

VisualAge Pacbase



Dialog Web Revamping Guide du Développeur

Version 3.5



Note

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section "Notices" après la Table des Matières.

En application de votre contrat de licence, vous pouvez consulter ou télécharger la documentation de VisualAge Pacbase, régulièrement mise à jour, à partir du site Web du Support Technique :

<http://www.ibm.com/software/awdtools/vapacbase/support.html>

La section Catalogue dans la page d'accueil de la Documentation vous permet d'identifier la dernière

3ème édition (octobre 2006)

La présente édition s'applique à :

- VisualAge Pacbase Version 3.5

Vous pouvez nous adresser tout commentaire sur ce document (en indiquant sa référence) via le site Web de notre Support Technique à l'adresse suivante : <http://www.ibm.com/software/awdtools/vapacbase/support.html> ou en nous adressant un courrier à :

IBM Paris Laboratory

1, place Jean-Baptiste Clément

93881 Noisy-le-Grand, France.

IBM pourra disposer comme elle l'entendra des informations contenues dans vos commentaires, sans aucune obligation de sa part.

© Copyright International Business Machines Corporation 1983,2003. Tous droits réservés.

Table des Matières

Préambule.....	5	Description de l'écran.....	33
Chapitre 1 : Présentation générale	7	Head	33
Dialog Web Revamping.....	7	Body	34
Principes de transformation d'une application TUI en application web J2EE.....	7	Structure des pages JSP	34
Chapitre 2 : Fonctionnement général de l'application lors de son utilisation via le Web	9	Structure des pages d'aide.....	35
Principe de fonctionnement.....	9	Aide contextuelle.....	36
Middleware.....	10	Aide générale (sommaire).....	36
Chapitre 3 : Extraction et transfert de fichiers	11	La barre de navigation	36
Extraction des Ecrans sur serveur	11	Chapitre 7 : Personnalisation des templates générés	37
Transfert de fichier.....	11	Les styles.....	37
Chapitre 4 : Lancement du générateur.....	13	Modification automatique des pages HTML.....	38
Version standalone	13	Les fonctions Javascript.....	40
Version plug-in sur WSAD.....	13	Présentation des fonctions JavaScript	40
Chapitre 5 : Habillage d'une application ..	14	Fonctions utilisées au niveau Général	40
Interface graphique du générateur.....	14	Fonctions utilisées au niveau Dialogue.....	42
Paramètres obligatoires.....	14	Fonctions utilisées au niveau Ecran	42
Options de génération.....	17	Utilisation des fonctions JavaScript comme point d'entrée	43
Communication.....	17	Les templates.....	44
Paramètres de configuration visuelle.....	20	Présentation des templates	44
Options Paramètres divers.....	22	Page d'accueil	44
Libellés des touches de fonction et du code Action22	22	Page commune aux écrans	44
Icônes	24	Page écran	45
Page Help	45	Page Help	45
Interface Ligne de commandes	25	Description des templates.....	45
Lancement du générateur.....	25	Modification des templates	47
Modification des paramètres de génération.....	25	Chapitre 8 : Annexe	48
Résultat de la génération.....	28	Configuration d'un Servlet Container.....	48
Chapitre 6 : Présentation des pages générées	31	Description des paramètres d'initialisation de la servlet PacWebServlet	50
La page d'accueil.....	31	Migration des applications (Versions 2.5, 3.0) vers le mode J2EE (Version 3.5)	51
Structure de la page HTML	32	Appel de la servlet	51
		Gestion des cookies pour le user/password.....	52
		Ajout des nouveaux fichiers	52
		Ajout du RootContext dans les pages.....	52
		Fichier de configuration du middleware VAP	52
		Exemple d'utilisation des proxys conversation.....	53
		Utilisation avec 'CICS Transaction Gateway'	54

NOTICES

Ce document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM. Cela ne signifie pas qu'IBM ait l'intention de les annoncer dans tous les pays où la compagnie est présente.

Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM.

Il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même les installations et applications réalisées avec des produits, logiciels ou services non expressément référencés par IBM.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous donne aucun droit de licence sur ces brevets ou demandes de brevet. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

Intellectual Property and Licensing
International Business Machines Corporation
North Castle Drive, Armonk, New-York 10504-1785
USA

Les détenteurs de licences du présent produit souhaitant obtenir des informations sur celui-ci à des fins : (i) d'échange d'informations entre des programmes développés indépendamment et d'autres programmes (y compris celui-ci) et (ii) d'utilisation mutuelle des informations ainsi échangées doivent s'adresser à :

IBM Paris Laboratory
Département SMC
1, place Jean-Baptiste Clément
93881 Noisy-le-Grand
FRANCE

De telles informations peuvent être mises à la disposition du Client et seront soumises aux termes et conditions appropriés, y compris dans certains cas au paiement d'une redevance.

IBM peut modifier ce document, le produit qu'il décrit ou les deux.

MARQUES

IBM est une marque d'International Business Machines Corporation, Inc.

AIX, AS/400, CICS, CICS/MVS, CICS/VSE, COBOL/2, DB2, IMS, MQSeries, PACBASE, RACF, RS/6000, SQL/DS, TeamConnection et VisualAge sont des marques d'International Business Machines Corporation, Inc. dans certains pays.

Java et toutes les marques et logos incluant Java sont des marques de Sun Microsystems, Inc. dans certains pays.

Microsoft, Windows, Windows NT et le logo Windows sont des marques de Microsoft Corporation dans certains pays.

UNIX est une marque enregistrée aux Etats-Unis et/ou dans d'autres pays et utilisée avec l'autorisation exclusive de la société X/Open Company Limited.

D'autres sociétés peuvent être propriétaires des autres marques, noms de produits ou logos qui pourraient apparaître dans ce document.

Préambule

Conventions typographiques

La **police courier New** (gras, violet) est utilisée pour toute chaîne de caractères à saisir, affichée ou correspondant à du code généré, les noms de fichiers, répertoires, fenêtres, menus ou noms de composants.

Les titres des documentations ainsi que les titres des chapitres ou paragraphes dans les renvois sont en *italique*.

Les symboles suivants sont utilisés :



note, remarque,



renvoi à un autre emplacement dans la documentation ou à une autre documentation,



Attention, remarque importante.

Conventions terminologiques

Vous trouverez dans le guide du développeur, les termes **utilisateur** et **développeur**, chacun utilisé dans un contexte particulier :

- l'utilisateur est la personne qui utilise l'application Web finalisée. Il est équipé d'un micro-ordinateur sur lequel un navigateur est installé.
- le développeur est la personne chargée de mettre en place Dialog Web Revamping, de générer le paramétrage d'habillage de l'application et d'installer l'application finalisée sur les postes des utilisateurs.

Le terme **écrans** désigne les écrans de l'application générée avec le module Dialogue.

Le terme **pages** désigne les pages HTML ou JSP générées par Dialog Web Revamping.

Le terme **page écran** désigne une page HTML contenant la description d'un écran de l'application.

Chapitre 1 : Présentation générale

Dialog Web Revamping

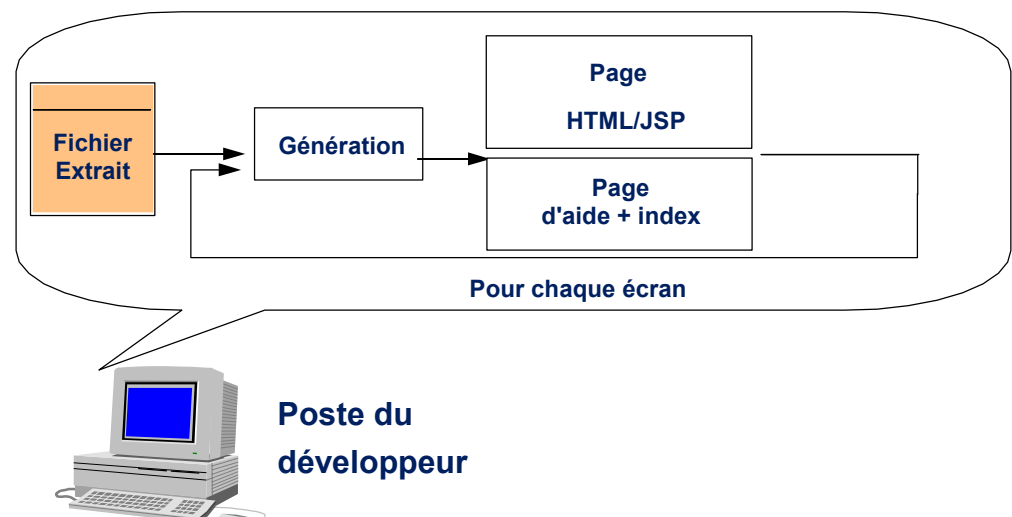
Dialog Web Revamping permet de générer automatiquement et à un moindre coût des applications Web J2EE à partir des applications VisualAge Pacbase existantes. Ainsi il offre à la fois une ouverture sur Internet/Intranet et un plus grand confort d'utilisation (utilisation de la souris, de listes déroulantes, d'une documentation interactive...).

Autre avantage, les pages HTML ou JSP générées peuvent être modifiées tant au niveau de l'interface utilisateur (ajout de boutons poussoirs, etc.) qu'au niveau des traitements (ajout de contrôles de champs par exemple) en utilisant le langage Java (Applets, Javascript ...) ou encore les éditeurs de pages HTML du marché.

Principes de transformation d'une application TUI en application web J2EE

Les principales étapes consistent à :

- extraire du Référentiel VisualAge Pacbase les fichiers qui contiennent la description des écrans et à les transférer sur le poste du développeur,
- positionner un certain nombre de paramètres de communication,
- définir les options d'habillage de l'application Dialogue et lancer la génération des templates HTML ou JSP, des pages d'index et d'aide.



