



Agence ou Service : NTIC

Projet : Outil de Changement de Repère

MANUEL D'INSTALLATION DE L'OUTIL CHANGEMENT DE REPERE TREPS

Rédigé par : Benjamin Renard	Diffusé à : CNES / IRAP 
Approuvé par : Chef de projet AKKA – N. Lormant Responsable projet CNES – N. Dufourg	

LISTE DES MODIFICATIONS DU DOCUMENT

Vers.	Date	Paragraphe	Description de la modification
01.0	09/12/13		Création du document
01.1	20/01/14		Prise en compte des modifications pour la version de livraison 1.1
01.2	13/02/14		Ajout de prérequis Mise à jour de la configuration (en relation avec la livraison 1.2)
01.3	14/03/14		Ajout de prérequis Ajout de la configuration de l'application (avec les valeurs par défaut) Corrections à propos de la plateforme FitNesse Ajout de la procédure d'installation détaillée Ajout de la procédure de mise à jour de l'application Ajout de la procédure de mise à jour des repères
01.4	03/04/14		Prise en compte des retours du CRE
01.5	10/02/15		Ajout de la nouvelle option de configuration « treps.source.maxfilerrecords »

SOMMAIRE

1	INTRODUCTION	5
2	DESCRIPTION DU PACKAGE D'INSTALLATION.....	6
3	PRE-REQUIS	7
3.1	Système.....	7
3.2	Packages.....	7
3.2.1	Apache (pour l'hébergement de TREPS-IHM et de TREPS-Com)	7
3.2.2	PHP et PHP-XML (pour l'exécution des scripts de TREPS-Com).....	7
3.2.3	GCC (pour la compilation de TREPS-Kernel et des COTS).....	7
3.2.4	GCC-C++ (pour la compilation de TREPS-Kernel et des COTS).....	7
3.2.5	CMake (pour la compilation de TREPS-Kernel)	7
3.2.6	Flex (pour la compilation du COTS gsoap).....	7
3.2.7	Bison (pour la compilation du COTS gsoap)	8
3.2.8	libxml2 (pour la compilation de TREPS-Kernel)	8
3.2.9	wget (pour l'exécution de TREPS-Kernel)	8
3.2.10	Java Runtime Environment >= V1.6 (uniquement pour la plateforme de validation : exécution de Fitnessse et de Sonar)	8
3.2.11	MYSQL (uniquement pour la plateforme de validation : base de données Sonar)	8
4	PROCEDURE D'INSTALLATION RAPIDE.....	9
4.1	Configuration de l'installation	9
4.2	Configuration de l'installation pour une mise en production	9
4.3	Compilation et installation	10
4.4	Aperçu de la structure du répertoire d'installation.....	10
5	PROCEDURE D'INSTALLATION DETAILLEE	11
5.1	Configuration de l'installation	11
5.2	Création des répertoires d'installation.....	11
5.3	Installation de TREPS-IHM	11
5.4	Installation de TREPS-Com	12
5.5	Compilation et installation des COTS	12

5.6	Compilation de gsoap en mode DEBUG (facultatif).....	12
5.7	Compilation et installation de TREPS-Kernel.....	13
5.8	Compilation et installation du serveur CSLIM de TREPS-Kernel.....	14
6	POST-INSTALLATION	15
6.1	Configuration de l'application TREPS	15
6.1.1	Configuration de TREPS-Kernel	15
6.1.2	Mise à jour de la liste des repères disponibles	16
6.1.3	Configuration de TREPS-Com.....	16
6.1.4	Configuration de TREPS-IHM.....	17
6.1.5	Modification des pages d'aide.....	17
6.2	Configuration apache et ouverture des ports	18
6.3	Passage des tests unitaires depuis Fitness	19
6.3.1	Ouverture du port pour Fitness	19
6.3.2	Lancement du serveur de tests Fitness	19
6.3.3	Arrêt du serveur de tests Fitness	20
6.3.4	Execution des tests	20
6.4	Utilisation de la plateforme SONAR	20
6.4.1	Création de la base de données MySQL.....	20
6.4.2	Ouverture du port pour Sonar	21
6.4.3	Lancement du serveur Sonar.....	21
6.4.4	Arrêt du serveur Sonar.....	21
6.4.5	Génération d'un rapport	21
6.4.6	Visualisation des rapports.....	21
7	DESINSTALLATION DE L'APPLICATION	22
8	MISE A JOUR DE L'APPLICATION	23
9	DOCUMENTS APPLICABLES ET DE REFERENCE (A/R)	24
10	ABREVIATIONS.....	25
10.1	Abréviations	25

