



Ari'i

Solution Open Source

d'Automatisation de traitements informatiques

Table des matières

Introduction.....	6
Machine virtuelle.....	7
Contenu.....	7
Composants.....	7
Logiciels.....	8
Intégration dans le S.I.....	8
VM Player.....	8
ESX/ESXi.....	8
Installation en 10 points.....	9
Administration.....	12
Connexion.....	12
Adresse IP.....	13
Open Source JobScheduler.....	14
Arrêt/relance.....	14
Symfony.....	15
Configuration.....	15
Utilisateurs.....	16
Ajouter un utilisateur.....	16
Rôle de l'utilisateur.....	17
Promouvoir un utilisateur.....	17
Changer un mot de passe.....	18
Mise à jour.....	19
Par internet (GitHub).....	19
Par dépôt d'Archive (SourceForge).....	19
Intégration de moteurs externes.....	20
Connexion sur le supervisor.....	20
Déploiement d'objet.....	20
Redirection.....	22
Compatibilité.....	25
Système d'exploitation.....	25
Serveur Web.....	25
Base de données.....	25
PHP.....	25
Modules.....	25
Montée en version PRO.....	28
Ari'i.....	28
JobScheduler.....	28
Portail.....	29
Configuration.....	29
Ergonomie.....	30
Modules.....	30
Barre latérale.....	30
Barre d'outil.....	31
La date de référence.....	31
Rafraîchissement.....	31
Site local.....	31
Espace de travail.....	31
Icônes.....	33
Mes informations.....	34
Mon compte.....	34
Changement de mot de passe.....	34
Session.....	34
Installation silencieuse.....	36

Formulaire.....	36
Licence.....	36
Moteur.....	37
Système.....	38
Base de données.....	39
Notifications.....	40
Traitements.....	40
Génération du fichier.....	41
Téléchargement.....	41
Audit.....	42
Suivi d'exploitation.....	43
Ergonomie.....	43
Vue globale.....	43
Traitements.....	45
Informations.....	49
Barre d'outil.....	50
Export RSS.....	50
Menu contextuel.....	51
Actions sur le traitement.....	51
Lancer un traitement.....	51
Arrêter un traitement en cours.....	52
Stopper un traitement.....	52
Diagnostic.....	53
Historique des traitements.....	54
Ordres.....	54
Informations.....	57
Détail d'un ordre.....	57
Journal d'un ordre.....	57
Historique de l'ordre.....	57
Actions sur une chaîne de traitements.....	58
Ajouter un ordre.....	58
Bloquer une chaîne.....	59
Actions sur un ordre.....	60
Déclencher un ordre.....	60
Suspendre un ordre.....	60
Débloquer un ordre.....	60
Actions sur une étape.....	62
Bloquer une étape.....	62
Débloquer une étape.....	63
Sauter une étape.....	63
Réactiver une étape.....	64
Plan.....	65
Moteurs.....	66
Actions.....	66
Vérifier le moteur.....	67
Mise en pause.....	67
Mise en pause.....	67
Redémarrer le moteur.....	67
Evénements.....	68
Messages.....	69
Focus.....	70
Principe.....	70
Etat des moteurs.....	70
Base de données.....	70

Visualisation.....	71
Opérations.....	72
Configuration.....	72
Ergonomie.....	72
Rapport.....	73
Cartographie.....	74
Configuration.....	74
Utilisation.....	75
Répertoires.....	75
Lignes.....	77
Suivi d'exploitation.....	78
Ergonomie.....	78
Vue globale.....	78
Traitements.....	80
Informations.....	84
Barre d'outil.....	85
Export RSS.....	85
Menu contextuel.....	86
Actions sur le traitement.....	86
Lancer un traitement.....	86
Arrêter un traitement en cours.....	87
Stopper un traitement.....	87
Diagnostic.....	88
Historique des traitements.....	89
Ordres.....	89
Informations.....	92
Détail d'un ordre.....	92
Journal d'un ordre.....	92
Historique de l'ordre.....	92
Actions sur une chaîne de traitements.....	93
Ajouter un ordre.....	93
Bloquer une chaîne.....	94
Actions sur un ordre.....	95
Déclencher un ordre.....	95
Suspendre un ordre.....	95
Débloquer un ordre.....	95
Actions sur une étape.....	97
Bloquer une étape.....	97
Débloquer une étape.....	98
Sauter une étape.....	98
Réactiver une étape.....	99
Plan.....	100
Moteurs.....	101
Actions.....	101
Vérifier le moteur.....	102
Mise en pause.....	102
Mise en pause.....	102
Redémarrer le moteur.....	102
Événements.....	103
Messages.....	104
GIT.....	105
Navigation.....	105
Nouveau référentiel.....	106
Timecode.....	107

Développement.....	108
Nouveau Module.....	108
Intégration Ari'i.....	110
Routage.....	110
Contrôleur.....	110
Vue.....	111
Pour plus d'informations.....	112
Communauté.....	112
Support.....	112

Introduction

Ari'i signifie le chef en tahitien, son rôle est d'orchestrer les traitements informatiques et piloter des automates Open Source JobScheduler. Les traitements informatiques considérés sont généralement qualifié de « batchs » et regroupe un vaste ensemble de commandes telles que les transferts de fichiers, les requêtes de base de données, les appels de services web. Cette liste étant loin d'être exhaustive puisque toute commande informatique peut être automatisée et devenir un batch.

Cette interface web permet le pilotage de l'automate d'exploitation Open Source JobScheduler mais aussi les divers composants proposés dans le cadre d'une exploitation informatique utilisant des produits open source.

C'est un portail constitué de modules activés en fonction des besoins ou des profils d'utilisation. Il est conçu sur Symfony2 et respecte les normes de ce dernier, cela permet à tout développeur Symfony d'intégrer facilement de nouveaux modules.

L'interface est directement fournie dans une machine virtuelle contenant l'ensemble des outils nécessaires aux modules. Solutions Open Source Paris garantit l'intégrité du système en livrant systématiquement un système complet.

Machine virtuelle

L'utilisation de la machine virtuelle est le moyen le plus simple de démarrer avec l'automate d'exploitation et ses différents composants. Elle est pré-configurée permet une intégration rapide puisqu'elle doit simplement être téléchargée et ouverte avec VM player ou intégrée dans un ESX. La finalité est un ensemble cohérent de composants interagissant entre eux.

Contenu

La machine repose sur un serveur LAMP qui représente l'infrastructure la plus utilisée dans un environnement web, ce standard permet de bénéficier de compétences sur site. Le serveur Linux est une Debian 7 64bits, un serveur Apache2, une base de données MariaDB et PHP 5.4. Si vous utilisez d'autres standards vous pouvez vous reporter au chapitre Compatibilités.

Composants

La machine virtuelle contient différents logiciels composants la solution, on peut distinguer 2 couches principales qu'il est possible de dissocier:

- l'infrastructure d'ordonnancement basé sur Open Source JobScheduler
- le portail web Symfony2 (Apache/MariaDB/PHP)

Logiciels

Les logiciels utilisés sont en licence open source, librement distribuables dans le même mode de licence.
La limite

Composant	Description
MariaDB	Base de données
Nginx	Serveur web
PHP	Langage de script pour la partie Web
Perl	Langage de script pour la partie système
Symfony2	Framework PHP
DHTMLx	Framework Javascript
Java 1.7	Environnement Java
Open Source JobScheduler 1.8	Ordonnanceur open source
Graphviz	Outil de création graphique
Git	Gestionnaire de versions
NTP	Serveur de temps

Attention ! Certains composants ne sont pas Open Source et sont soumis à licence comme DHTMLX, cette machine ne peut donc pas être distribuée en GPL.

Intégration dans le S.I.

L'intégration dépend du système d'information, on peut identifier trois niveaux d'utilisation en fonction du volume et des technologies présentes dans le parc informatique :

- pour les plus petits parc, la machine peut être exécutée dans un « VM Player », cet outil gratuit peut être installé sur différents systèmes d'exploitation
- pour les parcs disposant d'un ESX, la machine peut être directement intégrée et profiter des bénéfices de cette architecture
- pour les parcs avec des normes strictes sur les composants et les méthodes d'installation, il faut vérifier la compatibilité et installer manuellement

VM Player

Vous pouvez tester la machine directement sur votre station de travail en utilisant VM Player que vous pourrez télécharger gratuitement à cette adresse :

https://my.vmware.com/web/vmware/free#desktop_end_user_computing/vmware_player/7_0

ESX/ESXi

Pour une utilisation en entreprise, il est préférable d'importer la machine virtuelle dans un ESX ou un

ESXi. Le format étant de type « hosted » (-sxx), il sera nécessaire de convertir la machine virtuelle en un format utilisable sur l'ESX.

En ligne de commande :

Chargement du module multiextent:

```
vmkload_mod multiextent
```

Conversion:

```
vmkfstools -i ./Arii64.vmdk Arii64_esx.vmdk -d thin
```

Suppression de l'ancien disque:

```
vmkfstools -U Arii64.vmdk
```

Renommage du nouveau avec l'ancien nom:

```
vmkfstools -E Arii64_esx.vmdk Arii64.vmdk
```

Déchargement du module multiextent:

```
vmkload_mod -u multiextent
```

On ne doit plus avoir qu'un petit fichier Arii64.vmdk et un gros (le disque) Arii64-flat.vmdk.