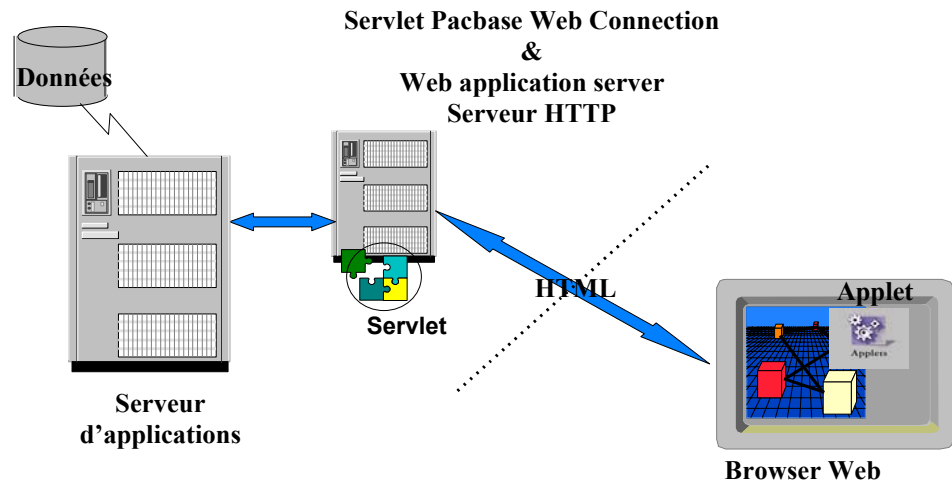
Chapitre 2 : Fonctionnement général de l'application lors de son utilisation via le Web

Principe de fonctionnement

Le fonctionnement général de l'application lors de son utilisation à travers un navigateur Web est le suivant :

L'utilisateur demande l'accès à l'application par la sélection d'un lien dans une page d'accueil HTML. Ce lien active la servlet **PacWebServlet** qui :

- établit un contexte pour l'utilisateur afin d'assurer la gestion de la conversation et initialise une session s'il s'agit d'une première requête ou rétablit un contexte existant dans le cas contraire.
- transmet la requête à l'application transactionnelle par l'intermédiaire des couches de communication et du moniteur transactionnel.
- L'application envoie un message logique correspondant à l'écran demandé. Ce message se compose d'informations générales relatives au contexte de la conversation ou à l'écran (identifiant de la conversation, numéro de session, code de l'écran, bibliothèque utilisée, code base...). La suite du message logique est constituée de l'ensemble des informations présentes sur l'écran (libellés, zones variables, ...) suivi de codes relatifs à ces informations (positionnement dans l'écran, ligne, colonne, longueur, nature, intensité, couleur, code rubrique, ...).
- Le message logique est ensuite décodé par la couche « proxy conversation ».
- Dans le cas HTML, la servlet va ensuite fusionner les données avec les templates HTML générés.
- Dans le cas JSP, la servlet transmet vers la JSP générée en lui fournissant les données.
- Au cas où une page n'aurait pas été générée, la servlet va générer « à la volée » cette page au format HTML.
- Le flot HTML résultant de cette fusion est alors transmis à l'utilisateur par l'intermédiaire du serveur HTTP.



Middleware

La communication entre la servlet et le mainframe se fait grâce au Middleware VisualAge Pacbase : le choix du type de Middleware et son paramétrage se fait lors de la génération.

Vous trouverez toutes les informations sur l'utilisation du Middleware VisualAge Pacbase dans le Guide du middleware.

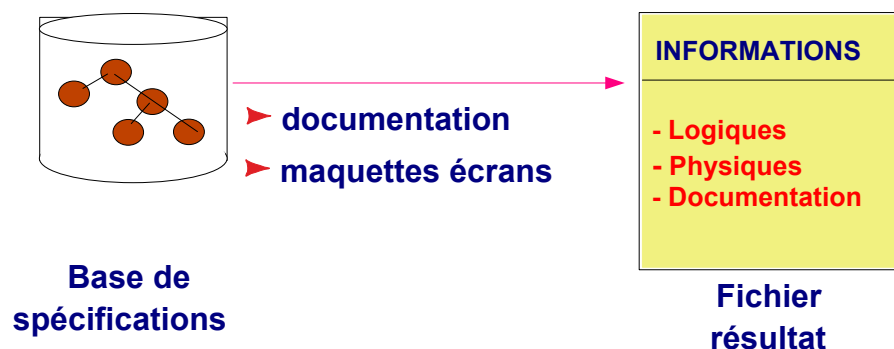
Chapitre 3 : Extraction et transfert de fichiers

Le générateur du Référentiel VA Pac génère les écrans de l'application TUI et réalise des fichiers contenant la description de ces écrans. Ces fichiers doivent être extraits, ils serviront alors de support à la création de templates HTML ou de JSP destinés à la présentation du contenu des écrans dans un navigateur Web.

Extraction des Ecrans sur serveur

Vous devez tout d'abord extraire depuis le Référentiel VisualAge Pacbase les informations nécessaires à la production du paramétrage local.

- Au niveau du Dialogue : titre de l'application, options, documentation.
- Au niveau de l'Ecran : Définition, Description et documentation.
- Au niveau des Rubriques : nom, code, longueur, valeurs autorisées, documentation.
- Aide du Dialogue.



Le fichier qui contient les descriptions des écrans développés avec le module Dialogue est produit par la procédure Va Pac **GPRT (GEO option C4)** sur le serveur. En sortie de la procédure, le développeur obtient le fichier **PAC7GT** (ou **GT** pour le système GCOS8) dont chaque enregistrement contient au maximum 180 caractères.

☞ Pour plus d'informations sur la procédure **GPRT**, consultez le manuel de référence *Guide de l'interface utilisateur Mode caractères*.

☞ vérifiez que les noms des écrans ne contiennent pas de caractères spéciaux interdits (*, ?, etc...), cela provoquerait un message d'erreur lors de la génération des pages web.

Transfert de fichier

Cette étape consiste à transférer le résultat de l'extraction sur le poste du développeur.

La descente du fichier **PAC7GT** ou **GT** sur micro-ordinateur est à la charge du développeur. L' utilitaire de transfert doit être paramétré de telle sorte que les caractères spéciaux et accentués soient bien conservés. Nous désignerons par **NOMFICH.EXT** le fichier local résultant du transfert.



Les informations concernant la modification des programmes Dialogue pour l'implémentation de Dialog Web Revamping sont documentées dans le manuel de référence *Module Dialogue*.

Chapitre 4 : Lancement du générateur

Il existe deux façons de lancer le générateur en fonction de votre environnement de travail selon que vous travaillez avec une version standalone (autonome) du module Dialog Web Revamping ou une version plug-in qui permet de l'intégrer à WebSphere Studio (WSAD).

Version standalone

Pour une version standalone, vous devez lancer l'exécution de **vapWebGen.bat**, soit directement depuis le gestionnaire de fichiers, soit par l'intermédiaire d'un raccourci.

Version plug-in à partir de WSAD

Pour lancer le générateur à partir de WebSphere Studio, vous devez suivre la procédure suivante :

- Lancez WebSphere Studio,
 - Choisissez une perspective Web,
 - Créez un projet Web en sélectionnant "Fichier" → "Nouveau" → "Autre".
- ou faites un clic droit sur un projet Web déjà créé et choisissez « Nouveau » → « Autre ».
- Enfin sélectionnez « VisualAge Pacbase eBusiness » → « Générateur d'habillage web pour dialogue » et cliquez sur "Next". La première fenêtre du générateur s'affiche à l'écran.

Chapitre 5 : Habillage d'une application

Cette étape consiste simplement à définir les options et paramètres permettant l'habillage de l'application VisualAge Pacbase. Ensuite le générateur transformera automatiquement les écrans de l'application en templates HTML ou JSP et les paramètres d'habillage définis seront sauvegardés sous forme d'un fichier. Le choix des paramètres d'habillage et le lancement du générateur peuvent se faire soit à partir d'une interface graphique, c'est-à-dire via la fenêtre du générateur, soit à partir d'un outil à interface textuelle.

Interface graphique du générateur

La génération des fichiers de paramétrage est effectuée automatiquement par le module Dialog Web Revamping qui traite des données extraites de la base et demande le minimum d'intervention de la part du développeur.



Les fenêtres qui vous sont présentées ci-après sont celles d'un générateur de version standalone. Les noms et la disposition des zones peuvent être légèrement différents dans les fenêtres du générateur en version plug-in, cependant, leurs fonctions sont identiques.

Paramètres obligatoires

Générateur d'habillage Web pour Dialogue

Paramètres obligatoires

Ajout de fichier(s)

Désallocation de fichier(s)

Désallocation totale

Repertoire de génération

Nom de l'application

Nom court de l'application: APPN

Format: HTML

Chemin d'accès contextuel

Génération seule du .INI Appliquer la sécurité

Repertoire de génération obli... <<Précédent Suivant>> Terminer Annuler

- **Fichiers d'extraction :**
 Vous devez sélectionner les fichiers d'extraction à partir desquels vous allez générer vos templates HTML ou vos JSP.
 - Le bouton **Ajout de fichier(s)** permet d'ajouter un ou plusieurs fichiers dans la liste.
 - Le bouton **Désallocation de fichier(s)** permet de désélectionner un ou plusieurs fichiers.
 - Le bouton **Désallocation totale** permet de désélectionner tous les fichiers.

- **Répertoire de génération :** Indiquez le répertoire dans lequel sera générée l'arborescence des fichiers. Par exemple, si le répertoire de génération est **c:\pacweb2** et le nom court de l'application, **WE** par exemple, on obtiendra :
 - **c:\pacweb2\WE\app** : répertoire des templates HTML de l'application.
 - **c:\pacweb2\WE\help** : répertoire des pages HTML de l'aide de l'application.
 - **c:\pacweb2\WE\icon** : répertoire contenant les icônes apparaissant dans les templates HTML.
 - **c:\pacweb2\WE\js** : répertoire des fichiers contenant les fonctions JavaScripts.

Le bouton **Parcourir** permet de sélectionner le répertoire de génération.

- **Nom de l'application :** Nom des fichiers
 - **<nom de l'application>.ini** dans lequel seront sauvegardés les paramètres de génération comme par exemple les paramètres Middleware utilisés par l'application.
 - **<nom de l'application>.war** qui correspond à l'application web au format Web Archive J2EE permettant le déploiement d'une application J2EE.

- **Nom court de l'application :** chaîne de caractères permettant de différencier les différentes applications générées et utilisée pour créer les différents répertoires propres à chaque application.

- **HTML/JSP :** ComboBox permettant de choisir le type de page générées :
 - soit **HTML** : dans ce cas, le code effectuant la fusion dans la page fait partie de la servlet
 - soit **JSP** : dans ce cas, le code effectuant la fusion se trouve dans la page JSP sous forme de scriptlet.

- **Chemin d'accès contextuel :** correspond à la racine de l'application déployée sur un serveur web (utile uniquement pour la génération HTML).

- **Génération seule du .INI :** seul le fichier contenant l'ensemble des paramètres est généré (version standalone uniquement).

- **Appliquer la sécurité** : si vous cochez cette case, la page d'accueil de l'application générée par Dialogue Web Revamping sera sécurisée. L'utilisateur autorisé devra entrer son code utilisateur et son mot de passe.

La sécurisation de l'application Web générée peut être effectuée via WebSphere Studio (définition des groupes utilisateurs et gestion des profils).

☞ Pour mettre en place la sécurité de l'application Web via WebSphere Studio , les étapes suivantes doivent être effectuées :

- ❑ importer dans WebSphere Studio le fichier généré par Dialogue Web Revamping, **<NomApplication>.war**,
- ❑ créer un serveur pour l'Application générée et le configurer,
- ❑ créer un ou plusieurs rôles de sécurité, ainsi que des contraintes de sécurité et identifier les utilisateurs autorisés pour ces rôles. Il faut pour cela compléter le fichier de description de déploiement de l'application Web ainsi que le fichier de description de déploiement EAR.

☞ Pour plus de détails, consultez sur l'Internet les documents IBM, notamment le manuel sur la sécurité WebSphere '*IBM WebSphere Security, WebSphere Handbook Series*' ainsi que l'aide en ligne de WebSphere Studio sur la mise en œuvre de la sécurité.

