



# [ HORNET ]

## Déploiement du framework Hornet



Mise en œuvre Hornet 3.10

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	O. Rousseil	Changement de nommage du fichier Mise à jour Hornet 3.10		

## DOCUMENTS DE REFERENCE

Titre
HORNET_GUI_Création d'un projet Hornet_3.10_1.0
HORNET_GUI_Guide de Paramétrage_1.4

## SOMMAIRE

<b>SUIVI DES MODIFICATIONS</b> .....	<b>2</b>
<b>DOCUMENTS DE REFERENCE</b> .....	<b>2</b>
<b>SOMMAIRE</b> .....	<b>3</b>
<b>1 INTRODUCTION</b> .....	<b>4</b>
<b>2 CONSTRUCTION DES LIVRABLES</b> .....	<b>5</b>
2.1 PRE-REQUIS.....	5
2.1.1 <i>Logiciels</i> .....	5
2.1.2 <i>Repository Ivy</i> .....	5
2.1.3 <i>Configuration Ivy</i> .....	5
2.2 HORNETSERVER.....	6
2.2.1 <i>Récupération des sources</i> .....	6
2.2.2 <i>Publication des livrables</i> .....	6
2.3 HORNETCLIENT .....	7
2.3.1 <i>Récupération des sources</i> .....	7
2.3.2 <i>Publication des livrables</i> .....	7
2.4 HORNETTEMPLATE .....	7
2.4.1 <i>Description</i> .....	7
<b>3 DEPLOIEMENT DES LIVRABLES</b> .....	<b>8</b>
3.1 DEPLOIEMENT DE HORNETSERVER DANS UN PROJET HORNET .....	8
3.2 DEPLOIEMENT DE LA PARTIE CLIENTE .....	8
3.2.1 <i>Schéma de l'arborescence</i> .....	8
3.2.2 <i>Création de l'arborescence</i> .....	8
3.2.3 <i>Configuration du serveur Web</i> .....	9

# 1 Introduction

Ce document précise les étapes pour la construction et le déploiement des livrables Hornet.

## 2 Construction des livrables

### 2.1 Pré-requis

#### 2.1.1 Logiciels

Se référer au guide de paramétrage (Java 6, Tomcat 6, Eclipse Kepler, Ant 1.9.2, Graphviz 2).

#### 2.1.2 Repository Ivy

Le repository compatible Hornet (disponible dans l'onglet Fichiers sur <https://adullact.net/projects/hornet/>) utilisé par Ivy doit être installé en local.

- Créer un répertoire local qui sert de référence dans la suite du document pour la configuration des propriétés Ivy. Le chemin vers ce répertoire est représenté par la variable <REPertoire REPOSITORY>.
- Récupérer l'archive contenant le repository compatible Hornet 3.10.
- Extraire l'archive et copier le contenu du répertoire « Repository\_AAAAMMJJ » dans <REPertoire REPOSITORY>.
  - L'arborescence obtenue sera la suivante :
    - <REPertoire REPOSITORY>/repository-technique-local
    - <REPertoire REPOSITORY>/repository-tiercesparties
    - <REPertoire REPOSITORY>/repository-integration
    - <REPertoire REPOSITORY>/repository-metier

#### 2.1.3 Configuration Ivy

Les fichiers ivysettings.xml et ivysettings.properties doivent être stockés dans le profil utilisateur.

Ex : sous windows 7 : **C:\Users\NomUtilisateur\ivy2**

Le fichier « **ivysettings.xml** » permet de configurer :

- Localisation des repositories
  - Fichier « **ivysettings.properties** »
- Accès aux repositories
- Cache
- Statuts

Le fichier ivysettings.properties contient la configuration des urls.

Note : il est fait référence au repository-technique-remote. Celui-ci représente les repos public type maven : <http://repo1.maven.org/maven2>

Il ne demande donc pas de configuration spécifique.

Exemple de ivysettings.xml :

```
<?xml version="1.0" encoding="UTF-8"?>
<ivy-settings>
  <properties file="ivysettings.properties" />

  <settings defaultResolver="main" />

  <resolvers>
    <chain name="main" returnFirst="true" >

      <filesystem name="repository-integration">
        <ivy pattern="${repository.integration.url}/${repository.integration.ivy.pattern}" />
        <artifact pattern="${repository.integration.url}/${repository.integration.artifact.pattern}" />
      </filesystem>

      <filesystem name="repository-technique-local">
        <ivy pattern="${repository.technique.local.url}/${repository.technique.local.ivy.pattern}" />
        <artifact
pattern="${repository.technique.local.url}/${repository.technique.local.artifact.pattern}" />
      </filesystem>
    </chain>
  </resolvers>
</ivy-settings>
```

```
<ibiblio name="repository-technique-remote" m2compatible="true"/>

<filesystem name="repository-technique-tiercesparties">
  <ivy
pattern="${repository.technique.tiercesparties.url}/${repository.technique.tiercesparties.ivy.pattern}" />
  <artifact
pattern="${repository.technique.tiercesparties.url}/${repository.technique.tiercesparties.artifact.pattern}" />
  </filesystem>

<filesystem name="repository-metier">
  <ivy pattern="${repository.metier.url}/${repository.metier.ivy.pattern}" />
  <artifact pattern="${repository.metier.url}/${repository.metier.artifact.pattern}" />
  </filesystem>

</chain>
</resolvers>

</ivy-settings>
```

