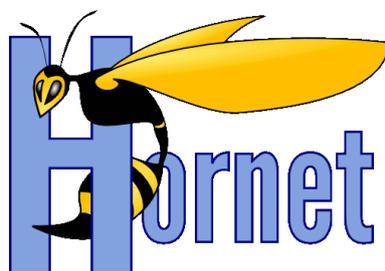




[Hornet] Mise en œuvre Hornet



Guide de déploiement du framework

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	A. Frigout	Initialisation	C. Rocheteau	01/03/12
2.0	A. Frigout F. Bernier- Malcoiffe	Modifications pour la version 2.0	S. Heurtematte	23/11/12
2.1	F. Bernier- Malcoiffe	Mise à jour du §2.1.2 Repository		04/03/12

DOCUMENTS DE REFERENCE

Version	Titre
1.0	Guide de création d'un projet Hornet

SOMMAIRE

SUIVI DES MODIFICATIONS	2
DOCUMENTS DE REFERENCE	2
SOMMAIRE.....	3
1 INTRODUCTION.....	4
2 CONSTRUCTION DES LIVRABLES	5
2.1 PRE-REQUIS.....	5
2.1.1 Logiciels.....	5
2.1.2 Repository Ivy.....	5
2.2 HORNETSERVER.....	5
2.2.1 Récupération des sources.....	5
2.2.2 Configuration Ivy.....	5
2.2.3 Construction des livrables.....	6
2.3 HORNETCLIENT.....	7
2.3.1 Récupération des sources.....	7
2.3.2 Configuration Ivy.....	7
2.3.3 Construction des livrables.....	7
2.4 HORNETTEMPLATE	8
2.4.1 Description.....	8
3 DEPLOIEMENT DES LIVRABLES	9
3.1 DEPLOIEMENT DE HORNETSERVER DANS UN PROJET HORNET	9
3.2 DEPLOIEMENT DE LA PARTIE CLIENTE	9
3.2.1 Schéma de l'arborescence.....	9
3.2.2 Création de l'arborescence.....	9
3.2.3 Configuration du serveur Web	10

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

- Eclipse Galiléo SR1 (v 3.5.1) et supérieur
- Ant 1.7.1 et supérieur

2.1.2 Repository Ivy

Le repository Hornet le plus récent, 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 <REPETOIRE REPOSITORY>.

Récupérer l'archive contenant le repository (de la forme « Repository_AAAAMMDD.zip »). Cette archive contient un répertoire « Repository_AAAAMMDD », qui contient lui-même les sous-répertoires « technique », « cache », « snapshot » et « metier ».

Extraire l'archive et copier le contenu du répertoire « Repository_AAAAMMDD » dans <REPETOIRE REPOSITORY>.

L'arborescence créée est la suivante :

- <REPETOIRE REPOSITORY>/technique
- <REPETOIRE REPOSITORY>/cache
- <REPETOIRE REPOSITORY>/snapshot
- <REPETOIRE REPOSITORY>/metier

2.2 hornetserver

Ce livrable constitue la partie serveur de Hornet, Il est constitué de hornetserver.

2.2.1 Récupération des sources

Dans Eclipse, veillez à désactiver au préalable l'option « Project > Build automatically ».

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

2.2.2 Configuration Ivy

Un environnement Ivy est inclus dans le projet Hornet. Vous devez utiliser cet environnement.

Le fichier « ivysettings.xml » permet de configurer :

- Localisation des repositories
 - fichier « **ivysettings.properties** » à la racine du projet
- Accès aux repositories
- Cache
- Statuts

ivysettings.xml :

```
<ivysettings>
  <properties file="ivysettings.properties" />
```

```
<settings defaultResolver="default" />

<cache default="repository-cache">
  <cache name="repository-cache" basedir="${repository.cache.url}"
    ivyPattern="${repository.cache.ivy.pattern}"
    artifactPattern="${repository.cache.artifact.pattern}"
    lockStrategy="no-lock">

  </cache>
</cache>

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

    <filesystem name="repository-technique" local="false" checkmodified="true" >
      <ivy pattern="${repository.technique.url}/${repository.technique.ivy.pattern}" />
      <artifact
pattern="${repository.technique.url}/${repository.technique.artifact.pattern}" />
    </filesystem>

  </chain>
</resolvers>

<statuses default="SNAPSHOT">
  <status name="SNAPSHOT" integration="false" />
  <status name="release" integration="true" />
  <status name="nightly" integration="true" />
</statuses>
</ivysettings>
```

Dans ivysettings.properties, modifiez les trois lignes suivantes pour indiquer le repository :

```
repository.technique.url=<REPERTOIRE REPOSITORY>/technique
...
repository.cache.url=<REPERTOIRE REPOSITORY>/cache
...
repository.snapshot.url=<REPERTOIRE REPOSITORY>/snapshot
```

2.2.3 Construction des livrables

- Pré-requis pour la génération JavaDoc :
 - o L'installation des outils de génération de graphiques (graphVIZ (graphviz-2.24.msi) pour la Javadoc.

