carte bancaire en le redirigeant vers un lien soi-disant sécurisé (il est à noter que les faux sites d'antivirus, ainsi que les alertes sont généralement en anglais). Parfois, et à la suite d'un téléchargement non réfléchi, il arrive que ce genre de malware arrive à s'installer sur notre système et là commencent d'interminables rappels nous annonçant que notre logiciel de protection n'est pas mis à jour et que nous encourons de graves dangers. Là encore, les pirates ont pour motivation de récupérer des numéros de cartes bancaires (afin de les utiliser ou de les revendre) et de continuer l'infection de façon discrète une fois que l'utilisateur s'est délesté de quelques euros. Ces infections serviront par la suite aux pirates afin de réaliser des attaques DOS (Denial Of Service), relayer des envois de spams et/ou récupérer des informations personnelles...

2.2.1.2. Contre les rogues

Rappelez-vous qu'un véritable site d'antivirus proposant un "scan Online" ne peut pas réaliser le scan de votre disque dur en direct, il faut avant tout que l'utilisateur signe numériquement une chartre en acceptant les conditions d'utilisation. De ce fait, l'éditeur d'antivirus ne force jamais l'installation de son logiciel de protection. Une fois l'installation et le scan terminés, l'antivirus vous signale les malwares présents sur votre système d'exploitation et peut vous proposer un bref aperçu de son encyclopédie virale en ligne. Pour connaître les scanners Online officiels, ainsi que des antivirus officiels, reportez-vous en fin d'article à la rubrique liens annexes.

2.2.2. Les faux codecs

Quelquefois, en naviguant de site en site, il nous arrive de tomber sur des vidéos humoristiques ou pornographiques qui sollicitent notre curiosité. Mais généralement, pour visualiser le contenu, le site nous invite à télécharger un codec ou un ActiveX en usurpant dans la plupart des cas le logo de Microsoft Windows, ou l'ergonomie d'un lecteur média connu. Ces codecs ou ces ActiveX ne sont ni plus ni moins que des malwares qui ont pour but d'infecter le système d'exploitation du visiteur.

Les fausses alertes de codecs ressemblent le plus généralement à ceci :



2.2.2.2. Contre les faux codecs

Face à cela, des moteurs de recherche comme Google indexent automatiquement les sites qui présentent un danger potentiel pour les internautes. Afin d'être averti d'un site potentiellement dangereux, il suffit de cocher dans Mozilla Firefox les deux cases "Signaler...." et la case "Prévenir...." qui se situe dans Outils > Options > Sécurité. Il en va de même pour Internet Explorer, avec un curseur mis sur "Haute" pour ce qui concerne la navigation Web ; le chemin à suivre est : Outils > Options Internet > Sécurité.



Exemple d'avertissement lié à une politique de sécurité stricte du navigateur Firefox

De plus, pour être sûr de pouvoir lire toutes sortes de fichiers multimédia sans que cela puisse poser problème, il existe des sites officiels qui proposent des packs regroupant un nombre important de codecs, et ceci de façon totalement gratuite et non dangereuse (lien bas).

2.2.3. Les vers informatiques

Cependant, même si vous n'êtes pas sur le net, il est possible d'être infecté par une pièce jointe que l'on peut recevoir depuis sa boîte mail ou tout simplement en utilisant une messagerie instantanée. En effet, comme bon nombre d'utilisateurs nous recevons beaucoup de mails ; certains, dont nous ne connaissons pas l'expéditeur, contiennent parfois une pièce jointe avec dans le corps du message un descriptif nous invitant à l'ouvrir, jouant ainsi sur la curiosité du lecteur. Quelquefois, il arrive que l'un de vos contacts soit lui même infecté et vous envoie sans le vouloir ou le savoir, un message qui vous invite à ouvrir cette pièce jointe contenant un fichier ".exe". Dans le meilleur des cas, il vaut mieux supprimer le message si l'expéditeur ne vous confirme pas que celui-ci a bien été envoyé par lui même et si votre antivirus ne détecte rien...

Il est bon de rappeler aussi que les grands noms tels que Orange, Microsoft ainsi que d'autres sociétés, n'invitent jamais leurs utilisateurs à télécharger des pièces jointes contenues dans les mails et que les fichiers texte (en .txt) -ou image (en .jpg ou .gif) ne peuvent propager de virus (sauf, bien sûr, si le virus est caché dans l'image...). De plus, certains utilisateurs sur les messageries instantanées véhiculent (sans le vouloir) à travers leurs connexions des fichiers malveillants ou des liens qui renvoient vers des sites à haut risque. Généralement, ces demandes ont lieu de manière inhabituelle et spontanée, c'est à dire qu'elles interviennent alors même que l'on ne discute pas avec la personne ou lors d'une communication entre elles, celle-ci se fait de façon incohérente... Avec l'ère des réseaux sociaux tels que Facebook, Twitter, etc. les pirates ont compris l'influence que peut avoir un compte piraté dans la diffusion de malwares et l'extension de leur(s) réseau(x) de machines zombies.



Exemple d'un compte piraté servant à la prolifération d'un malware

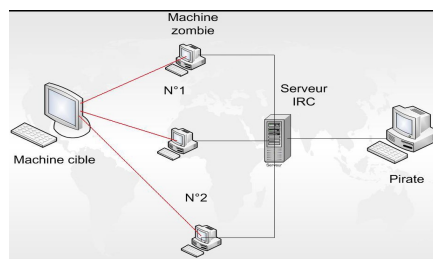
2.2.3.2. Contre les vers informatiques

Ayez toujours un regard critique la prochaine fois que l'un de vos contacts envoie sans le vouloir un tel message ou que vous tombez sur un lien suspect. Demandez-lui confirmation avant toute tentative de manipulation. Si le ver se diffuse à travers un réseau, c'est alors à l'administrateur de mettre en place un "honey pot" afin de pouvoir le capturer et l'étudier.

2.2.4. Les machines "zombies"

Au point où nous en sommes, il semble opportun d'expliquer ce que deviennent les machines qui ont été infectées. Comme dit plus haut, les motivations diffèrent selon le pirate. Cela peut couvrir la récupération des données personnelles, la récupération de données bancaires, le ransomware, c'est-à-dire que des fichiers sensibles se trouvant sur le système ciblé vont être chiffrés ou détruits si l'utilisateur ne paye pas une rançon (une sorte de chantage des temps modernes)... Cependant la motivation la plus importante reste l'emprise que peut avoir un pirate sur la toile ou/et sur un réseau, et pour mener à bien cette ambition, les machines ainsi infectées vont devenir des machines zombies, c'est-à-dire quelles deviendront ses esclaves afin de mener à bien ses intentions.

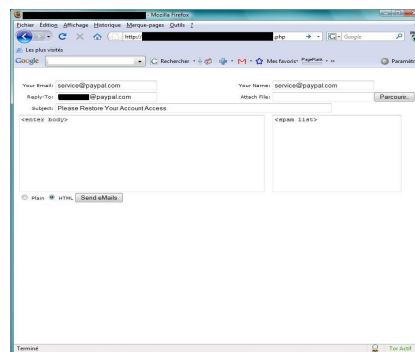
Cela peut donner lieu à l'envoi de millions de spams à travers le monde, à des attaques de types DOS (Denial of Service ou déni de service), c'est-à-dire que le pirate va utiliser son réseau dans le but de rendre une application informatique incapable de répondre aux requêtes de ses utilisateurs.



2.2.5. Le spam

Le spam ou pourriel, désigne l'envoi de courriers non sollicités par les destinataires qui sont envoyés en masse par un pirate à des fins publicitaires. Généralement, un pirate utilise deux méthodes pour pouvoir réaliser l'attaque :

- soit il passe par un réseau de botnet ;
- soit il exploite un ou des sites Web mal protégés (d'un particulier ou d'une entreprise).



Aperçu d'un site zombi permettant l'envoi de spam

Le contenu de ces pourriels est multiple. En effet, cela peut correspondre à l'incitation d'utilisation de médicaments (viagra, DHEA), de casinos en ligne, de (faux) crédits à la consommation, de passeports... Certains pirates n'hésitent pas à promouvoir les mérites de chaînes financières promettant un enrichissement rapide et efficace (bien entendu tout cela est faux !)

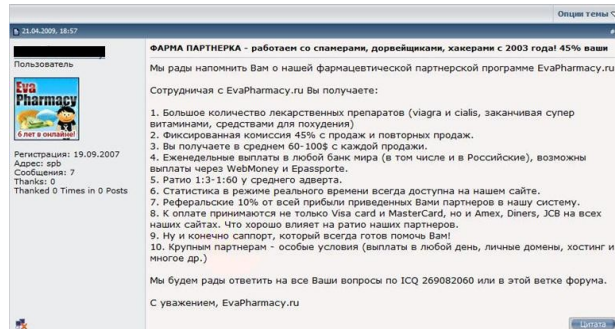
2.2.5.2. Comment est-on spamé et comment lutter ?

Toutes les adresses mails sont glanées par le pirate sur Internet en utilisant le ou les moteur(s) de recherche qui sont à sa portée tout en utilisant des programmes créés généralement par ses soins. S'il estime avoir un nombre assez important d'adresses mail, il peut les revendre sur des sites "pirates" afin d'en tirer un bénéfice (généralement cela se vend par des lots de mille pour quelques euros). Dans le cas contraire, il peut les garder pour lui et les utiliser pour infecter des utilisateurs.