### Ivysettings.properties :

```
default.artifact.pattern=[organisation]/[module]/[revision]/[type]s/[module]-[revision](-[classifier]).[ext]
default.ivy.pattern=[organisation]/[module]/[revision]/ivys/[module]-[revision]-ivy.xml

repository.url=${user.home}/.ivy2

# dépôts
repository.integration.url=${repository.url}/repository-integration
repository.integration.artifact.pattern=${default.artifact.pattern}
repository.integration.ivy.pattern=${default.ivy.pattern}

repository.metier.url=${repository.url}/repository-metier
repository.metier.artifact.pattern=${default.artifact.pattern}
repository.metier.ivy.pattern=${default.ivy.pattern}

repository.technique.local.url=${repository.url}/repository-technique-local
repository.technique.local.artifact.pattern=${default.artifact.pattern}
repository.technique.local.ivy.pattern=${default.ivy.pattern}

repository.technique.tiercesparties.url=${repository.url}/repository-tiercesparties
repository.technique.tiercesparties.artifact.pattern=${default.artifact.pattern}
repository.technique.tiercesparties.ivy.pattern=${default.ivy.pattern}
```

## 2.2 hornetserver

Ce livrable constitue la partie serveur de Hornet.

### 2.2.1 Récupération des sources

Dans Eclipse, bien veiller à travailler sur un workspace en **UTF-8** (Preferences>General>Workspace:UTF-8).

Récupérez le projet « **hornetserver** » sous Eclipse par un checkout SVN ou un import des sources.

### 2.2.2 Publication des livrables

Remarque concernant la génération JavaDoc : les binaires Graphviz doivent être dans le PATH (vérifiable avec la ligne de commande « **dot -help** »).

- Lancer la tâche « **publish-project-all-integration** » du build.xml à la racine du projet. Les artefacts seront créés dans le dossier « livrable », puis seront copiés dans le « **repository-integration** » du référentiel de composants d'Ivy en respectant l'arborescence utilisée :
  - Le dossier « **fr.gouv.diplomatie.hornet** » contient les différents composants générés.

- Pour chaque composant, on retrouve les dossiers contenant le jar, la Javadoc dans un fichier zip, ainsi que les sources archivées, les fichiers de tests éventuels ainsi que les rapports (ivy et de tests si présents).
- Pour une version release, lancer la tâche « **publish-project-all-release** » du build.xml.

## 2.3 hornetclient

---

Ce livrable constitue la partie client riche de Hornet.

### 2.3.1 Récupération des sources

---

Récupérer le projet « **hornetclient** » sous Eclipse par un Checkout SVN ou un import des sources.

### 2.3.2 Publication des livrables

---

La procédure suivante permet la génération du livrable des composants et du livrable du thème par défaut.

Elle permet aussi de générer des livrables YUIet gallery « nettoyés » de la version issue du repository. Ainsi les fichiers dont les extensions sont paramétrées dans le build.properties sont supprimés. Les extensions pré-paramétrées sont .swf, .psd, .old (insensibles à la casse).

```
#Patterns de fichiers à exclure des livrables YUI  
build.yui.excludes.fileextensions=**/*.swf,**/*.psd,**/*.old,**/*-coverage.js
```

#### Procédure :

En fonction d'une livraison en intégration, Lancez la tâche « **publish-project-all-integration** » ou « **publish-project-all-release** » du build.xml à la racine du projet.

Vérifier la présence de ces artefacts dans le repository-integration (version integration) ou repository-metier (version release).

## 2.4 hornettemplate

---

### 2.4.1 Description

---

Ce livrable constitue la partie génération de nouveau projet Hornet, il est constitué de hornettemplate, se référer au document « Guide de création d'un projet Hornet ».

## 3 Déploiement des livrables

### 3.1 Déploiement de hornetserver dans un projet Hornet

Le fichier de configuration du framework Ivy, pour une application, permet de définir les dépendances vers le framework hornetserver.