Par le VMware vCenter Converter Standalone :

Ce logiciel qui peut être téléchargé gratuitement à cette adresse

<http://www.vmware.com/products/converter>

permet de convertir une machine virtuelle pour l'importer sur l'ESX.

Installation en 10 points

Il est possible d'installer Ari'i sur une architecture web existante, la procédure se déroule comme suit :

- ✓ création d'un hôte virtuel Apache
- ✓ décompression de l'archive
- ✓ création de la base de données
- ✓ configuration du portail

Procédures :

- 1) Un serveur LAMP est disponible
- 2) Décompresser l'archive arii.zip
- 3) Créer un virtual host qui pointe sur le répertoire web
- 4) Configurer la base de données

- Editer le fichier app/config/parameters.yml

- Exemple Linux :

```
1. <VirtualHost *:80>
2.     ServerAdmin webmaster@localhost
3.
4.     DocumentRoot /home/arii/Symfony/web
5.     DirectoryIndex app.php
6.
7.     <Directory />
8.         Options FollowSymLinks
9.         AllowOverride None
10.    </Directory>
11.    <Directory /home/arii/Symfony/web/>
12.        Options Indexes FollowSymLinks MultiViews
13.        AllowOverride All
14.        Order allow,deny
15.        allow from all
16.    </Directory>
17.
18.    ErrorLog ${APACHE_LOG_DIR}/arii_error.log
19.
20.    # Possible values include: debug, info, notice, warn,
    error, crit,
21.    # alert, emerg.
22.    LogLevel warn
23.
24.    CustomLog ${APACHE_LOG_DIR}/arii_access.log combined
25. ServerAlias www.arii.org *.arii.org
26. </VirtualHost>
```

5) Tester l'url `http://<serveur>/app_dev.php`

6) Créer la base de données

```
php app/console doctrine:schema:create
ATTENTION: This operation should not be executed in a production
environment.
Creating database schema...
```

Database schema created successfully!

7) Créer un nouvel utilisateur

```
php app/console arii:user:create
Please choose a username:operator
Please choose an email:operator@sos-paris.com
Please choose a password:operator
Please choose a firstname:operator
Please choose a lastname:operator
Please choose an enterprise:sos-paris
Created user operator
```

8) Donner un rôle à l'utilisateur

```
php app/console fos:user:promote operator ROLE_OPERATOR
Role "ROLE_OPERATOR" has been added to user "operator".
```

9) Connexion

- Url `http://<serveur>/login`
- En tant qu'operator

10) Créer un filtre global

- Mon compte (en haut à droite)
- Onglet « Filtre »
- Bouton Nouveau
- Donner un titre
- Mettre * dans tous les champs

Administration

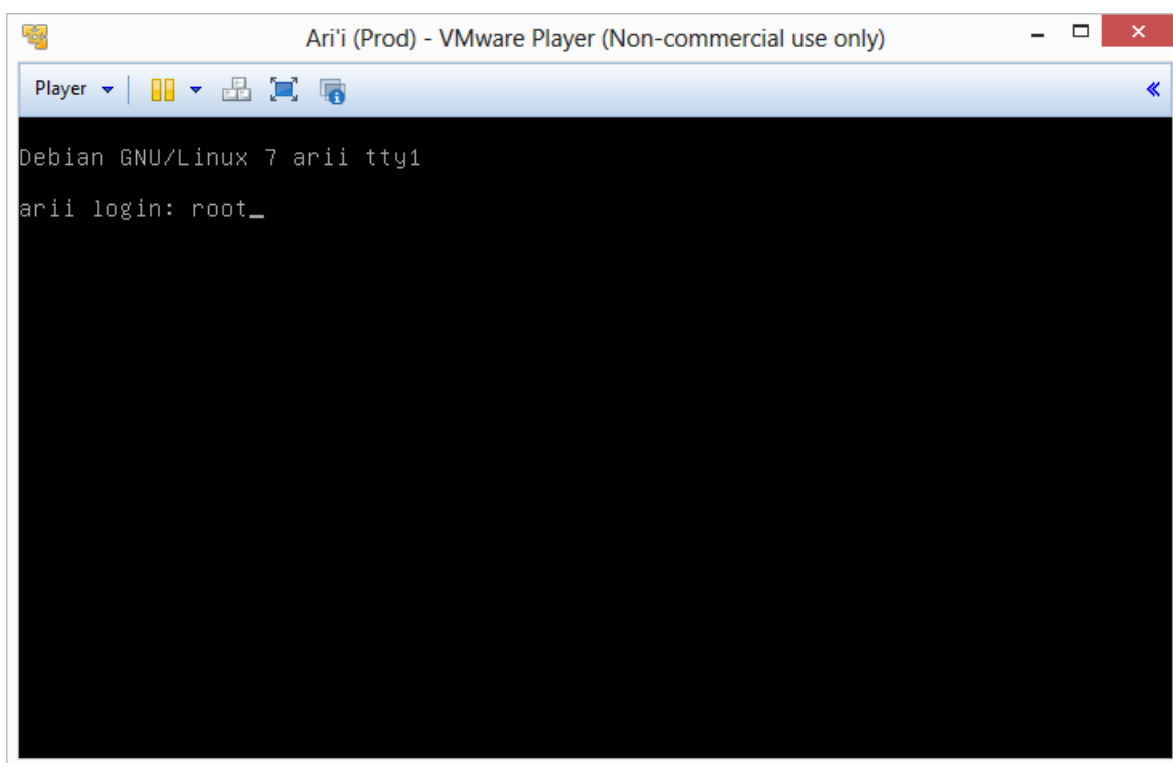
Ce chapitre fournit les procédures basiques pour l'administration de la machine virtuelle.

Connexion

Pour se connecter sur la machine, on dispose de différents comptes :

Login	Mot de passe	Description
root	root	Compte disposant de tous les droits
osjs	osjs	Compte utilisé pour la gestion de l'ordonnanceur
www-data		Compte pour la partie web

Il est préférable de changer les mots de passe, ce changement n'a pas d'impact sur la machine virtuelle.



Adresse IP

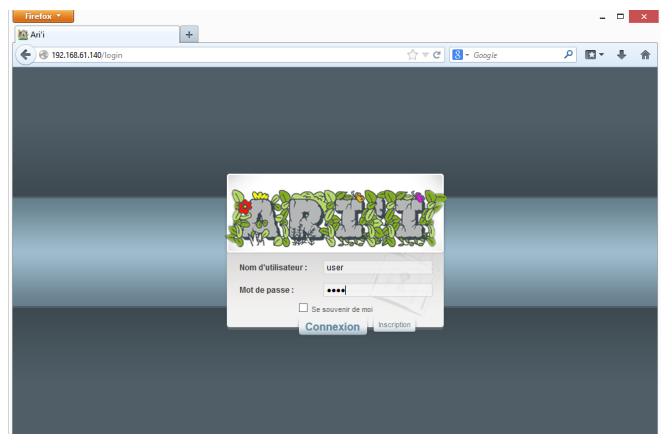
Si vous utilisez VM Player, la première chose est de connaître l'adresse IP utilisée pour la machine. Vous devez être connecté en root et taper la commande ifconfig.

```
root@arii:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:bb:1b:88
          inet adr:192.168.61.140  Bcast:192.168.61.255
Masque:255.255.255.0
          adr inet6: fe80::20c:29ff:febb:1b88/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:340529 errors:1 dropped:0 overruns:0 frame:0
          TX packets:348105 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:244483659 (233.1 MiB)  TX bytes:348854401 (332.6
MiB)
          Interruption:19 Adresse de base:0x2000

lo        Link encap:Boucle locale
          inet adr:127.0.0.1  Masque:255.0.0.0
          adr inet6: ::1/128 Scope:Hôte
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:133296 errors:0 dropped:0 overruns:0 frame:0
          TX packets:133296 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
          RX bytes:26737903 (25.4 MiB)  TX bytes:26737903 (25.4 MiB)
```

Cette adresse vous permet de vous connecter directement sur l'interface web avec l'adresse IP obtenue. Dans notre exemple :

<http://192.168.61.14>



Open Source JobScheduler

Le compte osjs doit être utilisé pour toutes les actions sur l'ordonnanceur.

