

# Complément 4

## PL/SQL Web Toolkit et PSP

Il est possible d'utiliser plusieurs technologies pour construire une application Web autour d'Oracle. La plus simple est l'offre APEX (*Oracle Application EXpress*) qui ne nécessite aucun codage, mais elle est limitée en fonctionnalités. Une autre alternative consiste à programmer des *servlets* qui utilisent l'API JDBC pour dialoguer avec la base.

Ce complément se focalise sur deux autres techniques natives qui ne sont probablement pas beaucoup utilisées.

- *PL/SQL Web Toolkit* offre la capacité d'écrire des programmes PL/SQL qui génèrent des pages HTML lorsqu'ils sont invoqués via un navigateur client.
- *PL/SQL Server Pages* (PSP) intègre des instructions PL/SQL dans des pages HTML (ou XML) par l'utilisation de balises spécifiques. Cette technique se rapproche des pages ASP (*Active Server Pages*) de Microsoft et de l'approche JSP (*Java Server Pages*) de Sun. Pour pouvoir développer des pages PSP, et si vous n'utilisez pas la configuration du Web, vous devez disposer d'un serveur 8.1.6 ou plus récent, et des cartouches PL/SQL d'Oracle sous Internet Application Server, WebDB, ou Oracle Application Server.

Le choix entre l'une de ces technologies sera guidé par le type d'application Web développée. Si l'application ne doit pas inclure un grand nombre d'instructions PL/SQL et contient beaucoup de code HTML, il vaut mieux utiliser *PL/SQL Server Pages*. Dans le cas suivant, *PL/SQL Web Toolkit* pourra être préféré. Si vous préférez le monde de l'Open Source, PHP sera un allié idéal pour concevoir vos sites interactifs.

### Configuration minimale d'Apache

---

Oracle inclut un serveur Apache. Dans notre jeu d'exemple (Oracle9i) nous avons reconfiguré le numéro du port d'écoute à 77 (80 par défaut). Pour ce faire, modifiez les fichiers :

- `httpd.conf` sous `\oracle\ora92\Apache\Apache\conf\` (modifiez les entrées `Port 77` et `Listen 77`);
- `ports.ini` sous `\oracle\ora92\Apache\Apache\` (modifiez l'entrée `s_apachePort = 77`).

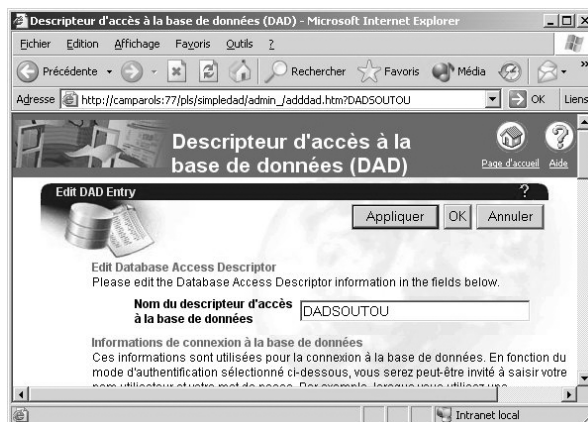
Arrêtez et redémarrez le service OracleOraHome92HTTPServer pour prendre en compte la nouvelle configuration.

Lancez l'explorateur en inscrivant le nom de votre machine suivi du numéro de port (exemple : `http://camparols:77/`), il vient :

**Figure C4-1** Menu général d'Apache sous Oracle



Choisissez `Mod_plsql Configuration Menu`, puis `Paramètres des DAD de la passerelle`, enfin `Ajout d'un descripteur par défaut` (configuration vide). Il vient l'écran suivant dans lequel vous devez nommer un DAD (*Data Access Descriptor*), ici `DADSOUTOU` :