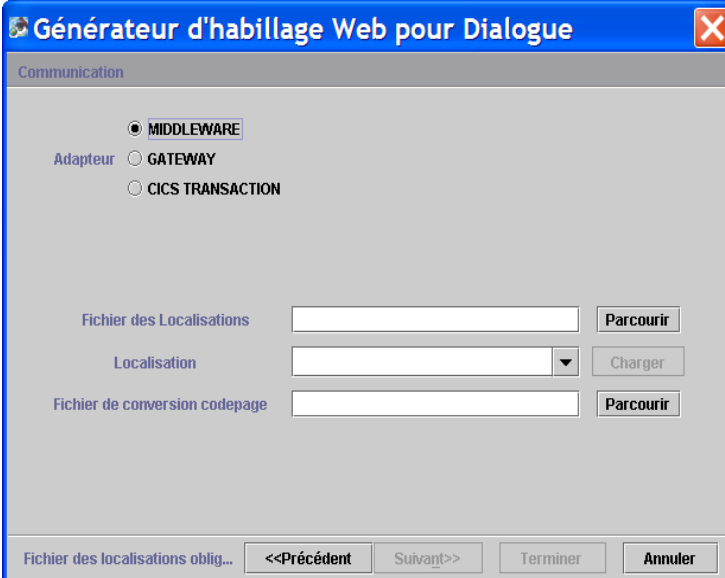
Options de génération

Les options proposées vous permettent de personnaliser vos pages. Vous avez la possibilité de sélectionner :

- la communication via un middleware, une gateway ou une transaction CICS, ainsi que tous les paramètres spécifiques à chaque communication
- les paramètres de configuration visuelle (messages d'erreur, fond d'écran, champs, libellés...).
- des paramètres divers (suffixe des fichiers contenant les pages HTML, nom de la première page HTML, titre de l'application, contrôles locaux, génération de commentaire).
- les libellés associés aux codes actions et aux touches fonction.
- le nom des icônes des barres de menu.

Toutes ces options de génération sont sauvegardées dans un fichier nommé **<nom de l'application>.ini**. Elles pourront être réutilisées lors d'une re-génération. Les informations indispensables au fonctionnement de la servlet **PacWebServlet** sont générées dans le fichier **web.xml**.

Communication



The screenshot shows a dialog box titled "Générateur d'habillage Web pour Dialogue" with a "Communication" tab. It contains the following elements:

- Radio buttons for communication type: **MIDDLEWARE** (selected), **GATEWAY**, and **CICS TRANSACTION**.
- Text label "Adapteur" followed by the selected radio button.
- Text input field for "Fichier des Localisations" with a "Parcourir" button.
- Text input field for "Localisation" with a dropdown arrow and a "Charger" button.
- Text input field for "Fichier de conversion codepage" with a "Parcourir" button.
- Footer area with "Fichier des localisations oblig...", navigation buttons "<<Précédent" and "Suivant>>", and "Terminer" and "Annuler" buttons.

Générateur d'habillage Web pour Dialogue

Communication

MIDDLEWARE
 Adapteur GATEWAY
 CICS TRANSACTION

Adresse IP

N° de Port

Localisation

N° de port obligatoire

Générateur d'habillage Web pour Dialogue

Communication

MIDDLEWARE
 Adapteur GATEWAY
 CICS TRANSACTION

Adresse IP

N° de Port

Nom du serveur

Code transaction

Adresse IP obligatoire

- **Adapteur** : permet d'indiquer le type d'adaptateur utilisé pour communiquer avec l'application host :
 - **Middleware** : Middleware direct (les dll du middleware doivent être accessibles par la variable d'environnement **PATH**)
 - **Gateway** : utilisation d'une gateway
 - ☞ Voir aussi le Manuel *Applications eBusiness – Interface de programmation des Proxies*, chapitre *Attribut*, sous-chapitre *Gestion des communications*, section *Choix de l'adaptateur de communication*.
 - **CICS Transaction** :



Voir aussi plus loin le chapitre *Annexe*, sous-chapitre *Utilisation avec 'CICS Transaction Gateway'*.

Lors de l'utilisation du mode **Middleware**

- **Fichier des Localisations** : le fichier contenant les localisations.
- **Localisation** : la localisation utilisée.
- **Fichier de conversion codepage** : le fichier de transcodification utilisé par le Middleware.

Lors de l'utilisation du mode **Gateway**

- **Adresse IP** : L'adresse TCPIP de la machine sur laquelle s'exécute la gateway.
- **N° de Port** : le N° de port sur laquelle la gateway est en écoute.
- **Localisation** : la localisation utilisée.

Lors de l'utilidation **CICS Transaction Gateway** :

- **Adresse IP** : l'adresse TCPIP de la machine sur laquelle s'exécute la gateway(CTG).
- **Port** : le N° de port sur laquelle la gateway (CTG) est en écoute.
- **Nom du serveur** : le nom du serveur pour CTG.
- **Code transaction** : le nom de la transaction.

Paramètres de configuration visuelle

Paramètres de configuration visuelle

Message d'erreur Fenêtre Javascript
 Champ
 Les deux

Fond d'écran est une couleur est un fichier

Champs Utiliser les paramètres du dialogue Utiliser une feuille de style

Libellés Utiliser les paramètres du dialogue Utiliser une feuille de style

Libellés pour les listes et les radio-boutons Court Long

Limite de génération pour les radio-boutons

Lien vers d'autres écrans

<<Précédent Suivant>> Terminer Annuler

- **Message d'erreur** : Cette option permet de paramétrer l'affichage des erreurs.
 - **Fenêtre Javascript** : le message d'erreur s'affiche dans une fenêtre directement dans le browser.
 - **Champ** : Le message s'affiche en fin de page comme indiqué dans la maquette.
 - **Les deux** : possibilité d'avoir les deux types d'affichage simultanément.
- **Fond d'écran** : Sélectionnez la couleur de fond de la page.
 - **est une couleur** : choisissez une couleur dans la liste déroulante.
 - **est un fichier** : Choisissez une bitmap qui se trouve dans le répertoire `<nom court de l'application>\icon, <nom court de l'application>` étant le nom entré dans la zone **Nom court de l'Application** de la page **Paramètres obligatoires**.
- **Champs** : partie variable de l'écran
 - **Utiliser les paramètres du Dialogue** : utilisation de l'affichage tel qu'il est décrit dans VisualAge Pacbase.
 - **Utiliser une feuille de style** : utilisation du style `.libvar` défini dans le fichier `style.css`. Ce fichier se trouve dans le répertoire `<nom court de l'application>\js`.
- **Libellés** : partie fixe de l'écran

- **Utiliser les paramètres du Dialogue**: utilisation de l'affichage tel qu'il est décrit dans VisualAge Pacbase.
- **Utiliser une feuille de style**: utilisation du style `.libfix` défini dans le fichier `style.css`.
- **Libellés pour les listes et les radio-boutons**:
 - **Court**: Utilise le libellé court comme valeur affichée par une liste déroulante ou un bouton radio.
 - **Long**: Utilise le libellé long comme valeur affichée par une liste déroulante ou un bouton radio.
- **Limite de génération pour les radio-boutons**: Cette valeur est la limite à partir de laquelle une liste de choix est maquetée sous forme de liste déroulante plutôt que de bouton radio.
- **Lien vers d'autres écrans**: Permet de spécifier le type de maquetage des liens entre écrans (liste déroulante, bouton ou boîte éditée).

Options Paramètres divers



- **Langue** : cette option permet de générer la page d'accueil et les messages standard des contrôles locaux en français ou en anglais. La page d'accueil contient un titre et les boutons déclenchant les applications générées.
- **Noms** :
 - **Suffixe** donne le suffixe utilisé pour les noms de fichiers HTML (.htm ou .html)
 - **Nom de la première page HTML** : nom de la page d'accueil
 - **Titre de l'application**: titre qui apparaîtra dans le bouton d'activation de l'application dans la page d'accueil.
- **Contrôle local** :
 - **Contrôle numérique**
 - **Contrôle de date**
 - **Contrôle de présence**

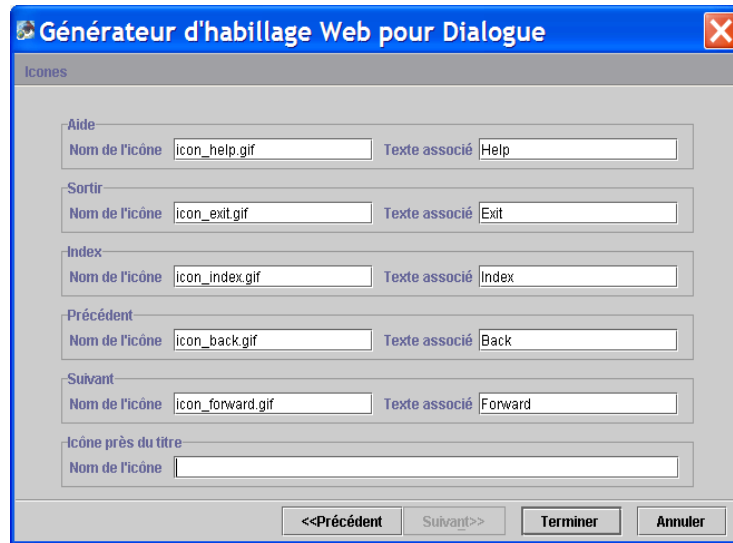
Ces fonctions sont décrites dans le fichier `public.js` du répertoire `js`.

- **Générer avec des commentaires** : Ajoute des commentaires dans les pages générées permettant à un post processeur de modifier ces pages.

Libellés des touches de fonction et du code Action

- **Commandes d'édition :**
Ces 4 zones de saisie permettent de générer une liste de choix pour les codes actions : créer, modifier, annuler, créer ou modifier.
- **Touches de fonction Aide standard :** lorsque cette option est cochée, les touches fonction utilisées pour activer l'aide standard et l'aide contextuelle standard, sont maquettées sous forme de boutons.
Les aides standards sont celles gérées directement par le serveur applicatif par opposition à l'aide des pages HTML générées à partir du fichier extrait du Référentiel VisualAge Pacbase.
- **Position des touches de fonction :** Indiquez l'emplacement des touches fonctions sur l'écran : en haut, en bas de l'écran ou dans le pied de page.
- **Touches de fonction :** vous pouvez spécifier les libellés qui apparaîtront dans les boutons associés aux touches fonctions en haut ou en bas de l'écran. Vous pouvez insérer jusqu'à 24 touches.
- **Libellés des touches de fonction :**
 - **Entree :** Libellé du bouton associé à la touche **Entree**
 - **Effacement :** Libellé du bouton associé à la touche **suppr**
 - **Page suivante :** Libellé du bouton associé à la touche **Pg dn**
 - **Page précédente :** Libellé du bouton associé à la touche **Pg Up**

Icônes



Ces options permettent de :

- saisir le nom des fichiers contenant les icônes des différents boutons standards.
- de saisir leur libellé par défaut, c'est-à-dire le texte de l'étiquette associée à l'icône.

Ces icônes sont contenues dans le répertoire **icon** de l'application.

- **Aide** : permet à l'utilisateur de consulter la page d'aide, l'index de la page d'aide et l'aide concernant la page d'aide.

☞ Pour plus de détails, consultez le chapitre *Présentation des pages générées*, section *Structure des pages d'aide*.

