

Scarab - Guide d'administration

Scarab pour administrateurs systèmes

The Scarab Development Team

Scarab - Guide d'administration: Scarab pour administrateurs systèmes

The Scarab Development Team

Table of Contents

1. Installer Scarab	1
Généralités	1
Comment installer Scarab	1
Le schéma d'une installation type	1
Préparer l'installation	1
Choisir un système d'exploitation	1
Choisir un système de base de données	1
Choisir un moteur de servlet	3
Récupérer les autres éléments nécessaires à l'installation de Scarab	3
Comprendre et paramétrer l'authentification par NTLM	4
Configurer votre base de données pour Scarab	6
Installer et configurer MySQL	6
Configurer PostgreSQL	12
Préparer et configurer Oracle	13
Construire Scarab	14
Etape 1: Déballer la distribution de Scarab	14
Etape 2: Migration d'une ancienne version de Scarab (pre-19)	14
Etape 3: Configurer Scarab	14
Etape 4: Construire Scarab	15
Etape 5a: Créer la base de données	16
Etape 5b: Migrez vos données existantes	16
Etape 6: Démarrer Tomcat	17
2. Administration système de Scarab après l'installation	18
Sauvegarde et restauration	18
Sauvegarde des informations de Scarab	18
Restauration des informations de Scarab à partir d'une sauvegarde	19
Déplacer une installation de Scarab vers un autre serveur	19
Modifier les données de la base lors d'une modification de serveur	20
3. Utiliser Scarab avec Subversion	21

Chapter 1. Installer Scarab

Généralités

Comment installer Scarab

En l'état actuel du développement -- et peut-être également dans le moyen terme -- Scarab ne s'installe pas tout assemblé, avec par exemple un programme d'installation.

Au lieu de cela, Scarab est construit dynamiquement pour chaque installation particulière, en fonction des paramètres locaux mais surtout du système de gestion de base de données utilisé pour stocker les fiches et les autres informations dont Scarab a besoin.

Cette perspective peut vous inquiéter de prime abord mais la procédure est raisonnablement simple et bien éprouvée : vous devriez donc y parvenir sans difficultés, comme bien d'autres avant vous.

Le schéma d'une installation type

Une installation type se déroule en quatre étapes:

- Préparer l'installation.
- Installer et régler le système de gestion de base de données
- Construire Scarab
- Construire la base de données de Scarab.

Préparer l'installation

Choisir un système d'exploitation

Scarab bénéficie de la portabilité de Java. Il peut donc en théorie être installé sur une grande variété de systèmes.

En pratique, Scarab a été fréquemment déployé sur trois sortes de systèmes:

- Linux: cas le plus fréquent, sur lequel le plus d'expérience a été accumulé -- et donc logiquement le système sur lequel vous êtes le plus susceptible d'obtenir de l'aide de la part de la communauté des utilisateurs en cas de difficulté.
- Windows
- MacOS X: Scarab est d'ailleurs partiellement développé dans cet environnement et le premier démonstrateur, la "runbox", a tourné pendant des mois sur cette plate-forme.

Choisir un système de base de données

Scarab requiert un système de gestion de base de données pour y stocker les informations structurées:

paramétrage, requêtes, états, modules mais surtout les fiches. Il convient donc à ce stade de choisir (puis d'installer, voir au chapitre suivant) un SGBD.

Dans le principe, Scarab est utilisable avec tous les SGBD disposant d'un driver JDBC.

En pratique, le fonctionnement correct de Scarab est aussi conditionné par certaines questions de compatibilité SQL et surtout par la qualité du code généré par Torque, le composant de persistance et de mapping objet/relationnel associé au framework Turbine sur lequel est construit Scarab.

SGBD testés et supportés par Scarab

MySQL

C'est le SGBD sur lequel Scarab est développé et donc entièrement testé. Si vous n'avez pas de contrainte particulière qui influence votre décision, MySQL devrait être logiquement votre premier choix de SGBD pour faire tourner Scarab.

Quelle version choisir? Scarab fonctionnera également avec une version 3.x ou 4.x. Il est possible que le support de la version 3 soit un jour supprimé pour tirer profit des possibilités transactionnelles de la version 4. Ici encore, si vous n'avez pas de raison particulière qui guide votre choix, optez pour la sécurité à long terme et partez directement sur la version 4.

MySQL est disponible gratuitement et en open-source sur Linux et sur Windows.

PostgreSQL

Scarab a été testé avec succès sur les versions 7.2.1 et supérieures de PostgreSQL.

PostgreSQL est disponible gratuitement et en open-source sur Linux et sur Windows.

Oracle 9i

La compatibilité de Scarab avec Oracle 9i est testée.

En raison d'un problème lié à la gestion des BLOBs par Torque, la mise en oeuvre de Scarab sur Oracle 9i est un peu plus délicate que sur d'autres SGBD et nécessite de faire un petit patch du code généré.

SGBD testés et incomplètement supportés par Scarab

Les SGBD suivants peuvent être paramétrés dans Scarab mais les tests effectués montrent que Scarab n'est pas complètement opérationnel sur ces plate-formes :

- Microsoft SQL Server¹
- HSQLDB
- DB2

En outre :

- De premiers essais ont eu lieu récemment avec Apache Derby et ne sont pas concluants pour l'instant mais ils seront poursuivis.
- Oracle 8i a été supporté dans les premières versions de Scarab mais n'est plus supporté et ne le sera ~~probablement plus jamais~~².

