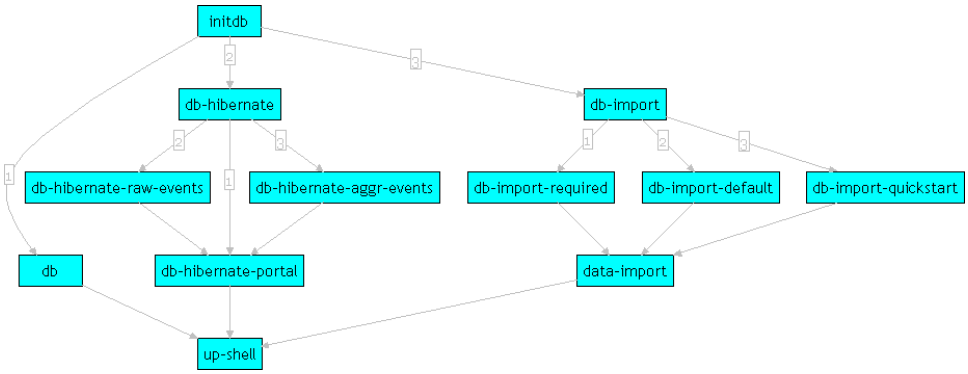
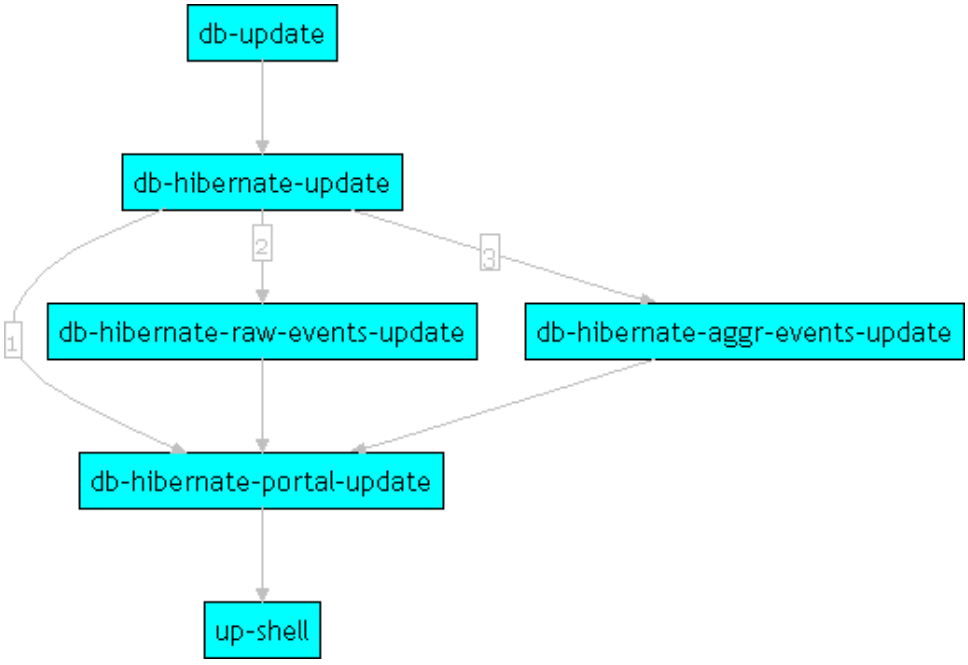
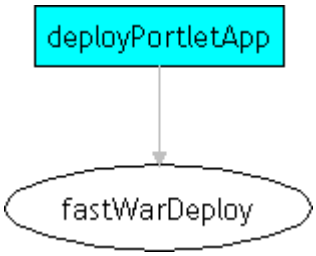
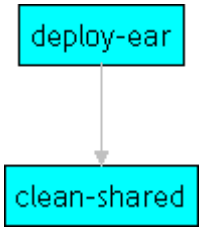
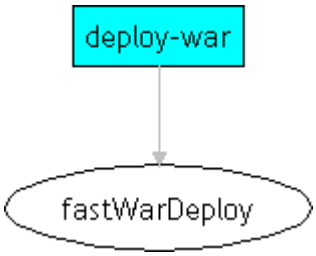
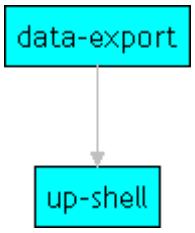
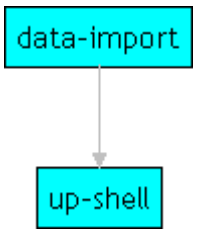
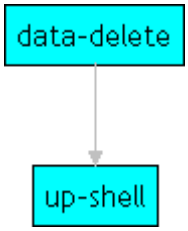
a) Architecture tâches ANT (esup 4)

- Les tâches en **gras** sont les tâches principales. Les tâches en gris sont des tâches secondaires ;
- Dans les schémas, les numéros représentent l'ordre d'appel.