- **Index** : permet à l'utilisateur de réafficher l'index.
- **Sortir** : permet à l'utilisateur de quitter l'application ou l'aide.
- **Précédent** : permet à l'utilisateur de naviguer dans les pages d'aide qu'il a déjà consultées. En cliquant sur le bouton, la page précédente s'affichera.
- **Suivant** : permet à l'utilisateur de naviguer dans les pages d'aide qu'il a déjà consultées. En cliquant sur le bouton, la page suivante s'affichera.
- **Icône près du titre** : nom du fichier contenant l'icône à placer dans l'en-tête de chaque page.

Interface Ligne de commandes

Lancement du générateur

Pour lancer le générateur par une ligne de commande, vous devez taper la commande suivante :

```
VapWebGenBatch.bat <rep_génération> <fichier_paramètres>  
<Geofile>
```

rep_génération désigne le répertoire sous lequel l'ensemble des pages HTML est généré.

fichier_paramètres désigne le fichier contenant les différents paramètres de génération. Par défaut le fichier **PacWebgen.ini** est utilisé.

Geofile désigne le fichier obtenu à l'issue de l'extraction du ou des fichiers contenant la description des écrans Pacbase (**GEO C4**).

Vous pouvez lancer la génération de plusieurs fichiers **Geofile** à la fois (7 maximum).

Modification des paramètres de génération

Le fichier **Pacwebgen.ini** est utilisé par défaut pour générer une application HTML. Vous devez renseigner obligatoirement certains champs de ce fichier avant de lancer la génération.

Les paramètres de génération :

Valeur de défaut	Autres valeurs possibles	Description
[Contrôles]		
Number control= NO	YES	Contrôle de numéricité
Date control= NO	YES	Contrôle date
Presence control= NO	YES	Contrôle de presence
[Format]		
Label format= STYLE	DIALOGUE	Format des libellés DIALOGUE : utilisation de l'affichage tel qu'il est décrit dans VisualAge Pacbase STYLE : utilisation d'une feuille de style
Field format= DIALOGUE	STYLE	Format des champs DIALOGUE : utilisation de l'affichage tel qu'il est décrit

Valeur de défaut	Autres valeurs possibles	Description
		dans VisualAge Pacbase STYLE : utilisation d'une feuille de style
Radio to list= 2		Cette valeur est la limite à partir de laquelle une liste de choix est maquetée sous forme de liste déroulante plutôt que de bouton radio
Comment= NOCOMMENT	COMMENT	Génération avec ou sans ajout de commentaires
SLabel= short	long	Type de libellé : court ou long comme valeur affichée par une liste déroulante ou un bouton radio.
[application]		
First page name= company		Nom de la page d'accueil
Suffix= .htm	.html	Suffixes utilisés pour nommer les pages HTML.
Language= ENGLISH	FRENCH	Langue de génération
ContextPath= /pacweb		Racine de l'application web pour génération HTML
Create_War_File= YES	NO	Permet d'inhiber la création du fichier .war
TypeGeneration= HTML	JSP	Type de génération
Short application title= APPN		Chaîne de caractères utilisée pour créer les différents répertoires propres à chaque application.
Application title= title		Libellé du bouton de lancement de l'application sur la page d'accueil
Instal_Dir		Répertoire contenant les templates générés
Error message= BOTH	WINDOW ou FIELD	Type d'affichage des messages d'erreur : affiché dans une fenêtre ou en fin de page ou les deux simultanément.
ShortInstal_Dir		Nom du répertoire de génération
[middleware]		
EPIConversationFactory.trans actionId		Transaction activée en mode CTG
EPIConversationFactory.geoF ile		Noms des fichiers d'extraction
EPIConversationFactory.addr ess		Adresse de la machine utilisée par la gateway CTG
EPIConversationFactory.serv er		Nom du serveur pour CTG

Valeur de défaut	Autres valeurs possibles	Description
EPIConversationFactory.port		N° de port de la gateway
ConversationFactory	<code>com.ibm.vap.webconnection.ctg.EPIConversationFactory com.ibm.vap.webconnection.pwcm.PacWebConversationFactory </code>	
PacWebConversationFactory.serverAdapter.codePageFile		Le fichier de transcodification
PacWebConversationFactory.serverAdapter= <code>com.ibm.vap.middleware.MiddlewareAdapter</code>	<code>com.ibm.vap.gateway.GatewayAdapter</code>	Le type d'adaptateur choisi : middleware direct ou Gateway
PacWebConversationFactory.serverAdapter.location		La localisation utilisée dans le fichier des localisations
PacWebConversationFactory.serverAdapter.locationsFile		Le fichier des localisations
PacWebConversationFactory.serverAdapter.host		Adresse de la machine sur laquelle s'exécute la Gateway
PacWebConversationFactory.serverAdapter.port		N° de port de la gateway
[CMVT]		
C CMVT string= <code>Create</code>		Libellé du code mouvement de création
M CMVT string= <code>Modify</code>		Libellé du code mouvement de modification
A CMVT string= <code>Cancel</code>		Libellé du code mouvement d'annulation
X CMVT string= <code>Create or Modify</code>		Libellé du code mouvement de création/modification.
[couleurs]		
Background= <code>White</code>		Couleur du fond de page
Link color= <code>Blue</code>		Couleur des liens hypertextes
A Link color= <code>Red</code>		Couleur des liens activés
Visited link color= <code>Red</code>		Couleur des liens déjà visités
[Ecrans]		
Common PageUp Key=		Libellé du bouton associé à la touche Pg AR(PageUp)
Common Enter Key=		Libellé du bouton associé à la touche Entrée (Enter)
Standard help PFKEY= <code>NO</code>	<code>YES</code>	Maquettage pour les touches d'aide standard
Function buttons= <code>TOP</code>	<code>BOTTOM</code> ou <code>FOOTER</code>	Emplacement des touches fonctions sur l'écran : en

Ce fichier est utilisé uniquement par le générateur afin de retrouver les paramètres de génération de l'application lors d'une re-génération

- une page d'accueil, nécessaire à l'initialisation du dialogue dont le nom est donné par le champ **Nom de la première page HTML**:
<Nom de la première page HTML>.htm
 - une page signalant la deconnexion à l'application : **EOTPAGE.htm**
 - une page affichant les erreurs internes de la servlet :
servletoexception.jsp
 - un fichier **<NomApplication>.war** au format Web Archive J2EE permettant le déploiement de l'application .
- dans le sous-répertoire **app** du répertoire **rep_génération / Nom_court_application**:
 - une page utilitaire, contenant des fonctions utilitaires Javascript ou les descriptions en **frames** (=zones) des écrans HTML, dans un fichier dont le nom est donné par le champ **Nom court de l'application** :
<Nom court de l'application>app.htm
 - une page par écran, stockée dans un fichier ayant pour nom :
<CodeEcran> (sur 6 caractères) et pour suffixe **htm** , **html** ou **jsp** (option de génération HTML/JSP) :
<CodeEcran>.<suffix>
 - les fichiers suivants:
editpl.htm, error.htm, fieldtpl.htm, hiddentpl.htm, labeltpl.htm, menubar.htm, notfoundtpl.htm, passwdtpl.htm
 - dans le sous-répertoire **help** du répertoire **rep_génération / Nom_court_application**:
 - une page d'aide de nom **<Nom court de l'application>help.htm**
 - une page d'aide par écran dans un fichier ayant pour nom **<Nom court de l'application>_help_<CodeEcran>** (sur 6 caractères) et pour suffixe **htm**.
 - une page d'index par écran dans un fichier ayant pour nom **<Nom court de l'application>_index_<CodeEcran>** (sur 6 caractères) et pour suffixe **htm**.
 - **footer.htm** : les icônes pour les écrans d'aide
 - **helphelp** : texte de l'aide sur l'aide
 - **<Nom court de l'application>_Index** : Index général
 - dans le sous-répertoire **js** du répertoire **rep_génération / Nom_court_application**
 - un fichier **public.js** pour les fonctions publiques utilisées au niveau Général
 - un fichier **private.js** pour les fonctions privées utilisées au niveau Général

- `<Nom court de l'application>private.js` : les fonctions privées utilisées au niveau du Dialogue
- `<Nom court de l'application>public.js` : les fonctions publiques utilisées au niveau du Dialogue
- `<CodeEcran>private.js` : les fonctions privées utilisées au niveau de l'Ecran
- `<CodeEcran>public.js` : les fonctions publiques utilisées au niveau de l'Ecran
- `style.css` : le fichier des styles
- `notfoundtplprivate.js` et `notfoundtplpublic.js`



Pour plus d'informations sur les fonctions utilisées au niveau Général, Dialogue et Ecran, consultez le chapitre *Personnalisation des templates Générés*, sous-chapitre *Les fonctions Javascript*.

- dans le sous-répertoire `icon` du répertoire `rep_génération / Nom_court_application` :
les bitmaps utilisées dans les différentes pages
- dans le sous-répertoire `WEB-INF` du répertoire `rep_génération` :
 - `web.xml` : fichier décrivant les paramètres de l'application web et permettant de modifier le comportement de la servlet :
 - Le paramètre `traceLevel`, positionné à `0` par défaut, peut être modifié pour obtenir des traces (`0` : pas de trace, `4` : trace de niveau debug). Ces traces sont redirigées sur la sortie standard du serveur d'application (console sous WAS).
 - Le paramètre `CacheAvailable`, positionné à `True` par défaut, peut être modifié afin d'inhiber la mémoire cache (`True` : mémoire cache activée, les pages html chargées sont mémorisées ; `False` : mémoire cache désactivée, les pages html chargées ne sont pas mémorisées). Lorsque le paramètre est positionné à `False`; les pages html chargées ne sont pas mémorisées, ainsi les modifications effectuées sur les pages sont prises en compte immédiatement sans avoir à relancer le serveur.
 - La valeur par défaut (positionnée à 60000) du paramètre `ConnectionCleaningInterval` du `middlewareAdapter` peut être modifiée.
 - répertoire `lib` contenant les différentes bibliothèques indispensables à l'application (`vaprun.jar`, `vapweb.jar` ou `vapwebctg.jar`).



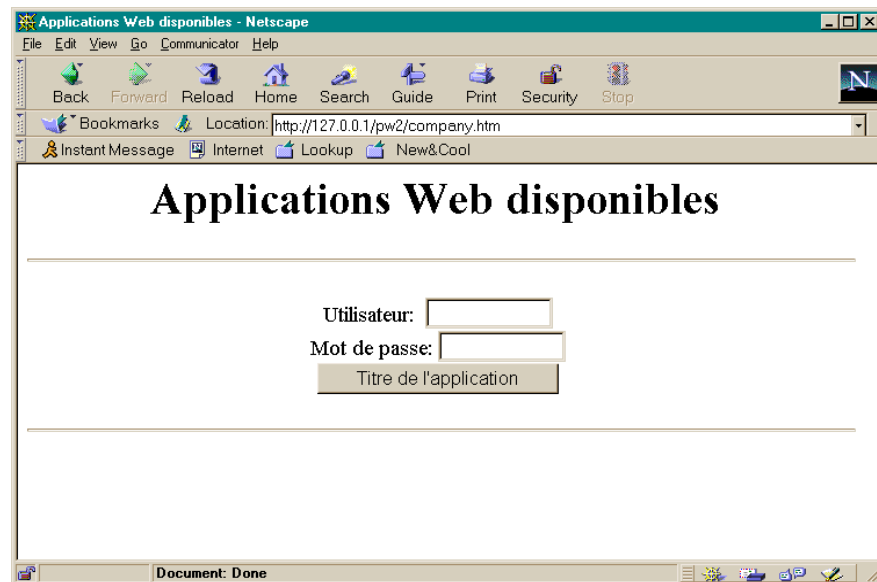
Lorsque vous régénérez une page, vous perdez toutes les modifications initialement apportées aux pages HTML.