Echantillon d'une liste de mails fonctionnels

Parfois, il arrive que ce soit des entreprises malveillantes qui font appel à des services de spammeurs afin d'augmenter leurs visites, et/ou d'obtenir un éventuel client. Ces techniques sont utilisées dans le domaine de la revente de faux médicaments, tel que la vente de viagra ou de protéines musculaires, comme le montre la capture d'écran suivante (traduction : « Bonjour, chers webmasters et les optimiseurs, je tiens à vous proposer la coopération avec le programme d'affiliation EvaPharmacy.ru Nous sommes sur le marché depuis plus de 6 ans et possédons une vaste expérience dans Pharma-convertir du trafic en ligne gastronomique. Avec la filiale pharmaceutique EvaPharmacy.ru vous obtenez: 1. Un grand nombre de médicaments (Viagra et Cialis, terminant avec un super vitamines, amaigrissement) 2. Fixe 45% de commission sur les ventes et les ventes répétées. 3. Vous obtenez une moyenne de 60-100 \$ par vente. 4. Les paiements hebdomadaires à toutes les banques du monde (y compris en Russie), mai payer par WebMoney et ePassporte. 5. Ratio 1:3-1:60 les publicités à la moyenne. 6. Statistiques en temps réel est toujours disponible sur notre site Web. 7. 10 Décision% de tous les profits étant donné vos partenaires dans notre système. 8. Le paiement peut être faite non seulement une carte Visa et MasterCard, et Amex, Diners, JCB, à tous nos sites. Comme pour influencer sur le ratio de nos partenaires. 9. Et bien sûr un support, qui est toujours prêt à vous aider! 10. Les grands partenaires - les conditions particulières (paiement en une journée, personnels des domaines, hébergement, et bien d'autres) »)



Annnonce de relayeur de spams sur un forum de l'Est

Il existe diverses techniques pour ne pas voir sa boîte mail spammée :

- la première consiste à écrire son adresse mail en langage humain ou de façon "masquée" (en utilisant des "[" ou des "]"). Par exemple, l'adresse JeanDupond@ladresse.com devient : "JeanDupond[arobase]ladresse [point/dot(en Anglais)]com", ou tout simplement "JeanDupond a l'adresse point com" ;

- la seconde consiste à écrire son adresse mail dans une image en y ajoutant en début et en fin d'adresse des éléments de ponctuations ou tout autre chose et ceci afin que l'adresse ne soit pas aspirée par un robot.

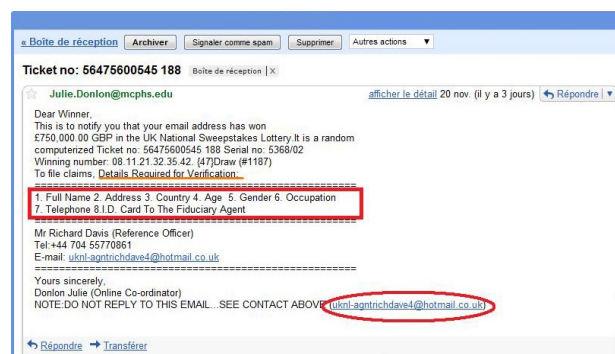
Enfin, il existe des sites permettant de référencer le contenu des spams ainsi que leur origine. En voici quelques-uns :

- [Lien87](#)
- [Lien88](#)
- [Lien89](#)

2.2.6. Le phishing

2.2.6.1. Le phishing de "bas niveau"

Enfin, pour clore cette partie, il est important d'énoncer un type d'attaque qui se base entièrement sur la naïveté humaine, à savoir le phishing (ou l'hameçonnage en français). Cette attaque consiste à faire croire à la victime qu'elle converse avec un site ou une entreprise de confiance, afin de lui soutirer des informations personnelles (numéro de carte bancaire, mots de passe) ou tout simplement à lui faire croire qu'elle est l'heureuse gagnante d'une loterie ou d'un héritage quelconque. Cela peut passer par le biais de mails ou la redirection vers un site vitrine.



Attaque de type phishing par mail

Dans la même idée, cela consiste à envoyer un mail aux

victimes potentielles en usurpant l'adresse mail de leur(s) banques ou à toute autre société à laquelle elles sont rattachées. Le contenu du mail leur annonce qu'elles ont perdu leurs identifiants et que cela risque d'être définitif. Dès lors, la seule alternative annoncée par les pirates est celle de la re-confirmation des identifiants sur une vitrine par l'intermédiaire d'un lien.



Cher client de Cr dit Agricole Centre France,

Le d partement technique de Cr dit Agricole proc de   une mise   jour de logiciel programm e de fa on   am liorer la qualit  des services bancaires.

Nous vous demandons avec bienveillance de cliquer sur le lien ci-dessous et de confirmer vos d tails bancaires.

<http://www.cantal.credit-agricole.fr/d73001733411/modext>

Nous nous excusons pour tout d sagr ment et vous remercions de votre coop ration.

  Cr dit Agricole 2007

Exemple de tentative de phishing sur les clients du Cr dit Agricole en 2007

Vous l'aurez donc compris, le pirate avancera toujours des arguments r alistes pr nant une am lioration des services de s curit , une panne technique qui a compromis la base de donn es des clients et supprimant par la m me occasion les identifiants, une mise   jour qui a besoin d' tre aid e par le client...

2.2.6.2. Le phishing de "haut niveau"

Le phishing de "haut niveau", qui est plus avanc  que les attaques de phishing normal, est appel  pharming. Cela consiste   falsifier des adresses IP appartenant   des domaines et   introduire ces informations dans la m moire cache du serveur DNS. De ce fait, si une victime potentielle tape les mots cl s sp cifiques   sa banque dans son navigateur Web, elle sera automatiquement redirig e vers la page vitrine du pirate. Il faut comprendre que le pharming est une attaque tr s dangereuse car elle ne force aucunement la victime   cliquer sur un lien ou   ex cuter un fichier, elle redirige simplement l'individu vers un lien sp cifique sans que celui-ci ne s'en rende compte.

2.2.6.3. Contre le phishing

Contre ce type de d sagr ment, il est bon de toujours garder   l'esprit que votre banque, votre fournisseur d'acc s   Internet, ainsi que divers types de jeux en ligne ne vous demanderont jamais vos diff rents identifiants suite   une pseudo panne technique ou   une mise   jour qui a mal tourn . Les banques, ainsi que diff rentes soci t s, ont de quoi se pr munir face   ce genre de probl me, donc les clients n'ont pas de soucis   se faire de ce c t -l . En revanche, il est vrai qu'un mot de passe trop faible augmente les risques de voir son compte pirat . Pour  viter ce genre de probl me concernant les mots de passe trop faibles, reportez-vous   la rubrique liens. Enfin, concernant les attaques de types phishing sur le net, les "vitrines" mises en place par les pirates sont g n ralement vite d tect es par les moteurs de recherche puis index es dans la liste noire (la fameuse Google Black-list). D s qu'un internaute tente d'acc der   une page contrefaite, celui-ci est automatiquement alert  par un " cran" lui annon ant le plagiat ou le site malveillant.

2.2.7. Conclusion

En conclusion, il semble bon de rappeler qu'il est parfois dangereux de naviguer sur des sites de Warez, sur des sites de "Super-Hacker" vous montrant comment devenir le ma tre du monde avec trois commandes MSDOS (bien s r cela est ironique), sur des sites pornographiques et plus encore sur des sites o  l'on peut trouver des keygens et des cracks, car g n ralement ces programmes apparemment inoffensifs cachent les 3/4 du temps un virus ou un trojan ( tude r alis e par la soci t  Kaspersky Lab en 2007)...

Si vous ne pouvez ou si vous ne voulez pas payer des logiciels, il existe l'alternative des logiciels open source ainsi que d'autres logiciels qui peuvent  tre utilis s gratuitement. Enfin, il faut toujours garder   l'esprit de ne jamais divulguer ses informations personnelles sur des forums ou des chats, plus encore si Internet est utilis  par des enfants. Dans ce cas l , il faut veiller   utiliser un contr le parental tout en limitant les installations (cf. "liens annexes").

3. La d tection

3.1. (E;5)...Touch !

Suite   une infection, il faut se poser quelques questions.

-Comment cela a t  possible ?

-Par quel moyen le malware a-t-il r ussi   s'installer ?

  ces deux questions, on peut  mettre l' ventualit  que le programme malveillant ait r ussi   s'installer en exploitant diverses vuln rabilit s, comme une application configur e par d faut (voire mal configur e), un oubli de mise   jour d'un  l ment de s curit , un mot de passe trop court ou pas assez complexe, etc.   cela, on peut aussi rajouter "le clic de trojan" qui a fait s'installer le malware avec un nom all chant, une confiance trop aveugle en l'exp diteur d'une pi ce jointe, une installation faite par ing nierie sociale.

