



[Hornet] Développement Hornet



Guide de migration d'un projet Acube vers Hornet

Cette création est mise à disposition selon le Contrat Paternité - Pas d'Utilisation Commerciale - Partage des Conditions Initiales à l'Identique disponible en ligne <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/> ou par courrier postal à Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA

SUIVI DES MODIFICATIONS

Version	Auteur	Description	Vérification	Date
1.0	F. Bernier-Malcoiffe	Document initial	C. Rocheteau	04/04/12
2.0	A. Caillaud	Application du nouveau modèle de document Mise à jour Hornet 2.0	F. Bernier-Malcoiffe	23/11/12

DOCUMENTS DE REFERENCE

Version	Titre
1.0	Guide de création d'un projet Hornet
1.0	Guide du développeur Hornet 2.0

SOMMAIRE

SUIVI DES MODIFICATIONS	2
DOCUMENTS DE REFERENCE	2
SOMMAIRE.....	3
TABLEAUX	3
FIGURES.....	3
1 INTRODUCTION.....	4
2 PROCESSUS DE MIGRATION	5
2.1 MOTIVATION ET PRINCIPE DE MIGRATION	5
2.2 PROCESSUS GENERAL	5
2.3 RECENSEMENT DES IMPACTS ET CHANGEMENTS COTE CLIENT	5
2.3.1 Règles de gestion.....	5
2.4 RECENSEMENT DES IMPACTS COTE SERVEUR	5
2.5 PRISE EN COMPTE DE LA NOUVELLE ARCHITECTURE	6
2.6 CREATION DE LA STRUCTURE ET DES MODELES DE PAGE CLIENTE	6
2.7 MISE A JOUR DE LA STRUCTURE DU PROJET SERVEUR.....	6
2.8 VERIFICATION DE FONCTIONNEMENT ET DE NON REGRESSION.	7
3 MIGRATION A PARTIR D'UN GABARIT HORNET.....	8
3.1 GENERATION D'UN PROJET HORNET	8
3.2 GESTION DES DEPENDANCES	8
3.3 ARBORESCENCE PROJET	8
3.4 CONFIGURATION DU PROJET HORNET	8
3.4.1 Struts	8
3.4.2 Tiles.....	8
3.4.3 iBATIS	8
3.5 REFONTE PARTIE CLIENTE	8
3.6 APPLICATION D'UN THEME.....	9

TABLEAUX

Aucune entrée de table d'illustration n'a été trouvée.

FIGURES

Aucune entrée de table d'illustration n'a été trouvée.

1 Introduction

Ce guide de migration présente les grandes étapes et principes d'une migration d'application Acube vers Hornet.

Hornet est un Framework, un outillage, une méthodologie qui permet de réaliser efficacement des applications Web accessibles (conformes au RGAA niveau AA) et performantes.

Hornet est composé de plusieurs parties :

- hornetclient pour les parties CSS et JavaScript sur les postes et navigateurs,
 - un thème par défaut
 - des composants (onglet, formulaire, menu horizontal, tableau, calendrier, ...)hornetclient s'appuie sur la bibliothèque de composants éprouvés « Yahoo User Interface Toolkit » (YUI).
- hornetserver pour faciliter le développement java côté serveur,
- hornettemplate pour créer un gabarit de projet Hornet

Dans la suite de ce document, FRED 2 désigne les versions 2.x du Framework ergonomique FRED Acube, et LISE 3 désigne les versions 3.x du Framework LISE Acube.

Le remplacement de Acube par Hornet apporte des solutions et des améliorations sur des points tels que l'accessibilité, l'ergonomie ou encore la facilité de développement.

2 Processus de migration

2.1 Motivation et Principe de migration

Il y a plusieurs motivations possibles pour une migration vers Hornet comme :

- Disposer d'une application avec une nouvelle IHM moderne,
- Satisfaire des exigences d'accessibilité,
- Ajouter des fonctionnalités à des écrans Acube existants.

La migration d'une application Acube FRED2 et LISE 3 vers Hornet peut se faire partiellement ou totalement.

Cependant vu les exigences d'accessibilité et la structure des applications FRED2, la coexistence d'Hornet et FRED2 au sein d'une application n'est pas souhaitable.

Dans la plupart des cas, la migration d'Hornet implique une refonte totale de la partie présentation d'une application.

Il est toujours possible de migrer partiellement une application ou d'utiliser Hornet et FRED2 dans une même page, mais ces cas doivent faire l'objet d'une étude spécifique.