Chapitre 6 : Présentation des pages générées

La page d'accueil

La page d'accueil contient les champs **Utilisateur** et **Mot de passe** pour les applications dont la partie serveur applicatif impose une identification (CICS par exemple).

Le libellé utilisé pour le bouton activant l'application est celui que vous avez entré dans le champ **titre de l'application** de la page **Paramètres divers** des options de génération.

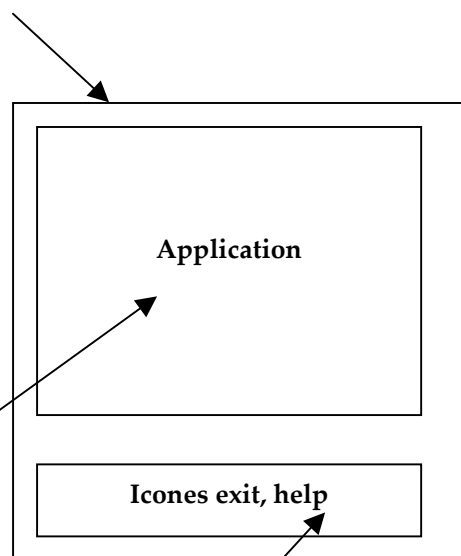


Structure de la page HTML

La page HTML générée est une page générale qui découpe la fenêtre en deux zones (frames), l'une pour afficher l'écran de l'application et l'autre pour afficher les boutons contrôlant la fenêtre de l'application et la fenêtre d'aide.

Nom du fichier : `<ShortApplicationTitle>App.htm`

Fonction : définit 2 frames `app` et `menubar`, contient les fonctions JavaScript



Nom du frame : `app`

Fonction du frame : affiche l'écran demandé

Nom du fichier de l'écran affiché dans le frame : `<NomEcran>.htm`

Nom du frame : `menubar`

Fonction : affiche la barre de contrôle de l'application

Nom du fichier de la barre de menu affichée dans le frame : `menubar.htm`

Description de l'écran

Le fichier `<NomEcran>.htm` généré contient la description de l'écran de l'application. L'écran est découpé en deux sections **HEAD** et **BODY**.

<code><HTML></code>
<code><HEAD></code> Commentaire indiquant que l'écran résulte d'une génération VisualAge Pacbase Informations META identifiant l'écran (code programme, date de génération, etc.) Titre de l'écran Appel de la feuille de style Chargement des fonctions JavaScript (niveau écran) <code></HEAD></code>
<code><BODY></code> <code><FORM METHOD=POST></code> Champ HIDDEN utilisé pour le message d'erreur, la position du curseur, la valeur de la touche fonction Description de l'écran : Littéraux Champs INPUT et SELECT etc. <code></FORM></code>
<code></BODY></code> <code></HTML></code>

Head

Cette section contient trois types d'informations :

- l'identification de la version de programme (instructions META)
- le titre de l'écran.
- le nom de la feuille de style
- le chargement des fonctions JavaScript

Exemple :

```
<TITLE>Titre de l'écran</TITLE>
<META NAME=" SESSI" CONTENT=" 5222 ">
<META NAME=" LIBRA " CONTENT=" XXX ">
<META NAME=" DATGN " CONTENT=" 15012000 ">
<META NAME=" TIMGN" CONTENT=" XXXXXXXX ">
<META NAME=" PROGE " CONTENT=" XXXXXXXX ">
<META NAME=" COBASE " CONTENT=" XXX ">
<LINK REL=STYLESHEET HREF="/zz/app/style.css" type="text/css">
<SCRIPT LANGUAGE="JavaScript" SRC="/zz/js/zzxxxxPrivate.js">
</SCRIPT>
<SCRIPT LANGUAGE="JavaScript" SRC="/zz/js/zzxxxxPublic.js">
</SCRIPT>
```

Body

Cette section contient la description proprement dite de l'écran. Elle contient une instruction **FORM** qui précise l'action '**POST**' (option de transfert des paramètres) et contient éventuellement la liste des contrôles à effectuer (**onSubmit**).

La section **Body** a les caractéristiques suivantes :

- Chaque nouvelle ligne entraîne l'ajout d'un saut à la ligne (instruction **
) dans le cas d'une génération de type « **Line ».
- Chaque libellé ou champ protégé est repris sous forme de texte, adapté en fonction des paramètres de la génération.
- Cas des champs variables (protégés ou saisissables) :
 - rubrique de type **F** (affichée et protégée à l'écran mais reçue par le programme) : il faut rajouter en plus du libellé sous forme de texte, un champ caché ayant pour valeur le texte du libellé
 - rubrique de type **P** (affichée et protégée à l'écran mais non reçue par le programme) : la rubrique est affichée en texte simple
 - rubrique de type **V** (saisissable) : chaque champ modifiable est décrit avec une instruction **<INPUT>** ou **<SELECT>**, contenant le nom de la rubrique correspondante (**attribut NAME**), la longueur du champ (**attribut MAXLENGTH**) et, pour les champs saisissables, la longueur affichée (égale à la longueur maximum lors de la génération, mais modifiable par le développeur). Cette instruction ne contient pas l'appel de la fonction de contrôle appropriée : ceci est fait au niveau du submit (Event Handler de type **onSubmit**).

Exemple :

```
<INPUT TYPE='TEXT' NAME='COPOS' CONTENT='#COPOS000101'  
MAXLENGTH=5 LENGTH=5>
```

- Les champs dont les valeurs ne sont pas connues (type **TEXT** ou **PASSWORD**) ont un attribut **CONTENT** (ou **VALUE**) dont la valeur est le nom du champ préfixé d'un # (cf. exemple ci-dessus).
- Chaque champ ayant une valeur par défaut est initialisé avec cette valeur (attribut **SELECTED** des champs **INPUT** de type **BUTTON**, instruction **SELECTED** des champs **SELECT**, etc.).

Structure des pages JSP

Les pages JSP sont composées d'une en-tête spécifique contenant des clauses d'import, des déclarations Java et des scriptlets suivie d'une partie HTML équivalente à celle des pages HTML : seule la génération des champs variables est différente.

L'affichage des champs variables se fait par l'appel d'une fonction Java (**DisplayField**) dont le code peut être consulté dans le fichier **Private.jspf**.

Structure des pages d'aide

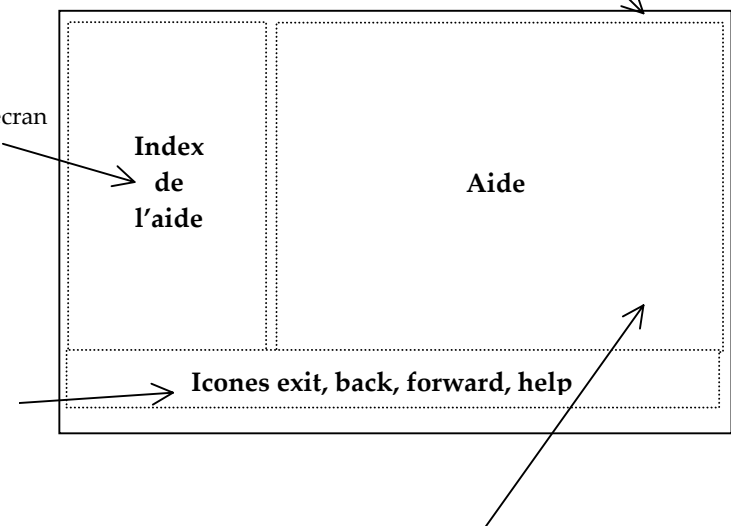
Les pages d'aide sont générées à partir du fichier extrait de VisualAge Pacbase.

L'aide est générée directement en HTML, et sauvegardée sur le serveur HTTP.
Une page d'aide est structurée de la manière suivante :

Nom du fichier : **<Nom court de l'application>help.htm**
Fonction : définit 3 frames **helpindex**, **helptext** et **footer**.

Nom du frame : **helpindex**
Fonction : affiche l'index général ou l'index d'un écran
Nom du fichier : **<Nom court de l'application>_Index_<NomEcran>.htm**

Nom du frame : **footer**
Fonction : Barre de navigation dans l'aide
Nom du fichier : **footer.htm**



Nom du frame : **helptext**
Nom du fichier : **<Nom court de l'application>_Help_<NomEcran>.htm**
Fonction : Affiche l'aide sur l'aide ou l'aide sur l'écran.

Agencement de l'aide par frames

Le fichier **<Nom court de l'application>help.htm** (**<Nom court de l'application>** étant défini dans l'option correspondante de la page **Paramètres obligatoires** des options de génération) définit la taille et l'emplacement des trois frames. Les trois frames sont à leur tour décrits dans trois fichiers : **footer.htm**, **<Nom court de l'application>_Index_<NomEcran>.htm** et **<Nom court de l'application>_help_<NomEcran>.htm**.

Aide contextuelle

Une page d'aide est générée pour chaque écran. Cette page contient :

- l'aide générale associée à l'écran : `<Nom court de l'application>_help_<NomEcran>.htm`
Elle s'affiche dans le frame `helptext`.
- la liste des rubriques de l'écran , avec leur libellé long, les valeurs autorisées et la documentation associée : `<Nom court de l'application>_Index_<NomEcran>.htm`
Elle s'affiche dans le frame `helpindex`.

Aide générale (sommaire)

L'aide générale est affichée dans le frame `helpindex`, c'est-à-dire le même que celui qui contient la liste des rubriques d'un écran particulier.

L'aide générale contient la liste des écrans. Pour chaque écran, un lien hypertexte permet d'accéder à l'aide associée à cet écran. Quand on clique sur le nom d'un écran, la page d'aide correspondante s'affiche dans le frame `helptext` et l'index de l'écran recouvre l'aide générale dans le frame `helpindex`.

La barre de navigation

La barre de navigation s'affiche dans le frame `footer`, elle est décrite dans le fichier `footer.htm`. Elle contient les icônes suivantes :

- `back` et `forward` : permettent de revoir les aides déjà appelées. Elles appellent les fonctions Javascript `history.back()` et `history.forward()` qui se trouvent dans les fonctions standards Javascript.
- `help` : permet d'afficher l'aide générale dans le frame `helpindex`.
- `exit` : permet de quitter l'aide.

Chapitre 7 : Personnalisation des templates générés

Les templates HTML ou JSP générés sont entièrement personnalisables. Vous avez la possibilité de modifier l'aspect d'une page (ajout de couleur, d'icônes, modification des libellés, modification de la présentation des champs ou de leur positionnement). Ces modifications peuvent être faites grâce à un éditeur de page HTML ou en modifiant directement la page HTML à l'aide d'un éditeur de texte.