**Figure C4-2** Création d'un DAD

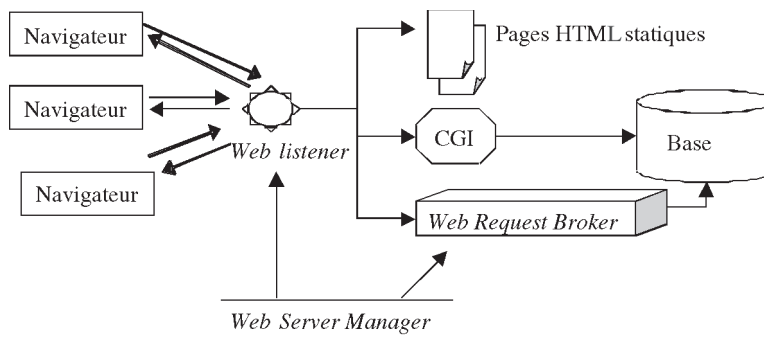
Si vous voulez vous identifier à chaque démarrage, ne renseignez que le schéma et la chaîne de connexion à votre base locale. Si vous voulez ne pas avoir à ressaisir un nom d'utilisateur et un

mot de passe, renseignez ces deux champs ainsi que la chaîne de connexion. Faites OK, Oracle doit signaler la création du nouveau DAD qui vous servira pour tester les exemples qui suivent.

## PL/SQL Web Toolkit

L'architecture mise en place par Oracle pour les applications Web met en jeu différents composants. Le navigateur envoie une requête sous la forme d'une URL qui invoque une procédure PL/SQL. Le processus *Web listener* analyse et route l'appel vers le composant adapté (page statique, CGI, procédure PL/SQL, etc.) afin de générer une page HTML.

**Figure C4-3** Architecture du Web Oracle



Le processus *Web Request Broker* est composé d'un ensemble de cartouches applicatives fournies en standard ou achetées à part (PL/SQL, Java, multimédia, etc.). Une cartouche contient du code qui permet la connexion à la base et l'exécution de procédures stockées.

Les paquetages HTP (*HyperText Procedures*) et HTF (*HyperText Functions*) regroupent un ensemble de sous-programmes permettant de générer des balises HTML. Il existe aussi d'autres fonctionnalités résidant dans le paquetage OWA\_UTIL et OWA (*Oracle Web Agent*) que nous n'étudierons pas ici.

### Détail d'une URL

Côté client, chaque URL doit être constituée de la manière suivante : `http://adresseMachine.domaine:numPort/accèsPLSQL/programme`. Le tableau ci-après détaille les composants d'une telle URL.

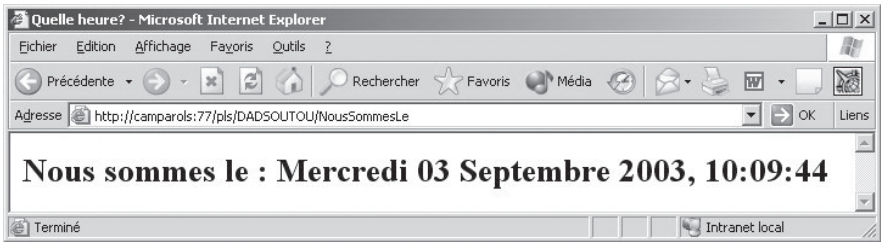
Tableau C4-1 Composition d'une URL

Composant	Commentaires
adresseMachine	Machine hébergeant les services Web.
domaine	Domaine internet.
numPort	Port UDP utilisé par le <i>Web listener</i> .
accèsPLSQL	Désigne le DAD, ( <i>Data Access Descriptor</i> ) qui réunit plusieurs paramètres (utilisateur, mot de passe, chaîne de connexion, pool de connexion, compatibilité SSO, etc.).
programme	Chemin d'accès à la procédure PL/SQL générant du HTML (de la forme [nomSchéma] . [nomPaquetage] . [nomProcédure]).

La figure suivante illustre le résultat généré par la procédure `NousSommesLe` détaillée plus loin. Les composant de l'URL sont : `http://camparols:77/pls/DADSOUTOU/NousSommesLe`.

- `camparols` est la machine qui héberge les services Web d'Oracle ;
- `77` est le port d'écoute UDP du *Web listener* ;
- `pls` précise la technologie utilisée (ici PL/SQL) ;
- `DADSOUTOU` désigne l'alias contenant les paramètres d'accès (utilisateur, mot de passe...) ;
- `NousSommesLe` est le nom de la procédure PL/SQL qui génère une page HTML.

Figure C4-4 Appel de la procédure cataloguée `NousSommesLe`



Paquetages HTP et HTF

Les tableaux suivants décrivent les principales procédures des paquetages HTP et HTF. Chaque procédure HTP possède une fonction HTF équivalente afin de pouvoir imbriquer deux balises.