1 INTRODUCTION

L'objectif de ce document est de présenter la procédure d'installation de l'Outil de Changement de Repère, qui sera désigné par son acronyme TREPS dans la suite de ce document.

TREPS est un outil Web « léger » permettant à un utilisateur de définir, à travers une IHM ergonomique et intuitive, une opération de changement de repère sur des données qu'il fournit en entrée, dans le but d'obtenir des données de sortie correspondantes au résultat de sa transformation.

Le calcul à proprement parlé d'une transformation de changement de repère est assuré par l'appel au Web Service de l'outil CDPP/3DView.

2 DESCRIPTION DU PACKAGE D'INSTALLATION

Le package d'installation est une archive « .tgz » contenant l'ensemble des sources et des COTS nécessaires à l'installation de TREPS.

Cette archive contient un répertoire principal, contenant lui-même trois sous-répertoires :

- **COTS** : qui contient une liste d'archives « tar.gz », ainsi que le script de compilation et d'installation « **build_cots.sh** » des COTS,
- **ihm** : qui contient l'ensemble des sources de la sous-application TREPS-IHM (cf. [A1]), ainsi que des ressources liées à cette application,
- **server** : qui contient lui-même deux répertoires :
 - **kernel** : avec l'ensemble des sources de la sous-application TREPS-Kernel (cf. [A1]), ainsi que des ressources liées à cette application,
 - **php** : avec l'ensemble des sources de la sous-application TRESP-Com (cf. [A1])

Le répertoire principal contient aussi un script « **compilAndInstall.sh** », dont le rôle est de compiler l'application TREPS, ainsi que ses COTS, puis de procéder à son installation.

Dans la suite de ce document, nous considérons que le package d'installation a été décompressé. On désignera le répertoire principal du package décompressé par **<PACKAGE_DIR>**.

3 PRE-REQUIS

3.1 SYSTEME

La compilation et l'installation de l'application TREPS nécessite un serveur 64 Bits sous CentOS 6.3.

La taille du package d'installation est de 170 Mo. Une fois que celui-ci est décompressé, et à la fin de l'installation, il peut prendre une taille atteignant jusqu'à 1.1 Go (en fonction de la configuration de l'installation).

Le répertoire d'installation, quant-à-lui, peut atteindre une taille de 1 Go (en fonction de la configuration de l'installation).

Il est donc recommandé d'avoir **un espace disque disponible d'au moins 3 Go** pour installer l'application TREPS.

3.2 PACKAGES

La compilation de l'outil TREPS nécessite l'installation de certains packages des dépôts CentOS.

3.2.1 Apache (pour l'hébergement de TREPS-IHM et de TREPS-Com)

```
yum install httpd
```

3.2.2 PHP et PHP-XML (pour l'exécution des scripts de TREPS-Com)

```
yum install php  
yum install php-xml
```

3.2.3 GCC (pour la compilation de TREPS-Kernel et des COTS)

```
yum install gcc
```

3.2.4 GCC-C++ (pour la compilation de TREPS-Kernel et des COTS)

```
yum install gcc-c++
```

3.2.5 CMake (pour la compilation de TREPS-Kernel)

```
yum install cmake
```

3.2.6 Flex (pour la compilation du COTS gsoap)

```
yum install flex
```

3.2.7 Bison (pour la compilation du COTS gsoap)

```
yum install bison
```

3.2.8 libxml2 (pour la compilation de TREPS-Kernel)

```
yum install libxml2  
yum install libxml2-devel
```

3.2.9 wget (pour l'exécution de TREPS-Kernel)

```
yum install wget
```

3.2.10 Java Runtime Environment >= V1.6 (uniquement pour la plateforme de validation : exécution de Fitnessse et de Sonar)

```
yum install java-1.6.0-openjdk.x86_64
```

3.2.11 MYSQL (uniquement pour la plateforme de validation : base de données Sonar)

```
yum install mysql mysql-server  
chkconfig --levels 235 mysqld on  
/etc/init.d/mysqld start  
mysql_secure_installation
```


4 PROCEDURE D'INSTALLATION RAPIDE

La procédure d'installation rapide consiste en la compilation et l'installation de l'application TREPS, y compris ses COTS, à partir de l'exécution du seul script « `compilAndInstall.sh` ».