La deuxi me partie de cet article ne pr tend pas d tecter tous les malwares, mais simplement vous aider dans la recherche puis la suppression de malwares, gr ce   de simples  v nements qui vont trahir la pr sence de logiciels malveillants,   savoir :

- l' cran bleu, cela r sulte d'un bug dans le fonctionnement du syst me Windows. Ce genre d'erreur se produit g n ralement lorsqu'il manque le driver d'une carte graphique ou de tout autre composant de l'ordinateur, mais si cela se produit assez fr quemment et depuis quelques temps, il se peut que ce soit un malware (ou un rootkit) qui soit   l'origine de ce probl me ;

- message d'erreurs r current ;

- activation du mat riel tel que l'ouverture et la fermeture du lecteur de c d rom, message provenant de la barre des t ches annon ant la d sactivation d'un  l ment de s curit , ou le d placement non voulu de la souris. Ces sympt mes trahissent la pr sence de trojans sur votre syst me d'exploitation ;

- la présence de fichiers inconnus tels que des musiques, des parties de film ou des fichiers avec une extension non reconnue. Si la présence de ces fichiers n'est pas due à l'utilisation de votre entourage, cela signifie que le système sur lequel vous travaillez est compromis et qu'il est vraisemblablement utilisé comme machine zombie servant aux téléchargement ou à l'hébergement de fichiers illégaux. Cela peut être dangereux si ces fichiers ont un caractère pédo-pornographique, et en cas de constat avéré, il est sage d'alerter les autorités compétentes (liens) ;

- l'apparition répétitive de la boîte de dialogue UAC (User Account Control) qui est propre au système d'exploitation de Windows Vista (et ses successeurs). Cette boîte de dialogue apparaît à chaque fois qu'il y a besoin des droits d'administrateurs pour installer un programme (la chercheuse en sécurité informatique, Joanna Rutkowska, a réussi à démontrer qu'il était possible d'infecter Windows Vista en outrepassant cette sécurité) ;

- un fort ralentissement du système d'exploitation, qui se traduit par une utilisation du processeur qui tend vers les 100 % et/ou une mémoire qui est saturée après un certain laps de temps d'utilisation. Ces symptômes trahissent la présence d'un malware qui est trop gourmand en ressource ;

- une connexion Internet lente ; cela peut être dû à une connexion physique mal entretenue ou un malware qui exploite la connexion Internet.

3.2. Jouer les "Sherlock Holmes"

Lorsqu'un des symptômes énoncé plus haut est présent sur votre système d'exploitation, la première chose à faire est d'essayer de lancer un rapport du logiciel Hijackthis (descriptif et utilisation du logiciel : [Lien90](#)). Ce logiciel permet de détecter les installations malveillantes sur les navigateurs Web et les malwares qui se lancent à chaque démarrage de Windows. Pour cela, une fois Hijackthis lancé, il faut sélectionner l'onglet "Do a system scan and save a logfile". À la suite du scan, le programme fournit un log qui peut être interprété par le site Web du créateur de Hijackthis, à savoir : [Lien91](#). Lors de l'interprétation du log, il faut faire attention aux fausses alertes et porter une attention toute particulière aux points d'interrogation jaunes, ainsi qu'aux croix rouges.

ID	Description	Statut	Remarque
016	Service (C:\WINDOWS\system32\smssvc.exe) - Unknown owner	?	Cet élément a été classé comme bonne par nos visiteurs.
017	Service (C:\WINDOWS\system32\smssvc.exe) - Unknown owner	?	Éléments à vérifier sur le domaine.
023	Service (C:\WINDOWS\system32\smssvc.exe) - Unknown owner	?	Ce service (smssvc.exe) a été identifié comme étant légitime.
023	Service (C:\WINDOWS\system32\smssvc.exe) - Unknown owner	?	Ce service (smssvc.exe) a été identifié comme étant légitime.
023	Service (C:\WINDOWS\system32\smssvc.exe) - Unknown owner	?	Ce service (smssvc.exe) semble être légitime.
023	Service (C:\WINDOWS\system32\smssvc.exe) - Unknown owner	?	Ce service (smssvc.exe) semble être légitime.
023	Service (C:\WINDOWS\system32\smssvc.exe) - Unknown owner	?	Ce service (smssvc.exe) a été identifié comme étant légitime.
023	Service (C:\WINDOWS\system32\smssvc.exe) - Unknown owner	?	Ce service (smssvc.exe) a été identifié comme étant légitime.
023	Service (C:\WINDOWS\system32\smssvc.exe) - Unknown owner	?	Ce service (smssvc.exe) a été identifié comme étant légitime.

Aperçu d'un rapport Hijackthis

Une fois l'origine de l'infection ciblée, il suffit de se pencher sur le problème en recherchant d'éventuels détails..... À l'heure actuelle, le meilleur outil permettant la recherche de renseignements sur un fichier (et sur tout autre caractère) reste sans aucun doute le moteur de recherche Google. En effet, il suffit de taper dans la barre de recherche le nom du fichier suspect pour tomber sur des analyses qui ont eu lieu et/ou sur des commentaires divers (forum, éditeurs d'antivirus, etc). Dès lors, l'utilisation du moteur de recherche permet de confirmer les suspicions d'un fichier malveillant.

[Viruslist.com - Backdoor Win32 Rbot.bni](#)
 %System%\Irdvxc.exe. Après quoi, la backdoor exécute cette copie toutes les deux secondes avec les paramètres de ligne de commande suivants : ...
[www.viruslist.com/fr/virus/encyclopedia?virusid=141920-23k - En cache - Pages similaires](#)

Exemple de recherche concluante sur un fichier suspect

Il existe aussi sur la toile des sites spécialisés qui permettent l'étude d'un fichier suspect (lien). On peut citer le site d'analyse dynamique Virus Total qui, une fois le fichier suspect uploadé, le soumet à une trentaine d'antivirus. Cependant, il faut rester vigilant aux fausses alertes qui peuvent être émises par le moteur de recherche (un patch officiel d'un élément sécurité peut être vu par un antivirus comme un élément néfaste, cela est rare mais ça arrive quelquefois).

Fichier LiteVideocodec.exe reçu le 2008.11.29 19:49:48 (CET)
 Situation actuelle: **terminé**
 Résultat: **9/37 (24.33%)**

Antivirus	Version	Dernière mise à jour	Résultat
AhnLab-V3	2008.11.28.2	2008.11.29	-
AntiVir	7.9.0.36	2008.11.29	TR/Crypt.XPACK.Gen
Authentium	5.1.0.4	2008.11.28	-
Avast	4.8.1281.0	2008.11.28	-
AVG	8.0.0.199	2008.11.29	-
BitDefender	7.2	2008.11.29	MemScan:Trojan.FakeAlert.APF
CAT-QuickHeal	10.00	2008.11.29	(Suspicious) - DNAScan
ClamAV	0.94.1	2008.11.29	-
DrWeb	4.44.0.09170	2008.11.29	Trojan.Download.13276
eSafe	7.0.17.0	2008.11.27	-
eTrust-Vet	31.6.6234	2008.11.28	-
Ewido	4.0	2008.11.29	-
F-Prot	4.4.4.56	2008.11.28	-
F-Secure	8.0.14332.0	2008.11.29	-
Fortinet	3.117.0.0	2008.11.29	-
GData	19	2008.11.29	MemScan:Trojan.FakeAlert.APF

Extrait d'un rapport d'analyse d'un faux codec vidéo à partir du site Virus Total

3.3. Les sites Web suspects

3.3.1. Les "recherches passives"

Lorsqu'on est sur le Web, il nous arrive parfois de douter sur l'authenticité d'un site. Afin de pallier ce doute, il existe sur la toile quelques sites proposant le scan online d'une page Web (lien bas). Il faut comprendre ici par scan online le fait que le site dynamique va scanner la page ciblée en vérifiant le code source de la page douteuse. Généralement, les codes sources qui sont utilisés par le pirate pour permettre des infections, ne sont que guère modifiés.

exploit
 PREVENTION LABS

Home
 Products
 Resource Center
 Support
 About Us
 Media Center
 Contact Us

Quick Links
 • [LinkScanner.Pro](#)
 • [ThreatCenter](#)
 • [EXP Blog](#)

LinkScanner ONLINE
 Keep Your Surfing Safe

There was 1 threat found.

DANGEROUS: LinkScanner Online has found [Rogue spyware scanner]

Detail: Exploit: Trojan installer
 This is code that is used to trick victims into installing potentially unwanted software.

Risk Category: Exploit
Description: XPL's Intelligence Network has detected an exploit. An exploit is a piece of malware code that takes advantage of a vulnerability in a software application, usually the operating system or a web browser to infect a computer. Exploits usually target a computer by means of a drive-by-download - the user has no idea that a download has even taken place. XPL recommends not visiting this web site regardless if your computer has been patched for the vulnerability.

Scanned: Friday, February 20, 2009

Résultat du site LinkScanner concernant une page frauduleuse

De plus, si l'on souhaite obtenir d'avantage d'informations sur un site suspect, il est alors possible d'utiliser des outils de collecte d'informations dite "passive". À cela, on peut citer le site ripnet.com ([Lien92](#)) ou encore "le couteau suisse" l'internet Robtex.com ([Lien93](#)). Il rassemble à lui tout seul bon nombre d'outils permettant la collecte de différents types, tels que :

- RBL
- DNS
- IP-number
- route
- AS numbers
- BGP annouements
- AS macros

à l'heure de la publication, l'exemple suivant est périmé.