Tableau C4-2 Balises générales

Procédures HTP	Balises HTML
HTP.htmlOpen, HTP.htmlClose	<HTML>, </HTML>
HTP.headOpen, HTP.headClose	<HEAD>, </HEAD>
HTP.bodyOpen, HTP.bodyClose	<BODY>, </BODY>
HTP.comment	<!-- et -->

Tableau C4-3 Balises d'en-tête

Procédures HTP	Balises HTML
HTP.base	<BASE>
HTP.title	<TITLE>
HTP.meta	<META>
HTP.script	<SCRIPT>
HTP.style	<STYLE>
HTP.isindex	<ISINDEX>

Tableau C4-4 Balises pour les applets Java

Procédures HTP	Balises HTML
HTP.appletopen, HTP.appletclose	<APPLET>, </APPLET>
HTP.param	<PARAM>

Tableau C4-5 Balises pour les listes

Procédures HTP	Balises HTML
HTP.olistOpen, HTP.olistClose	<OL>, </OL>
HTP.ulistOpen, HTP.ulistClose	<UL>, </UL>
HTP.dlistOpen, HTP.dlistClose	<DL>, </DL>
HTP.dlistTerm	<DT>
HTP.dlistDef	<DD>
HTP.dirlistOpen, HTP.dirlistClose	<DIR>, </DIR>
HTP.listHeader	<LH>
HTP.listingOpen, HTP.listingClose	<LISTING>, </LISTING>
HTP.menulistOpen, HTP.menulistClose	<MENU>, </MENU>
HTP.listItem	<LI>

Tableau C4-6 Balises pour les formulaires

Procédures HTP	Balises HTML
HTP.formOpen, HTP.formClose	<FORM>, </FORM>
HTP.formCheckbox	<INPUT TYPE="CHECKBOX">
HTP.formHidden	<INPUT TYPE="HIDDEN">
HTP.formImage	<INPUT TYPE="IMAGE">
HTP.formPassword	<INPUT TYPE="PASSWORD">
HTP.formRadio	<INPUT TYPE="RADIO">
HTP.formSelectOpen, HTP.formSelectClose	<SELECT>, </SELECT>
HTP.formSelectOption	<OPTION>
HTP.formText	<INPUT TYPE="TEXT">
HTP.formTextarea, HTP.formTextarea2	<TEXTAREA>
HTP.formTextareaOpen,	<TEXTAREA>, </TEXTAREA>
HTP.formTextareaOpen2,	
HTP.formTextareaClose	
HTP.formReset	<INPUT TYPE="RESET">
HTP.formSubmit	<INPUT TYPE="SUBMIT">

Tableau C4-7 Balises pour les tables

Procédures HTP	Balises HTML
HTP.tableOpen, HTP.tableClose	<TABLE>, </TABLE>
HTP.tableCaption	<CAPTION>
HTP.tableRowOpen, HTP.tableRowClose	<TR>, </TR>
HTP.tableHeader	<TH>
HTP.tableData	<TD>
HTP.format_cell	<TD>

Tableau C4-8 Balises images et ancrs

Procédures HTP	Balises HTML
HTP.line, HTP.hr	<HR>
HTP.img, HTP.img2	<IMG>
HTP.anchor, HTP.anchor2	<A>
HTP.mapOpen, HTP.mapClose	<MAP>, </MAP>

Tableau C4-9 Balises de paragraphes

Procédures HTP	Balises HTML
HTP.header	<H1> à <H6>
HTP.para, HTP.paragraph	<P>
HTP.print, HTP.prn	Génère du texte passé en paramètre.
HTP.prints, HTP.ps	Génère du texte en caractères spéciaux.
HTP.preOpen, HTP.preClose	<PRE>, </PRE>
HTP.blockquoteOpen	<BLOCKQUOTE> ,
HTP.blockquoteClose	</BLOCKQUOTE>
HTP.div	<DIV>
HTP.nl, HTP.br	 
HTP.nobr	<NOBR>
HTP.wbr	<WBR>
HTP.plaintext	<PLAINTEXT>
HTP.address	<ADDRESS>
HTP.mailto	<A> avec l'attribut MAILTO.
HTP.area	<AREA>
HTP.bgsound	<BGSOUND>