Ce fichier de configuration « **ivy.xml** » implémente des dépendances vers le framework hornetserver avec les profils définis dans la configuration.

```
<dependencies>
  <dependency org="fr.gouv.diplomatie.hornet"
    name="hornetserver-all"
    rev="3.10.2"
    conf="compile->core,libDependances;runtime->runtime;test->test"
    transitive="true" />
</dependencies>
```

### 3.2 Déploiement de la partie cliente

La partie cliente est déposée sur un serveur web : apache.

Dans le répertoire htdocs pour une installation windows et /var/www pour une version linux.

#### 3.2.1 Schéma de l'arborescence

L'arborescence finale attendue est la suivante :

Le répertoire parent sera appelé répertoire [PARENT]. C'est le contexte minimal au bon fonctionnement dans le sens où n'importe quelle sur-arborescence à [PARENT] est possible.

- [PARENT]
  - hornetclient
    - [HORNETCLIENT\_VERSION]
      - fwk
      - themes
        - hornet-skin-defaut
        - hornet-skin-defaut-xcombine
        - hornet-skin-[THEME\_A\_NOM]-[THEME\_A\_VERSION]
        - hornet-skin-[THEME\_A\_NOM]-xcombine-[THEME\_A\_VERSION]
        - hornet-skin-[THEME\_B\_NOM]-[THEME\_B\_VERSION]
        - hornet-skin-[THEME\_B\_NOM]-xcombine-[THEME\_B\_VERSION]
  - yui
    - yui
      - [VERSION]
    - gallery
      - [VERSION]

Les arborescences hornet-skin-[THEME\_A\_NOM]-[THEME\_A\_VERSION] et hornet-skin-[THEME\_B\_NOM]-[THEME\_B\_VERSION] sont des éventuelles arborescences de thèmes supplémentaires.

#### 3.2.2 Création de l'arborescence

Les zips de hornetclient précédemment créés sont :

- hornetclient-fwk-X.X.X.zip
- hornetclient-themes-X.X.X.zip
- hornetclient-yui-X.X.X-gallery.zip
- hornetclient-yui-X.X.X-yui.zip

##### 3.2.2.1 Pré-requis



Avoir l'arborescence suivante :

- [PARENT]
  - hornetclient
  - yui
    - yui
    - gallery

### 3.2.2.2 *hornetclient*

Dans le répertoire « [PARENT]/hornetclient », créer le répertoire qui correspond au numéro de version [HORNETCLIENT\_VERSION] ex : « 3.10.2 ».

- Arborescence des composants

Dans le répertoire [PARENT]/hornetclient/[HORNETCLIENT\_VERSION]/fwk, décompressez le contenu de l'archive

- hornetclient-fwk-X.X.X.zip

NB : Le composant « hornetconfig », à intégrer dans chaque page web, contient la configuration avec des chemins d'accès relatifs vers les répertoires des thèmes et de YUI.

- Arborescence des thèmes

Dans le répertoire [PARENT]/hornetclient/[HORNETCLIENT\_VERSION]/themes, décompressez le contenu des archives :

- hornetclient-themes-X.X.X.zip
- [THEME\_A\_NOM].zip
- [THEME\_B\_NOM].zip

### 3.2.2.3 *YUI (librairies)*

- YUI
  - Dans le répertoire « [PARENT]/yui/yui », le répertoire qui correspond au numéro de version [VERSION], ex : « 3.17.2 » doit être présent. Sinon, le créer.
  - Dans ce répertoire, décompressez le contenu de l'archive :
    - hornetclient-yui-X.X.X -yui.zip.
- gallery
  - Dans le répertoire « [PARENT]/yui/gallery », le répertoire qui correspond au numéro de version [VERSION], ex : « 2014.02.13-03-13 » doit être présent. Sinon, le créer.
  - Dans ce répertoire, décompressez le contenu de l'archive :
    - hornetclient-yui-X.X.X -gallery.zip.

## 3.2.3 Configuration du serveur Web

Pour profiter pleinement d'Hornet, il faut configurer une politique de cache très longue durée sur le serveur hébergeant les ressources hornetclient. Les entêtes HTTP à mettre en œuvre ont une expiration à long terme et un Cache-control public.

### 3.2.3.1 *Cross-Domain fonts*

Pour récupérer correctement les polices de caractères présentes sur le cdn, il est nécessaire de paramétrer le serveur web pour autoriser ce genre de requêtes.

Ex : configuration apache :

Activation du module « headers » :

- a2enmod headers

```
<FilesMatch ".(ttf|ttc|otf|eot|woff|font.css|css)">
  Header set Access-Control-Allow-Origin "*"
</FilesMatch>
```

---

**FIN DU DOCUMENT**