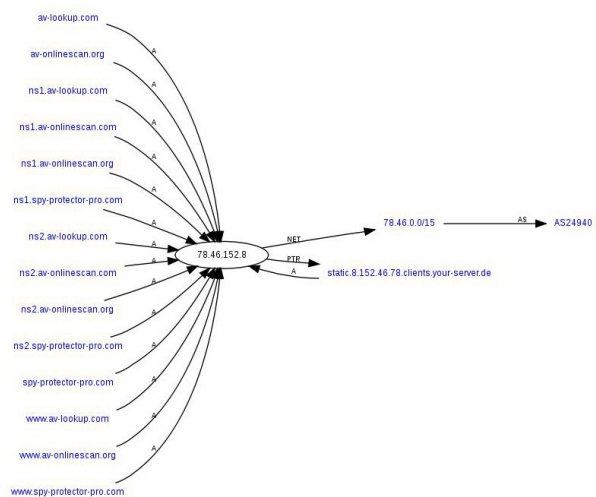
Par exemple, si nous recherchons des informations sur la localisation géographique d'un site qui nous paraît suspect (ici ? adresse ?), on peut utiliser l'un des outils proposés pour connaître l'adresse IP du serveur qui héberge le site, leurs noms, l'AS ([11](#)) ainsi que l'adresse mail qui a servi à enregistrer le site. Si l'on creuse un peu plus, on apprend que le serveur en question est localisé en Russie, et plus particulièrement à Saint-Petersburg. On remarque également le (faux) nom personnel utilisé par le pirate, qui a permis d'enregistrer le nom de domaine.

```
% This is the RIPE Whois query server #1.
% The objects are in RPSL format.
% % Rights restricted by copyright.
% See http://www.ripe.net/db/copyright.html
% Note: This output has been filtered.
% To receive output for a database update, use the "-B"
flag.
% Information related to '78.46.152.0 - 78.46.152.63'
```

```
inetnum: 78.46.152.0 - 78.46.152.63
netname: FASTVPSRU
descr: FastVPS.ru
country: DE
admin-c: OL203-RIPE
tech-c: OL203-RIPE
status: ASSIGNED PA
mnt-by: HOS-GUN
source: RIPE # Filtered
person: Oleg Lyubimov
address: Leninskiy pr. 96-1-128
address: 198332 Saint-Petersburg
address: RUSSIAN FEDERATION
phone: +79219707212
fax-no: +79219707212
e-mail: oleg.lyubimov@gmail.com
nic-hdl: OL203-RIPE
mnt-by: HOS-GUN
source: RIPE # Filtered
% Information related to '78.46.0.0/15AS24940'
route: 78.46.0.0/15
descr: HETZNER-RZ-NBG-BLK5
origin: AS24940
org: ORG-HOA1-RIPE
mnt-by: HOS-GUN
source: RIPE # Filtered
organisation: ORG-HOA1-RIPE
org-name: Hetzner Online AG
```

```
org-type: LIR
address: Hetzner Online AG
Attn. Martin Hetzner
Stuttgarter Str. 1
91710 Gunzenhausen
Germany
phone: +49 9831 610061
fax-no: +49 9831 610062
e-mail: info@hetzner.de
admin-c: GM834-RIPE
admin-c: HOAC1-RIPE
admin-c: MH375-RIPE
admin-c: RB1502-RIPE
admin-c: SK2374-RIPE
mnt-ref: HOS-GUN
mnt-ref: RIPE-NCC-HM-MNT
mnt-by: RIPE-NCC-HM-MNT
source: RIPE # Filtered
```

De plus, grâce aux fonctionnalités de robtex, il est possible de visualiser l'arborescence des noms de sites qui composent l'enregistrement de domaine.

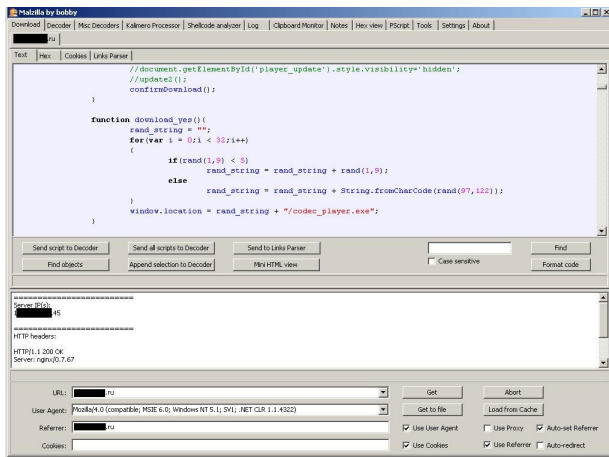


Visualisation de l'arborescence

Enfin, si l'on étudie la source de l'une de ces pages vitrines, on remarque que les malwares ne sont pratiquement jamais hébergés sur le même serveur; ils sont téléchargés à partir d'un lien annexe. C'est ce que nous allons voir tout de suite dans le chapitre suivant.

3.3.2. Étude plus approfondie

Malzilla est un utilitaire bien utile lorsqu'il s'agit d'analyser en profondeur une page suspecte. En effet ce petit logiciel permet de décoder bon nombre de fonctions qui sont parfois utilisées dans des applications Web, comme l'encodage en base 64. Il permet aussi d'afficher la page en format texte ou hexadécimal et prend en charge l'analyse des cookies, des logs...



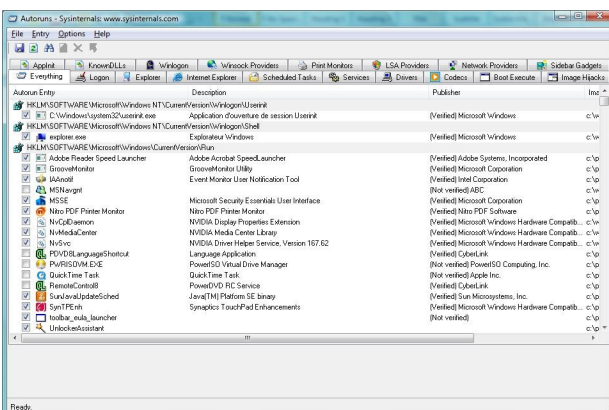
Dans l'exemple précédent, nous pouvons voir distinctement que le site héberge un malware (dans mon cas, le logiciel s'est avéré être un ransomware) intitulé "codec_player.exe".

4. Les solutions

4.1. Vérifier les programmes lancés au démarrage et vérifier la mémoire

4.1.1. Les programmes qui se lancent à chaque démarrage

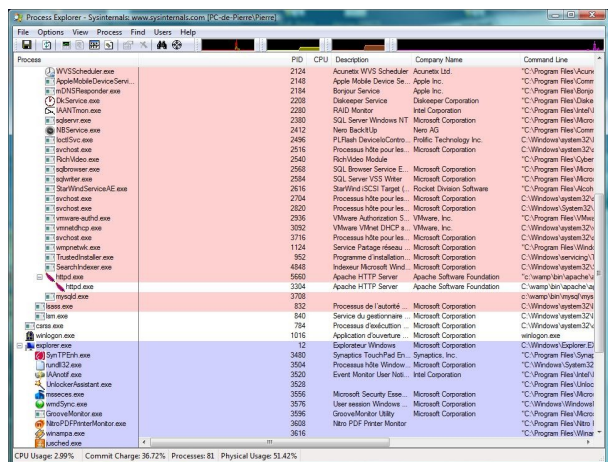
Pour être toujours actifs, les malwares ont besoin d'être lancés à chaque démarrage du système et, pour cela, ils vont devoir modifier la base de registre, ainsi que les divers paramètres qui sont liés au démarrage des fichiers. L'outil Autoruns de chez Microsoft liste tous les programmes et bibliothèques qui sont automatiquement exécutés à chaque démarrage de Windows. L'avantage majeur de cet outil est qu'il permet de voir les programmes qui se lancent pour tous les utilisateurs, ainsi que ceux lancés par l'utilisateur lui-même. Les programmes lancés au démarrage sont classés par types (Logon, Explorer, Internet Explorer, etc...). Afin de ne pas surcharger l'affichage, il faut veiller à cocher les paramètres suivants : Verify Code Signature et Hide Microsoft Signed Entries, se trouvant dans l'option du menu. Il est à noter que l'onglet Logon liste tous les programmes exécutés à l'ouverture d'une session.



Aperçu de l'outil Autoruns

4.1.2. Vérifier la mémoire

Après avoir listé les programmes qui se lancent au démarrage, il est bon de continuer l'investigation dans l'analyse du système. Pour ce faire, on peut utiliser l'outil Process Explorer, et ceci afin de lister les processus tout en donnant les caractéristiques, les bibliothèques utilisées, port utilisé...



Aperçu du logiciel Process Explorer

Afin de faciliter l'exploration, il faut avant tout configurer le logiciel. Pour cela il suffit d'aller dans l'onglet View de sélectionner "Select Columns" pour cocher les cases suivantes : Company Name, Command Line, Description, Version et Verified Signer. Toutes ces options permettent de nous renseigner sur l'origine du programme (cf. Company Name), sur son chemin d'accès (cf. Command Line) qui, s'il est différent des chemins habituels peut être considéré comme un programme suspect. Dès lors, il suffit d'utiliser le moteur de recherche pour en savoir plus sur un fichier suspect...