Tableau C4-10 Balises de caractères

Procédures HTP	Balises HTML
HTP.basefont	<BASEFONT>
HTP.big	<BIG>
HTP.bold	<B>
HTP.center	<CENTER>, </CENTER>
HTP.centerOpen, HTP.centerClose	<CENTER>, </CENTER>
HTP.cite	<CITE>
HTP.code	<CODE>
HTP.dfn	<DFN>
HTP.get_download_files_list	<EM>
HTP.fontOpen, HTP.fontClose	<FONT>, </FONT>
HTP.italic	<I>
HTP.keyboard, HTP.kbd	<KBD>, </KBD>
HTP.s	<S>
HTP.sample	<SAMP>
HTP.small	<SMALL>
HTP.strike	<STRIKE>
HTP.strong	<STRONG>
HTP.sub	<SUB>
HTP.sup	<SUP>
HTP.teletype	<TT>
HTP.underline	<U>
HTP.variable	<VAR>

Tableau C4-11 Balises pour le multifenêtrage

Procédures HTP	Balises HTML
HTP.frame	<FRAME>
HTP.framesetOpen, HTP.framesetClose	<FRAMESET>, </FRAMESET>
HTP.noframesOpen, HTP.noframesClose	<NOFRAMES>, </NOFRAMES>

Générer du HTML avec HTP

Le code suivant (NousSommesLe.sql) décrit la procédure PL/SQL NousSommesLe dont l'appel est illustré dans la section précédente. Chaque procédure HTP génère une balise HTML.

Tableau C4-12 Génération d'une page HTML

Code PL/SQL	Page HTML générée
CREATE PROCEDURE NousSommesLe AS BEGIN HTP.htmlOpen; HTP.headOpen; HTP.title('Quelle heure?'); HTP.headClose; HTP.bodyOpen; HTP.header(1, 'Nous sommes le : '    TO_CHAR(SYSDATE, 'Day DD Month YYYY, HH24:MM:SS') ); HTP.bodyClose; HTP.htmlClose; END;	<HTML> <HEAD> <TITLE>Quelle heure?</TITLE> </HEAD> <BODY> <H1>Nous sommes le : Mercredi 03 septembre 2003, 10:09:44</H1> </BODY> </HTML>

Générer du HTML avec HTF

Les fonctions HTF sont utiles pour imbriquer deux balises HTML. Le résultat d’une fonction HTF est passé en paramètre d’une procédure HTP comme le montre le code suivant :

Tableau C4-13 Imbrication de balises HTML

Code PL/SQL	Page HTML générée
HTP.htmlOpen;	<HTML>
HTP.bodyOpen;	<BODY>
HTP.header(1, HTF.italic('Coucou'));	<H1><I>Coucou</I></H1>
HTP.bodyClose;	</BODY>
HTP.htmlClose;	</HTML>

Pose d’hyperliens

Il est possible de poser des hyperliens (ancres) statiques ou dynamiques. Les ancres statiques sont composées d’URL figées dans le code. Les ancres dynamiques sont construites au cours de l’exécution de la procédure.

Hyperliens statiques

Le code suivant (Ancre.sql) présente l’exemple de deux ancres statiques. La première est studieuse, l’autre invite aux voyages :