JobScheduler est installé dans le répertoire /opt/jobscheduler/scheduler

Les fichiers utilisateurs sont dans /home/osjs/jobscheduler

Pour vérifier le bon fonctionnement de l'ordonnanceur, il faut se connecter sur l'url du serveur web embarqué : `http://<adresse IP>:4444`

Arrêt/relance

Pour Démarrer l'ordonnanceur :

- taper `/etc/init.d/jobscheduler start`
- indiquer le mot de passe root

```
osjs@arii:~$ /etc/init.d/jobscheduler start
Starting JobScheduler...
Mot de passe :
```

Pour l'arrêter : `/etc/init.d/jobscheduler stop`

```
osjs@arii:~$ /etc/init.d/jobscheduler stop

-----

Job Scheduler instance: scheduler
..... version: 1.6-SNAPSHOT
..... operated for: arii:4444
..... running since: 2014-02-27T03:02:27Z
..... state: running
..... cpu used:
..... job chains: 6
..... jobs: 20
..... orders: 3
..... remote instances: 0, connected instances:

-----

No jobs are running for this instance
Shutting down JobScheduler...
<?xml version="1.0" encoding="ISO-8859-1"?><spooler><answer
time="2014-02-28T02:04:57.465Z"><ok/></answer></spooler>
```

Symfony

Symfony est installé dans le répertoire /usr/share/symfony2

Le compte utilisé pour cette partie est www-data

```
root@arii:~# su - www-data
Pas de répertoire, connexion avec HOME=/
```

Configuration

L'interface est configurée pour dialoguer avec l'ordonnanceur de la machine virtuelle. Il est possible de modifier ces paramètres pour connecter l'interface sur une base de données et un JobScheduler existant afin de contrôler un installation existante.

Le fichier de paramètres est /usr/share/symfony2/app/config/parameters.yml

Paramètre	Description	Valeur par défaut
database_driver	Type de base de données	pdo_mysql
database_host	Hôte de la base	localhost
database_port	Port de connexion	3306
database_name	Nom de la base	arii
database_user	Utilisateur de connexion	root
database_password	Mot de passe	null
database_path	Répertoire	null
mailer_transport	Type de messagerie	smtp
mailer_host	Hôte de message	127.0.0.1
mailer_user	Utilisateur de connexion	null
mailer_password	Mot de passe	null
locale	Langue par défaut	fr

Pour une configuration plus complexe permettant de gérer de multiples sites, bases de données ou moteurs, il faut utiliser le module d'administration qui permet de définir son environnement technique.

Utilisateurs

La gestion des utilisateurs utilise les mécanismes Symfony et son mode console, il est suffisant pour ajouter rapidement quelques utilisateurs. Pour le traitement d'un plus grand volume de comptes, il est préférable d'utiliser le module d'administration.

Toutes commandes sont exécutées avec le compte www-data. Pour utiliser la console, il faut se déplacer dans le répertoire de Symfony puis préfixer les commandes par php app/console

```
$ cd /usr/share/symfony2
```

Pour avoir une liste des commandes disponibles, on peut indiquer un début de commande, si celle-ci est inconnue, la console affiche les arguments similaires :

```
$ php app/console arii:user
```

```
[InvalidArgumentException]  
Command "arii:user" is not defined.
```

```
Did you mean one of these?
```

```
arii:user:create
```

```
fos:user:promote
```

```
fos:user:activate
```

```
fos:user:change-password
```

```
fos:user:demote
```

```
fos:user:create
```

```
fos:user:deactivate
```

Ajouter un utilisateur

La commande arii:user:create permet d'ajouter un utilisateur en mode interactif ou en mode batch.

Pour le mode interactif, on indique la commande et on répond aux questions :

```
$ php app/console arii:user:create  
Please choose a username:eric  
Please choose an email:eric@sos-paris.com  
Please choose a password:eric  
Please choose a firstname:eric  
Please choose a lastname:ochet  
Please choose an enterprise:SOS Paris  
Created user eric
```

Le mode batch permet de passer l'ensemble des arguments afin de créer l'utilisateur en une seule ligne de

commande.

```
$ php app/console arii:user:create username mail@sos-paris.com
password prenom nom entreprise
Created user username
```

Rôle de l'utilisateur

Le rôle de l'utilisateur permet de modifier l'interface en fonction du profil. Le rôle est utilisé pour afficher ou cacher des options sur l'interface web.

Le rôle ne permet pas d'autoriser les actions sur les objets d'ordonnancement, pour réaliser cela il faut se reporter aux groupes et aux filtres du module d'administration.

Les rôles disponibles actuellement sont les suivants :

Rôle	Description
ROLE_READER	Le lecteur ne dispose d'aucun droit de création ou de modification. Il s'agit généralement d'un membre extérieur à l'équipe d'exploitation à qui on fournit une simple vue des traitements.
ROLE_OPERATOR	L'opérateur dispose des droits d'exécution pour le lancement ou les reprise des traitements.
ROLE_DEVELOPER	Le rôle du développeur est de concevoir les traitements et les scénarios d'exécution.
ROLE_USER	L'utilisateur conçoit et exécute les traitements pour lesquels il dispose des droits nécessaires.
ROLE_ADMIN	L'administrateur dispose de l'ensemble des droits fonctionnels sur l'exploitation. Il peut intervenir à tout niveau sur l'ordonnancement mais ne peut modifier l'infrastructure.
ROLE_MANAGER	Le manager gère l'infrastructure de l'exploitation et les groupes d'utilisateurs. Il définit les composants techniques comme les connections et les communications entre les composants.

Les rôles peuvent être hiérarchisés, par exemple un ADMIN bénéficier automatiquement des droit d'un opérateur qui est lui-même utilisateur.

Les rôles et les hiérarchies peuvent être définies suivant vos besoins en éditant le fichier `app/config/security.ym` :

```
role_hierarchy:
  ROLE_ADMIN:      [ROLE_USER, ROLE_OPERATOR, ROLE_DEVELOPPER]
  ROLE_MANAGER:    [ROLE_USER, ROLE_OPERATOR, ROLE_DEVELOPPER]
  ROLE_OPERATOR:   [ROLE_USER]
  ROLE_DEVELOPPER: [ROLE_USER]
```

Promouvoir un utilisateur

La commande `fos:user:promote` ajoute un rôle à un utilisateur :

```
$ php app/console fos:user:promote eric ROLE_ADMIN
Role "ROLE_ADMIN" has been added to user "eric".
```

Inversement, la commande fos:user:demote permet de le rétrograder :

```
$ php app/console fos:user:demote eric ROLE_ADMIN  
Role "ROLE_ADMIN" has been removed from user "eric".
```

Un utilisateur peut avoir plusieurs rôles, dans ce cas il faudrait le promouvoir avec autant de commandes. Il est inutile de spécifier les rôles d'une même hiérarchie, un utilisateur disposant du rôle ROLE_ADMIN a automatiquement le rôle OPERATOR et DEVELOPPER qui sont eux mêmes USER.

Changer un mot de passe

La commande fos:user:change-password permet de modifier le mot de passe d'un utilisateur :

```
$ php app/console fos:user:change-password eric new_password  
Changed password for user eric
```

Si un utilisateur souhaite changer son mot de passe, il doit utiliser le formulaire adéquat dans son espace de travail, pour plus d'informations il faut se reporter à la section Changement de mot de passe p34.

Mise à jour

Il existe différentes méthodes pour mettre à jour le site web en fonction des droits et des normes mise en place sur votre site.

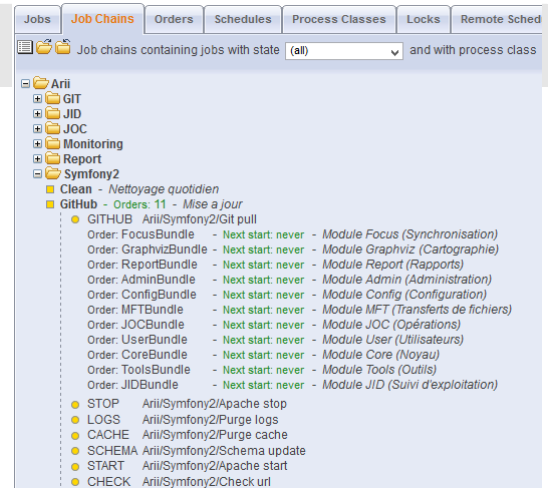
Par internet (GitHub)

La mise à jour est réalisée par une chaîne JobScheduler exécutée par le moteur Aarii. Les processus est une chaîne /Aarii/Symfony2/GitHub déclenchée par autant d'ordres qu'il existe de modules.

Chaque module peut être mis à jour à la demande.

La chaîne exécute les étapes suivantes :

- **Git pull** : récupération des sources sur Git Hub
- **Apache stop** : Arrêt du serveur web
- **Purge logs** : Suppression des fichiers dans app/logs
- **Purge cache** : Suppression des fichiers dans app/cache
- **Schema update** : Mise à jour du schéma de la base de données
- **Apache start** : Démarrage du serveur web
- **Check url** : Vérification du démarrage par appel de l'url



Par dépôt d'Archive (SourceForge)

SourceForge contient l'archive complète du site web, l'archive est à décompresser et à déposer dans un répertoire de publication.

Intégration de moteurs externes

Si Open Source JobScheduler est déjà déployé sur le site, on peut choisir de rapatrier ses traitements dans le hot folder du job scheduler de la machine virtuelle ou conserver le JobScheduler existant et le connecter à la machine virtuelle pour déployer de nouveaux traitements et sécuriser les moteurs distants.

Arii est un superviseur livré avec la machine virtuelle, si un supervisor est déjà installé il est préférable de ne conserver que celui qui est sur Arii car ses objets sont directement accessibles à travers l'interface. Sinon, il faudra partager le répertoire config du superviseur distant.

Connexion sur le supervisor

Pour connecter le moteur sur un superviseur, il faut ajouter modifier le fichier **scheduler.xml** et ajouter le nom ou l'adresse IP du superviseur ainsi que son port.