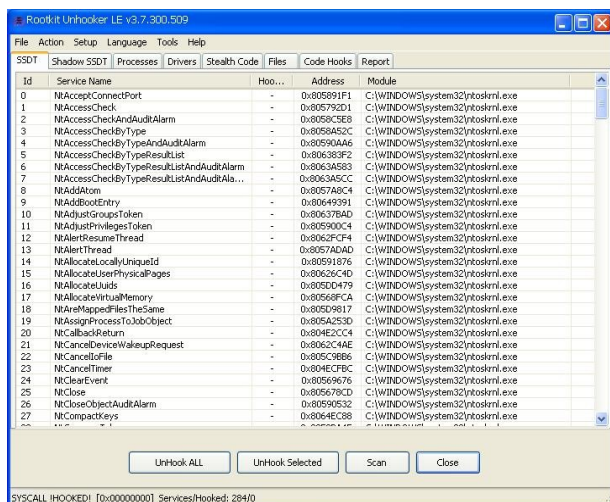
4.2. Voyage au centre du système: le noyau

Le noyau est la pièce maîtresse de tout système d'exploitation.

En effet, il gère les ressources de l'ordinateur et permet aux différents éléments qui composent l'ordinateur de communiquer entre eux ; il est par exemple responsable du bon déroulement des programmes qui sont en cours d'exécution. De ce fait, lorsqu'un malware ou plus particulièrement un rootkit arrive à compromettre le fonctionnement du noyau, il est alors possible que celui-ci dissimule des informations à l'utilisateur du système compromis. Il peut en effet cacher un accès à une porte dérobée ou cacher un programme en cours d'exécution, ce qui rend particulièrement difficile la détection de ce genre de malwares, car les antivirus ont une confiance aveugle en tout ce qui concerne les appels système ou plus particulièrement les pilotes de périphérique (cf. "Liens annexes : Revue, MISC").

À l'heure actuelle, il n'existe aucun moyen fiable pour détecter la présence d'un rootkit, mais certaines compromissions du noyau trahissent la présence de malwares. De ce fait, le site antirootkit.com ([Lien94](#)) recense un nombre important de rootkits et présente de nombreux logiciels permettant de les détecter et de les supprimer. On peut citer le logiciel "Aries Rootkit Remover" permettant de supprimer les rootkits de la

société Sony, suite à l'écoute de l'un de leur CD musicaux, ou encore "RootKit UnHooker" qui permet de détecter les processus cachés ainsi que les Hooks...



Aperçu du logiciel Rootkit UnHooker

Le logiciel Rootkit UnHooker présente la particularité d'être assez simple d'emploi pour tous ceux qui désirent vérifier la présence d'éventuels rootkits sur leur système d'exploitation.

Il faudra veiller à rester vigilant sur les chemins des exécutable dans les onglets "Drivers" et "Processus", principalement si les chemins soulignent un répertoire temporaire ou qui est caché, il est alors bon de faire une recherche plus approfondie.

4.3. "Élémentaire mon cher Watson !" : la suppression des malwares

Suite à une étude approfondie d'un poste compromis, il est bon de prendre en note le chemin du malware. Face à cela, il convient de supprimer au plus vite le malware afin qu'il ne puisse plus se relancer aux prochains redémarrages du système, cependant et outre la propre protection du fichier malveillant, Windows bloque à chaque fois la suppression d'un fichier lorsque celui est en cours d'exécution. Ces deux méthodes combinées offrent aux pirates l'opportunité de voir leur malware persister sur le système compromis. Toutefois, il existe un outil de suppression nommé Unlocker qui passe outre ces restrictions.



Aperçu de l'outil Unlocker

Parfois, il arrive que Unlocker et/ou l'antivirus ne puissent supprimer un malware car le pirate qui a codé le programme a fait en sorte de pallier ce genre de suppression. Il est alors bon se tourner vers Linux, en démarrant l'ordinateur à partir d'un Live CD et de le supprimer depuis ce système d'exploitation.

5. Comment éviter les problèmes

5.1. Outils élémentaires d'un poste sécurisé

Un poste sécurisé est synonyme de logiciels de sécurité et de mises à jour.

En effet, cela doit absolument passer par l'utilisation d'un antivirus qui jouera le rôle de détecteur de malwares ainsi que d'un pare-feu permettant de restreindre l'accès à un système d'exploitation depuis l'Internet. Bien évidemment, tous ces éléments doivent régulièrement être mis à jour lorsque le logiciel ou le système le rappelle. Il est bon d'éviter d'installer des logiciels de peer to peer, car généralement, ils recèlent des publicités que l'on ne souhaite pas forcément recevoir lorsque l'on est sur le net (ex. Kaza), il en va de même pour les pseudos logiciels permettant de personnaliser notre messagerie instantanée, etc. De plus, il est souhaitable de nettoyer de temps à autre le système d'exploitation avec des logiciels appropriés tels que Ccleaner.

5.2. Technique de cryptage

L'analyse du trafic réseau est une activité qui plait énormément aux spécialistes de la sécurité ainsi qu'aux Hackers. Grâce à des logiciels spécialisés dans l'écoute du trafic réseau (exemple : wireshark [\(Lien95\)](#)), ils peuvent alors stocker les paquets du trafic pour par la suite, lire le contenu des communications. Il est à noter que la plupart de ces outils se basent sur le fait que certains protocoles font transiter des informations en clair sur le réseau. Dès lors, si le réseau est partagé par un HUB, alors de simples outils d'écoutes suffisent pour sniffer et déchiffrer le contenu. Cependant si le réseau commuté est configuré avec un switch, le pirate ou le spécialiste mettra alors en place une attaque de type MAC flooding ou un ARP spoofing. Afin de pallier ces faiblesses, il existe des protocoles qui utilisent une stratégie cryptographique.

Avant de rentrer dans le vif du sujet, je vous suggère l'article suivant, écrit par l'un des membres du site : Introduction aux réseaux TCP IP [\(Lien96\)](#)

5.2.1. Le réseau et cryptographie

OSI	TCP/IP	Protocoles
7-Application	HTTP, FTP, Telnet	PGP, GnuPG, SET, S-HTTP, S/MIME
6-Présentation		
5-Session		
4-Transport	UDP, TCP	SSL, TLS, STLP, PCT, IPSEC
3-Réseau	Interconnexion de réseau IP ICMP	IPsec - IP MPLS couche 3 (BGP)
2-Liaison de données	Interface réseau Ethernet, Frame Relay, ATM, PPP	
1-Physique		

5.2.2. Quelques procédés de chiffrement

PGP (Pretty Good Privacy ([Lien97](#)))

PGP est une combinaison des meilleures fonctionnalités de la cryptographie de clés publiques et de la cryptographie conventionnelle.

C'est aussi un système de "cryptographie hybride". Lorsqu'un utilisateur utilise la cryptographie PGP pour chiffrer un texte, les données sont d'abord compressées. Cela permet de réduire le temps de transmission, d'économiser l'espace sur le disque et, surtout, de renforcer la sécurité cryptographique.

Il s'ensuit, une opération de chiffrement qui se déroule en deux étapes :

- PGP crée une clé secrète IDEA de manière aléatoire, et chiffre les données avec cette clé ;
- ensuite, PGP crypte la clé secrète IDEA et la transmet au moyen de la clé RSA publique du destinataire.

Lors de la phase de décryptage, cela se passe également en deux étapes :

- PGP déchiffre la clé secrète IDEA grâce à la clé RSA privée ;
- PGP déchiffre les données avec la clé secrète IDEA obtenue précédemment.

SSH (Secure Shell ([Lien98](#)))

Mis au point en 1995 par le Finlandais Tatu Ylönen, ce protocole permet à un client d'ouvrir une session sur une machine distante afin d'envoyer des commandes ou des fichiers de manière sécurisée.

Les données qui circulent entre le client et la machine distante sont chiffrées, permettant ainsi une grande confidentialité et rendant impossible l'écoute de la communication.

IPSec (Internet Protocol Security ([Lien99](#)))

C'est un ensemble de RFC (Request For Comments ([Lien100](#))) destinées à utiliser des algorithmes permettant l'acheminement de données sécurisées au sein d'un réseau IP.

Cela permet donc d'authentifier et de chiffrer les données, ce qui aura pour conséquence de rendre le flux de données que par le destinataire final tout veillant à l'intégrité de celle-ci.

SSL (Secured Socket Layer ([Lien101](#)))

Élaboré au début par Netscape, le protocole SSL offre une communication sécurisée au niveau de la 4e couche (transport).

Il se base sur les algorithmes RSA et utilise des certificats pour identifier l'émetteur et le récepteur.

TLS (Transport Layer Security ([Lien102](#)))

Basé sur SSLv3, il permet de remplacer le SSL. Ne dépendant pas des algorithmes déposés par la RSA, il comporte beaucoup de problèmes légaux.