Tâche ANT	Description sommaire
initportal	**Exécute toutes les tâches nécessaires au déploiement du portail et prépare la base de données du portail :** 1. Fait appel à la tâche "deploy-ear" 2. Fait appel à la tâche "initdb"
initdb	**Supprime toutes les tables et prépare la base de données du portail :** 1. Fait appel à la tâche "db" 2. Fait appel à la tâche "db-hibernate" 3. Fait appel à la tâche "db-import" 
db-update	**Met à jour une base de données existante pour passer à la version supérieure :** 1. Fait appel à la tâche "db-hibernate-update" 

db	**Charge certaines tables et données :** 1. Charge le fichier des tables "/properties/db/tables.xml" 2. Charge le fichier des données "/properties/db/data.xml" (Ces deux fichiers sont décrits ici) 3. À l'aide des deux fichiers précédemment chargés, créé un script de création de tables en faisant appel à la méthode "db" de la classe "po 4. Fait appel à la tâche "up-shell" avec comme paramètre le script précédemment créé Cette tâche et les taches db-* suivantes utilisent la classe portalShellBuildHelper décrite ici.
db-hibernate	**Supprime et créé les tables gérées par Hibernate :** 1. Fait appel à la tâche "db-hibernate-portal" 2. Fait appel à la tâche "db-hibernate-raw-events" 3. Fait appel à la tâche "db-hibernate-aggr-events"
db-hibernate-update	**Effectue les modifications de la base de données (dans le cas des montées de version du portail) pour le portail et les schémas d'évé** 1. Fait appel à la tâche "db-hibernate-portal-update" 2. Fait appel à la tâche "db-hibernate-raw-events-update" 3. Fait appel à la tâche "db-hibernate-aggr-events-update"
db-hibernate-portal	**Supprime et créé les tables gérées par Hibernate pour le portail :** 1. Créé un script de suppression de tables en faisant appel à la méthode "hibernateDrop" de la classe "portalShellBuildHelper" du package 2. Complète le script avec la création de tables en faisant appel à la méthode "hibernateCreate" de la classe "portalShellBuildHelper" du pac 3. Fait appel à la tâche "up-shell" avec comme paramètre le script précédemment créé
db-hibernate-raw-events	**Supprime et créé les tables gérées par Hibernate pour le stockage des évènements bruts (à usage statistique) du portail :** 1. Fait appel à la tâche "db-hibernate-portal" avec comme paramètre "databaseQualifier" la valeur "RawEventsDb"
db-hibernate-aggr-events	**Supprime et créé les tables gérées par Hibernate pour le stockage des évènements agrégés (à usage statistique) du portail :** 1. Fait appel à la tâche "db-hibernate-portal" avec comme paramètre "databaseQualifier" la valeur "AggrEventsDb"
db-hibernate-portal-update	**Effectue les changements demandés (sur la base de données du portail - lors des montées de version) pour le schéma du portail :** 1. Créé un script de modification de base en faisant appel à la méthode "hibernateUpdate" de la classe "portalShellBuildHelper" du package 2. Fait appel à la tâche "up-shell" avec comme paramètre le script précédemment créé
db-hibernate-raw-events-update	**Effectue les modifications de la base de données (dans le cas des montées de version du portail) pour le stockage des évènements l** 1. Fait appel à la tâche "db-hibernate-portal-update" avec comme paramètre "databaseQualifier" la valeur "RawEventsDb"
db-hibernate-aggr-events-update	**Effectue les modifications de la base de données (dans le cas des montées de version du portail) pour le stockage des évènements é** 1. Fait appel à la tâche "db-hibernate-portal-update" avec comme paramètre "databaseQualifier" la valeur "AggrEventsDb"
db-import	**Importe le contenu de fichier XML par défaut dans la base de données :** 1. Fait appel à la tâche "db-import-required" 2. Si la variable "noDefaultData" n'est pas valuée, fait appel à la tâche "db-import-default" 3. Si la variable "noQuickstartData" n'est pas valuée, fait appel à la tâche "db-import-quickstart" En fonction de la tâche, les fichiers d'un repertoire *uportal-war/src/main/data/***_entities* (cf. ci-dessous) seront importées en base de données de l'import. Ces données correspondent à des fichiers XML contenant les valeurs par défaut à entrer dans la base et nécessaires au bon déma Les fichiers XML des répertoires "entities" sont répartis dans des dossiers pour plus de lisibilité mais sont tous traités, et appellent des scripts C Il est possible de jouer sur les données importées en modifiant les fichiers XML ou directement les scripts Cernunnos pour modifier les donnée Pour plus d'informations sur le fonctionnement de Cernunnos : https://wiki.jasig.org/display/UPM30/Cernunnos+Overview http://code.google.com/p/cernunnos/
db-import-required	**Importe en base de données les entités requises :** 1. Fait appel à la tâche "data-import" avec comme répertoire des données le répertoire "uportal-war/src/main/data/required_entities" (décrit i