Dans notre, le superviseur écoute sur l'IP **192.168.61.130** et sur le port **44444** :

```
<spooler>

    <config mail_xslt_stylesheet = "config/scheduler_mail.xsl"
        port = "1801"
    supervisor="192.168.61.130:44444">
```

Après l'arrêt/relance du JobScheduler, on peut se connecter sur le superviseur (<http://192.168.61.130:44444>) pour voir apparaître la machine dans l'onglet « Remote Schedulers » du superviseur :

Jobs	Job Chains	Orders	Schedules	Process Classes	Locks	Remote Schedulers
2 Scheduler(s) (2 connected)						
IP		Hostname		Port	Id	Last Update
127.0.0.1		localhost			scheduler	2015-03-01 16:43:06
192.168.61.1					18RC1	2015-03-17 14:55:59

Déploiement d'objet

Dans notre exemple, le moteur distant apparaît avec l'adresse **192.168.61.1** et le port indiqué dans le fichier **scheduler.xml** est **1801**. Pour déployer des objets à partir du superviseur, il faut simplement créer un répertoire dédié dans **config/remote**.

On se connecte sur le serveur avec le compte arii et on vérifie que le moteur distant est accessible en utilisant, par exemple, une commande **wget** :

```
ariii@ariii64:/tmp$ wget http://192.168.61.1:1801
--2015-03-17 15:53:42-- http://192.168.61.1:1801/
Connexion vers 192.168.61.1:1801...connecté.
```

```

requête HTTP transmise, en attente de la réponse...200 OK
Longueur: non spécifié [text/html]
Sauvegarde en : «index.html»

[ <=> ] 7 764 --.-K/s
ds 0s

2015-03-17 15:53:42 (367 MB/s) - «index.html» sauvegardé [7764]

```

Si le superviseur ne peut pas se connecter, on devrait avoir un message de ce type :

```

arii@arii64:/tmp$ wget http://192.168.61.1:1802
--2015-03-17 15:55:03-- http://192.168.61.1:1802/
Connexion vers 192.168.61.1:1802...échec: Connexion refusée.

```

Dans ce cas, il faut vérifier :

- ✓ que l'adresse IP et le port sont corrects
- ✓ que le moteur distant n'est pas protégé par un parefeu
- ✓ qu'il autorise bien les connexions externes

Pour ce dernier point, l'adresse IP du superviseur peut être ajoutée dans le fichier scheduler.xml de la manière suivante :

```

<security ignore_unknown_hosts = "yes">
    <allowed_host host = "192.168.61.130" level = "all"/>
...
</security>

```

Pour ajouter un répertoire de déploiement sur cette machine en particulier, on va créer un répertoire avec les paramètres de connexions, adresse IP et port, séparés par un dièse (#) :

```

arii@arii64:~/jobscheduler/arii/config/remote$ mkdir
192.168.61.1#1801

```

Pour vérifier le bon fonctionnement, on crée un nouveau répertoire qui apparaîtra sur l'interface web du moteur distant :

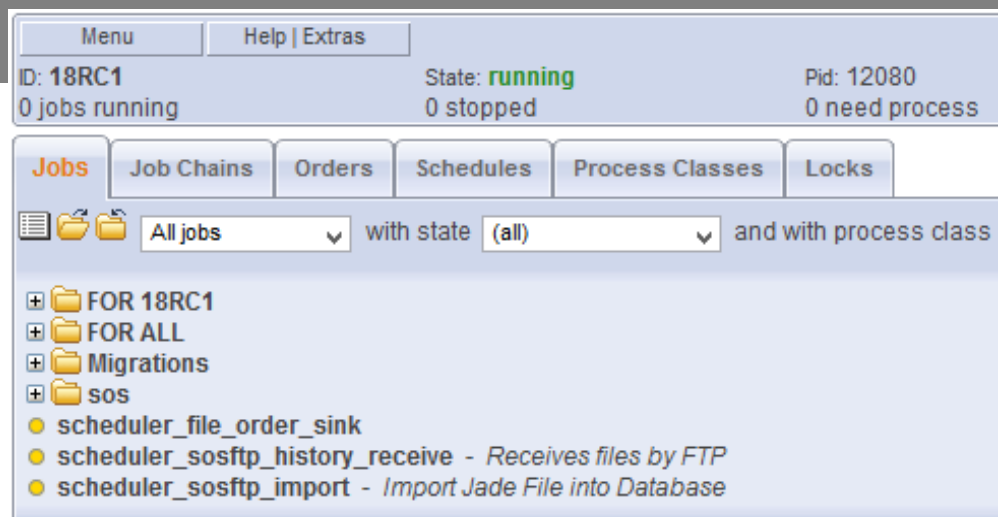
```

arii@arii64:~/jobscheduler/arii/config/remote/192.168.61.1#1801$
mkdir « FOR
18RC1 »

```

Si on se connecte sur le moteur distant, on peut voir qu'un répertoire « FOR 18RC1 » a été créé dans config/cache

On peut aussi vérifier le déploiement à travers _all



qui permet de déployer sur l'ensemble des abonnés de ce supervisor :

```
arii@arii64:~/jobscheduler/arii/config/remote$ cd _all
arii@arii64:~/jobscheduler/arii/config/remote/_all$ mkdir "FOR ALL"
```

Redirection

Pour simplifier les accès aux moteurs distants, on utilise la fonctionnalité proxy du serveur Apache pour offrir à l'utilisateur une url de type `http://<Arii>/js/<scheduler>/` qui évite d'avoir à connaître les adresses et le port de chaque moteur.

Pour ajouter une nouvelle adresse, on doit se connecter avec le compte root sur la machine virtuelle et aller dans la configuration apache :

```
arii@arii64:/tmp$ cd /etc/apache2/sites-enabled/
```

puis éditer le fichier arii dans lequel on ajoutera les 2 lignes suivantes :

```
ProxyPass /js/18RC1/ http://192.168.0.246:1801
ProxyPassReverse /js/18RC1/ http://192.168.0.246:1801/
```

On sauvegarde le fichier et on redémarre le serveur web :

```
root@arii64:/etc/apache2/sites-enabled# service apache2 restart
```

On peut ensuite vérifier l'url simplifiée : <http://192.168.61.130/js/18RC1/>

Si le moteur est une grille de calcul dans une configuration d'équilibrage de charges, on peut déclarer les moteurs de la manière suivante :

```
<Proxy balancer://grid/ >
BalancerMember http://127.0.0.1:5555/
BalancerMember http://127.0.0.1:5556/
BalancerMember http://127.0.0.1:5557/
</Proxy>
```

Puis indiquer le nom du groupe de machines :

```
ProxyPass /js/grid/ balancer://grid/
ProxyPassReverse /js/grid/ balancer://grid/
```

Si le moteur est en mode haute disponibilité, on indique les machines du groupe en précisant status=+H pour la machine qui doit être utilisée si et seulement si le serveur primaire ne répond plus :

```
<Proxy balancer://scheduler/ >
BalancerMember http://127.0.0.1:4444
BalancerMember http://127.0.0.1:4445 status=+H
</Proxy>
```

L'utilisation du « balancer » se fait de la même manière que pour la grille de calcul :

```
ProxyPass /js/scheduler/ balancer://scheduler/
ProxyPassReverse /js/scheduler/ balancer://scheduler/
```

Contenu du fichier `/etc/apache2/sites-enabled/arii` :

```
27. <Proxy balancer://grid/ >
28. BalancerMember http://127.0.0.1:5555/
29. BalancerMember http://127.0.0.1:5556/
30. BalancerMember http://127.0.0.1:5557/
31. </Proxy>
32.
33. <Proxy balancer://scheduler/ >
34. BalancerMember http://127.0.0.1:4444
35. BalancerMember http://127.0.0.1:4445 status=+H
36. </Proxy>
37.
38.
39. <VirtualHost *:80>
40.     ServerName arii64
41.     ServerAdmin webmaster@localhost
42.
43.     ProxyPass /js/18RC1/ http://192.168.0.246:1801/
44.     ProxyPassReverse /js/18RC1/ http://192.168.0.246:1801/
45.
46.     ProxyPass /js/arii/ http://127.0.0.1:44444/
47.     ProxyPassReverse /js/arii/ http://127.0.0.1:44444/
48.
49.     ProxyPass /js/scheduler/ balancer://scheduler/
50.     ProxyPassReverse /js/scheduler/ balancer://scheduler/
51.
52.     ProxyPass /js/backup/ http://127.0.0.1:4445/
53.     ProxyPassReverse /js/backup/ http://127.0.0.1:4445/
```


Compatibilité

Pour une intégration sur un Système d'Information, il peut être nécessaire de réinstaller les composants dans le respect des normes du site.

Système d'exploitation

Ari'i a été testé sur Windows, Linux Debian ou RedHat/Centos que ce soit en 32 ou 64bits.

Serveur Web

Il est possible d'utiliser Nginx au lieu d'Apache, il faudra se référer à la documentation Symfony2 pour l'installation du portail.

Aucun test n'a été effectué sur IIS, nous déconseillons donc ce serveur web.

Base de données

Cette base de données est un cache, elle peut être effacé à tout moment et reconstruite par les mécanismes Symfony2.