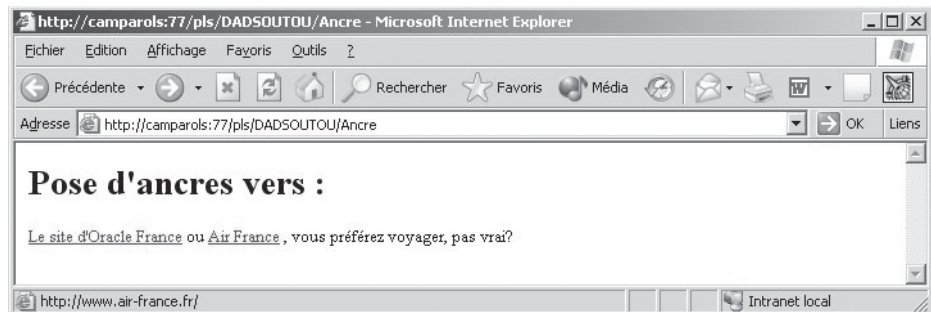
Tableau C4-14 Pose d’ancres statiques

Code PL/SQL	Page HTML générée
CREATE OR REPLACE PROCEDURE Ancre AS	
BEGIN	
HTP.htmlOpen;	<HTML>
HTP.bodyOpen;	<BODY>
HTP.header(1,'Pose d'ancres vers : ');	<H1>Pose d'ancres vers : </H1>
HTP.anchor('http://www.oracle.fr', 'Le site d'Oracle France');	<A HREF="http://www.oracle.fr">Le site d'Oracle France</A>
HTP.print(' ou ');	ou
HTP.anchor('http://www.air-france.fr', 'Air France');	<A HREF="http://www.air-france.fr">Air France</A>
HTP.print(' , vous préférez voyager, pas vrai? ');	, vous préférez voyager, pas vrai?
HTP.bodyClose;	</BODY>
HTP.htmlClose;	</HTML>
END;	

L’appel de cette procédure dans le navigateur produit l’affichage suivant.



Figure C4-5 Pose d'hyperliens statiques



### Hyperliens dynamiques

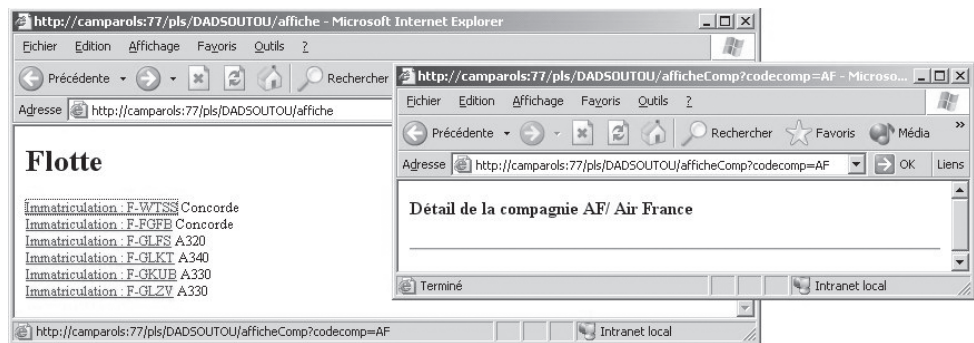
Le code suivant (affiche.sql) décrit la construction dynamique d'une liste d'ancres (ici en fonction de la table Avion). Chaque ancre permettra d'appeler la procédure afficheComp (afficheComp.sql) qui affichera les caractéristiques de la compagnie propriétaire.

Tableau C4-15 Hyperliens dynamiques

Code PL/SQL appelant	Code PL/SQL appelé
<pre> CREATE PROCEDURE affiche IS CURSOR curs IS SELECT immat, typeavion, comp FROM Avion; BEGIN HTP.htmlOpen; HTP.bodyOpen; HTP.header(1, 'Flotte'); FOR ligne IN curs LOOP   HTP.anchor('afficheComp?codecomp='        TO_CHAR(ligne.comp), 'Immatriculation     : '    ligne.immat);   HTP.print(' '    ligne.typeavion);   HTP.br; END LOOP; HTP.bodyClose; HTP.htmlClose; END; </pre>	<pre> CREATE PROCEDURE afficheComp (codecomp IN VARCHAR2 DEFAULT NULL) IS   v_nom Compagnie.nomComp%TYPE; BEGIN HTP.htmlOpen; HTP.bodyOpen; SELECT nomComp INTO v_nom FROM Compagnie WHERE comp = codecomp; HTP.header(3, 'Détail de la compagnie '    codecomp    '/'    v_nom); HTP.hr; HTP.bodyClose; HTP.htmlClose; END; </pre>

L'appel de la procédure affiche et la sélection du premier avion déclenchent l'exécution de la procédure afficheComp comme le montre l'enchaînement des écrans suivants.

Figure C4-6 Hyperliens dynamiques



Formulaires

Le code suivant (saisieComp.sql) illustre la création d'un formulaire qui saisit le numéro d'une compagnie avant d'afficher ses avions par l'intermédiaire de la procédure afficheAvions.