Pour les templates HTML, vous devez, toutefois, veiller à ne pas supprimer les chaînes de caractères commençant par #.

Exemple : Personnalisation de la page d'accueil

*La page d'accueil est systématiquement générée. Elle comporte par défaut une zone de titre et un bouton qui permet d'activer une application mais elle est modifiable. Vous pouvez, par exemple, ajouter des boutons qui vous permettront de lancer plusieurs applications à partir de la même page d'accueil. Pour cela vous devez recopier le paragraphe HTML suivant que vous trouverez dans la description de la page d'accueil générée, dans le fichier portant le nom que vous aurez saisi dans la zone **Nom de la première page HTML** de la page **Paramètres divers**.*

```
<INPUT TYPE = "button" NAME = "WE" VALUE = "Titre de l'application" onClick = "openNewWin('/WE/app/', 'WEApp')">
```

Les styles

La présentation (taille, police, couleur ...) des différents types de champs d'une page se fait en utilisant le mécanisme des styles CSS1 (Cascading style sheet). La déclaration de ces styles est faite dans le fichier **style.css** sous le répertoire **js** généré.

- Il y a un style par type de champ :
 - **text** pour les champs de type texte
 - **passwd** pour les champs de type secret
 - **libfix** pour les libellés fixes
 - **libvar** pour les libellés variables
 - **radio** pour les boutons radio
 - **screentitle** pour le titre des écrans
 - **dropdown** pour les listes déroulantes.

- 3 styles utilisés lors du maquetage des zones répétitives :
 - **libtbl** pour l'entête du tableau,
 - **rowodd** pour les lignes paires,
 - **roweven** pour les lignes impaires.
- et n styles permettant de simuler l'affichage des Dialogues.

Le nom de ces n styles est calculé à partir des attributs de présentation VisualAge Pacbase. On obtient 56 styles différents de nom **.libXYC** avec **X** pouvant prendre les valeurs **N** ou **B** (Normal, Bold), **Y** pouvant prendre les valeurs **N, B, U, R** (Normal, Bold, Underline, Reverse) et **C** prenant les valeurs **W, R, P, Y, G, T** et **B** (White, Red, Purple, Yellow, Green, Turquoise, Blue).

Les styles peuvent être modifiés, ils ne sont pas écrasés lors d'une nouvelle génération.

Modification automatique des pages HTML

Le générateur peut ajouter des tags commentaires lors de la génération (option **Générer avec des commentaires** de la page **Paramètres divers**). Ces tags sont ajoutés pour permettre une modification automatique des pages générées, les tags permettant à un outil de se repérer dans la page générée. De cette façon, il est possible d'appliquer la même modification sur un ensemble de pages ou de ré-appliquer les mêmes changements après une re-génération.

Les tags générés sont les suivants :

En début de ligne de l'écran **<!--!PWC!LINEx-->** avec **x** = N° de ligne

Chaque champ est entouré par

<!--!PWC!nom du champ-->

ou

<!--!PWC!xCy--> (pour les libellés)

Exemple : Suppression automatique d'une ligne et modification de l'attribut **SIZE** de certains champs.

Cet exemple est écrit en Perl.

```
# print current line until arg1
sub printuntil
{
    unless (/${_}[0]/)
    {
        print OUT;
        while (<IN>) {
            last if (/${_}[0]/);
            print OUT;
        }
    }
}

# replace value of size if greater than 10
sub replacesize {
    &printuntil("SIZE=");
    s/SIZE=[0-9]{2,}/SIZE=15/;
}

# suppress a line
sub supline()
{
    local($wrk) = "<!--!PWC!LINE";
    local($search) = $wrk . $_[0] . "-->";

    if (/${search}/)
    {
        while (<IN>) {
            last if (/${wrk}/);
        }
    }
}

($filein) = @ARGV;
open (IN, $filein) || die "Cannot open file $filein
:$filein \n";
open (OUT, ">result.htm");
while(<IN>) {
    &supline("19");
    if (<!--!PWC!NOMRUE/)
    {
        &replacesize();
        &printuntil("<!--!PWC!NOMRUE");
    }
    print OUT;
};
```

Les fonctions Javascript

Présentation des fonctions JavaScript

Vous avez la possibilité par l'écriture de programmes JavaScript, d'ajouter des contrôles et de personnaliser la présentation au niveau de chaque page.

Les fonctions Javascript générées pour gérer les pages HTML sont divisées en deux catégories, privée et publique, en fonction de leur type d'utilisation.



Les fonctions dites privées sont les fonctions indispensables au bon fonctionnement de l'ensemble, elles ne doivent pas être modifiées.

Les fonctions publiques sont des fonctions générées vides, elles servent de points d'entrée pour le développeur qui pourra ainsi ajouter ses propres traitements. Seul leur nom ne doit pas être modifié.

Contrairement aux fonctions publiques, les fonctions privées sont régénérées systématiquement lors d'une demande de génération; les modifications qui auraient pu être faites sont écrasées. De plus, ces fonctions sont divisées en 3 niveaux : Général, Dialogue et Ecran.

Les fonctions dites du niveau Général sont les fonctions génériques communes à tous les Dialogues, les fonctions dites Dialogue sont des fonctions spécifiques à un Dialogue particulier et enfin les fonctions dites du niveau Ecran sont celles qui sont spécifiques à un écran particulier.

Si l'on a choisi **WX** par exemple comme **Nom court de l'application**, lors de la génération, la nomenclature des fichiers utilisés pour stocker les fonctions sera :

	Générale	Dialogue	Ecran
Publique	Public.js	WXPublic.js	xxxxxxPublic.js
Privée	Private.js	WXPrivate.js	xxxxxxPrivate.js

Fonctions utilisées au niveau Général

Fonctions privées

openNewWin(strPath, strAppName)

Ouvre une nouvelle fenêtre navigateur : utilisé pour afficher l'aide.

Go(str)

Fonction activée lors de la validation d'une page soit par touche fonction soit par touche « Enter »

ReadCook ()

Lecture du nom et du mot de passe de l'utilisateur saisis dans la page d'accueil.

UpdatePosCursor (aObjectText)

Mémoire le champ où se trouve le point d'insertion.

SetPosCursor ()

Positionne le point d'insertion.

DisplayError ()

Affiche un message d'erreur dans une fenêtre JavaScript si ce type d'affichage a été demandé à la génération.

PWCLoad ()

Fonction activée lors du chargement d'une page.

PWCUnload ()

Fonction activée lors du déchargement d'une page.

PWCChange (objet)

Fonction activée lors de la modification d'un champ.

PWCKeyPressed (e)

Cette fonction est exécutée à chaque pression d'une touche du clavier. Elle permet de soumettre une page en tapant la touche Enter.

PWCSubmit ()

Fonction activée lors de la soumission d'une page.

Fonction de traces

Les fonctions **TraceInit**, **Trace** et **TraceView** permettent d'ajouter un mécanisme de trace.

TraceInit initialise une variable dans laquelle sont ajoutées des messages grâce à la fonction **TraceView**. L'affichage du contenu de cette variable, et donc des traces, se fait par l'appel de **TraceView**.

Ces fonctions peuvent être utilisées dans toutes fonctions publiques ou privées.

Fonctions publiques

chkDate (type, field)

Teste si le champ est de type date

chkNum (field)

Teste si le champ est de type numérique

chkHere (field)

Contrôle de présence du champ

displayErr (str, field)

Fonction réalisant l'affichage des erreurs détectées par les fonctions de contrôles ci-dessus.

`initArray()`, `isEmpty(str)`, `isANumber(str)`

Fonctions utilisées pour l'implémentation des fonctions de contrôle.

Fonctions utilisées au niveau Dialogue

Fonctions privées

Pas de fonction mais un fichier est prévu pour d'éventuelles évolutions

Fonctions publiques

Toutes ces fonctions sont vides. Ce sont des points d'entrée pour ajouter des traitements. Elles sont appelées par l'intermédiaire des fonctions du niveau Général.

PWCLoadA()

Fonction appelée lors du chargement d'une page HTML et **après** les traitements spécifiques définis au niveau Général (Cf. **PWCLoad()**).

PWCUnloadB()

Fonction appelée lors du déchargement d'une page HTML et **avant** les traitements spécifiques définis au niveau Général (Cf. **PWCUnload()**).

PWCChangeA(objet)

Fonction appelée lors de la modification d'un champ d'une page HTML et **après** les traitements spécifiques définis au niveau Général (Cf. **PWCChange(objet)**).

PWCChangeB(objet)

Fonction appelée lors de la modification d'un champ d'une page HTML et **avant** les traitements spécifiques définis au niveau Général (Cf. **PWCChange**).

PWCKeyPressedB()

Fonction appelée quand on appuie sur une touche clavier et **avant** les traitements spécifiques définis au niveau Général (Cf. **PWCKeyPressed(e)**).

PWCSubmitB()

Fonction appelée lors de la soumission d'une page HTML et **avant** les traitements spécifiques définis au niveau Général (Cf. **PWCSubmit()**).

Fonctions utilisées au niveau Ecran

Fonction privée

check()

Cette fonction est lancée avant la soumission d'une page : elle effectue les tests de numéricité, de présence et de date qui auraient pu être demandés à la génération.

Fonctions publiques

PWCSCRLoadA ()

Fonction appelée lors du chargement d'une page HTML, **après** les traitements spécifiques définis au niveau Général (Cf. **PWCUnload ()**) et **après** **PWCLoadA ()**.

PWCSCRUnloadB ()

Fonction appelée lors du déchargement d'une page HTML et **avant** les traitements spécifiques définis au niveau Général (Cf. **PWCUnload ()**) et **avant** l'appel de **PWCUnloadB**

PWCSCRChangeA (objet)

Fonction appelée lors de la modification d'un champ d'une page HTML, **après** les traitements spécifiques définis au niveau Général (Cf. **PWCChange (objet)**) et après **PWCChangeA**.

PWCSCRChangeB (objet)

Fonction appelée lors de la modification d'un champ d'une page HTML, **avant** les traitements spécifiques définis au niveau Général (Cf. **PWCChange (objet)**) et avant **PWCChangeA**.

PWCSCRKeyPressedB ()

Fonction appelée lors de l'utilisation de la touche clavier, **avant** les traitements spécifiques définis au niveau Général (Cf. **PWCKeyPressed (e)**) et avant **PWCKeyPressedB**.

PWCSCRSubmitB ()

Fonction appelée lors de la soumission d'une page HTML, **avant** les traitements spécifiques définis au niveau Général (Cf. **PWCSubmitB ()**) et avant **PWCSubmitB**.

Utilisation des fonctions JavaScript comme point d'entrée

Vous pouvez modifier le comportement standard des pages en utilisant les fonctions JavaScript publiques décrites dans la section précédente.

En fonction du niveau auquel elles s'appliquent (Général, Dialogue ou Ecran), ces modifications se répercuteront sur une ou plusieurs pages.