Le portail ne peut utiliser que MySQL ou MariaDB car les mécanismes de synchronisation utilisent les mécanismes d'auto-increment et de requêtes SQL multiples.

La base de données de l'automate d'exploitation peut utiliser MySQL, la validation sur Oracle est en cours.

PHP

La version utilisée est une 5.4.

Modules

<http://arii/arii/info.php>

```
arii@arii64:~/Symfony$ php app/check.php
*****
*
*   Symfony requirements check   *
*
*****
```

* Configuration file used by PHP: /etc/php5/cli/php.ini

**** ATTENTION ****

* The PHP CLI can use a different php.ini file

* than the one used with your web server.

* To be on the safe side, please also launch the requirements check

* from your web server using the web/config.php script.

**** Mandatory requirements ****

OK PHP version must be at least 5.3.3 (5.4.34-0+deb7u1 installed)

OK PHP version must not be 5.3.16 as Symfony won't work properly with it

OK Vendor libraries must be installed

OK app/cache/ directory must be writable

OK app/logs/ directory must be writable

OK date.timezone setting must be set

OK Configured default timezone "Europe/Paris" must be supported by your installation of PHP

OK json_encode() must be available

OK session_start() must be available

OK ctype_alpha() must be available

OK token_get_all() must be available

OK simplexml_import_dom() must be available

OK APC version must be at least 3.1.13 when using PHP 5.4

OK detect_unicode must be disabled in php.ini

OK PCRE extension must be available

**** Optional recommendations ****

OK Requirements file should be up-to-date

OK You should use at least PHP 5.3.4 due to PHP bug #52083 in earlier versions

OK When using annotations you should have at least PHP 5.3.8 due to PHP bug #55156

OK You should not use PHP 5.4.0 due to the PHP bug #61453

OK When using the logout handler from the Symfony Security Component, you should have at least PHP 5.4.11 due to PHP bug #63379 (as a workaround, you can also set invalidate_session to false in the security logout handler configuration)

OK You should use PHP 5.3.18+ or PHP 5.4.8+ to always get nice error messages for fatal errors in the development environment due to PHP bug #61767/#60909

```
OK      PCRE extension should be at least version 8.0 (8.3
installed)
OK      PHP-XML module should be installed
OK      mb_strlen() should be available
OK      iconv() should be available
OK      utf8_decode() should be available
OK      posix_isatty() should be available
OK      intl extension should be available
OK      intl extension should be correctly configured
OK      intl ICU version should be at least 4+
OK      a PHP accelerator should be installed
OK      short_open_tag should be disabled in php.ini
OK      magic_quotes_gpc should be disabled in php.ini
OK      register_globals should be disabled in php.ini
OK      session.auto_start should be disabled in php.ini
OK      PDO should be installed
OK      PDO should have some drivers installed (currently
available: mysql)
```

Montée en version PRO

Dans le cadre d'une production informatique, un contrat de support est généralement nécessaire. Le passage en version PRO permet de disposer d'un support et d'outils plus adaptés à une utilisation professionnelle.

Ari'i

La version PRO permet d'utiliser les composants commerciaux de DHTMLx tels que les grilles arborescentes (treegrid) ou la pagination. Il est nécessaire de disposer d'une licence DHTMLx pour utiliser Ari'i dans ce mode, si vous n'en disposez pas, vous pourrez en acquérir une directement chez l'éditeur ou dans le cadre d'un contrat support SOS Paris qui vous permettra d'utiliser notre licence multi-entreprises.

Attention ! Le passage en version pro signifie que votre installation n'est plus en mode

JobScheduler

SOS Berlin fournit un support en anglais qui peut être intéressant lorsqu'on a besoin de nouvelles évolutions. Pour les questions fonctionnelles, il est nécessaire de souscrire aux « service requests ».

Le contrat de support SOS Paris couvre le niveau 1 et 2 ainsi que toute demande fonctionnelle.

Portail

Le portail est le module de base, il permet de gérer l'ensemble des fonctions communes utilisées par les autres modules :

- l'authentification des utilisateurs
- la gestion des langues
- les mécanismes de session
- les accès à la base de données
- la gestion d'erreur et les audits
- l'accès aux autres modules

Configuration

La configuration du portail se limite à indiquer les modules publiés dans le paramètre arii_modules en précisant le rôle nécessaire pour le visualiser :

Paramètre	Valeur par défaut
arii_modules	JID(ROLE_USER),MFT(ROLE_USER),GVZ(ROLE_USER)

Les codes des modules sont les suivants :

Code	Module
JID	Suivi des traitements
JOC	Centre des opérations
MFT	Transferts de fichiers
GVZ	Cartographie
JOE	Edition des traitements
REP	Gestion des rapports
BAM	Suivi de processus
TC	Calcul de planifications complexes

Ergonomie

Le portail permet la navigation entre les modules, les principes de navigation sont repris par chacun d'eux pour s'inscrire dans une interface homogène.



Modules

Les modules sont accessibles à partir de l'écran principal, un système d'onglet permet de passer d'un module à l'autre sans passer par la page principale du page.

Un module est un programme à part entière, il peut être ajouté ou supprimer du portail par l'administrateur du site web.

Barre latérale

Chaque module dispose de la partie central du portail, cette partie est constituée de menus de liens sur la partie gauche et d'un écran principal sur la droite.



Barre d'outil




La barre d'outil du portail prend en charge les paramètres communs à l'ensemble des modules. Elle évolue en fonction du profil et de la complexité de l'environnement technique. Une barre d'outil minimum contient :

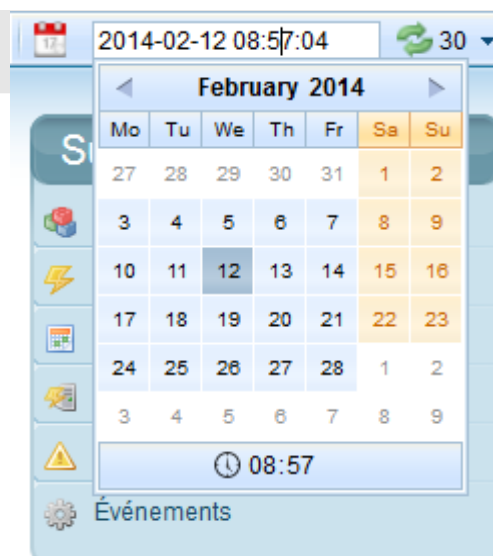
- La gestion de la date de référence
- Le délai de rafraîchissement
- Le site de l'utilisateur
- la gestion de l'espace de travail

La date de référence

La date de référence est la situation dans le temps, elle est particulièrement importante dans le cadre du suivi puisqu'elle permettra de se positionner dans le passé pour afficher les traitements exécutés les jours précédents ou au contraire de se projeter dans le futur pour afficher planification d'un jour donné.

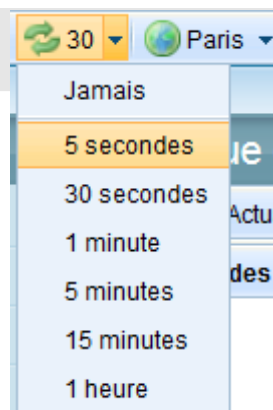
Pour modifier cette date, on peut la saisir dans le champ texte ou la sélectionner dans un calendrier.

Un clic sur l'icône du calendrier  positionne la date de référence à l'heure actuelle.



Rafraîchissement

Le rafraîchissement indique la fréquence de mise à jour des données sur l'écran. Les modules ne sollicitent le moteur que lors de demandes d'exécutions de commandes, une fréquence élevée n'aura donc pas d'incidence sur les performances de l'ordonnanceur.