¹Des problèmes subsistent au niveau de l'utilisation de la fonction SQL CONCAT et dans le mapping de certains types par Torque.

Choisir un moteur de servlet

Scarab est en principe compatible avec tout moteur de servlet compatible avec la norme Servlet 2.3 ; un patch permet théoriquement de le rendre compatible avec des moteurs de servlet compatibles avec le standard 2.2, notamment WebSphere Application Server 4.0 -- mais ceci n'a plus été testé depuis longtemps.

En pratique, la plupart des utilisateurs de Scarab le mettent en oeuvre avec un serveur Tomcat; d'ailleurs, le processus de construction de Scarab construit également un serveur Tomcat allégé, "the sandbox", que vous pouvez utiliser pour tester ou faire fonctionner Scarab dès qu'il est construit.

Vous pouvez également choisir de déployer Scarab derrière une installation existante de Tomcat; par ailleurs, des essais ont été également menés de manière satisfaisante sur Resin.

Le déploiement de Scarab sur des serveurs d'applications J2EE complets est assez délicat et rarement concluant; les essais menés sur JBoss3 et sur WebLogic Server n'ont pas abouti pour l'instant.

Récupérer les autres éléments nécessaires à l'installation de Scarab

Pour procéder à l'installation de Scarab, vous aurez encore besoin de trois choses: un SDK Java, le système de construction Ant et bien sûr Scarab lui-même.

Télécharger et installer un SDK Java

Si vous ne disposez pas d'un SDK Java (pré-installé sur certains systèmes, par exemple MacOS X), vous devez en installer un. Attention: vous avez besoin d'un SDK Java 2 [notez que l'appellation JDK est périmée] et pas seulement d'un JRE (Java Runtime Environment), car il vous faudra compiler Scarab.

La version du SDK doit être au moins 1.3.1.

Vous pouvez télécharger un SDK à partir de <http://www.java.com> ou <http://java.sun.com>.

Assurez-vous que la variable d'environnement JAVA_HOME pointe sur le répertoire dans lequel le SDK est installé.

Télécharger et installer Ant

Pour construire Scarab, vous aurez également besoin du logiciel de construction automatisée Ant.

Vous pouvez télécharger Ant à partir de <http://ant.apache.org>.

La version peut être 1.4 ou 1.5 (si vous vouliez recompiler la version de Torque embarquée dans Scarab, vous devriez utiliser une version 1.4; cette manoeuvre est de toute façon réservée aux développeurs patentés).

Assurez-vous que la variable d'environnement ANT_HOME pointe sur le répertoire dans lequel Ant est installé.

Caution

La version de ant fournie par RedHat avec RHEL3 (dont la dernière mise à jour est ant-1.5.2-23) lance une exception dans la cible create-db.

²Il s'agit d'un problème relatif aux *right outer joins*.

³Sur JBoss il s'agit probablement d'un problème de paramétrage de log4j.

L'exception est la suivante :

```
BUILD FAILED
file:/export/home/admin/scarab/target/scarab/WEB-INF/conf/runtime-torque.xml:72: taskdef class org.apache.torque.task.TorqueDocumentationTask cannot be found
```

Pour contourner ce problème, utilisez la version de ant fournie avec Scarab :

- Copiez l'archive de `$(SCARAB_ROOT)/www/repository/ant/jars/ant-1.5.3-1.jar` vers `$(SCARAB_ROOT)/build`
- Dans le répertoire `$(SCARAB_ROOT)/build` lancez la commande `java -jar ant-1.5.3-1.jar create-db`

Télécharger Scarab et préparer son installation

Il vous faut bien sûr finalement télécharger Scarab lui-même. Vous trouverez la dernière version (1.0-b21) à cette adresse: <http://scarab.tigris.org/servlets/ProjectDocumentList>.

Décompressez l'archive de Scarab dans le répertoire que vous choisirez.

L'essentiel de votre travail lors de l'installation portera dans un premier temps sur le répertoire `./build`.

Comprendre et paramétrer l'authentification par NTLM

L'authentification par NTLM permet de se loguer automatiquement (Single-Sign On) dans un domaine Windows NT. C'est une fonctionnalité qui n'est utile que dans des environnements bien contrôlés et administrés ; si vous ne pensez pas l'utiliser, sautez sans hésiter cette section.

Comprendre l'authentification par NTLM dans Scarab

Le développement de cette fonctionnalité a été entrepris pour permettre de se loguer automatiquement lorsqu'on travaille au sein d'un réseau Windows. Elle est basée sur du code fourni par Peter Nei (merci à lui) mais l'implémentation actuelle est pour l'essentiel basée sur JCIFS (voir à la fin de cette section).

L'authentification NTLM est implémentée sous forme d'une nouvelle valve (`NtlmLoginValve`) dans le pipeline, qui tente d'obtenir l'identification de l'utilisateur depuis le navigateur en utilisant le protocole de challenge NTLM.

Les éléments d'identité renvoyés par le navigateur ne sont pas acceptés directement, ils sont authentifiés dans le domaine NTLM.