Procédure (testée avec la version 2.24):

- Ouvrez l'exécutable « **graphviz-2.24.msi** »
- Suivre l'assistant en cliquant 3 fois sur « **next** », puis close.
- Lancez une invite de commande et tapez :

```
dot
```

- Puis valider, la commande doit être connue.

- Faire un « Project > Clean... » sur le projet avec l'option « Start a build immediately ». Ceci lancera la tâche « **generation-classpath** » du build.xml à la racine du projet.

L'option « Project > Build automatically » peut alors être ré-activée.

- Lancez la tâche « **livraison** » du build.xml à la racine du projet.
 - Les artefacts sont créés dans le dossier « **livrable** ».
 - Puis ils sont copiés dans le repository « **technique** » 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.

2.3 hornetclient

Ce livrable constitue la partie client riche de Hornet, il est constitué de hornetclient.

2.3.1 Récupération des sources

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

2.3.2 Configuration Ivy

Le fichier « **ivy.xml** » permet de définir les points suivants :

- Information du projet
- Plan de configuration
- Artefacts à publier
- Dépendances

Le dossier de configuration d'Ivy est « **ivysettings** ».

Le fichier « **ivy.xml** » s'appuie sur le fichier « **common.xml** » présent dans le dossier de configuration.

« **common.xml** » est utilisé par le build : tâche "getYUIAndTools"

« **common.xml** » est un script de gestion des dépendances du projet qui s'appuie sur les fichiers « **ivysettings.xml** » et « **ivysettings.properties** » présents dans le dossier de configuration.

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

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

Ivysettings.xml :

```
<ivysettings>
  <properties file="ivysettings.properties" />
  <settings defaultResolver="default" />

  <caches default="repository-cache">
    <cache name="repository-cache" basedir="${repository.cache.url}"
      ivyPattern="${repository.cache.ivy.pattern}"
      artifactPattern="${repository.cache.artifact.pattern}"
      lockStrategy="no-lock">

    </cache>
  </caches>

  <resolvers>
    <chain name="default" returnFirst="true">
      <filesystem name="repository-technique" local="false" checkmodified="true" >
        <ivy pattern="${repository.technique.url}/${repository.technique.ivy.pattern}" />
        <artifact
          pattern="${repository.technique.url}/${repository.technique.artifact.pattern}" />
        </filesystem>
      </chain>
    </resolvers>
  </ivysettings>
```

Dans **ivysettings.properties**, modifiez les deux lignes suivantes pour indiquer le repository :

```
repository.technique.url=<REPERTOIRE REPOSITORY>/technique
...
repository.cache.url=<REPERTOIRE REPOSITORY>/cache
...
```

2.3.3 Construction des livrables

Pour générer le livrable des composants et le livrable du thème par défaut :

- Lancez la tâche « **livraison** » du build.xml à la racine du projet.

Cette tâche automatise l'enchaînement de l'exécution des tâches 1, 2, 3, 4 et 5 décrites ci-après dans le même chapitre.

1. Tâche « **getYUIAndTools** » du build.xml à la racine du projet.
 - Les répertoires « **2in3** », « **builder** », « **gallery** » et « **yui3** » sont créés.
 - Le répertoire « **zip-yui** » contient les zips de ces répertoires.

2.3.3.1 Composants

2. Tâche « **all** » du build.xml du répertoire « **fwk** » :
 - Une fois l'exécution terminée, rafraîchir.
 - Pour chaque composant, un répertoire est créé dans « **fwk/build/hornet-NOM_COMPOSANT** ». Il contient les fichiers JavaScript en version
 - normale « **hornet-NOM_COMPOSANT.js** »
 - minimum « **hornet-NOM_COMPOSANT-min.js** »
 - debug « **hornet-NOM_COMPOSANT-debug.js** »

NB : Le composant « **hornetconfig** » sert à générer le fichier de configuration « **hornetconfig.js** »

3. Tâche « **livraison-fwk** » du build.xml à la racine du projet.
 - Une fois l'exécution terminée, rafraîchir.
 - Une archive **livrable-hornetclient-fwk-[HORNETCLIENT_VERSION]-[DATE].zip** est créée dans le répertoire « **livrable** ».

2.3.3.2 Thème par défaut

4. Tâche « **all** » du build.xml du répertoire « **themes** » :
Pour chaque thème, un répertoire est créé dans « **themes/build** ». Il contient les fichiers css en version

- normal « **hornet-skin-NOM_THEME.css** »
- minimum « **hornet-skin-NOM_THEME-min.css** »
- et le css principal dans les deux versions : « **theme.css** » et « **theme-min.css** ».