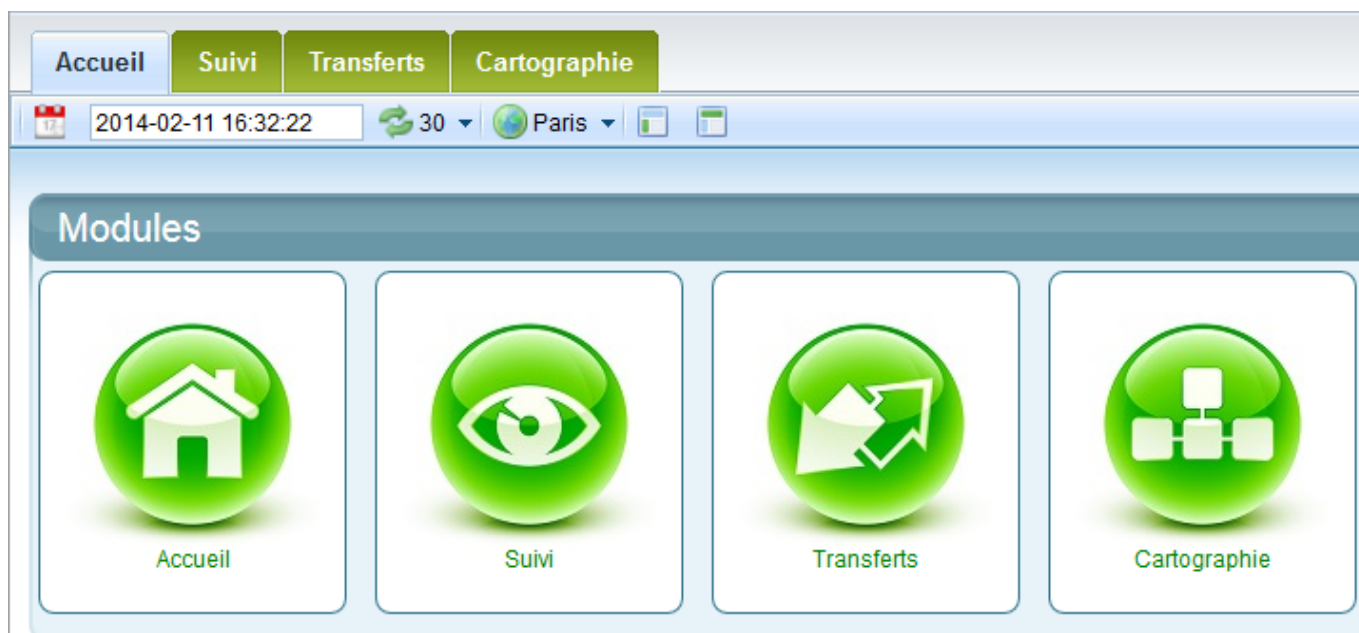
Site local

Le site permet d'indiquer le fuseau horaire du poste client pour afficher les dates à l'heure locale. Dans le cas d'entreprises multi-sites, la liste donne l'ensemble des sites de l'entreprise.

Espace de travail

Le module peut utiliser l'ensemble de l'écran en utilisant l'espace occupé par la barre latérale et par l'entête.

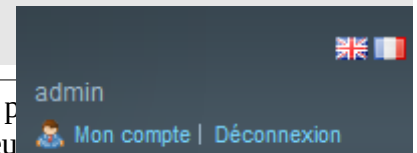
Enfoncer le bouton  montre la barre latérale, le bouton  affiche l'entête. Lorsque les entêtes et la barre latérale sont cachés, la navigation reste accessible par les onglets du portail passer d'un module à un autre ou par les onglets internes au module.












Pour utiliser votre écran de manière optimale et ceci quelque soit sa résolution, vous devez utiliser les fonctionnalités de votre navigateur. Pour afficher une plus grande quantité d'informations, vous pouvez utiliser la fonction zoom (touche Control + le signe + ou -).

La touche F11 permet de supprimer les menus du navigateur, libérant ainsi plus de place pour les données d'exploitation.

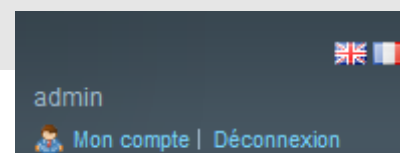
Icônes



	Ce symbole représente la définition d'un traitement constitué par des commandes complétées par les fonctions de l'ordonnanceur (heure d'exécution, machine, etc...).
	Une chaîne de traitement précise les étapes à suivre lors de l'exécution des traitements. En cas de succès, on passe à l'étape suivante, en cas d'erreur on suit un chemin dégradé pour automatiser la reprise.
	Un ordre est le déclencheur de la chaîne de traitement. Il peut être ajouté manuellement ou planifier dans le temps.
	Un traitement est dit chaîne lorsqu'il est conçu pour faire partie d'un scénario d'exécution. Il ne peut être exécuté qu'à travers une chaîne de traitements qui sera elle-même déclenché par un ordre.
	Lorsque le traitement est indépendant, on peut l'exécuter directement. Ces traitements sont à considérer comme des traitements de Cron Unix ou du Schedule Windows.
	La planification permet de spécifier les heures de lancement en fonction des mois, des jours du mois ou des jours de la semaine. Cette planification prend le compte les jours fériés.
	Le moteur d'ordonnancement gère l'ensemble des événements : calcul de planification ou de dépendances, soumissions des traitements et communication avec les composants externes.
	Un traitement peut être exécuté à travers une file d'attente et/ou sur une machine distante. On parle alors de classe de processus.
	Un verrou permet de gérer les ressources virtuelles afin de limiter les exécutions concurrentes des traitements.

Mes informations

Le coin supérieur gauche récapitule les informations de l'utilisateur : son compte et, dans un cadre professionnel, l'entreprise à laquelle il appartient.



Un clic suffit pour changer la langue de l'interface. L'interface est disponible en français et en anglais. Les traductions sont en cours pour les langues suivantes : Espagnol, Allemand, Chinois et Japonais.

Le lien déconnexion ferme la session en cours.

Mon compte

Cette section permet de voir et de modifier les paramètres de son compte. L'onglet profil concerne les informations administratives, l'onglet « Mot de passe » propose un formulaire dédié au changement de son mot de passe.

La première ligne indique le rôle de l'utilisateur, cette information ne peut être modifiée que par un administrateur.

Profile	Mot de passe
Utilisateur Operateur Concepteur Gestionnaire Administrateur Installateur	
À propos de moi	
First name	Eric
Nom	Ochet
Email	eo@sos-paris.com
Sauvegarder les changements	

Changement de mot de passe

Le changement de mot de passe est effectif à la prochaine connexion.

Profile	Mot de passe
Changer le mot de passe	
Nouveau mot de passe	••••••
Sauvegarder le nouveau mot de passe	

Session

La session est l'ensemble des informations de

Actualiser	30
Date	
• Date de référence	2014-02-26 20:41:32
• De	2014-02-22 20:32:08 (-4j)
• à	2014-02-28 20:32:08 (+2j)
• Fuseau horaire	
Site	
Filtre	
Moteur	
Référentiel	

l'environnement de l'utilisateur.

Installation silencieuse

L'installation silencieuse est un formulaire qui génère le fichier XML qui pourra être utilisé par l'installateur pour créer une nouvelle instance de l'ordonnanceur sur un serveur dépourvu d'interface graphique.

Le script peut être utilisé pour des machines Windows mais l'intérêt est moindre car celle-ci dispose toujours d'un environnement graphique. Pour une installation en masse, il est préférable d'utiliser le module Packages dont le fonctionnement est plus proche d'un outil de déploiement

Formulaire

La partie gauche est un formulaire à remplir, il contient les valeurs couramment utilisées qu'il faut simplement modifier en fonction de ses besoins spécifiques.

Pour générer directement un script pour un agent, il suffit de cliquer sur l'acceptation des termes du contrat et cliquer sur le bouton Télécharger. Un agent est moteur qui reçoit les ordres de soumission d'un autre moteur, il ne nécessite donc pas de base de données.

Licence

L'acceptation de la licence est un champ obligatoire.

Licence	Moteur	Système	Base de données	Notifications	Traitements
<div>Licence mode</div> <div>GNU General Public License 2.0</div> <div>For the GNU GPL 2.0 licensed version the distribution includes a free license key (file config/sos.ini).</div> <div>Installation support is offered free of charge. Continued operations support is restricted to the commercial version.</div> <div>The GNU GPL 2.0 licensed version supports only Windows and Linux. For other operating systems the commercial license is required.</div> <div>Further information at www.gnu.org/licenses</div> <div><input type="checkbox"/> J'accepte les termes du contrat de licence GPL V2 *</div> <div><div>GNU GENERAL PUBLIC LICENSE Version 2, June 1991</div><div>Copyright (C) 1989, 1991 Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.</div><div>Preamble</div><div>The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free</div></div>					
<div>Visualiser Télécharger</div> <div>Remplir le formulaire et générer le fichier d'installation silencieuse.</div>					

Moteur

L'onglet moteur définit les paramètres de connexion de l'ordonnanceur. Par défaut, le moteur écoute sur le port 4444 de la machine locale et démarre automatiquement en fin d'installation.

Licence	Moteur	Système	Base de données	Notifications	Traitements
Moteur					
Id *		scheduler			
		L'identifiant doit être unique par serveur et ne doit pas contenir les caractères suivants : / \ : ; * ? ! \$ % & " < > () ^			
Type		Serveur Autonome			
Démarrer le moteur		<input checked="" type="checkbox"/>			
		Launch the job scheduler after installation.			
Gestionnaire d'évènements		<input type="checkbox"/>			
		Cocher cette option si le moteur doit traiter des évènements.			
Réseau					
Hôte		localhost			
Port *		4444			
Machines autorisées		0.0.0.0			
		Il est recommandé d'activer la restriction d'accès à un nombre limité de machines. Pour autoriser toutes les machines, il faut saisir 0.0.0.0			
Jetty					
Port HTTP		44440			
Port HTTPS		8443			

Système

L'onglet système donne le choix entre Unix (Linux, AIX, HP-UX, Solaris) et Windows.

Dans tous les cas, on indique le répertoire d'installation et le répertoire utilisateur, il est possible de stocker les fichiers dans un seul et même répertoire.

Licence	Moteur	Système	Base de données	Notifications	Traitements
Installation					
Système d'Exploitation		Unix			
Répertoire d'installation		/opt/jobscheduler <small>Le répertoire d'installation contient les binaires et les bibliothèques.</small>			
Répertoire Utilisateur		/home/osjs/jobscheduler <small>Le répertoire utilisateur contient la configuration.</small>			
Connecteur CRON		<input type="checkbox"/> <small>Le connecteur CRON peut être utilisé pour configurer des traitements à partir d'un fichier crontab.</small>			
Chemin		/etc/crontab			
Timeout (s)		600			
<input type="checkbox"/> Système Cron					
Changement de l'utilisateur		Custom			
Changement de commande					

Pour Unix, le connecteur Cron ajoute un traitement qui scrute régulièrement la crontab pour extraire les planifications et les transformer en traitement Open Source JobScheduler.