Cette procédure requiert toutefois une courte étape de configuration en préambule.

4.1 CONFIGURATION DE L'INSTALLATION

Dans le script « `compilAndInstall.sh` », trois variables sont à définir :

- **TREPS_INSTALL_DIR** : répertoire dans lequel l'installation de TREPS et de ses COTS va s'effectuer (noté `<INSTALL_DIR>` dans la suite du document),
- **TREPS_BUILD_DEBUG** : contient le type de build à effectuer (s'appliquant au noyau) :
 - **1** : pour compiler en mode Debug
 - **0** : pour compiler en mode Release
- **TREPS_VALIDATION** : précise si l'installation doit se faire avec la plateforme de validation
 - **1** : pour installer la plateforme de validation (Fitnessse, Sonar, et les différents COTS nécessaires)
 - **0** : pour se contenter de l'installation de l'application TREPS

Dans le script « `server/php/Constants.php` », deux variables sont à modifier :

- **TREPS_INSTALL_DIR** : doit faire référence au chemin complet vers le répertoire d'installation de TREPS (ie. Doit correspondre à « `<INSTALL_DIR>` »),
- **TREPS_DEBUG** : lorsque cette variable est à « `true` », les différentes réponses aux requêtes faites à TREPS-Com incluent le log généré par l'appel à TREPS-Kernel.

Dans le fichier « `server/kernel/CMakeLists.txt` », quatre variables sont modifiables :

- **TREPS_SONAR_PORT** : précise le port à utiliser pour le serveur Sonar (par défaut 8090)
- **TREPS_MYSQL_PORT** : précise le port à utiliser pour le serveur mySQL (par défaut 3306)
- **TREPS_FITNESSE_PORT** : précise le port à utiliser pour le serveur Fitnessse (par défaut 8081)
- **TREPS_3DVIEW_BASEDIR** : précise l'URL de base du Web Service 3DView à utiliser

4.2 CONFIGURATION DE L'INSTALLATION POUR UNE MISE EN PRODUCTION

Pour une mise en production de l'application, il est conseillé de suivre les recommandations suivantes :

- L'application doit être installée en mode Release,
- L'application doit être installée sans la plateforme de validation,
- La variable **TREPS_DEBUG** du script « `server/php/Constants.php` » doit être à false.

4.3 COMPILATION ET INSTALLATION

Une fois l'étape de configuration effectuée, il suffit d'exécuter le script « compilAndInstall.sh ».

Attention : cette commande **ne doit pas** être exécutée en mode super utilisateur.

```
cd <PACKAGE_DIR>
./compilAndInstall.sh
```

4.4 APERÇU DE LA STRUCTURE DU REPERTOIRE D'INSTALLATION

L'installation aura pour effet la création du répertoire <INSTALL_DIR>, contenant les trois répertoires :

- **bin** : contenant les binaires qui ont été compilés, répartis dans les deux répertoires :
 - COTS : contenant les binaires compilés des COTS
 - kernel : contenant le binaire compilé de l'application TREPS-Kernel
- **test** : contenant tout ce qui concerne les plateformes de test de l'application
- **www** : contenant l'ensemble des sources devant être gérées par un serveur Apache

Attention : Le répertoire <INSTALL_DIR> **ne doit pas être** déplacé après installation.

5 PROCEDURE D'INSTALLATION DETAILLEE

Cette partie explique les différentes étapes du processus d'installation de l'application TREPS (le script `compilAndInstall.sh`, utilisé dans la section 4, ne fait qu'appliquer ce processus automatiquement).