Étant donné que cette fonctionnalité est spécifique à chaque navigateur (elle nécessite un peu de configuration dans Firefox, par exemple, comme on verra plus loin) et qu'elle n'a pas été testée sur tous les navigateurs, cela n'a probablement de sens de l'utiliser que dans des environnements réseau bien contrôlés.

Comportement

Le login par NTLM est désactivé par défaut ; il est activé en définissant deux propriétés Scarab :

```
scarab.login.ntlm.active=true|false
```

Positionner cette propriété à `true` nécessite de spécifier le domaine d'authentification.

```
scarab.login.ntlm.domain=mydomaincontroller
```

Lorsque l'utilisateur arrive dans Scarab, NTLMLoginValve va tenter d'obtenir ses éléments d'identification et va les valider dans le domaine NTLM. S'ils sont valides, NTLMLoginValve va tenter de loguer l'utilisateur dans Scarab. **Pour que cela fonctionne, le nom d'utilisateur NTLM doit correspondre à un nom d'utilisateur Scarab.** Si l'utilisateur peut être logué, un message est affiché, comme

```
You've been automatically logged in using your user 'YourUser'
```

Si les éléments d'identification fournis par NTLM ou le nom d'utilisateur Scarab correspondant ne sont pas valides, la valve "passe son tour" et le reste du système d'identification normal de Scarab se déroule (formulaire d'identification, login anonyme).

Lorsqu'un utilisateur authentifié (même Anonyme) clique sur le lien "Déconnexion", la session est terminée et l'écran de Login (Login.vm) est affiché pour permettre à un autre utilisateur (identifié par NTLM ou non) de tenter de se connecter au système avec d'autres identifiants.

Single-Sign-On

Un utilisateur peut toujours se connecter en utilisant le formulaire habituel ; ceci pose problème parce que les mots de passe Scarab et NTLM peuvent ne plus être synchronisés.

Solutions possibles :

1. Désactiver le login au moyen du formulaire habituel. Ceci pose problème si certains utilisateurs n'appartiennent pas au domaine Windows.
2. Utiliser une préférence utilisateur (UserPreference) pour choisir entre login NTLM et login par formulaire pour chaque utilisateur. Ceci restreint chaque utilisateur à un mode donné d'authentification, Scarab ou NTLM mais ajoute une tâche d'administration.

Ordre de précedence

Les différentes méthodes de login possibles sont sélectionnées dans cet ordre :

1. L'identification utilisant la page de login de Scarab a priorité sur toutes les autres..
2. Si le formulaire de login n'a pas été utilisé, le système essaiera l'authentification par NTLM (si elle est activée, bien sûr).
3. Un login anonyme sera accordé si c'est permis.

Firefox

Les utilisateurs de Firefox recevront par défaut une boîte de dialogue demandant un nom d'utilisateur et un mot de passe. Pour se loguer avec les éléments d'identification du domaine Windows, l'utilisateur doit répondre en entrant les données au format DOMAIN\User.

Exemple: Username: "MyDomain\mywindowsuser", Password: myplainpassword

Si les utilisateurs de Firefox veulent fournir les éléments d'identification du système d'exploitation Windows, ils doivent :

- Taper "about:config" dans la zone de saisie d'URL.
- Rechercher la clé de configuration 'network.automatic-ntlm-auth.trusted-uris'.
- Ajouter le nom du serveur Scarab à la valeur correspondante.

Autres navigateurs

Cette fonctionnalité n'a pas été testée avec d'autres navigateurs. Les autres navigateurs présentent probablement à l'utilisateur une boîte de dialogue demandant un nom d'utilisateur et un mot de passe mais cela peut varier.

Préparer le build de Scarab avec l'authentification par NTLM

Il y a une petite manipulation à faire sur les bibliothèques (archives) avant de construire Scarab ; c'est pourquoi cette longue discussion sur l'authentification par NTLM a pris place ici.

jcifs

Pour implémenter l'authentification par NTLM, Scarab utilise la bibliothèque java jcifs, qui fait partie du projet Samba (<http://jcifs.samba.org>).

La licence de jcifs est la LGPL, c'est pourquoi son jar ne peut pas être inclus dans la distribution de Scarab.

Certains projets Apache (Lenya) utilisent déjà jcifs, en fournissant un **jar factice** pour permettre la compilation et en demandant à l'administrateur de télécharger lui-même le jar réel. Cette méthode convient aussi pour Scarab, il vous faudra donc télécharger vous-même jcifs à l'une de ces adresses :

- <http://jcifs.samba.org/>
- <http://www.ibiblio.org/maven/org.samba.jcifs/jars/jcifs-1.2.6.jar>

Une fois que vous avez téléchargé le jar, vous devez remplacer le jar fourni avec Scarab. Attention, si vous utilisez Maven, il est possible que vous ayez déjà la version factice dans votre repository Maven local (normalement sous `$HOME/.maven/repository/org.samba.jcifs/jars`).

Warning

Vous devez remplacer le jar factice par le jar réel que vous aurez téléchargé.

Configurer votre base de données pour Scarab

Installer et configurer MySQL

Guide pour les distributions linux utilisant des RPM

Récupérer une distribution de MySQL

Vous trouverez une distribution appropriée sur le site web de MySQL [<http://www.mysql.com>].

Suivez les instructions de la documentation de MySQL. On ne donnera ici que des informations résumées.