Tableau C4-16 Formulaire

Code PL/SQL	Page HTML générée
<pre>CREATE PROCEDURE <b>saisieComp</b>(comp IN VARCHAR2 DEFAULT NULL) IS BEGIN   HTP.htmlOpen; HTP.bodyOpen;   HTP.header(1,'Avions d'une compa- gnie','CENTER');   HTP.hr;   HTP.formOpen('saisieComp');   HTP.print('Numéro de la compagnie : ');   HTP.formText('comp');   HTP.hr;   HTP.formSubmit(cvalue=&gt;'Executer');   HTP.formReset(cvalue=&gt;'Annuler');   IF comp IS NOT NULL THEN     <b>afficheAvions</b>(comp);   ELSE     HTP.print('&lt;BR&gt;Saisir un code compagnie &lt;BR&gt;');   END IF;   HTP.formClose;   HTP.bodyClose;   HTP.htmlClose; END;</pre>	<p><b>Figure C4-7</b> Formulaire de saisie</p> <p>The image shows a screenshot of the 'Avions d'une compagnie' form in Internet Explorer. The form has a title 'Avions d'une compagnie', a text input field for 'Numéro de la compagnie' with the value 'AF', and two buttons: 'Executer' and 'Annuler'. Below the buttons is the text 'Saisir un code compagnie'. The address bar shows the URL 'http://camparols:77/pls/DADSOUTOU/saisieComp'.</p>

Tables

Les procédures HTP qui permettent de construire des tables sont intéressantes pour afficher des résultats de requêtes. Le code suivant (`afficheAvions.sql`) décrit la procédure appelée par le formulaire précédent. Un curseur permet de récupérer l'ensemble des avions appartenant à une compagnie dont le code est passé en paramètre (ici AF). La table produite est composée de trois colonnes.

Tableau C4-17 Création d'une table HTML

Code PL/SQL

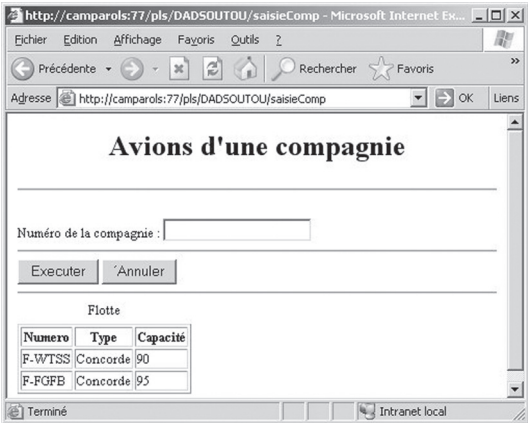
Page HTML générée

```
CREATE PROCEDURE
afficheAvions(cp IN VARCHAR2)
IS
CURSOR curs IS SELECT immat,
typeAvion, cap FROM Avion WHERE
comp = cp;
begin
HTP.htmlOpen; HTP.bodyOpen;
HTP.line;
HTP.tableOpen('BORDER');
HTP.tableCaption('Flotte', 'CE
NTER');
HTP.tableRowOpen;
    HTP.tableHeader('Numero');
    HTP.tableHeader('Type');

HTP.tableHeader('Capacité');
HTP.tableRowClose;
FOR ligne IN curs LOOP
    HTP.tableRowOpen;

HTP.tableData(ligne.immat);
HTP.tableData(ligne.typeAvion);
    HTP.tableData(ligne.cap);
    HTP.tableRowClose;
END LOOP;
HTP.tableClose;
HTP.bodyClose; HTP.htmlClose;
END;
```


Figure C4-8 Table HTML



Listes

Le code suivant (`affichelisteAvions.sql`) décrit l'affichage des avions d'une compagnie dont le code passe en paramètre au niveau de l'URL (testée dans l'écran suivant avec le code 'AF'). La procédure génère une liste HTML.

Tableau C4-18 Création d'une liste HTML

Code PL/SQL	Page HTML générée
<pre>CREATE PROCEDURE <b>affichelisteAvions</b> (cp IN VARCHAR2) IS   CURSOR curs IS SELECT immat, typeAvion, cap FROM Avion WHERE comp = cp;   v_immat Avion.immat%TYPE;   v_type Avion.typeAvion%TYPE;   v_cap Avion.cap%TYPE; BEGIN   HTP.htmlOpen; HTP.bodyOpen;   HTP.header(1,'Flotte de la compagnie '    cp);   HTP.line;   <b>HTP.ulistOpen;</b>   OPEN curs;   LOOP     FETCH curs INTO     v_immat,v_type,v_cap;     <b>HTP.listItem( v_immat    ' type : '    v_type    ' capacité : '    v_cap);</b>     EXIT WHEN curs%NOTFOUND;   END LOOP;   <b>HTP.ulistClose;</b>   HTP.bodyClose;   HTP.htmlClose; END;</pre>	<p><i>Figure C4-9</i> Liste générée</p> 