db-import-default	**Importe en base de données les entités par défaut :** 1. Fait appel à la tâche "data-import" avec comme répertoire des données le répertoire "default_entities.location" défini dans le fichier build.p
db-import-quickstart	**Importe en base de données les entités de "démarrage rapide" :** 1. Fait appel à la tâche "data-import" avec comme répertoire des données le répertoire "quickstart_entities.location" défini dans le fichier buil ci)
deployPortletApp	**Déploie le portlet (dans le conteneur de servlet) dont le .war est précisé en paramètre :** 1. Assemble le portlet à l'aide de la tâche "AssembleTask" de pluto 2. Fait appel à la tâche "fastWarDeploy" avec comme paramètre le portlet 
deploy-ear	**Déploie le portail, les librairies et les portlets dans le conteneur de servlet :** 1. Si demandé (cleanShared à true), fait appel à la tâche "clean-shared" 2. Ajoute "uportal-ant-tasks/pom.xml" à la liste des dépendances maven 3. Créé le répertoire "shared/lib" à la racine de tomcat 4. Déploie l'ear dans tomcat (à l'aide de la tâche précédemment chargée dans uportal-ant-tasks)  Plus d'informations ici.
clean-shared	**Supprime le contenu du répertoire "shared/lib" de tomcat (supprime les librairies partagées) :** 1. Purge le répertoire concerné
deploy-war	**Déploie la webapp uPortal dans le conteneur de servlet :** 1. Fait appel à la tâche "fastWarDeploy" avec comme paramètre le war de uPortal 
fastWar Deploy	**Déploie le .war en paramètre dans le conteneur de servlet :** 1. Si spécifié (removeExisting à true), supprime le .war existant 2. Copie le .war dans le conteneur de servlet (extractWars à false) ou extrait le .war à côté des autres webapp (extractWars à true). Note : Techniquement le traitement est différent sous Windows et sous Centos.

data-export	**Exporte <u>toutes</u> les données vers un fichier XML :** 1. Créé un script de sélection de données en faisant appel à la méthode "dataExport" de la classe "portalShellBuildHelper" du package 2. Fait appel à la tâche "up-shell" avec comme paramètre le script précédemment créé  **Exemple** : ant data-export -dir=some/export/directory
data-import	**Importe en base de données le ou les fichier(s) XML en paramètre :** 1. Créé un script d'insertion en base en faisant appel à la méthode "dataImport" de la classe "portalShellBuildHelper" du package 2. Fait appel à la tâche "up-shell" avec comme paramètre le script précédemment créé  **Exemple 1** : ant data-import -Ddir=some/directory **Exemple 2** : ant data-import -Dfile=some/path/to/xml
data-delete	**Supprime les données spécifiées (Type et Id en paramètre) :** 1. Créé un script de suppression de la base en faisant appel à la méthode "dataDelete" de la classe "portalShellBuildHelper" du package 2. Fait appel à la tâche "up-shell" avec comme paramètre le script précédemment créé  **Exemple** : ant data-delete -Dtype=MATABLE -Dsysid=MONIDTECHNIQUE
up-shell	**Exécute le script en paramètre :** 1. Exécute la classe "PortalShell" et donne le script en paramètre (délègue l'exécution du script)

Note sur la méthode de génération des schémas

```
Logiciel utilisé : grand-ui-0.7.2 (fichier en PJ => https://www.esup-portail.org/download/attachments/257949796
/grand-ui-0.7.2.7z?api=v2)
Il faut donner le fichier build.xml en entrée

Notes :
- Le logiciel ne fonctionne que sur machine 32 bits
- Il est préférable de travailler sur un copie du build.xml (afin de supprimer les "depends='prodPrompt'") afin
de ne pas polluer le schéma
```