Installer la distribution

Si, par exemple, vous avez téléchargé la version 4.0.22, vous pouvez installer MySQL comme suit :

```
rpm -i MySQL-server-4.0.22-0.i386.rpm MySQL-shared-4.0.22-0.i386.rpm
```

Initialiser et paramétrer votre serveur de base de données

Installez la base de données root d'abord et démarrez le démon mysql. Il semble que cette étape ne soit pas absolument requise, parce que l'installation des rpm's le fait déjà. Mais juste au cas où elle ne l'aurait pas fait, vous pouvez procéder manuellement :

```
mysql_install_db  
safe_mysqld
```

Configurer MySQL pour Scarab

La recette suivante détaille une manière d'éviter soigneusement les problèmes de connexion entre MySQL et Scarab, problèmes qui s'avèrent presque toujours relever de privilèges mal configurés. Supposons que vous avez une installation toute propre de MySQL sur votre système et que vous voulez exécuter MySQL et Scarab sur la même machine, à savoir localhost. Chaque étape est expliquée jusqu'à un certain niveau de détail. Il est possible que toutes les étapes ne soient pas nécessaires dans votre environnement.

Étape I (unix seulement) - Vérifier le fichier /etc/hosts

Si vous prévoyez d'exécuter MySQL sur localhost, vous devez vérifier l'entrée correspondante dans votre fichier /etc/hosts. Nous avons trouvé des problèmes lorsque l'entrée localhost est définie comme dans les exemples suivants :

```
127.0.0.1    localhost.localdomain localhost    (vu sur Debian)  
127.0.0.1    linux localhost      (vu sur Suse)
```

Si vous trouvez ce genre d'entrée, inversez l'ordre de sorte que "localhost" apparaisse en premier :

```
127.0.0.1    localhost localhost.localdomain (sur Debian)  
127.0.0.1    localhost linux              (sur Suse)
```

Il se peut que vous ayez à redémarrer la machine (mais ce n'est pas certain). Après le redémarrage, assurez-vous que le système d'exploitation n'a pas manipulé l'entrée de votre fichier /etc/hosts. Nous n'avons pas d'idée à vous suggérer si c'était le cas (mais si vous en aviez une, nous serions heureux de la connaître !).

Étape II - S'assurer que MySQL fonctionne correctement

Supposons que vous venez d'installer une instance de MySQL. MySQL est fourni avec un administrateur nommé "root", pour lequel il n'y a pas de mot de passe associé. Oui, c'est un gros problème de sécurité. Mais cela aide à démarrer sur MySQL rapidement. Nous reviendrons sur la sécurité à l'étape suivante.

Pour l'instant, essayez juste de lancer l'administration de la base. Vous pouvez faire cela en ligne de commande :

```
mysql -u root mysql
```

Vous devez maintenant avoir accès à la base de données d'administration de votre instance de MySQL. Si votre demande de connexion est rejetée, quelque chose d'inattendu a dû se produire. Soit votre base de données est corrompue, soit quelqu'un d'autre a installé la base et a paramétré la sécurité. Dans ce cas, vous devrez peut-être demander à votre administrateur de vous donner accès à la base de données "mysql" pour pouvoir passer à l'étape suivante.

Étape III - Sécuriser votre instance de MySQL (optionnel) (optional)

Une fois que vous êtes connecté(e) à la base de données, soit en tant que root, soit en tant qu'un administrateur (selon votre site), vous pouvez vérifier les droits :

```
select host, user, password from user;
```

```
+-----+-----+-----+
| host      | user  | password |
+-----+-----+-----+
| localhost | root  |          |
+-----+-----+-----+
1 row in set (0.01 sec)
```

Sécurisons maintenant l'instance. D'abord, nous voulons sécuriser l'utilisateur root. Nous lui donnons donc un mot de passe, ici "secret".

Note

Cette étape n'est PAS nécessaire pour faire fonctionner Scarab; si vous ne souhaitez pas sécuriser votre base de données maintenant, n'exécutez pas l'instruction suivante.

```
grant all on *.* to root@localhost identified by 'secret';
```

A partir de maintenant et pour toujours vous DEVEZ vous rappeler ce mot de passe! Voici à quoi ressemble maintenant la table user:

```
select host, user, password from user;
+-----+-----+-----+
| host      | user  | password |
+-----+-----+-----+
| localhost | root  | *14E65567ABDB5135D0CFD9A70B3032C179A49EE7 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Étape IV - Ajouter l'utilisateur administrateur de Scarab

Nous choisissons *admin* comme administrateur avec le mot de passe *secret* et nous prenons *scarab* comme nom de base de données. Mais vous pouvez bien sûr en choisir d'autres.

Note

Vous aurez besoin plus tard de TOUS les noms et identifiants choisis ci-dessus car ils devront figurer dans le fichier `build.properties` de Scarab !