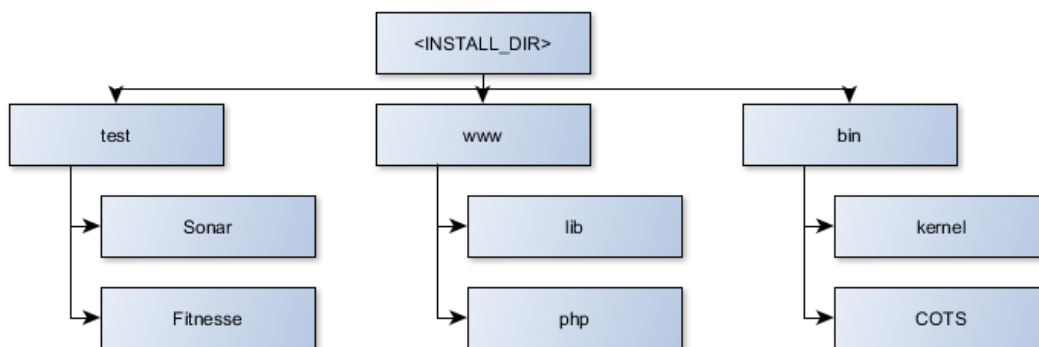
Elle présente également des options de configuration supplémentaires, pouvant être utiles pour un développeur.

5.1 CONFIGURATION DE L'INSTALLATION

La configuration de l'installation est identique à celle présentée lors de la procédure d'installation rapide (cf. 4.1).

5.2 CREATION DES REPERTOIRES D'INSTALLATION

La hiérarchie suivante de répertoires doit être construite afin de recevoir les différents éléments constituant l'application TREPS :



Configuration des droits d'écriture sur ces répertoires :

```
chmod -Rf 775 <INSTALL_DIR>
```

5.3 INSTALLATION DE TREPS-IHM

Installation de la librairie ExtJS :

```
cd <INSTALL_DIR>/www/lib
tar xzf <PACKAGE_DIR>/COTS/ext-4.2.1-gpl.tar.gz
```

Installation de la librairie sampjs :

```
cd <INSTALL_DIR>/www/lib
tar xzf <PACKAGE_DIR>/COTS/sampjs.tar.gz
```

Installation des sources de TREPS-IHM :

```
cp -Rf <PACKAGE_DIR>/ihm/* <INSTALL_DIR>/www/
```

5.4 INSTALLATION DE TREPS-COM

Installation des sources de TREPS-Com :

```
cp -Rf <PACKAGE_DIR>/server/php/* <INSTALL_DIR>/www/php/
```

Configuration des droits d'écriture sur le répertoire de travail de TREPS-Com :

```
chmod 775 <INSTALL_DIR>/www/php/data
```

5.5 COMPILATION ET INSTALLATION DES COTS

La compilation et l'installation des différents COTS de l'application est gérée par le script « <PACKAGE_DIR>/COTS/build_cots.sh ».

```
cd <PACKAGE_DIR>/COTS/  
export TREPS_COTS_INSTALL_DIR=<INSTALL_DIR>/bin/COTS/  
chmod 775 build_cots.sh  
./build_cots.sh 1
```

Pour ne pas installer les COTS relatifs à la plateforme de validation, remplacez la dernière ligne par :

```
./build_cots.sh 0
```

5.6 COMPILATION DE GSOAP EN MODE DEBUG (FACULTATIF)

Afin de suivre les contenus transmis et reçus par le client gsoap du Web Service CDPP/3DView, vous pouvez compiler gsoap en mode DEBUG.