5. Tâche « **livraison-themes** » du build.xml à la racine du projet.
 - Une fois l'exécution terminée, rafraîchir.
 - Une archive **livrable-hornetclient-themes-[HORNETCLIENT_VERSION]-[DATE].zip** est créée dans le répertoire « **livrable** ».

Après exécution de la tâche livraison, rafraîchir le projet

Le répertoire livrable doit contenir les archives :

- **livrable-hornetclient-fwk-[HORNETCLIENT_VERSION]-[DATE].zip**
- **livrable-hornetclient-themes-[HORNETCLIENT_VERSION]-[DATE].zip**

2.4 hornettemplate

2.4.1 Description

Ce livrable constitue la partie génération de projet de Hornet.

Il est constitué des **éléments** suivants :

- **hornettemplate**

Se référer au « **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="2.0.0"
    conf="compile->core,libDependances;runtime->runtime;test->test"
    transitive="true" />
</dependencies>
```

3.2 Déploiement de la partie cliente

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-[HORNETTHEMEXXX_NOM]-[HORNETTHEMEXXX_VERSION]
 - hornet-skin-[HORNETTHEMEYYY_NOM]-[HORNETTHEMEYYY_VERSION]
 - yui
 - yui
 - [VERSION]
 - 2in3
 - [VERSION]
 - gallery
 - [VERSION]

Les arborescences • hornet-skin-[HORNETTHEMEXXX_NOM]-[HORNETTHEMEXXX_VERSION] et hornet-skin-[HORNETTHEMEYYY_NOM]-[HORNETTHEMEYYY_VERSION] sont des éventuelles arborescences de thèmes supplémentaires.

3.2.2 Création de l'arborescence

Les zips de hornetclient précédemment créés sont :

- livrable-hornetclient-fwk-[HORNETCLIENT_VERSION]-[DATE].zip
- livrable-hornetclient-themes-[HORNETCLIENT_VERSION]-[DATE].zip

Les 3 derniers zip « 2in3 », « gallery » et « yui », sont à récupérer dans le répertoire « **zip-yui** » :

- 2in3-[VERSION].zip
- gallery-[VERSION].zip
- yui-[VERSION].zip

3.2.2.1 Pré-requis

Avoir l'arborescence suivante :

- [PARENT]
 - hornetclient
 - yui
 - yui
 - 2in3
 - gallery

3.2.2.2 hornetclient

Dans le répertoire « **[PARENT]/hornetclient** », créez le répertoire qui correspond au numéro de version [HORNETCLIENT_VERSION] ex : « **2.0.0** ».

3.2.2.2.1 Arborescence des composants

- Dans le répertoire **[PARENT]/hornetclient/[HORNETCLIENT_VERSION]/fwk**, décompressez le contenu de l'archive :
 - **livrable-hornetclient-fwk-[HORNETCLIENT_VERSION]-[DATE].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.

3.2.2.2.2 Arborescence des thèmes

- Dans le répertoire **[PARENT]/hornetclient/[HORNETCLIENT_VERSION]/themes**, décompressez le contenu des archives
 - **livrable-hornetclient-themes-[HORNETCLIENT_VERSION]-[DATE].zip**
 - **livrable-[HORNETTHEMEXXX_NOM]-[HORNETTHEMEXXX_VERSION]-[DATE].zip**
 - **livrable-[HORNETTHEMEYYY_NOM]-[HORNETTHEMEYYY_VERSION]-[DATE].zip**

3.2.2.3 YUI (librairies)

3.2.2.3.1 YUI

- Dans le répertoire « **[PARENT]/yui/yui** », le répertoire qui correspond au numéro de version [VERSION], ex : « **3.4.1** » doit être présent. Sinon, le créer.
- Dans ce répertoire, décompressez le contenu de l'archive :
 - **yui-[VERSION].zip**.

3.2.2.3.2 gallery

- Dans le répertoire « **[PARENT]/yui/gallery** », le répertoire qui correspond au numéro de version [VERSION], ex : « **2011.06.29-23-18** » doit être présent. Sinon, le créer.
- Dans ce répertoire, décompressez le contenu de l'archive :
 - **gallery-[VERSION].zip**.

3.2.2.3.3 2in3

- Dans le répertoire « **[PARENT]/yui/2in3** », le répertoire qui correspond au numéro de version [VERSION], ex : « **2.9.0** » doit être présent. Sinon, le créer.
- Dans ce répertoire, décompressez le contenu de l'archive :
 - **2in3-[VERSION].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 sont une expiration à long terme et un Cache-control public.

FIN DU DOCUMENT