## PL/SQL Server Pages



La technologie PSP (*PL/SQL Server Pages*) permet d’intégrer un script PL/SQL dans une source HTML. Une fois écrit, le fichier d’extension psp doit être chargé dans la base comme une procédure stockée.

### Généralités

Les fichiers PSP peuvent être composés :

- de balises HTML ;
- d’une procédure PL/SQL générant toutes les balises HTML (sans inclure de procédures HTP) ;
- d’instructions PL/SQL intégrées au code HTML.

La structure générale d'un programme PSP est décrite dans la figure suivante. Les instructions PL/SQL se trouvent entre les balises <% et %>.

Figure C4-10 Structure générale d'un programme PSP

```
<%@ page language="PL/SQL" %>                                Exemple1.psp
<%@ plsql procedure="nomProcédure" %>
<%@ plsql parameter=" nomVariable" %>
...
<%! variableLocale typevariable; %>
...
<HTML>
<BODY>
  <% instructions PL/SQL %>
  < Balises HTML >
  <% ... %>
</BODY>
</HTML>
```

Balises

Le tableau suivant présente les principales balises propres à PSP :

Tableau C4-19 Imbrication de balises HTML

Directive	Signification
<%@ page language="PL/SQL" %>	Identification d'un fichier de type PSP.
<%@ plsql procedure="nomProcédure" %>	Nom de la procédure stockée.
<%@ plsql parameter="nomVariable" %>	Paramètre de la procédure stockée.
<!-- commentaire -->	Commentaire non visible dans la page HTML produite.
<!-- commentaire -->	Commentaire visible dans la page HTML produite.
<%@ include file = " nomFichier " %>	Inclure le contenu d'un autre fichier.
<%! nomvariable typevariable; %>	Déclarer (et initialiser) une variable.
<%! nomVariable typeVariable := valeurInitiale; %>	
<% instructions PL/SQL %>	Bloc d'instruction PL/SQL.
<%= expression %>	Extraction d'une expression PL/SQL (exemple : valeur d'une variable).

Le type d'un paramètre d'une procédure stockée est VARCHAR2. Pour spécifier un autre type, utiliser la directive TYPE="nomType". Il est aussi possible de préciser une valeur par défaut avec la directive DEFAULT="expression".

Le contenu d'un fichier PSP est généralement traduit en page HTML. Il est toutefois possible de générer un résultat d'un type différent à l'aide de l'instruction `<%@ page contentType = "typeMIME" %>`. Le type définit la nature du fichier produit (`text/html`, `text/xml`, `text/plain`, `image/jpeg`...).

Une variable PL/SQL n'est visible que dans un seul bloc (entre `<%` et `%>`).

## Chargement d'un programme PSP

Les sources PSP doivent être chargées (compilées) dans la base de données en tant que procédures stockées. Pour ce faire, Oracle fournit l'utilitaire `loadpsp`. Cet utilitaire s'utilise en ligne de commande de la manière suivante :

```
loadpsp -replace -user utilisateur/motdepasse@chaineconnexion
nomSource.psp
```

La procédure stockée générée (de nom *nomSource*) contient les instructions HTP et HTF qui permettent de construire la page HTML finale.

## Appel

Après avoir chargé la source PSP dans la base, il est possible de l'appeler directement dans l'URL d'un navigateur (comme pour les procédures écrites sous PL/SQL *Web Toolkit*).

```
http://adresseMachine.domaine:numPort/appliPLSQL/accès
```

- *appliPLSQL* : alias de l'application et de la cartouche. Les paramètres sont constitués par le DAD (*Data Access Descriptor*) qui identifie l'accès à la base ;
- *accès* : Chemin d'accès à la procédure PL/SQL avec d'éventuels paramètres (de la forme *nomSchéma.nomProcédure?paramètres=valeurs*).