Nous partons sur le principe que l'ensemble des COTS ont été compilés et installés tel que décrit en 5.5 auparavant.

```
cd <PACKAGE_DIR>/COTS/gsoap-2.8.17  
./configure --prefix=<INSTALL_DIR>/bin/COTS/gsoap --disable-ssl --enable-debug  
make clean  
make  
make install
```

Le client gSOAP de TREPS-Kernel générera alors trois fichiers de logs lors de son exécution :

- SENT.log : contenant les informations transmises au Web Service
- RECV.log : contenant les informations reçues du Web Service
- TEST.log : contenant différentes informations concernant l'activité du client

5.7 COMPILATION ET INSTALLATION DE TREPS-KERNEL

Quelques prérequis :

```
cd <PACKAGE_DIR>/server/kernel/  
chmod 775 scripts/generate3DViewSOAPClient.sh  
export TREPS_INSTALL_DIR=<INSTALL_DIR>
```

Si vous souhaitez installer les éléments de TREPS-Kernel relatifs à la plateforme de validation (valable que si vous avez compilé les COTS relatifs à la compilation):

```
export TREPS_VALIDATION=1
```

Dans le cas contraire :

```
export TREPS_VALIDATION=0
```

Puis ensuite, pour compiler le noyau en mode DEBUG :

```
cmake -G"Unix Makefiles" -DCMAKE_BUILD_TYPE=Debug  
make
```

ou bien, pour le compiler en mode RELEASE:

```
cmake -G"Unix Makefiles" -DCMAKE_BUILD_TYPE=Release  
make
```

Et pour installer TREPS-KERNEL:

```
make install CMAKE_INSTALL_ALWAYS=1
```

Enfin, pour configurer les droits d'écriture :

```
chmod 775 <INSTALL_DIR>/server/kernel/
```

5.8 COMPILATION ET INSTALLATION DU SERVEUR CSLIM DE TREPS-KERNEL

Cette compilation n'est valable que si vous avez auparavant compilé TREPS-Kernel avec les éléments relatifs à la plateforme de validation (cf. 5.7).

```
cd <PACKAGE_DIR>/server/kernel/test/Fitness/CSlimServer/  
cmake -G"Unix Makefiles" -DCMAKE_BUILD_TYPE=Debug  
make  
make install CMAKE_INSTALL_ALWAYS=1
```

6 POST-INSTALLATION

6.1 CONFIGURATION DE L'APPLICATION TREPS

6.1.1 Configuration de TREPS-Kernel

Le fichier de configuration de l'application TREPS-Kernel est un fichier ASCII éditable, que vous trouverez au chemin suivant :

```
<INSTALL_DIR>/bin/kernel/config/app.config
```

C'est un fichier ASCII de type clef=valeur suivi d'un retour chariot.

Le caractère # en première colonne du fichier indique que la ligne est un commentaire.

Voici les différentes options disponibles :

- `treps.cache.use` : 1 pour activer le système de cache, 0 sinon (attention ! le système de cache a été désactivé, cette option n'aura pour le moment aucune influence sur l'application),
- `treps.tmp.dir` : répertoire (relatif au répertoire de l'application) dans lequel les répertoires temporaires des différentes opérations seront créés (par défaut « `tmp/` »),
- `treps.xsd.dir` : répertoire (relatif au répertoire de l'application) contenant les fichiers XSD (par défaut « `xsd/` »),
- `treps.data.dir` : répertoire (relatif au répertoire de l'application) contenant les fichiers de données (par défaut « `data/` »),
- `treps.data.frames` : nom du fichier de données de définition des repères (situé dans `treps.data.dir`) (par défaut « `frames.xml` »),
- `treps.data.times` : nom du fichier de données de définition des formats de temps (situé dans `treps.data.dir`) (par défaut « `times.xml` »),
- `treps.data.exports` : nom du fichier de données de définition des structures d'exportation (situé dans `treps.data.dir`) (par défaut « `exports.xml` »),
- `treps.data.formats` : nom du fichier de données de définition des formats de fichier (situé dans `treps.data.dir`) (par défaut « `formats.xml` »),
- `treps.xsd.request` : nom du fichier xsd servant à la validation des fichiers de requêtes (situé dans `treps.xsd.dir`) (par défaut « `request.xsd` »),
- `treps.proxy.username` (facultatif) : à configurer si l'application tourne derrière un proxy (en commentaire par défaut),
- `treps.proxy.password` (facultatif) : à configurer si l'application tourne derrière un proxy (en commentaire par défaut),