```
grant all on scarab.* to admin@localhost identified by 'secret';
grant grant option on scarab.* to admin@localhost;
grant all on mysql.* to admin@localhost;
```

Note

La seconde instruction est nécessaire parce que l'administrateur devra créer l'utilisateur associé à Scarab à l'exécution (voir plus bas) ! Le terme `grant grant option` ci-dessus est intentionnel, ce n'est pas une coquille ! La troisième instruction est nécessaire parce que l'utilisateur de la base *scarab* utilise un mot de passe pour s'identifier. Pour pouvoir définir les paramètres correspondants, l'administrateur doit avoir accès aux tables de contrôle contenues dans la base *mysql*.

Voici à quoi ressemble la table `user` maintenant:

```
select host, user, password from user;
+-----+-----+-----+
| host      | user  | password |
+-----+-----+-----+
| localhost | root  | *14E65567ABDB5135D0CFD9A70B3032C179A49EE7 |
| localhost | admin | *14E65567ABDB5135D0CFD9A70B3032C179A49EE7 |
+-----+-----+-----+
```

Étape V - Tester le paramétrage

Caution

Par sécurité, gardez le shell courant ouvert et NE QUITTEZ PAS la session `mysql`. Ça peut aider si quelque

chose se passe mal ;-)

OK, ouvrez maintenant un nouveau shell et essayez de créer une base de données à partir de l'utilisateur admin:

```
mysqladmin -u admin -h localhost -p create scarab
Enter password: *****
```

Vous avez maintenant créé votre base de données scarab, même si elle est vide pour l'instant. Si la base de données était déjà présente, vous auriez reçu un message d'erreur approprié. Dans ce cas, ne continuez pas le processus sans avoir vérifié d'où vient cette base !

Note

Cette étape n'est PAS requise par le paramétrage de Scarab mais procéder ainsi permet de s'assurer que les privilèges définis pour l'administrateur de Scarab sont corrects et fonctionnent comme prévu.

Étape VI - Configurer scarab

Copiez le fichier `minimal.properties` (ou `build.properties`) et nommez la copie `build.properties`. Vous devez maintenant définir un utilisateur de base de données pour votre instance. Prenons par exemple "scarab" comme nom d'utilisateur MySQL et "baracs" comme mot de passe. Ces deux propriétés sont utilisées maintenant en plus de celles de l'utilisateur d'administration définies à l'étape IV ci-dessus (gardez tout cela à l'esprit quelques instants).

Éditez maintenant `build.properties`. Pour les paramètres de la base de données, repérez les propriétés suivantes :

```
scarab.database.username=scarab           #changed from **GENERATED**
scarab.database.password=baracs           #changed from **GENERATED**
scarab.database.admin.username=admin      #changed from **GENERATED**
scarab.database.admin.password=secret    #changed from **GENERATED**
scarab.database.host=localhost            #unchanged
scarab.database.name=scarab                #unchanged
```

Juste pour être complets: il y a d'autres propriétés qui peuvent être utilisées mais d'habitude elles sont générées à partir des paramètres ci-dessus. Les deux seules propriétés auxquelles vous devriez faire attention sont :

```
scarab.database.type=mysql                #unchanged
scarab.database.port=**GENERATED**        #unchanged
```

La première vaut `mysql` par défaut et la seconde sera positionnée à 3306 (le port standard utilisé par MySQL pour accepter des connexions).

Mais vraiment, la manière la plus sûre de procéder est encore de tout reconstruire :

```
ant clean
ant
ant create-db
```

En cas de problème

Si ça ne marche toujours pas...

Voici deux liens supplémentaires vers des ressources qui peuvent vous aider à résoudre les problèmes de permissions :

- La [documentation MySQL](http://www.mysql.com/documentation/mysql/bychapter/manual_MySQL_Database_Administration.html#Access_denied) [http://www.mysql.com/documentation/mysql/bychapter/manual_MySQL_Database_Administration.html#Access_denied]
- sourceforge [http://sourceforge.net/docman/display_doc.php?docid=8968&group_id=15923]

En dernier recours (s'applique à certaines distributions linux)

Si ça ne marche toujours pas, vérifiez encore ceci ⁴. Le connecteur MySQL Connector/J ne peut communiquer avec MySQL qu'en utilisant TCP/IP, puisque Java ne supporte pas les Unix domain sockets. La communication TCP/IP avec MySQL peut être affectée si MySQL a été démarré avec l'option `--skip-networking` ou s'il y a un firewall.

Si MySQL est démarré avec l'option `--skip-networking` (le package Linux Debian, par exemple, procède ainsi), il vous faudra mettre en commentaire cette option dans le fichier `/etc/mysql/my.cnf` ou `/etc/my.cnf`. Le fichier `my.cnf` peut aussi se trouver dans le répertoire `data` de votre serveur MySQL ou n'importe où ailleurs (cela dépend de la manière dont MySQL a été compilé pour votre système). Les binaires créés par MySQL AB cherchent toujours ce fichier dans `/etc/my.cnf` et `[datadir]/my.cnf`. Si votre serveur MySQL est derrière un firewall, vous devrez aussi penser à configurer le firewall pour permettre les connexions TCP/IP depuis la machine qui héberge Scarab jusqu'au serveur MySQL sur le port sur lequel MySQL écoute (par défaut, 3306).

Configurer PostgreSQL

Suite à une bogue dans le framework de persistance que nous utilisons, Scarab ne fonctionnera pas tel quel avec PostgreSQL. Par chance, la solution de ce problème est simple, il suffit d'exécuter un script pour contourner le problème.

This script must be run by a user with sufficient privileges, which generally means the user that originally created and initialised the PostgreSQL database. Typically this user will be 'postgres'.

Ce script se trouve dans la distribution de Scarab sous `src/sql/postgresql/hack-db.sql`; il peut être exécuté de différentes manières, par exemple en ligne de commande :