Exemple : Ajout d'une fonction transformant les minuscules en majuscules :
Si l'on veut faire passer systématiquement de minuscule en majuscule les valeurs du champ **NOVOL1000101** pour un écran donné, il suffit de remplacer le code de la fonction **PWCSCRChangeA** définie dans le fichier **<nom_écran>Public.js**.

```

function PWCSRChangeA(objet) {
    var wfrom = objet.name;

    if (wfrom == "NOVOL1000101")
        objet.value = objet.value.toUpperCase()
    }

```

Les templates

Présentation des templates

Les fichiers template HTML générés par Dialog Web Revamping sont construits à partir de fichiers templates livrés dans Dialog Web Revamping.

Ces fichiers template de génération sont placés sous les répertoires **htmltpl** et **jsptpl** : chacun de ces répertoires étant utilisé en fonction du type de génération choisi (HTML ou JSP).

Les fichiers template HTML utilisés lors de la génération contiennent des étiquettes qui sont remplacées de façon dynamique par le contenu des fichiers de l'application.

Page d'accueil

Page HTML contenant le bouton d'activation de l'application.

Template utilisé	Etiquette du template	Remplacée par le contenu du fichier suivant
Firsthtmlpage.tpl	#firsthtmlpagetitle	firsthtmlpagetitle_xx.tpl
	#errorcompat	errorcompat_xx.tpl
	#user	User_xx.tpl
	#password	Password_xx.tpl
	#suffix	Le suffixe donné à la génération

xx correspond au choix de la langue : **fr** pour français et **us** pour anglais.

Page commune aux écrans

Page contenant le découpage en frames et les contrôles locaux communs à chacune des pages écrans.

Template utilisé	Etiquette	Remplacée par
app.tpl	#shortapp	Le nom court de l'application donné à la génération
	#applicationtitle	Le titre de l'application donné à la génération
	#servletcall	servletcall.tpl
	#menubar	Menubar.tpl
	#suffix	Le suffixe donné à la génération
	#msgnoframes	Msgnoframes_xx.tpl

xx correspond au choix de la langue : **fr** pour français et **us** pour anglais.

Page écran

Page principale contenant la description propre à un écran. Deux types de templates sont disponibles selon le format choisi pour la génération de la page.

Template utilisé	Etiquette	Remplacée par
Screen.tpl	#title	
	#shortapp	Le nom court de l'application donné à la génération
	#screenname	Le nom de l'écran VisualAge Pacbase
	#body	Outfield.tpl Label.tpl Radio.tpl Editbox.tpl Hiddenfield.tpl Dropdownbox.tpl Table.tpl
	#enter	Bouton submit Submit_xx.tpl

Page Help

Page contenant le découpage en frames de l'aide

Template utilisé	Etiquette	Remplacée par
help.tpl	#msgnoframes	Msgnoframes_xx.tpl

Page contenant le texte de l'aide

Template utilisé	Etiquette	Remplacée par
helpscreen.tpl	#label	helplabel_us
	#label	helplabel_fr
	#bodytag	bodytag.tpl
	#bg	backgroundfile.tpl

Description des templates

Templates	Description
Privatejspf.tpl	Fichier Private.jspf contenant les scriptlets Java
app.tpl	Page commune contenant le découpage en frames et le chargement des fonctions utilisées pour les contrôles locaux
backgroundfile.tpl	Utilisé pour générer le nom de fichier contenant le fond d'écran dans le cas d'un fond d'écran de type fichier
beginindex.tpl	Debut des pages d'index
bodytag.tpl	Debut de la partie BODY d'une page
bodytbl.tpl	Body set d'une page générée avec option « Table »
bodytitle.tpl	Titre d'une page écran
button.tpl	Bouton submit
cell.tpl	Cellule d'un tableau
chkdate.tpl	Fonction contrôle de date
chkhere.tpl	Fonction contrôle de présence
chknum.tpl	Fonction contrôle de numéricité

Templates	Description
createwarfile.tpl	Contient la commande activée pour fabriquer le fichier .war
dropdownbox.tpl	Liste déroulante
editbox.tpl	Boite éditable
edittpl.tpl	Champ éditable lors d'une génération « à la volée »
endindex.tpl	Fin des pages d'index
eot_xx.tpl	Message de fin de transaction
eotpage.tpl	Contient la page « Fin de transaction »
Epilibjarfile.tpl	Liste des fichiers copiés lors de l'utilisation CTG
errorcompat_fr.tpl	Message d'erreur contrôle du navigateur en français
errorcompat_us.tpl	Message d'erreur contrôle du navigateur en anglais
fieldtpl.tpl	Champ libellé pour une génération à la volée
firsthtmlpage.tpl	Page d'accueil
firsthtmlpagetitle_fr.tpl	Titre français de la page d'accueil
firsthtmlpagetitle_us.tpl	Titre anglais de la page d'accueil
footer.tpl	Pied de page des pages d'aide
help.tpl	Découpage en frames des pages d'aide
helphelp_fr.tpl	Texte de l'aide sur l'aide en français
helphelp_us.tpl	Texte de l'aide sur l'aide en anglais
helpindex.tpl	Page index
helplabel_fr.tpl	Titre de l'aide en français
helplabel_us.tpl	Titre de l'aide en anglais
helplineindex.tpl	Lien utilisé par les index d'aide
helprub.tpl	Ancre pour l'aide sur une rubrique
helpscreen.tpl	Page contenant l'ensemble de l'aide d'un écran
hiddenfield.tpl	Champ invisible
hiddentpl.tpl	Template pour génération à la volée des champs cachés
indextitle_fr.tpl	Titre pour l'index en français
indextitle_us.tpl	Titre pour l'index en anglais
initparam.tpl	Declaration paramètre dans web.xml
itemradio.tpl	Génération d'un item de bouton radio
label.tpl	Libellé d'une rubrique sur un écran
labeltpl.tpl	Libellé lors d'une génération à la volée
libjarfile.tpl	Liste des fichiers copiés lors de l'utilisation Middleware ou Gateway
lineindex.tpl	Lien dans l'index de l'aide HTML
menubar.tpl	Pied de page des écrans
menubarpf.tpl	Pied de page des écrans avec touches fonction
msgdate_fr.tpl	Message pour les contrôles de date en français
lineindex.tpl	Lien utilisé par l'index général
msgdate_us.tpl	Message pour les contrôles de date en anglais
msgday_fr.tpl	Message pour les contrôles de date (jour) en français
msgday_us.tpl	Message pour les contrôles de date (jour) en anglais
msgfield_fr.tpl	Champ
msgfield_us.tpl	Field
msglongdate_fr.tpl	Message long pour les contrôles de date en français
msglongdate_us.tpl	Message long pour les contrôles de date en anglais
msgmonth_fr.tpl	Message pour les contrôles de date (mois) en français
msgmonth_us.tpl	Message pour les contrôles de date (mois) en anglais
msgnoframes_fr.tpl	Message pour signaler l'utilisation de frames en français
msgnoframes_us.tpl	Message pour signaler l'utilisation de frames en anglais
msgnumber_fr.tpl	Message pour les contrôles de numéricité en français
msgnumber_us.tpl	Message pour les contrôles de numéricité en anglais
msgvalue_fr.tpl	Message demande d'une valeur en français
msgvalue_us.tpl	Message demande d'une valeur en anglais
notfound.tpl	Template principal pour la génération « à la volée »
onclick.tpl	Action activée par le clic sur une zone

Templates	Description
onfocus.tpl	Action activée par la prise de focus sur une zone
option.tpl	Choix pour une liste déroulante
outfield.tpl	Champ
passwdtpl.tpl	Champ secret lors d'une génération à la volée
password_fr.tpl	Libellé mot de passe sur écran d'accueil en français
password_us.tpl	Libellé mot de passe sur écran d'accueil en anglais
pfkey.tpl	Touche fonction
private.tpl	Les fonctions JavaScript privées
privatedia.tpl	Les fonctions JavaScript privées niveau Dialogue
privatescr.tpl	Les fonctions JavaScript privées niveau Ecran
public.tpl	Les fonctions JavaScript publiques
publicdia.tpl	Les fonctions JavaScript publiques niveau Dialogue
publicscr.tpl	Les fonctions JavaScript publiques niveau Ecran
radio.tpl	Bouton radio
row.tpl	Ligne dans une table
screen.tpl	Squelette général pour un écran
screenbutton.tpl	Bouton
screenbuttonbar.tpl	Barre de boutons lors de l'enchaînement par boutons
servletcall.tpl	Appel de la servlet
servletexception.tpl	Page affichée pour signaler une exception dans la servlet
starteditor.tpl	Commande utilisée pour lancer l'éditeur de localisation
style.tpl	La feuille de style
submit_fr.tpl	Libellé du bouton submit en français
submit_us.tpl	Libellé du bouton submit en anglais
table.tpl	Table
titleicon.tpl	Ajout d'une icône de titre
user_fr.tpl	Libellé utilisateur pour écran d'accueil en français
user_us.tpl	Libellé utilisateur pour écran d'accueil en anglais
Webxml.tpl	Génération du fichier web.xml

Modification des templates

Dialogue Web Revamping génère les pages HTML à partir de fichiers templates HTML.

Les fichiers templates HTML utilisés lors de la génération contiennent des blocs de syntaxes HTML modifiables qui permettent de construire un paragraphe élémentaire.

Il est possible de modifier ces templates de façon à appliquer ces modifications à l'ensemble des pages générées.

Si vous décidez de modifier un template, plutôt que de modifier directement les fichiers se trouvant sur le répertoire `template.htm`, il est préférable de les dupliquer sous le répertoire `template.usr`.

Chapitre 8 : Annexe

Vous trouverez dans ce chapitre toutes les informations sur les modifications à effectuer sur une application générée avec une version antérieure à Dialog Web Revamping 3.5 pour pouvoir l'utiliser avec cette version 3.5.

Configuration d'un Servlet Container

La servlet **PacWebServlet** est configurée dans le Servlet Container grâce au fichier **web.xml**.