- `treps.log4cxx.config` : nom du fichier de configuration du logger (situé dans `treps.data.dir`) (par défaut « `log4cxx.config` »),
- `treps.log4cxx.logger` : nom du logger à utiliser (par défaut « `TREPS-Kernel` »),
- `treps.source.maxfilesize` : taille maximale en octets du fichier de données (par défaut 20 Mo),
- `treps.source.maxfilerecords` : nombre d'enregistrement maximal contenu par le fichier de données (par défaut 100000),
- `treps.ascii.headerchar` : caractère utilisé pour désigner une ligne de commentaire dans un fichier de données ASCII Tabulaire (par défaut « `#` »),
- `treps.frames.tags` : définition des mots clés désignant un repère (les différents mots clés sont séparés par une virgule) (par défaut « `COORDINATE_SYSTEM,FRAME_SYSTEM` »),
- `treps.tmp.lifetime` : temps de vie maximal (en secondes) pour un répertoire temporaire après lequel il est automatiquement supprimé (par défaut « `86400` »).

6.1.2 Mise à jour de la liste des repères disponibles

Un script permettant l'exécution d'une requête de mise à jour de la liste des repères est disponible :

```
cd <INSTALL_DIR>/bin/kernel/scripts  
  
./framesUpdateList.sh
```

Le fichier XML de base contenant la liste des repères (désigné par `treps.data.frames` dans le fichier de configuration de `TREPS-Kernel`, cf. 6.1.1) ne sera pas modifié.

La requête créera le fichier « `treps.data.frames.new` » (`treps.data.frames` représentant le nom du fichier de base) avec la liste des repères à jour.

Il est à noter que les repères devenus obsolètes ne sont pas supprimés automatiquement de la liste et sont simplement indiqués dans le log d'exécution de la requête (par exemple : « `Frame obsolete (no deleted) : MSO` »).

Dans l'éventualité où aucune modification n'est détectée, le fichier « `treps.data.frames.new` » ne sera pas généré.

Enfin, et après avoir complété manuellement le fichier généré « `treps.data.frames.new` », renommez le en « `treps.data.frames` » (`treps.data.frames` représentant le nom du fichier de base).

6.1.3 Configuration de TREPS-Com

Le fichier de configuration de `TREPS-Com` est situé au chemin `<INSTALL_DIR>/www/php/Constants.php`.

La seule et unique option configurable dans ce fichier est :

`TREPS_DEBUG` : `true` pour renvoyer le log d'exécution de `TREPS-Kernel` dans la réponse JSON à la requête (par défaut « `true` »).

6.1.4 Configuration de TREPS-IHM

Le fichier de configuration de TREPS-IHM est situé au chemin <INSTALL_DIR>/www/app/Constants.js.

Dans ce fichier, vous pouvez configurer :

- APP_TITLE : le titre de l'application, qui s'affiche dans le header (par défaut « TREPS - Coordinate Transformations »),
- APP_DESCRIPTION : une courte description de l'application (notamment transmise à un hub SAMP au moment de la connexion de TREPS-IHM) (par défaut « Space physics coordinate transformation tool by CDPP. »),
- CDPP_URL : l'URL vers le site du CDPP (par défaut « http://cdpp.eu/ »),
- PHP_DIR, DIRECT_API et SAMP_LIB : **ces options ne doivent pas être modifiées**, elles sont liées à l'installation de l'application TREPS,
- WS_DOC : chemin vers la documentation du Web Service CDPP/3DView, qui sera accessible depuis le module 'Help' de l'application (par défaut « ressources/doc/IMPEX-IF-1-2-GFI_00_04.pdf »),
- DROP_MAX_FILESIZE : la taille maximale, en octets, acceptable pour un fichier qui est « glissé / déposé » (par défaut « 2097152 »),
- GRID_BUFF_PAGESIZE : nombre d'enregistrements constituant une « page ». Il s'agit du nombre d'enregistrements lu par requête pour la visualisation sous forme de grille (par défaut « 400 »),
- GRID_BUFF_LEADINGZONE : nombre d'enregistrements supplémentaires à charger en cache pour la visualisation sous forme de grille (par défaut « 800 »),
- PLOT_MAX_POINTS : le nombre d'enregistrements à charger pour la visualisation sous forme de plots (les autres enregistrements sont accessibles via un module de navigation « Next », « Previous ») (par défaut « 50 »),
- VECTOR_COMP1_COLOR, VECTOR_COMP2_COLOR et VECTOR_COMP3_COLOR : couleurs utilisées pour représenter les composantes d'un vecteur dans la grille de visualisation des données source (par défaut « #FFCECE », « #D6F8DE » et « #CACAFF »),
- TIME_COLOR : couleur utilisée pour représenter le champ « temps » dans la grille de visualisation des données source (par défaut « #EEF093 »),
- VECTOR_COMP1_PLOT_COLOR, VECTOR_COMP2_PLOT_COLOR et VECTOR_COMP3_PLOT_COLOR : couleurs utilisées pour représenter les composantes d'un vecteur dans un plot (par défaut « #FF0000 », « #00FF00 » et « #0000FF »),