6. Conclusion générale

Bon nombre d'éditeurs de systèmes de protection vendent leur produit pour répondre à une demande des utilisateurs ainsi que pour pallier le nombre croissant des menaces liées à Internet. Cependant, il ne faut pas se croire entièrement en sécurité derrière un simple antivirus, car celui-ci n'est qu'un maillon parmi tous les éléments de sécurité (pare-feu, filtre, accès restreint par mot de passe, etc.). Si le matériel informatique est à disposition de tierces personnes, comme dans les bibliothèques, dans les lycées avec le CDI... il est alors bon de rappeler les bonnes pratiques d'une utilisation d'Internet en y affichant des rappels d'utilisation (post-it), à travers les écrans de veille ou sur le fond d'écran de l'ordinateur et d'en limiter leur utilisation (ex. filtrage des adresses MAC si cela est dans un réseau Wi-Fi, droits d'administrateur restreints, téléchargement interdit, logiciel de contrôle à distance pour l'administrateur, etc). Ces rappels passent aussi par une sensibilisation de l'utilisation de l'ordinateur et d'Internet dans les écoles ou les lycées. Les chats ouverts à tout publique sont fortement à proscrire, surtout lorsque l'ordinateur est utilisé par des enfants. Enfin, il est intéressant de se tenir informé des dernières vulnérabilités en visitant des sites comme Zataz.com, Vulnérabilite.com, voire Developpez.com.

7. Liens annexes

- Club des développeurs : [Lien103](#)
- Portail de la Sécurité : [Lien104](#)
- Journal de l'actualité informatique et multimédia : [Lien105](#)
- Site regroupant les dernières vulnérabilité afin de se tenir à jours des découvertes : [Lien106](#)

Microsoft

- Qu'est-ce que Microsoft ? : [Lien107](#)
- Site officiel : [Lien108](#)
- Microsoft TechNet : [Lien109](#)
- FAQ Windows sur Developpez.com : [Lien110](#)

Linux

- Qu'est-ce que Linux ? : [Lien111](#)
- Distribution Linux : [Lien112](#)
- FAQ Linux sur Developpez.com : [Lien113](#)

Protections

- Débat sur les meilleurs antivirus : [Lien114](#)
- Débat sur les meilleurs par-feux : [Lien115](#)
- L'ABC de la sécurité, fait par l'un des membres de developpez.com (Shoky) : [Lien116](#)
- FAQ Sécurité de developpez.com : [Lien117](#)
- Vulgarisation de la sécurité informatique : [Lien118](#)

Scanner Online

- Trend Micro : [Lien119](#)
- BitDefender : [Lien120](#)
- Kaspersky lab : [Lien121](#)

Sites web dynamiques proposant d'analyse d'un fichier suspect

- UploadMalware : [Lien122](#)
- VirusTotal : [Lien123](#)
- Anubis : [Lien124](#)
- Jotti's malware scan : [Lien125](#)
- ThreatExpert (site développé par une équipe de professionnels qui sont à l'origine du site PC Tools') : [Lien126](#)
- VirSCAN Online AV Scanner : [Lien127](#)
- Filterbit : [Lien128](#)

Logiciels Utiles

- Les meilleurs outils gratuits pour Windows : [Lien129](#)
- Malzilla : [Lien130](#)
- codec video pour pouvoir lire tous les formats video : [Lien131](#)

Protection de l'enfance

- Protection des mineurs : [Lien132](#)
[Lien133](#)
[Lien134](#)
- Contrôle parental : [Lien135](#)
[Lien136](#)
- Signaler un site: [Lien137](#)
- spots publicitaires : [Lien138](#)
[Lien139](#)
- Internet sans craintes : [Lien140](#)

Pour aller plus loin

- The New York Times, 20 Octobre 2008, John Markoff, "A Robot Network Seeks to Enlist Your Computer" : [Lien141](#)
- Google Black-list : [Lien142](#)
- Anti-phishing et vie privée : [Lien143](#)

- Interview de Eric Filliol : "L'Etat doit s'appuyer sur les hackers" ([Lien144](#))

Reportage

- National Geographic, Hackers
- ARTE, Cyber Guérilla
- Envoyé Spécial, Cybercriminalité
- ARTE, Tous fichés

Livres

- La Taz, Peter Lamborn Wilson dit Hakim Bey, pour comprendre en partie comment fonctionne la mentalité des pirates.
- Tsun zu, l'art de la guerre
- Sécurité informatique, principe et méthode à l'usage des DSI, RSSI et administrateur
- Chaînes d'exploits, scénarios de Hacking avancé et prévention

Revue

- Science et vie, Décembre 2008, n° 1095, Internet au bord de l'explosion
- Misc n°41, Dossier: La cybercriminalité..ou quand le net se met au crime organisé

- Misc n°47, Dossier: La lutte antivirale, une cause perdue ?

- Hakin9 n°31, Secrets de cybercriminels, Phisers, spammeurs, voleurs d'identité

- Courrier International, 2 au 22 Aout 2007, n°874-875-876, rubrique Europe > Russie, Les Hackers investissent le champ politique

- Courrier International, 5 au 11 Mars 2009, n°957, rubrique Europe > Multimédia, Hackers en uniforme

- Courrier International, /////////////// 2009, n°//, rubrique Science, cyberguerre

Retrouvez cet article de Pierre Therrode en ligne : [Lien145](#)

Les dernières news

Un chercheur expose des failles du noyau de Linux pour l'améliorer, Red Hat et surtout la NSA verraient ses efforts d'un très mauvais œil

Expert reconnu du Kernel, le chercheur Brad Spengler a présenté hier sa vision de la sécurité de Linux durant la conférence LinuxCon qui se déroule cette semaine à Boston.

Spengler y a, par exemple parlé de « **Enlightenment** », un exploit de son cru qui permet - ni plus ni moins - de désactiver la politique de sécurité de Linux, y compris SELinux et AppArmor.

« Enlightenment » qui prouve, selon lui, que **les contrôles**

d'accès ne sont pas la protection ultime, mais doivent plus être considérés comme une dernière ligne de défense.

Spengler affirme avoir codé 11 autres exploits du Kernel durant les dernières années. Des efforts qui ont aidé à **rendre Linux plus sûr**.

Il explique son acharnement à produire autant d'exploits par la passivité des entreprises qui développent Linux, à commencer par **Red Hat qui n'a réagi à aucun de ses sept premiers exploits** qui menaçaient la sécurité des utilisateurs de Red Hat Entreprise Linux.

Ce n'est qu'au bout du huitième que l'entreprise aurait finalement réagi en corrigeant toutes les failles. Pour lui, **seules les failles rendues publiques semblent pouvoir produire un changement** dans la perception de la sécurité (lire par ailleurs : Google veut réinventer les règles de divulgation des failles ([Lien146](#)))

Si l'on se doute que Red Hat ne sera pas ravi par ces affirmations, d'autres sont aussi passablement énervés par les travaux de Spengler. **La NSA (Agence de Sécurité Nationale), qui a participé au développement de SELinux, voit ses efforts d'un très mauvais œil.**

Elle aurait même réagi en contactant, non pas le chercheur, mais son employeur afin de **nuire à sa carrière.**

Mais pour lui la fin justifie les moyens et SELinux serait à présent un bien meilleur produit.

Spengler présente aussi **un ensemble de pistes pour améliorer la sécurité** de Linux.

Il suggère en premier lieu l'usage de la « **randomisation de l'espace d'adressage** » (ASLR), une technique d'adressage aléatoire des objets partagés dans la mémoire. Cette technique pourrait compliquer les attaques qui exploitent les adressages mémoires écrits en dur.

Il propose également de **retirer du noyau ce qu'il appelle des « infoleaks »** (fuites d'informations) comme la fonction « slabinfo » qui rapporte la taille des blocks de mémoire et peut être potentiellement utilisée par les pirates.

D'autres éléments du Kernel mériteraient également d'être mieux sécurisés, comme « syscall », la table des appels système qui doit être, selon lui, passée en lecture seule.

Mais selon Spengler, **la sécurité ne concerne pas que les développeurs.** Des mesures prises par les administrateurs et les utilisateurs peuvent aussi mener à des systèmes plus sûrs.

Pour mémoire, Brad Spengler est l'expert en sécurité à l'origine du projet « grsecurity », un patch permettant d'ajouter des fonctionnalités de sécurité au noyau Linux.

Commentez cette news de Idelways en ligne : [Lien147](#)

Un groupe de chercheurs anonymes attaque Microsoft en dévoilant une vulnérabilité de Windows pour se venger de sa campagne anti-Ormandy

Un groupe anonyme de chercheurs en sécurité a mis en ligne, la semaine dernière, des informations sur une faille non corrigée de Windows.

Cet acte est une réponse —selon leur communiqué— à l'hostilité chronique dont fait preuve Microsoft envers les experts en sécurité, avec récemment l'affaire Tavis Ormandy, l'ingénieur de Google qui a publié un PoC aujourd'hui exploité sur plus de 10.000 machines (lire les détails de l'affaire ([Lien148](#))).

Ils ont également annoncé la création du MSRC, Microsoft-Spurned Research Collective (la convention des chercheurs méprisés par Microsoft), allusion évidente au Microsoft Security Response Center qui porte les mêmes initiales (MSRC).

L'annonce de cette convention a été postée anonymement à partir d'un compte Hushmail. Elle liste six collaborateurs dont les noms sont tous représentés par des « X ». Le groupe invite d'autres chercheurs à s'y inscrire et assure avoir mis en place un système de contrôle anti-employés de Microsoft.