*Exemple de fichier **web.xml***

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app
  PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
  "http://java.sun.com/j2ee/dtds/web-app_2_2.dtd">

<web-app id="APPN">
  <display-name>vols new</display-name>
  <servlet>
    <servlet-name>PacWebServlet</servlet-name>
    <servlet-class>com.ibm.vap.webconnection.html.servlet.PacWebServlet</servlet-
class>
    <init-param>
      <param-name>Suffix</param-name>
      <param-value>.htm</param-value>
    </init-param>
    <init-param>
      <param-name>Error message</param-name>
      <param-value>BOTH</param-value>
    </init-param>
    <init-param>
      <param-name>Instal_Dir</param-name>
      <param-value>/APPN/app</param-value>
    </init-param>
    <init-param>
      <param-name>traceLevel</param-name>
      <param-value>0</param-value>
    </init-param>
    <init-param>
      <param-name>PacWebConversationFactory.serverAdapter.codePageFile</param-
name>
      <param-value>D:\user\VapGateway\Charconv.txt</param-value>
    </init-param>
    <init-param>
      <param-name>PacWebConversationFactory.serverAdapter</param-name>
```



```

        <param-value>com.ibm.vap.middleware.MiddlewareAdapter</param-value>
    </init-param>
    <init-param>
        <param-name>PacWebConversationFactory.serverAdapter.location</param-
name>
        <param-value>PacWeb-LOCAL</param-value>
    </init-param>
    <init-param>
        <param-
name>PacWebConversationFactory.serverAdapter.locationsFile</param-name>
        <param-value>D:\user\VapGateway\vapLocat_pacweb.ini</param-value>
    </init-param>
    <init-param>
        <param-name>ConversationFactory</param-name>
        <param-
value>com.ibm.vap.webconnection.pwcm.PacWebConversationFactory</param-value>
    </init-param>

    <init-param>
        <param-name>endOfTransactionPage</param-name>
        <param-value>/EOTPAGE.htm</param-value>
    </init-param>
</servlet>

<servlet-mapping>
    <servlet-name>PacWebServlet</servlet-name>
    <url-pattern>/PacWebServlet</url-pattern>
</servlet-mapping>

<session-config>
    <session-timeout>
        30
    </session-timeout>
</session-config>
<welcome-file-list>
    <welcome-file>company.htm</welcome-file>
</welcome-file-list>
<error-page>
    <exception-type>javax.servlet.ServletException</exception-type>
    <location>/servletexception.jsp</location>
</error-page>
</web-app>

```

Description des paramètres d'initialisation de la servlet PacWebServlet

Ces paramètres sont les suivants :

- **Suffix**: le suffixe utilisé pour le nommage des templates HTML
- **Error message**: la façon d'afficher les erreurs
- **Instal_Dir**: le répertoire sous lequel se trouve les templates
- **TraceLevel**: niveau de trace de la servlet **0** : pas de trace jusqu'à **4** : trace de debug
- **PacWebConversationFactory.serverAdapter.codePageFile**: le path du fichier de transcodification
- **PacWebConversationFactory.serverAdapter**: l'adaptateur choisi pour le middleware
- **PacWebConversationFactory.serverAdapter.location**: la localisation utilisée dans le fichier des localisations
- **PacWebConversationFactory.serverAdapter.locationsFile**: le path du fichier des localisations
- **ConversationFactory** et **PacWebConversationFactory.serverAdapter**: utilisés pour instancier les classes relatives au middleware choisi.
- **EndOfTransactionPage**: Page affichée lors de la reception d'une fin de transaction
- **EPIConversationFactory.transactionId**: code transaction en mode CTG
- **EPIConversationFactory.geoFile**: liste des fichiers d'extraction utilisés par l'application
- **EPIConversationFactory.address**: adresse de la machine exécutant CTG
- **EPIConversationFactory.server**: le nom du serveur pour CTG
- **EPIConversationFactory.port** : le N° de port de la gateway CTG
- **HtmlFiller**: ce paramètre permet d'inhiber la modification dynamique de la présentation des champs. Dans ce cas, il faut ajouter le bloc suivant :

```
<init-param>  
<param-name>HtmlFiller</param-name>
```

```
<param-  
value>com.ibm.vap.webconnection.PacWebHtmlFiller</para  
m-value>  
</init-param>
```

Migration des applications (Versions 2.5, 3.0) vers le mode J2EE (Version 3.5)

La version 3.5 de Dialog Web Revamping étant implémentée par une servlet, un simple serveur HTTP ne suffit plus : il est indispensable d'utiliser un moteur de servlet ou serveur d'application Web supportant les servlets (par exemple Websphere Application Server, Tomcat...)

Prenons le cas d'une ancienne application que l'on voudrait déployer en utilisant un **Root Context =/oldpacweb** :

Appel de la servlet

Les pages HTML générées doivent être modifiées afin de remplacer l'appel du script perl par l'appel de la servlet **PacWebServlet**.

Cette modification intervient dans 2 fichiers de l'application (**private.js** et **<AppCode>app.htm**) ainsi que dans la première page HTML générée (ie **company.htm**)

Dans ces 2 fichiers, il faut remplacer l'appel du script cgi :

Soit pour **<AppCode>app.htm** :

```
wrk = "<frame src=" + '/cgi/perl.exe?cgicgi.pl%20' +  
today.getTime() + '%20YOYO%20' + MyUser + '%20' + MyPasswd  
+ '%20localhost' + "name='app' scrolling=yes>"
```

par l'appel de la servlet

```
wrk = "<frame src=" + '/pacweb/servlet/PacWebServlet' +  
"name='app' scrolling=yes>"...
```

et pour **private.js** la ligne

```
top.app.document.forms[0].action='/pacweb/pacweb/servlet/Pa  
cWebServlet ...etc
```

est remplacée par la ligne

```
top.app.document.forms[0].action='/pacweb/servlet/PacWebSer  
vlet';
```



L'appel de la fonction javascript **ReadCook()** n'est plus utile et **doit être supprimée** de la page **<AppCode>app.htm**

Gestion des cookies pour le user/password

Dans la page d'authentification (ex : `company.html`), il est nécessaire de modifier la constitution du cookie en lui donnant le nom "pacweb=".

Exemple :

```
document.cookie="pacweb=@pwc@" + user + "@pwc@" + passwd;
```

Ajout des nouveaux fichiers

Sous la racine de l'application là où se trouve la page d'accueil, il faut ajouter les fichiers `EOTPAGE.<suffix>` et `servletexception.jsp`. Ces pages peuvent être retrouvées sous les répertoires `htmltpl/template.htm/EOTPAGE.tpl` et `htmltpl/servletexception.tpl`. Si l'on utilise le template `EOTPAGE.tpl`, il faudra remplacer la chaîne `#eot` par la chaîne de caractères que l'on veut voir apparaître lors de la fin de transaction.

Un nouveau répertoire `WEB-INF` doit être créé contenant le fichier de déploiement `web.xml` et le répertoire `lib` contenant les fichiers utilisés par la servlet `PacwebServlet` : `vaprun.jar` et `vapweb.jar`.

Vous devez adapter les paramètres `PacWebConversationFactory.serverAdapter.locationsFile` et `PacWebConversationFactory.serverAdapter.location` à votre fichier de configuration de middleware (cf. plus bas).

Ajout du RootContext dans les pages

Chaque appel à une ressource, que ce soit une feuille de style ou l'appel des fichiers contenant les javascripts, doit être modifié de façon à ce que l'URL soit préfixée par le Root Context qui sera choisi pour déployer l'application.

Exemple :

```
LINK REL=STYLESHEET HREF="/APPN/js/style.css" type="text/css">
```

Sera remplacé par :

```
LINK REL=STYLESHEET HREF="/oldpacweb/APPN/js/style.css" type="text/css">
```

Cette modification peut être, bien entendu, facilement automatisée par l'utilisation d'un script Perl par exemple.

Pour déployer facilement cette application, on peut fabriquer un fichier `.war` qui sera ensuite importé sur le serveur d'applications Web.

Exemple de la commande

```
Jar -cf New.jar APPN WEB-INF servletexception.jsp EOTPAGE.htm
```

Fichier de configuration du middleware VAP

L'utilisation des interfaces Java permettant l'utilisation du middleware VAP nécessite la création d'un fichier reprenant la configuration middleware.

Exemple :

```

<Location-PacWeb-CICS>
MWARE= TCPMVS
MESSAGE_LENGTH=14200
MONITOR=WE1S
MWTIMEOUT= 2000
MWADDRESS= 9.134.0.0 9999
MWCODEPAGE= 297
MWTRANSID=WE1S

```

Exemple d'utilisation des proxies conversation

Indépendamment de la servlet `PacWebServlet`, il est possible d'utiliser directement la couche Proxy conversation responsable de la mise en œuvre de la communication vers l'application host et donc d'utiliser la couche proxy conversation comme source de données provenant d'application Dialogue.

```

public static void main(String[] args) {
    ConversationFactory conversationFactory;
    java.util.Properties properties = new java.util.Properties();
    // Initialisation pour utilisation GatewayAdapter
    properties.put(SERVER_ADAPTER, "com.ibm.vap.gateway.GatewayAdapter");
    properties.put(CONVERSATION_FACTORY,
        "com.ibm.vap.webconnection.pwcm.PacWebConversationFactory");
    properties.put("PacWebConversationFactory.serverAdapter.locationsFile",
        "D:/user/VapGateway/vapLocat_pacweb.ini");
    properties.put("PacWebConversationFactory.serverAdapter.location",
        "PacWeb-LOCAL");
    properties.put("PacWebConversationFactory.serverAdapter.codePageFile",
        "D:/user/VapGateway/CharConv.txt");
    properties.put("PacWebConversationFactory.serverAdapter.host",
        "127.0.0.1");
    properties.put("PacWebConversationFactory.serverAdapter.port", "8888");
    conversationFactory = new PacWebConversationFactory();
    conversationFactory.initialize(properties);

    Conversation conversation = conversationFactory.createConversation();

    int conversationState = -1;
    Screen newScreen = null;
    while(true)
    {
        conversationState = conversation.getState();
        try
        {
            switch (conversationState)
            {
                case Conversation.BEGIN :
                    { //lancement de la conversation, récupération du premier écran
                        newScreen = conversation.begin();
                        break;
                    }
                case Conversation.MIDDLE :
                    { //mise a jour de l'écran avec les nouvelles valeurs et poursuite
de la conversation
                        newScreen = conversation.nextScreen();
                    }
            }
        }
    }
}

```

```

        break;
    }
    case Conversation.END :
    { //fin de la conversation
        newScreen = conversation.end();
        break;
    }
}

if (newScreen != null)
{
    displayScreen(newScreen);
    if (newScreen.getCode().equalsIgnoreCase("WE00ME"))
    {

    }
    else
    {
        // Affiche les valeurs d'une iterative
        Field field;
        ListIterator ListedesPilotes =
newScreen.findUpdatableFieldsMatching(Screen.ITERATIVE_CATEGORY, "NOMPIL");
        while (ListedesPilotes.hasNext()) {
            field = (com.ibm.vap.webconnection.Field) ListedesPilotes.next();
            System.out.println("NOMPIL[" + field.getCategoryLineIndex() + "]= "
                + field.getStringValue());

            }
            // Mettre "FT" dans le code choix
            Field codeChx = newScreen.findUpdatableFieldMatching("COCHX");
            codeChx.setStringValue("FT");
        }
    }
} catch (ConversationException ce)
{
    System.out.println("ConversationException : " + ce.getMessage());
}
if (newScreen == null) break;
} //while
}

```

Utilisation avec 'CICS Transaction Gateway'

C'est une fonctionnalité disponible uniquement pour les applications générées sous CICS en mode natif, c'est-à-dire avec prise en compte des maps BMS. L'application envoie un message physique correspondant à la map BMS d'où la présence dans le message du code de l'écran mais l'absence d'informations relatives à certains champs (code du champ, catégorie, nature...). Ces informations sont donc recherchées dans le fichier obtenu à l'issue de l'extraction du ou des fichiers contenant la description des écrans VA Pac (**GE0C4**) (S'assurer, par conséquent, que le fichier soit bien en phase avec la partie serveur).

La version 4.0 au minimum de Cics Transaction Gateway est nécessaire pour utiliser cette fonctionnalité.



Le moteur de servlet doit avoir été configuré de manière à avoir accès au fichier **ctgclient.jar**.



Référence : DSOWB000353F