Il n'existe pas d'équivalent pour Windows mais un module Ari'i offrira un système équivalent à partir du Planificateur de tâches.

Licence	Moteur	Système	Base de données	Notifications	Traitements
Installation					
Système d'Exploitation		Windows			
Répertoire d'installation		%PROGRAMFILES%/jobscheduler <small>Le répertoire d'installation contient les binaires et les bibliothèques.</small>			
Répertoire Utilisateur		%COMMONAPPDATAROOT%/jobscheduler <small>Le répertoire utilisateur contient la configuration.</small>			

Base de données

L'onglet Base de données permet de connecter le moteur à une base de données. La base de données permet de stocker les historiques et les états moteur en cas d'arrêt brutal du système.

La base de données est indispensable pour utiliser le module de suivi.

Licence	Moteur	Système	Base de données	Notifications	Traitements
Base de données					
<input checked="" type="checkbox"/> Utiliser une base de données					
Type <input type="text" value="MySQL"/>					
Hôte * <input type="text" value="localhost"/>					
Port * <input type="text" value="3306"/>					
Schéma * <input type="text" value="root"/>					
Utilisateur * <input type="text" value="root"/>					
Mot de passe <input type="password"/>					
<input checked="" type="radio"/> Pilote MariaDB					
<input type="radio"/> Pilote MySQL					
Connecteur * <input type="text"/>					
<input checked="" type="checkbox"/> Créer les tables					

Pour Mysql, on peut utiliser le pilote MariaDB fournit avec le produit ou prendre le pilote natif MySQL, dans ce deuxième cas, il est nécessaire de télécharger le pilote chez Oracle puis d'indiquer sa localisation.

Notifications

L'onglet Notifications sert à la configuration des mails. On indique la politique d'envoi de mail et les paramètres de connexion au serveur de messagerie.

Licence	Moteur	Système	Base de données	Notifications	Traitements
Notifications					
<input type="checkbox"/> Envoi de mail					
<input checked="" type="checkbox"/> sur erreur					
<input checked="" type="checkbox"/> sur avertissement					
<input type="checkbox"/> sur Succès					
<input type="checkbox"/> Messagerie					
Hôte <input type="text"/>					
Port <input type="text" value="25"/>					
Expéditeur <input type="text"/>					
Destinataire(s) <input type="text"/>					
en copie <input type="text"/>					
en copie cachée <input type="text"/>					
<input type="checkbox"/> Authentification					
Compte <input type="text"/>					
Mot de passe <input type="text"/>					

Traitements

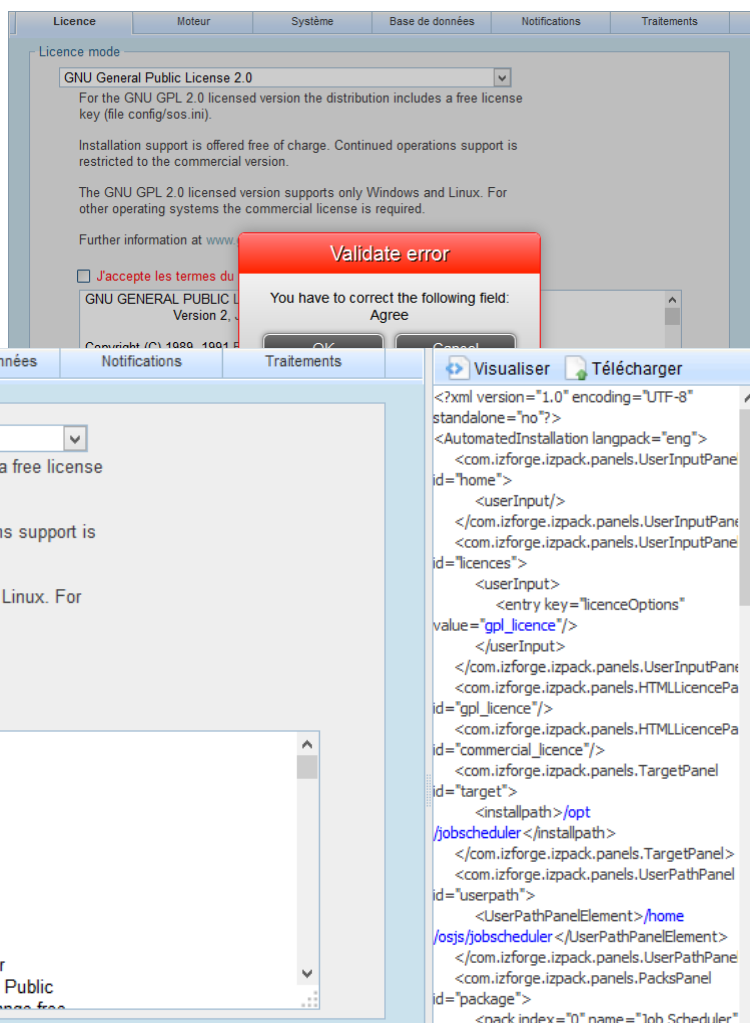
L'onglet traitements permet d'ajouter des traitements pré-définis dans le moteur. Il n'existe actuellement que la vérification des nouvelles versions.

Licence	Moteur	Système	Base de données	Notifications	Traitements
Traitements					
<input type="checkbox"/> Vérifier la mise à jour					
Jour de départ <input type="text" value="Lundi"/>					
Heure de départ <input type="text" value="13:00"/>					
<input type="checkbox"/> Automatic download					

Génération du fichier

Les paramètres obligatoires sont indiqués par une astérisque rouge, comme par exemple l'acceptation des termes du contrat GPL.

Si un champ obligatoire est manquant, un message d'alerte apparaît en indiquant le champs concerné. Le champs est lui-même indiqué e rouge pour plus de lisibilité.

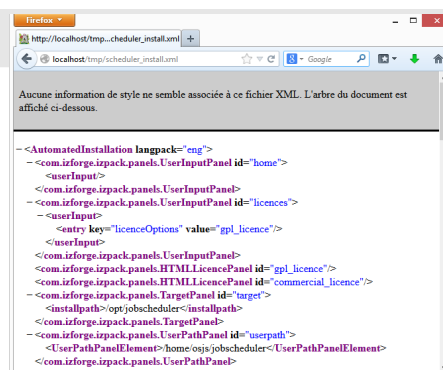


Téléchargement

Le bouton télécharger donne un accès direct au fichier pour le copier/coller dans le fichier de destination ou faire un « enregistrer sous » puis le transférer sur la cible.

L'url peut être utilisé pour récupérer le fichier XML à partir de la machine à installer en utilisant curl ou wget.

```
wget http://<adresse  
IP>/tmp/scheduler_install.xml
```



Audit

L'audit affiche l'ensemble des actions réalisées par l'utilisateur. Un double clic affiche le détail de l'action et le message de réponse.

Si une commande a été correctement exécutée, on obtient le message de réponse du serveur. Dans le cas contraire, l'origine de l'erreur est indiquée.

Heure du journal	Module	Action	Statut	Message	Utilisateur	IP address
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
2014-02-28 21:02:10	JID	<modify_job job="peakup/Deuxieme" cmd="stop"/>	SUCCESS		admin	::1
2014-02-28 20:57:53	JID	<modify_job job="peakup/Deuxieme" cmd="stop"/>	ERROR	!ERROR: Can not Open URL: http://sosparis:4444/%	admin	::1
2014-02-28 13:51:44	JID	<job_chain_node.modify job_chain="peakup/Job Ch	INFO	http://sosparis:4444/%3Cjob_chain_node.modify%2	admin	::1
2014-02-28 13:51:16	JID	<job_chain_node.modify job_chain="peakup/Job Ch	INFO	http://sosparis:4444/%3Cjob_chain_node.modify%2	admin	::1
2014-02-28 13:49:52	JID	<modify_order job_chain="peakup/Job Chain" order	INFO	http://sosparis:4444/%3Cmodify_order%20job_chai	admin	::1
2014-02-26 19:36:29	JID	<job_chain_node.modify job_chain="peakup/Job Ch	INFO	http://sosparis:4444/%3Cjob_chain_node.modify%2	admin	::1
2014-02-26 19:29:49	JID	<modify_order order="Reprise sur erreur 1" job_cha	INFO	http://sosparis:4444/%3Cmodify_order%20order%3l	admin	::1
2014-02-26 19:29:37	JID	<job_chain_node.modify job_chain="peakup/Job Ch	INFO	http://sosparis:4444/%3Cjob_chain_node.modify%2	admin	::1
2014-02-26 19:27:02	JID	<modify_order order="Reprise sur erreur 1" job_cha	INFO	http://sosparis:4444/%3Cmodify_order%20order%3l	admin	::1
2014-02-26 19:24:07	JID	<modify_order job_chain="peakup/Job Chain" order	INFO	http://sosparis:4444/%3Cmodify_order%20job_chai	admin	::1
De 1 à 10 Page 1 10 Lignes par page						
Heure du journal	2014-02-28 21:02:10			Message	183	
Statut	SUCCESS			Module	JID	
Action	<modify_job job="peakup/Deuxieme" cmd="stop"/>					
Message	<?xml version="1.0" encoding="ISO-8859-1"?> <spooler> <answer time="2014-02-28T20:02:10.818Z"> <ok/> </answer> </spooler>					