2.2 Processus général

Le processus général d'une migration est :

- Recensement des impacts et des changements côté client,
- Recensement des impacts côté serveur,
- Prise en compte de la nouvelle architecture,
- Création de la structure et des modèles des pages clientes,
- Mise à jour de la structure de la partie serveur,
- Vérification de fonctionnement et de non régression.

2.3 Recensement des impacts et changements côté client

Ce recensement consiste à lister les impacts de l'utilisation d'Hornet à la place de FRED.

Hornet permet plus de souplesse sur les ergonomies des applications, par exemple :

- Les menus ne sont plus obligatoirement à gauche de la page, et peuvent être situés à l'horizontal sous l'entête,
- Il est possible d'afficher des fenêtres de dialogue modales,
- Le tableau éditable à la FRED2 est remplacé au choix par :
 - o deux pages, le tableau et le formulaire d'édition.
 - o le formulaire d'édition en pop-in

2.3.1 Règles de gestion

Il faut également s'assurer qu'aucune règle de gestion décrite dans les spécifications ne soit effectuée uniquement au sein de la partie cliente. Si le cas s'avère être présent, le plan de test de l'application doit impérativement contenir les cas permettant de tester ces règles de gestion avec la version Hornet.

2.4 Recensement des impacts côté serveur

Côté serveur, les impacts concernent essentiellement la construction des pages HTML et les flux de données.

Avec Hornet, les pages HTML sont générées côté serveur (JSP) et contiennent les données. La présentation et l'affichage sont contrôlés par des feuilles de styles CSS.

Avec FRED2, les pages HTML contiennent uniquement la présentation et les données sont échangées avec le serveur via des flux XML et des requêtes Ajax.

Dans le cas où votre application est basée sur LISE 2, la mise en œuvre d'Hornet client nécessitera probablement une réécriture complète des classes actions et des flux associés et la migration du code serveur vers Hornet serveur.

2.5 Prise en compte de la nouvelle architecture

L'architecture Hornet diffère de manière importante de celle de ACube; plus particulièrement s'agissant de la partie présentation.

Basiquement, avec ACube, les pages HTML ne contiennent que des conteneurs HTML C'est le JavaScript (via FRED) qui enrichit dynamiquement ces conteneurs. Il crée des fragments de HTML (composants formulaires, tableaux, ...) et les initialise avec les données applicatives récupérées de manière asynchrone sur le serveur. Avec un navigateur ne supportant pas le JavaScript, les pages sont, pour ainsi dire, vides.

Hornet a pour principe majeur le « Progressive Enhancement » (ou l'amélioration progressive).

Les pages web sont ainsi composées de trois parties :

- La partie sémantique en html ou xhtml (le fond),
- La partie présentation avec des styles CSS (la forme),
- La partie interactive, dynamique et événementielle avec du javascript (la forme).

Le service à l'utilisateur doit être « augmenté et enrichi » progressivement par chaque couche. Généralement la partie interactive du « *Progressive Enhancement* » inclut des échanges de type Ajax, d'où l'appellation « *Hijax* » fréquemment rencontrée pour la combinaison du *Progressive Enhancement* et d'Ajax.

De plus, dans Hornet, les scripts client (YUI, hornetclient, thèmes) sont externes à l'application, sur un serveur distant.

L'architecture Hornet est détaillée dans les deux premiers chapitres du « Guide du développeur Hornet ».

2.6 Création de la structure et des modèles de page cliente

Avec Acube, le « layout » des pages de l'application est effectué à l'aide du « frameset » de FRED 2.

Avec Hornet, il est réalisé à l'aide de JSP et Apache Tiles.

Cette étape consiste à supprimer l'ensemble de la partie cliente liée à FRED 2 (soit le contenu du répertoire « Client ») et créer les modèles Tiles équivalents.

Hornet Server inclut des TagLib Struts 2 pour faciliter la création de page fonctionnant avec Hornet client, ces Taglib sont à privilégier dans les JSP.

2.7 Mise à jour de la structure du projet serveur

Côté serveur, les mises à jour concernent :

- L'arborescence globale
- Les tâches de construction (build.xml)
- La gestion des dépendances avec Ivy :
 - Il faut dans un premier temps, supprimer le contenu du répertoire « lib » puis ajouter la librairie propre à Ivy.
 - De même, il faut supprimer le répertoire « WEB-INF/lib », les librairies doivent être récupérées par les tâches ivy.
 - Pour la mise en place et l'utilisation d'Ivy, il faut se référer au chapitre dédié à la gestion des dépendances dans le « Guide du développeur Hornet ».
- L'ajout de Tiles comme plugin Struts pour la factorisation des pages HTML.