```
postgres> psql scarab < hack-db.sql
```

⁴Merci à Josh Howe pour sa contribution.

où 'scarab' est le nom de la base de données de Scarab, et 'postgres' est l'utilisateur "système" PostgreSQL. Sinon, on peut aussi lancer cette commande à la console interactive psql :

```
scarab=> \i hack-db.sql
```

Préparer et configurer Oracle

Récupérer les drivers JDBC

Pour des raisons de licence, le jar contenant le pilote JDBC d'Oracle ne peut pas être inclus dans la distribution de Scarab.

Vous devez le récupérer vous-même dans votre distribution Oracle (ou depuis Oracle Technology Network: http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html). À moins d'avoir une raison impérative de faire autrement, prenez les tout derniers pilotes: ils sont compatibles avec d'anciennes versions d'Oracle et sont beaucoup plus performants que les anciennes versions.

Le jar dont vous avez besoin s'appelle probablement `classes12.jar` (pour les JDK 1.2 et 1.3) ou `ojdbc14.jar` (pour les SDK 1.4 et 1.5). Copiez ce jar dans le répertoire `$SCARAB/lib` (c'est-à-dire le répertoire lib directement à la racine de votre distribution Scarab. Dans une installation à neuf de Scarab, ce répertoire est normalement vide, à l'exception d'un fichier README. C'est normal, il est rempli au cours du processus de build.

Configurer l'accès à votre base Oracle

Dans le fichier `build.properties` que vous allez constituer à l'étape 3 ci-dessous, vous devrez configurer l'accès à votre base de données Oracle.

```
scarab.database.host=localhost
scarab.database.port=
scarab.database.url=
scarab.database.admin.url=
scarab.database.type=oracle
scarab.database.jdbc.driver=oracle.jdbc.driver.OracleDriver
scarab.database.username=scarab
scarab.database.password=secret
scarab.database.admin.username=scott
scarab.database.admin.password=tiger
```

Résoudre un problème relatif aux LOBs

Pour utiliser des Large Objects (LOBs), tels que des BLOBs (Binary Large Objects) ou des CLOBs (Character Large Objects) il est nécessaire d'instancier des classes Oracle (pas les classes génériques JDBC) et la couche de persistance que Scarab utilise pour l'instant, Torque, ne le permet pas. Un moyen de contourner ce problème est d'utiliser des VARCHARs en lieu et place des LOBs. Pour cela, il vous faudra éditer le fichier `webapps/scarab/WEB-INF/sql/turbine.sql` après avoir construit Scarab (voir la section suivante), et remplacer `BLOB` par `VARCHAR2(4000)`

^sVous pouvez aussi éditer `webapps/scarab/WEB-INF/src/torque/templates/sql/base/oracle/db.props` et changer `VARIABLE = BLOB` en `VARIABLE = VARCHAR2(4000)` -- ça revient au même mais survit aux rebuilds.

Utiliser Oracle 10g XE sur la même machine Scarab

Juste un petit avertissement si vous comptez utiliser Scarab sur la même machine qu'Oracle 10g XE. Scarab et la web app d'administration d'Oracle écoutent par défaut sur le même port (8080), ce qui va provoquer un conflit.

Pour forcer Scarab à écouter sur un autre port, donnez à la propriété `scarab.http.port` une valeur de votre choix et éditez le fichier `./tomcat/conf/server.xml` pour configurer Tomcat et le faire écouter sur un autre port.

Pour changer le port sur lequel écoute la web app d'administration d'Oracle, consultez la documentation d'Oracle 10g XE.

Construire Scarab

Note

Cette section est une version formatée du fichier `INSTALL` situé à la racine de la distribution de Scarab.

Etape 1: Débaler la distribution de Scarab

Une distribution de Scarab peut être téléchargée comme archive `.zip` ou `.tar.gz`. Décompressez l'une ou l'autre de ces archives dans un répertoire pour lequel vous avez les droits d'écriture.

Etape 2: Migration d'une ancienne version de Scarab (pre-19)

Veuillez lire la section 2.1 du fichier `MIGRATION` pour plus d'instructions sur la migration de votre paramétrage actuel.

Si vous n'avez pas d'installation de Scarab antérieure à la version 19, ou si vous ne voulez pas garder vos paramètres et vos données, vous pouvez sauter cette étape.

Etape 3: Configurer Scarab

Si vous avez déjà un ancien fichier `build.properties` d'une version précédente de Scarab, copiez ce fichier `build.properties` à la racine (`$$SCARAB_ROOT`) de votre nouvelle installation (PAS dans le répertoire `build`).

Si vous n'avez pas un ancien fichier `build.properties`, créez-en un comme suit:

1. Allez dans le répertoire racine (`$$SCARAB_HOME`)
2. Copiez (ne renommez pas !) un des deux fichiers suivants comme `build.properties`:
`project.properties` (paramétrage complet)
`minimal.properties` (paramétrage minimal)

Choisissez `project.properties` comme point de départ si vous êtes un développeur très impliqué dans le développement de Scarab. Dans tous les autres cas, `minimal.properties` est un meilleur choix, croyez-nous ! La plupart d'entre vous feront donc ceci :