6.1.5 Modification des pages d'aide

Les pages d'aide de l'application TREPS-IHM sont présentent dans le répertoire <INSTALL_DIR>/www/help/.

Dans ce répertoire, le fichier « pages.xml » référence l'ensemble des pages d'aide disponibles. Chaque page est décrite de la manière suivante :

```
<page>

    <id>page_id</id>

    <title>page_title</title>

    <file>help/page_html</file>

</page>
```

Avec :

- page_id : un identifiant unique désignant la page,
- page_title : le titre de la page,
- help/page_html : le chemin (relatif au répertoire « <INSTALL_DIR>/www ») vers le fichier html du contenu de la page.

Le contenu d'une page est écrit au format html.

6.2 CONFIGURATION APACHE ET OUVERTURE DES PORTS

Un virtual host Apache doit être créé pour l'accès à l'IHM de l'application depuis un navigateur.

Ce virtual host doit pointer sur le répertoire « <INSTALL_DIR>/www ». Voici un exemple :

```
NameVirtualHost *:80

<VirtualHost *:80>

    ServerName treps.akka.eu

    DocumentRoot "<INSTALL_DIR>/www/"

    ErrorLog /var/log/httpd/treps_error_log

    <Directory "<INSTALL_DIR>/www">

        Order allow,deny

        Allow from all
```

```
</Directory>
```

```
</VirtualHost>
```

Le répertoire « <INSTALL_DIR> » doit être accessible pour l'utilisateur apache (noté <APACHE_USER>) et le groupe apache (noté <APACHE_GROUP>). Pour se faire, le plus simple est de modifier le propriétaire et le groupe de <INSTALL_DIR> :

```
sudo chown -Rf <APACHE_USER>:<APACHE_GROUP> <INSTALL_DIR>
```

Ensuite, il faut redémarrer le service httpd :

```
sudo service httpd restart
```

Enfin, il faut s'assurer de l'ouverture du port apache, dans « /etc/sysconfig/iptables ». En cas de modification de ce fichier, il faut redémarrer le service :

```
sudo service iptables restart
```

6.3 PASSAGE DES TESTS UNITAIRES DEPUIS FITNESS

6.3.1 Ouverture du port pour Fitness

Il faut s'assurer de l'ouverture du port défini pour Fitness lors de la configuration de l'installation (cf. 4.1), dans « /etc/sysconfig/iptables ». En cas de modification de ce fichier, il faut redémarrer le service :

```
sudo service iptables restart
```

6.3.2 Lancement du serveur de tests Fitness

Un script permet de lancer facilement le serveur de tests :

```
cd <INSTALL_DIR>/test/Fitness/scripts  
  
./fitness_start_server.sh
```

6.3.3 Arrêt du serveur de tests Fitnessse

```
ps -ef | grep fitnessse
```

Repérer l'id du process du serveur Fitnessse (noté PID ci-dessous), puis :

```
kill -9 PID
```

6.3.4 Execution des tests

6.3.4.1 En ligne de commande

Un script permet l'exécution des tests unitaires en ligne de commande:

```
cd <INSTALL_DIR>/test/Fitnessse/scripts  
  
./fitnessse_run_suite.sh
```

Ce script exécute les tests en utilisant Valgrind afin de générer une validation dynamique du code. Il est appelé lors de la génération d'un rapport Sonar.