Une modification de la configuration Struts est requise afin d'y inclure Tiles. De plus, d'autres fichiers de configuration tels que « struts-tiles.xml » et « tiles-config.xml » seront nécessaires.

- Les pages et scripts de référence

Pour ces changements, le plus simple est de créer un nouveau projet Hornet server à partir du template et de comparer avec le projet à migrer.

2.8 Vérification de fonctionnement et de non régression.

Comme montré précédemment, la migration d'une application de ACube vers Hornet représente un effort important dont le résultat doit faire l'objet d'une vérification de fonctionnement et de non régression complète.

Une vigilance particulière doit être portée sur le point soulevé en §2.3.1 Règles de gestion.

3 Migration à partir d'un gabarit Hornet

3.1 Génération d'un projet Hornet

Le delta entre une arborescence de type Acube et une arborescence Hornet est très important. Aussi, il est nécessaire de partir d'un projet Hornet bien organisé. Pour cela, on utilise un gabarit Hornet permettant de créer un nouveau projet. La création de ce projet doit être effectuée en suivant le « Guide de création d'un projet Hornet » basé sur l'utilisation de `hornettemplate`.

3.2 Gestion des dépendances

Dans Hornet, la gestion de configuration est assurée par Ivy. Il faut donc utiliser un fichier de configuration listant les dépendances plutôt que d'embarquer les librairies dans le répertoire « lib » du projet.

Un projet Hornet généré par `hornettemplate` contient déjà un ensemble de librairies de référence (Spring, Struts, Apache Commons, JTDS, ...). Chaque projet utilise ensuite ses propres librairies qui sont renseignées dans le fichier de configuration d'Ivy.

3.3 Arborescence projet

Côté serveur, lorsqu'on a une arborescence de projet Acube ainsi qu'une bonne organisation des packages Java, la correspondance avec la nomenclature Hornet est assez simple. On peut facilement transposer les classes d'un package Acube vers un package Hornet.

3.4 Configuration du projet Hornet

Dans un projet Hornet, les fichiers de configuration doivent être placés dans `/src/config`.

3.4.1 Struts

La principale difficulté réside dans l'utilisation d'un fichier Struts fortement lié au Framework client FRED. La différence la plus importante concerne la gestion d'erreurs. Dans un projet Hornet, on emploie un interceptor alors que dans un projet Acube, on utilise un résultat global. Utilisé conjointement à Tiles, le mécanisme Hornet permet d'afficher les erreurs métier et technique directement dans la page courante sans redirection, simplifiant ainsi leur prise en charge.

3.4.2 Tiles

Tiles n'est pas utilisé dans les projets Acube. La configuration présente dans le fichier `tiles.xml` du projet Hornet est le socle de base et sert de modèle afin de configurer l'ensemble des pages.

3.4.3 iBATIS

Le fichier référençant les configurations de mapping iBATIS est renommé en `ibatis.xml` et placé dans le répertoire `/src/config`. La nomenclature Hornet préconise de créer un fichier de mapping par table en base de données et de le suffixer par « `_SqlMap.xml` ».

3.5 Refonte partie cliente

La modification la plus importante concerne la partie cliente. Le passage à Hornet implique un changement profond de fonctionnement de l'IHM. De ce fait, une refonte complète est nécessaire.

Dans un projet Acube, la partie commune à toutes les pages de l'application (entête et pied de page) est dupliquée dans chaque page. La migration est l'occasion de mettre en commun ces éléments grâce à l'utilisation de Tiles et des layouts de base fournis par Hornet.

Dans tous les cas, les anciennes pages HTML sont entièrement réécrites sous forme de JSP éventuellement complétées par les composants YahooUI et hornetclient.

La structure des pages est revue. Dans un projet Acube, celle-ci est réalisée à l'aide de tableaux HTML. Dans un projet Hornet, il est préconisé d'utiliser YUI Grids CSS (basé sur des blocs (<div>) et des styles CSS pour positionner les éléments de la page).

3.6 Application d'un thème

Un projet créé à l'aide de hornettemplate utilise le thème par défaut (cf. Guide de création d'un projet Hornet).

Si un autre thème est nécessaire, il est tout à fait possible de le développer. Il faut alors faire attention à ce qu'il corresponde au layout fourni par hornettemplate.

FIN DU DOCUMENT