Microsoft, de son côté, minimise la faille dévoilée. Pour la société, un hacker doit disposer d'un accès physique ou avoir réussi un autre exploit pour l'utiliser. Elle juge ainsi le risque au dessous du seuil nécessitant une alerte de sécurité, habituellement la première étape dans la correction des problèmes de sécurité.

Reste qu'au delà du débat sur la gravité de la vulnérabilité, l'initiative du groupe d'expert est très polémique.

Aussi, et surtout, parce qu'elle est anonyme.

Un léger manque de courage ?

Commentez cette news de Idelways en ligne : [Lien149](#)

Masterminds of Programming

Masterminds or Programming features exclusive interviews with the creators of several historic and highly influential programming languages. In this unique interview collection, you'll learn about the processes that led to specific design decisions, including the goals these pioneers had in mind, the trade-offs they had to make, and how their experiences have made an impact on programming today. This book includes interviews with :

- Adin D. Falkoff: **APL**
- Thomas E. Kurtz: **BASIC**
- Charles H. Moore: **FORTH**
- Robin Milner: **ML**
- Donald D. Chamberlin: **SQL**
- Alfred Aho, Peter Weinberger, and Brian Kernighan: **AWK**
- Charles Geschke and John Warnock: **PostScript**
- Bjarne Stroustrup: **C++**
- Bertrand Meyer: **Eiffel**
- Brad Cox and Tom Love: **Objective-C**
- Larry Wall: **Perl**
- Simon Peyton Jones, Paul Hudak, Philip Wadler, and John Hughes: **Haskell**
- Guido van Rossum: **Python**
- Luiz Herique de Figueiredo and Roberto Ierusalimsky: **Lua**
- James Gosling: **Java**
- Grady Booch, Ivar Jacobson, and James Rumbaugh: **UML**
- Anders Hejlsberg: Delphi inventor and lead developer of C#

If you're interested in the people whose vision and hard work helped shape the computer industry, you'll find *Masterminds of Programming* fascinating.

Critique du livre par Matthieu Brucher

Quand on met ensemble une vingtaine de créateurs de langage, le résultat ne peut être qu'intéressant. C'est un bon révélateur de caractère, comme ils tendent à ouvrir leur cœur. Je pense que c'est exactement ce qui s'est produit ici.

Pas besoin de lister les langages proposés, ils font partie des langages à succès, en termes de qualité ou en termes de renommée, ce qui veut dire que certains peuvent être peu connus. Chaque chapitre est très différent des autres et, comme je l'ai dit auparavant, est très révélateur de son auteur.

En fonction de la personne interviewée, les topics abordés sont différents. Pour certains auteurs, on a des questions sur des thèmes proches de leur langage. D'autres, parlent de leur vision du futur de la programmation. Certains vont nous dire à quel point leur langage est superbe et que c'est une honte qu'il ne soit pas plus utilisé ou qu'il pourrait tout remplacer... D'autres encore vont vous dire que le langage leur a échappé des mains.

Il n'y a pas une seule ligne de code, c'est surtout un livre de pensées, et c'est le cœur du livre : on n'a pas de conseil sur comment écrire son code ou designer son application. On récupère les raisons fondamentales derrière un langage et leur succès. Des pistes sont données sur le futur de nos programmes, comment les langages vont évoluer et sur quoi il faut potentiellement se focaliser. Malheureusement, le chapitre que j'ai presque le plus apprécié est le dernier (sur Eiffel). C'est drôle que dans ce celui-ci, Bertrand Meyer parle des pépites qu'il a trouvées dans le dernier chapitre d'un autre ouvrage, Structured Programming (Dijkstra et al).

D'un point de vue culturel, ce livre est une mine d'or. D'un point de vue des ego, ce livre est un révélateur. Toutes choses étant égales, un grand nombre de choses peuvent être apprises en lisant ce livre sur l'approche philosophique d'un langage.

Critique du livre par Doeraene Sébastien

Lors que j'ai lu le résumé de ce livre, j'ai su que je voulais le lire. Et à la fin de ma lecture, je ne suis absolument pas déçu, bien au contraire !

Au contraire des ouvrages habituels sur la programmation, celui-ci ne vous apprendra pas le moins du monde à programmer, ni même à concevoir, ni ne vous apprendra un quelconque nouveau langage. Mais il vous fera découvrir les états d'esprit, de synthèse, de conception, de ce que l'auteur a nommé les *masterminds* (les *génies*) de la programmation.

Il est difficile de critiquer ce livre sur la base de son écriture ou de sa pédagogie, qui sont tout simplement sans objet ici. En revanche, je soulignerai la qualité du choix des questions posées en interview, ainsi que l'apport global que le lecteur pourra en tirer, qui sont exceptionnels.

Je recommande en particulier ce livre à tous ceux qui conçoivent - ou qui ont l'intention de concevoir - un langage de programmation. Les conseils, prodigués par les maîtres en la matière, que cet ouvrage nous livre leur seront d'une aide incontestable. Ceci est valable également pour les étudiants pour qui ce serait un projet de cours ou de mémoire.

Ceci ne veut pas dire pour autant que je ne le conseille pas aux autres développeurs. Les discussions avec ces génies

de la programmation seront bénéfiques à tous, ne serait-ce que pour comprendre les raisons d'existence des langages qu'ils utilisent au quotidien.

En bref, un livre à lire au moins une fois dans sa vie - si pas plus !

*Retrouvez cette critique de livre sur la page livres
Conception : [Lien150](#)*

Liens

- Lien1 : <http://netapsys.developpez.com/tutoriels/java/tests-junit4-spring/>
- Lien2 : <http://commonsware.com/mmurphy>
- Lien3 : <http://developer.android.com/index.html>
- Lien4 : <http://www.pearson.fr/livre/?GCOI=27440100347170>
- Lien5 : <http://java.developpez.com/livres/>
- Lien6 : <http://uml.developpez.com/>
- Lien7 : <http://julien-pauli.developpez.com/tutoriels/php/phpunit-avance/>
- Lien8 : <http://julien-pauli.developpez.com/tutoriels/php/liaisons-uml/>
- Lien9 : <http://www.xdebug.org/>
- Lien10 : <http://www.php.net/gc>
- Lien11 : <http://julien-pauli.developpez.com/tutoriels/php/variables/>
- Lien12 : <http://www.symfolive.com/6-plugins-symfony-indispensables/>
- Lien13 : <http://www.developpez.net/forums/d930713/php/bibliotheques-frameworks/symfony/points-forts-d-framework-php-tel-symfony/>
- Lien14 : <http://flxhr.flensed.com/>
- Lien15 : <http://www.kezen.fr/wp-content/uploads/2010/07/flXHR-1.0.5.zip>
- Lien16 : <http://flxhr.flensed.com/download.php>
- Lien17 : http://www.adobe.com/devnet/articles/crossdomain_policy_file_spec.html
- Lien18 : <http://ajax.developpez.com/tutoriels/ajax/ajax-cross-domain-avec-flxhr/>
- Lien19 : <http://www.w3.org/TR/css3-multicol/>
- Lien20 : <http://plambert.developpez.com/tutoriel/css/mise-page-colonnes-full-css/fichiers/mise-en-page-avec-tableau.html>
- Lien21 : <http://plambert.developpez.com/tutoriel/css/selecteurs-class-id/>
- Lien22 : <http://www.mammouthland.net/parisweb2007/atelier-CSS-positionnement.pdf>
- Lien23 : <http://www.mammouthland.net/parisweb2007/>
- Lien24 : <http://2007.paris-web.fr/>
- Lien25 : <http://plambert.developpez.com/tutoriel/css/style-base/#I-D>
- Lien26 : <http://plambert.developpez.com/tutoriel/css/mise-page-colonnes-full-css/fichiers/mise-en-page-css-sans-tableau.html>
- Lien27 : <http://web.accessibilisation.net/www.maxdesign.com.au/presentation/workshop/slide6.htm>
- Lien28 : http://openweb.eu.org/articles/ Problemes_tableaux
- Lien29 : http://www.openweb.eu.org/articles/initiation_flux/
- Lien30 : <http://www.css4design.com/blog/design-css-liquide-ou-fluide-encadre-d-images-repetees-sans-table>
- Lien31 : <http://plambert.developpez.com/tutoriel/css/principe/>
- Lien32 : <http://plambert.developpez.com/tutoriel/css/balise-xhtml-base/>
- Lien33 : <http://plambert.developpez.com/tutoriel/css/style-base/>
- Lien34 : <http://plambert.developpez.com/tutoriel/css/mettre-forme-texte/>
- Lien35 : <http://plambert.developpez.com/tutoriel/css/gestion-marges/>
- Lien36 : <http://plambert.developpez.com/tutoriel/css/resoudre-probleme-marges/>
- Lien37 : <http://plambert.developpez.com/tutoriel/css/bordures/>
- Lien38 : <http://plambert.developpez.com/tutoriel/css/images-fond/>
- Lien39 : <http://plambert.developpez.com/tutoriel/css/effet-rollover/>
- Lien40 : <http://plambert.developpez.com/tutoriel/css/alignement-float/>
- Lien41 : <http://plambert.developpez.com/tutoriel/css/puce-image/>
- Lien42 : <http://plambert.developpez.com/tutoriel/css/citation-retrait-texte/>
- Lien43 : <http://plambert.developpez.com/tutoriel/css/mise-page-colonnes-full-css/>
- Lien44 : <http://plambert.developpez.com/tutoriel/css/bouton/fichiers/test-bouton.html>
- Lien45 : <http://plambert.developpez.com/tutoriel/css/bouton/fichiers/test-bouton2.html>
- Lien46 : http://openweb.eu.org/articles/bouton_css/
- Lien47 : <http://plambert.developpez.com/tutoriel/css/bouton/>
- Lien48 : <http://plambert.developpez.com/tutoriel/css/plusieurs-images-fond/fichiers/test.html>
- Lien49 : <http://plambert.developpez.com/tutoriel/css/plusieurs-images-fond/>
- Lien50 : <http://sgbd.developpez.com/livres/>
- Lien51 : <http://projets.developpez.com/users/33090>
- Lien52 : <http://projets.developpez.com/projects/qextend/activity?from=2010-07-06>
- Lien53 : <http://projets.developpez.com/projects/qextend>
- Lien54 : <http://qt.developpez.com/doc/qextend/0.0.1/fr/>
- Lien55 : <http://qt.developpez.com/doc/qextend/0.0.1/en/>
- Lien56 : http://projets.developpez.com/attachments/download/478/QExtend_V0.0.1.7z
- Lien57 : <http://projets.developpez.com/projects/qextend/issues>
- Lien58 : <http://qt.developpez.com/doc/4.6/QWidget>
- Lien59 : <http://qt.developpez.com/doc/4.6/qt/>
- Lien60 : <http://qt.developpez.com/doc/4.6/threads>
- Lien61 : <http://qt.developpez.com/doc/4.6/liste-des-pages-traduites/>
- Lien62 : <http://qt.developpez.com/doc/latest/qprocess/>
- Lien63 : <http://qt.developpez.com/doc>
- Lien64 : <http://qt.developpez.com/doc/latest>
- Lien65 : <http://qt.developpez.com/doc/4.6/qwidget>
- Lien66 : http://www.developpez.net/forums/u83713/jacques_jean/
- Lien67 : <http://www.developpez.net/forums/u330467/hornetbzz/>
- Lien68 : <http://www.developpez.net/forums/u263471/Wachter/>
- Lien69 : <http://www.developpez.net/forums/u299822/furr/>
- Lien70 : <http://www.developpez.net/forums/u44506/johnlamerica/>
- Lien71 : <http://www.developpez.net/forums/u254882/dourouc05/>
- Lien72 : <http://www.developpez.net/forums/u341511/abdelite/>
- Lien73 : <http://www.developpez.net/forums/u333286/greattux/>
- Lien74 : <http://www.developpez.net/forums/u60889/XHeLL/>
- Lien75 : <http://www.developpez.net/forums/u292096/Amnell/>
- Lien76 : <mailto:doc-qt@redaction-developpez.com>
- Lien77 : <http://www.developpez.net/forums/private.php?do=newpm&u=44506>