Il est aussi possible de l'appeler dans une page HTML ou dans une autre procédure PSP (hyperlien ou via la méthode POST d'un formulaire).

## Interaction avec la base

L'utilisation d'instructions PL/SQL dans un script PSP permet de manipuler facilement des données de la base. Les balises HTML permettront de développer l'interface graphique (affichages et saisies).

### Formulaire

Les lignes suivantes (*codecompagnie.psp*) décrivent le code de la procédure PSP *codecompagnie* qui effectue, par l'intermédiaire d'un formulaire, la saisie du code d'une compagnie aérienne. L'affichage des avions de cette compagnie sera réalisé par la procédure

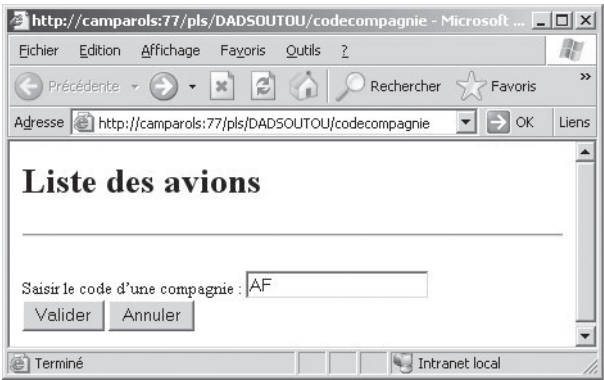
listeavions appelée par la méthode POST du formulaire (quand l'utilisateur actionnera le bouton Valider). La procédure codecompagnie ne contient pas d'instructions PL/SQL.

Tableau C4-20 Formulaire généré par une page PSP

Code PSP	Commentaires
<pre>&lt;%@ page language="PL/SQL" %&gt; &lt;%@ plsql procedure="codecompagnie" %&gt; &lt;HTML&gt;&lt;BODY&gt; &lt;H1&gt;Liste des avions &lt;/H1&gt; &lt;HR&gt; &lt;FORM METHOD="POST" ACTION = "listeavions"&gt; &lt;p&gt;Saisir le code d'une compagnie : &lt;INPUT TYPE = "text" NAME = "codecomp"&gt; &lt;BR&gt; &lt;INPUT TYPE = "submit" VALUE = "Valider"&gt; &lt;INPUT TYPE = "reset" VALUE = "Annuler"&gt; &lt;/FORM&gt;&lt;/BODY&gt;&lt;/HTML&gt;</pre>	<p>Nom de la procédure stockée.</p> <p>Définition de la procédure appelée.</p> <p>Définition du paramètre.</p>

Après compilation de ces deux procédures, l'appel de codecompagnie produit l'écran suivant. Supposons que l'utilisateur saisisse 'AF' dans la zone de texte.

Figure C4-11 Appel de la procédure codecompagnie



Affichage des résultats

Le code suivant (listeavions.psp) décrit la procédure listeavions qui réalise l'affichage des avions d'une compagnie dont le code passe en paramètre.

Tableau C4-21 Source PSP définissant un formulaire

Code PSP	Commentaires
<pre>&lt;%@ page language="PL/SQL" %&gt;  &lt;%@ plsql procedure="listeavions" %&gt; &lt;%@ plsql parameter="codecomp" %&gt; &lt;HTML&gt;&lt;BODY&gt; &lt;H1&gt; Avions de la compagnie &lt;%= codecomp %&gt; &lt;/H1&gt; &lt;BR&gt; &lt;% FOR ligne IN (SELECT immat, typeAvion, cap FROM Avion WHERE comp = &lt;b&gt;codecomp&lt;/b&gt;) LOOP %&gt; &lt;LI&gt; Immatriculation : &lt;%= ligne.immat %&gt; - &lt;%= ligne.typeA- vion %&gt; - &lt;%= ligne.cap %&gt; places &lt;BR&gt; &lt;/LI&gt; &lt;% END LOOP; %&gt;&lt;/BODY&gt;&lt;/HTML&gt;</pre>	<p>Nom de la procédure stockée.</p> <p>Paramètre de la procédure.</p> <p>Affichage du paramètre.</p> <p>Curseur parcourant la table Avion.</p>

L'activation du bouton Valider au niveau de la page précédente produit l'écran suivant :

Figure C4-12 Affichage des résultats (appel de listeavions)