```
cd $SCARAB_HOME
```

```
cp minimal.properties build.properties
```

3. Parcourez votre fichier `build.properties` et suivez les instructions. Si vous avez vraiment besoin d'une propriété et qu'elle n'est pas disponible dans votre `build.properties`, ajoutez-la, c'est tout !

Franciser votre installation de Scarab

La langue de l'interface utilisateur de Scarab n'est pas définie par les paramètres d'installation ou de configuration mais par les préférences du navigateur de l'utilisateur qui accède à l'application web. Il n'y a donc rien de particulier à configurer ici pour que Scarab "parle" français.

Il y a par contre deux réglages que vous voudrez faire à ce stade.

Paramétrage du jeu de caractères utilisé pour le courrier électronique

Vous devez pour cela définir la propriété `scarab.email.encoding`, par exemple

```
scarab.email.encoding=ISO-8859-1
```

```
scarab.email.encoding=UTF-8
```

Le courrier électronique fonctionnera probablement sans ce réglage mais vous risquez de voir les noms d'utilisateur précédés de préfixes linguistiques particulièrement inesthétiques.

Définition de la langue par défaut

Vous devez pour cela définir les propriétés `scarab.locale.default.language` et `scarab.locale.default.country`

```
scarab.locale.default.language=fr
```

```
scarab.locale.default.country=FR
```

Ce paramétrage agit au moins sur la langue dans laquelle les données par défaut sont installées au moment de la création de la base.

Etape 4: Construire Scarab

Allez dans le répertoire `$SCARAB_ROOT/build`

Lancez Ant de la manière suivante :

```
ant
```

Note

Si vous êtes un développeur, en particulier un développeur de Scarab, vous pouvez aussi construire Scarab avec Maven 1.

Cette façon de faire vous donne plus d'options et de possibilités, par exemple la faculté de lancer des tests unitaires.

Pour plus de détails, vous pouvez vous référer au chapitre 3 du Guide du Développeur.

Etape 5a: Créer la base de données

Warning

SEULEMENT DANS LE CAS OU VOUS NE MIGREZ PAS UNE ANCIENNE VERSION DE SCARAB !

Lisez le fichier MIGRATION si vous voulez conserver vos données.

Vous pouvez maintenant créer et peupler la base de données de Scarab.

Exécutez une des commandes ci-dessous.

Si vous utilisez Ant :

```
ant create-db
```

ou

```
ant -Dskip.seed.data=true create-db
```

La seconde commande évite de peupler votre base de données avec les données d'exemple. Si vous voulez juste tester Scarab, nous vous conseillons d'utiliser la première option et de charger les données d'exemple dans votre base de données.

Des commandes similaires existent si vous utilisez Maven:

```
maven scarab:create-db
```

```
maven -Dskip.seed.data=true scarab:create-db
```

Gabarits JIRA

Par défaut, un module est créé avec des types de fiches correspondant à la structure de JIRA, ce qui vous permet de transférer des fiches de JIRA à Scarab. Si vous ne voulez pas ce module additionnel dans votre système, vous pouvez passer l'option `-Dskip.jira.templates.data=true` à Ant ou à Maven. Par exemple :

```
ant -Dskip.jira.templates.data=true create-db
```

or

```
maven -Dskip.jira.templates.data=true scarab:create-db
```

Note

Cette option est indépendante de `skip.seed.data`, vous pouvez donc spécifier l'une, l'autre ou les deux.

Etape 5b: Migrez vos données existantes

Note

SEULEMENT SI VOUS MIGREZ UNE INSTALLATION EXISTANTE DE SCARAB !!

Référez-vous au fichier MIGRATION pour des instructions complémentaires sur la migration de vos données existantes.

Etape 6: Démarrer Tomcat

Sur Windows, faites un double clic sur le fichier `$SCARAB_HOME/tomcat/bin/startup.bat`

Sur Unix/Linux, exécutez le script shell `$SCARAB_HOME/tomcat/bin/startup.sh`

L'application web sera alors disponible comme il a été spécifié en utilisant les propriétés suivantes (voir à l'étape 3) !:

- `scarab.http.scheme`
- `scarab.http.domain`
- `scarab.http.port`
- `scarab.context`

Scarab sera alors accessible avec un navigateur à l'URL `${scarab.http.scheme}://${scarab.http.domain}:${scarab.http.port}/${scarab.context}`

Si vous n'avez pas modifié les valeurs par défaut, vous trouverez donc Scarab à cette adresse: `http://localhost:8080/scarab`

Si votre paramétrage est correct, Scarab vous présentera la page de login après une redirection. Félicitations ! Scarab est installé et fonctionne sur votre système.

Si l'une des étapes ci-dessus échoue, lisez d'abord le fichier `README.txt` dans son intégralité. Si cela ne suffit pas à résoudre votre problème, inscrivez-vous à la liste de diffusion des utilisateurs de Scarab en envoyant un courriel à `users-subscribe@scarab.tigris.org` [<mailto:users-subscribe@scarab.tigris.org>].

Lorsque vous êtes abonné(e), envoyez un courriel à `users@scarab.tigris.org` [<mailto:users@scarab.tigris.org>] en expliquant précisément votre problème.

Chapter 2. Administration système de Scarab après l'installation

Sauvegarde et restauration

Pour remettre en état une installation de Scarab, en cas de défaillance du système par exemple, vous devez procéder à des sauvegardes régulières.

Pour cela, il vous faut bien sûr sauvegarder les données de la base de données Scarab mais aussi les répertoires qui contiennent les pièces jointes, les index utilisés pour la recherche plein texte avec Lucene et les objets sérialisés par Intake.

Sauvegarde des informations de Scarab

Sauvegarde de la base de données

MySQL

En ligne de commande, tapez par exemple:

```
mysqldump scarab > nom_de_fichier.sql
```

Si votre serveur MySQL requiert une authentification, il vous faudra passer les arguments correspondants dans la commande, par exemple:

```
mysqldump scarab --user=nom --password=mot_de_passe > nom_de_fichier.sql
```

Sauvegarde des autres informations (non stockées dans la base)

Les répertoires utilisés pour stocker les pièces jointes et les index Lucene ainsi que le fichier qui contient les objets sérialisés par Intake sont définis au moment de la construction de Scarab par les paramètres suivants, respectivement :

- `scarab.attachments.path`
- `scarab.lucene.index.path`
- `scarab.intake.serialize.file`

Il vous faut donc sauvegarder ces deux répertoires et ce fichier en même temps que les informations de la base de données.

Si vous n'avez pas redéfini les trois paramètres ci-dessus lorsque vous avez construit Scarab, tous ces objets se trouvent dans l'arborescence de fichiers de Scarab sur le serveur sous le répertoire `/WEB-INF`.

Sur Windows, le plus simple est probablement d'archiver ces deux répertoires et le fichier intake dans une archive zip.

Sur Linux, vous pouvez faire de même avec la commande suivante, par exemple :

```
tar zcf scarab.tgz attachments index intake-xml.ser
```

Si l'un ou plusieurs des trois paramètres a/ont été redéfini(s), il vous faudra déterminer la localisation des répertoires et du fichier correspondants pour pouvoir les archiver.

Restauration des informations de Scarab à partir d'une sauvegarde

Restauration de la base de données

MySQL

Il vous faut vous connecter à MySQL, par exemple :

```
mysql
```

ou en spécifiant votre nom d'utilisateur et votre mot de passe si votre serveur MySQL requiert une authentification :

```
mysql --user=nom --password=mot_de_passe
```

Le cas échéant, détruire si nécessaire la base de données Scarab corrompue :

```
drop database scarab;
```

Créer une nouvelle base de données Scarab et la sélectionner :

```
create database scarab;
```

```
use scarab;
```

Réimporter les données que vous aviez sauvegardées précédemment avec mysqldump :

```
source nom_de_fichier.sql;
```

Et voilà !

Sauvegarde des autres informations (non stockées dans la base)

Décompactez l'archive que vous avez constituée précédemment avec les pièces jointes, les index Lucene et les objets sérialisés par Intake dans les répertoires définis lors de la construction de Scarab.

Votre installation de Scarab est maintenant prête à repartir.

Déplacer une installation de Scarab vers un autre serveur

En cas de défaillance du serveur ou pour toute autre raison d'exploitation, vous pouvez être amené à déplacer une installation existante de Scarab sur un autre serveur.

A moins que le paramétrage et le nommage réseau de la machine destination ne soient rigoureusement identique à ceux de la machine source, vous devrez faire une nouvelle installation sur la machine cible en adaptant éventuellement certains paramètres de la configuration.

Si le nom réseau de la machine cible n'est pas celui de la machine de départ, c'est-à-dire si la nouvelle

installation ne répond pas à la même URL que l'installation d'origine, il vous faudra également faire deux modifications dans les données de la base.

Modifier les données de la base lors d'une modification de serveur

Modifier la table SCARAB_GLOBAL_PARAMETER

Si vous changez l'un des paramètres `scarab.http.domain`, `scarab.http.scheme`, `scarab.http.scriptname` ou `scarab.http.port` dans la nouvelle installation, vous devrez mettre à jour les entrées correspondantes de la table `SCARAB_GLOBAL_PARAMETER`. Par exemple :

```
update SCARAB_GLOBAL_PARAMETER set VALUE='newserver.example.com' where  
NAME='scarab.http.domain' ;
```

```
update SCARAB_GLOBAL_PARAMETER set VALUE='8081' where  
NAME='scarab.http.port' ;
```

Modifier la table SCARAB_MODULE

Vous pourrez avoir à modifier le nom de domaine dans la table `SCARAB_MODULE` :

```
update SCARAB_MODULE set DOMAIN='newserver.example.com' ;
```

Modifier la table SCARAB_ISSUE

Vous pourrez avoir à modifier le nom de domaine dans la table `SCARAB_ISSUE` :

```
update SCARAB_ISSUE set ID_DOMAIN='newserver.example.com' ;
```

Chapter 3. Utiliser Scarab avec Subversion

L'intégration d'un outil de suivi de faits techniques (Scarab) avec un outil de gestion de configuration (Subversion) est une quête du Graal pour la plupart des développeurs.

Il est possible de faire fonctionner ensemble ces deux outils.

En attendant que ce chapitre soit rédigé, vous pouvez vous référer à la documentation disponible à cette URL : <http://www.wever.org/java/space/java/Integrating+Subversion+and+Scarab>