Lien78 : <http://www.developpez.com/redirect/123>
Lien79 : <http://www.developpez.net/forums/d941430/c-cpp/bibliotheques/qt/documentation-qt-4-6-traduite-francais/>
Lien80 : <http://qt.nokia.com/downloads>
Lien81 : <http://labs.trolltech.com/blogs/2010/05/06/qt-47-scope-change-regarding-qt-multimedia/>
Lien82 : <http://labs.trolltech.com/blogs/2010/07/08/qt-47-beta2-and-qt-creator-21-snapshots-available/>
Lien83 : <http://www.developpez.net/forums/d825667/c-cpp/bibliotheques/qt/sortie-qt-4-7-release-candidate-bientot-celle-qt-creator-2-1-a/#post5334703>
Lien84 : <http://www.developpez.net/forums/d825667/c-cpp/bibliotheques/qt/sortie-qt-4-7-release-candidate-bientot-celle-qt-creator-2-1-a/#post5433378>
Lien85 : <http://securite.developpez.com/livres/>
Lien86 : <http://www.developpez.net/forums/d74061/webmasters-developpement-web/general-conception-web/securite/meilleures-pratiques-securite-protoger-site-hackers/>
Lien87 : <http://www.stopforumspam.com/>
Lien88 : <http://www.spamhaus.org/>
Lien89 : <http://fspamlist.com/>
Lien90 : <http://www.developpez.net/forums/d530859/systemes/windows/securite/hijackthis-lutliser/>
Lien91 : <http://www.hjackthis.de/fr>
Lien92 : <http://www.ripnet.com/>
Lien93 : <http://www.robtext.com/>
Lien94 : <http://antirookit.com/>
Lien95 : <http://www.wireshark.org/>
Lien96 : <http://ram-0000.developpez.com/tutoriels/reseau/tcp-ip/>
Lien97 : http://fr.wikipedia.org/wiki/Pretty_Good_Privacy
Lien98 : http://fr.wikipedia.org/wiki/Secure_Shell
Lien99 : <http://fr.wikipedia.org/wiki/IPsec>
Lien100 : http://fr.wikipedia.org/wiki/Request_for_comments
Lien101 : <http://fr.wikipedia.org/wiki/SSL>
Lien102 : http://fr.wikipedia.org/wiki/Transport_Layer_Security
Lien103 : <http://www.developpez.com/>
Lien104 : http://fr.wikipedia.org/wiki/Portail:Sécurité_informatique
Lien105 : <http://www.zataz.com/>
Lien106 : <http://www.vulnerabilite.com/>
Lien107 : <http://fr.wikipedia.org/wiki/Microsoft>
Lien108 : <http://www.microsoft.com/downloads/search.aspx?displaylang=fr>
Lien109 : <http://technet.microsoft.com/fr-fr/sysinternals/default.aspx>
Lien110 : <http://windows.developpez.com/faq/>
Lien111 : <http://fr.wikipedia.org/wiki/Linux>
Lien112 : <http://www.livecdlist.com/>
Lien113 : <http://linux.developpez.com/faq/>
Lien114 : <http://www.developpez.net/forums/d294988/hardware-systemes-logiciels/windows/securite/s-antivirus-trouvez-efficace>
Lien115 : <http://www.developpez.net/forums/d294994/hardware-systemes-logiciels/windows/securite/firewall-s-trouvez-efficace-s>
Lien116 : <http://www.developpez.net/forums/d356579/hardware-systemes-logiciels/windows/securite/labc-securite>
Lien117 : <http://securite.developpez.com/faq>
Lien118 : http://www.securite-informatique.gouv.fr/gp_mot4.html
Lien119 : <http://housecall.trendmicro.com/fr>
Lien120 : <http://www.bitdefender.com/scan8/ie.html>
Lien121 : <http://www.kaspersky.com/virusscanner>
Lien122 : <http://www.uploadmalware.com/>
Lien123 : <http://www.virustotal.com/>
Lien124 : <http://analysis.seclab.tuwien.ac.at/>
Lien125 : <http://virusscan.jotti.org/>
Lien126 : <http://www.threatexpert.com/default.aspx>
Lien127 : <http://virscan.org/>
Lien128 : <http://www.filterbit.com/>
Lien129 : <http://windows.developpez.com/freewares/>
Lien130 : <http://malzilla.sourceforge.net/>
Lien131 : <http://www.01net.com/telecharger/windows/Multimedia/codecs/fiches/26950.html>
Lien132 : <http://www.delegation.internet.gouv.fr/mineurs/index.htm>
Lien133 : <http://www.actioninnocence.org/suisse/index.aspx>
Lien134 : <http://www.internet-mineurs.gouv.fr/>
Lien135 : <http://www.aig-filtra.org/web/resultats.aspx?nav=3>
Lien136 : <http://www.orange.fr/bin/frame.cgi?u=http%3A//assistance.orange.fr/755.php>
Lien137 : <https://internet-signalment.gouv.fr/>
Lien138 : <http://www.youtube.com/watch?v=FtuOpNy4Bj4>
Lien139 : <http://www.youtube.com/watch?v=O8sMchZD4D0>
Lien140 : <http://www.internetsanscrainte.fr/>
Lien141 : <http://www.nytimes.com/2008/10/21/technology/internet/21botnet.html>
Lien142 : <http://fr.securityvibes.com/mcafee-france-press-601.html>
Lien143 : http://blog.developpez.com/adiguba?title=anti_phishing_et_vie_privée
Lien144 : <http://bugbrother.blog.lemonde.fr/2010/05/24/eric-filliol-letat-doit-sappuyer-sur-les-hackers/>
Lien145 : <http://thpierre.developpez.com/articles/pieges-internet/>
Lien146 : <http://www.developpez.net/forums/d954828/club-professionnels-informatique/actualites/google-veut-reinventer-regles-divulgateion-failles-microsoft-vise/>
Lien147 : <http://www.developpez.net/forums/d962215/club-professionnels-informatique/actualites/chercheur-expose-failles-noyau-linux-ameliorer-securite-los/>
Lien148 : <http://www.developpez.net/forums/d938372/club-professionnels-informatique/actualites/fallait-publier-preuve-faisabilite-faille-xp-ingenieur-google/>
Lien149 : <http://www.developpez.net/forums/d948820/club-professionnels-informatique/actualites/groupe-chercheurs-anonymes-attaque-microsoft-apres-laffaire-ormandy/>
Lien150 : <http://conception.developpez.com/livres/>