6.3.4.2 Par navigateur

L'accès à la plateforme de test peut se faire via un navigateur, en indiquant le nom du serveur et le port utilisé (par exemple <http://treps.cesr.fr:8081>).

Cliquez sur :

- Tests sans Valgrind : pour exécuter les tests unitaires sans Valgrind
- Tests avec Valgrind : pour exécuter les tests unitaires avec Valgrind (validation dynamique du code)

Puis dans le menu à gauche, cliquez sur « Suite ».

6.4 UTILISATION DE LA PLATEFORME SONAR

6.4.1 Création de la base de données MySQL

Avant d'exécuter pour la première fois le serveur Sonar, il est nécessaire de créer la base de données MySQL qui contiendra l'ensemble des résultats des différents rapports générés.

Pour se faire :

```
cd <INSTALL_DIR>/test/Sonar/mysql  
  
mysql -u root -p < create_sonar_treps.sql
```

6.4.2 Ouverture du port pour Sonar

Il faut s'assurer de l'ouverture du port défini pour Sonar lors de la configuration de l'installation (cf. 4.1), dans « /etc/sysconfig/iptables ». En cas de modification de ce fichier, il faut redémarrer le service :

```
sudo service iptables restart
```

6.4.3 Lancement du serveur Sonar

```
cd <INSTALL_DIR>/test/Sonar/scripts  
  
./sonar_start_server.sh
```

6.4.4 Arrêt du serveur Sonar

```
cd <INSTALL_DIR>/test/Sonar/scripts  
  
./sonar_stop_server.sh
```

6.4.5 Génération d'un rapport

```
cd <INSTALL_DIR>/test/Sonar/scripts  
  
./sonar_create_report.sh
```

Lors de la génération d'un rapport Sonar :

- Plusieurs warning du type « cannot find the sources for ...» s'affichent, concernant les "include" vers un COTS ou vers une librairie C. Cela est normal, seul le code de l'application TREPS-Kernel doit être analysé.
- Plusieurs warning du type « Cannot find the rule class_X_Y, skipping violation» s'affichent. Ils concernent le code généré par gSOAP pour le client du Web Service CDPP/3DView, il n'est donc pas gênant que ces violations ne soient pas remontées par Sonar.

6.4.6 Visualisation des rapports

La visualisation des rapports Sonar se fait via un navigateur, en indiquant le nom du serveur et le port utilisé (par exemple <http://treps.cesr.fr:8090>).

7 DESINSTALLATION DE L'APPLICATION

Se référer à la section 6.3.3 pour l'arrêt du serveur Fitnessse.

Se référer à la section 6.4.4 pour l'arrêt du serveur Sonar.

Ensuite, il suffit de supprimer le répertoire d'installation de l'application :

```
sudo rm -Rf <INSTALL_DIR>
```

8 MISE A JOUR DE L'APPLICATION

Une mise à jour de l'application s'effectue en suivant les étapes suivantes :

- Réaliser une sauvegarde des différents fichiers de configuration listés dans 6.1 (la reconfiguration de l'application se fera manuellement),
- Désinstaller l'application TREPS en suivant la procédure décrite dans la section 7. Cette étape est facultative étant donné que plusieurs installations de l'outil peuvent cohabiter sur une même machine (en utilisant un <INSTALL_DIR> différent),
- Installer la nouvelle version de l'application TREPS, en suivant la procédure décrite dans la section 4,
- Réaliser la procédure de post-installation telle que décrite dans la section 6 (la configuration pourra se faire en s'aidant des fichiers sauvegardés lors de la première étape de cette procédure de mise à jour).

9 DOCUMENTS APPLICABLES ET DE REFERENCE (A/R)

A/R	Référence	Titre
[A1]	CDPP-CD-32100-482-SIL	Dossier de conception de l'outil de changement de repère TREPS

10 ABREVIATIONS

10.1 ABREVIATIONS

Abréviation	Nom détaillé
CDPP	Centre de Données de la Physique des Plasmas
CNES	Centre National d'Etudes Spatiales
IRAP	Institut de Recherche en Astrophysique et Planétologie
TREPS	Transformation de REpère pour la Physique Spatiale
SAMP	Simple Application Messaging Protocol