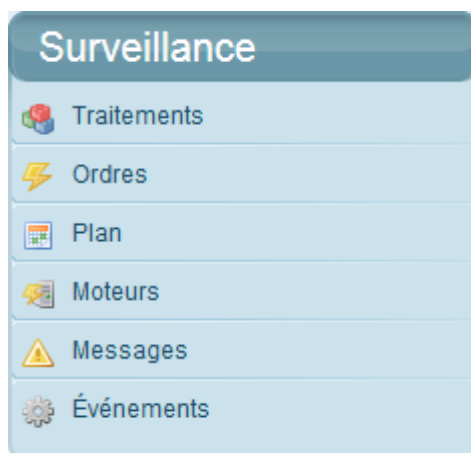
Suivi d'exploitation

Ce module permet de suivre les traitements passés et en cours d'exécution. Il est directement connecté à la base de données et ne sollicite le moteur qu'au moment de l'exécution des commandes évitant ainsi la consommation de ressources. Ce module est dédié au pilotage ou aux utilisateurs souhaitant visualiser l'état des traitements.

Le suivi reprend la configuration du portail pour se connecter sur la base de données et communiquer avec le moteur d'ordonnancement.



Ergonomie



Open Source JobScheduler stocke dans la base de données les historiques d'exécutions. Il stocke aussi les états afin de reprendre les processus en cas d'arrêt imprévu du système.

Le module de suivi reprend l'ensemble de ces données pour les afficher sous forme de graphiques, de tableaux ou bien encore de chronogrammes.

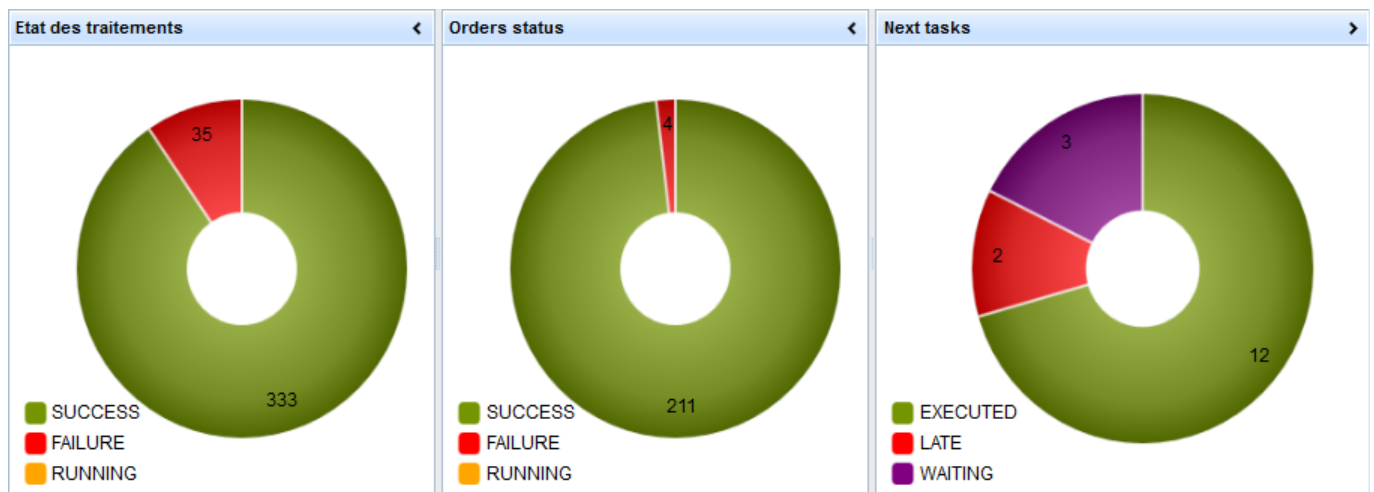
La vue calendrier permet de spécifier la plage de dates à visualiser par rapport à la date de référence. Pour rappel, la date de référence est celle indiquée dans la barre d'outil du portail.

Dans l'exemple ci-contre, on ne prendra en compte que les données datant de 4 jours avant la date de référence jusqu'à 2 jours dans le futur.



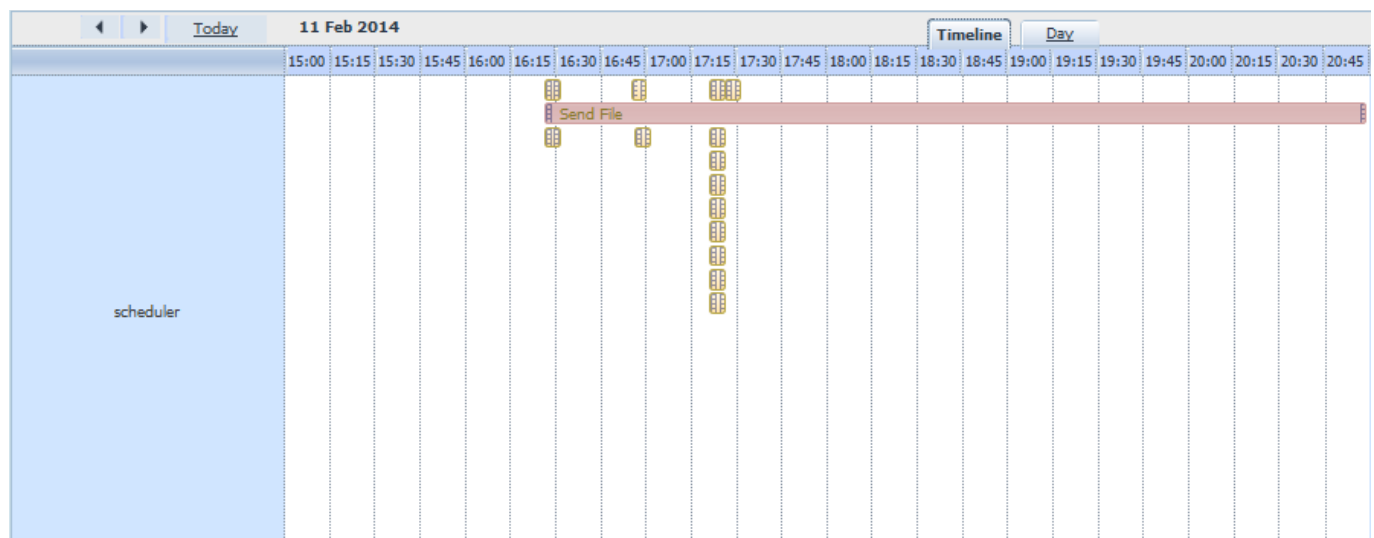
Vue globale

La vue globale permet d'avoir une vision graphique des traitements sur la période concernée. On affiche les 3 grands domaines : les traitements (indépendants ou ordonnés), les ordres et les prochaines planifications.





Attention ! Il s'agit d'un historique, il est donc possible que les traitements en erreur aient été corrigés au moment de la visualisation. Pour état en temps réel, il est nécessaire d'utiliser le module **Opérations**.

Ce graphique est complété par un chronogramme qui donne l'état des traitements sur une ligne de temps.







Traitements

La zone de visualisation des traitements est découpée en deux zones : une zone de données et une zone de détail qui donne accès aux informations du traitement courant, à son dernier journal d'exécution et à son historique.

Les traitements sont des tâches informatiques indépendantes ou utilisées dans des séquences. On les distingue par l'icone utilisé, le symbole  signifie que le traitement est indépendant, alors qu'un traitement utilisé dans une chaîne sera symbolisé comme suit .

Chaque colonne peut être filtrée pour un accès rapide à l'information. Ces filtres sont de 3 types : texte, liste déroulante et nombre. Le premier cas permet de retrouver un traitement par une partie de son nom, la liste déroulante sert à la sélection d'un statut et le dernier filtre permet de sélectionner un exit code.


























L'organisation reprend l'arborescence des fichiers, on distingue trois niveaux : le moteur, les répertoires et

Localisation	Statut	Heure de départ	Heure de fin
<input type="text" value="ftp"/>	<input type="text" value="v"/>	<input type="text"/>	<input type="text"/>
 scheduler	RUNNING 	2014-02-12 07:52:36	
 scheduler_sosftp_import	SUCCESS 	20:36:06	20:41:15

les traitements.

Un quatrième niveau est ajouté lorsque le traitement est en cours, il est représenté par un suivi du numéro de processus sur la machine concernée. Dans l'exemple suivant, le traitement running job est en cours d'exécution avec le PID 17396.



Localisation	Statut	Heure de départ	Heure de fin	Durée	Exit	Prochain départ
<input type="text"/>	<input type="text" value="v"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
 scheduler	RUNNING 	2014-02-12 07:52:36				
 Jade	SUCCESS 	2014-02-11 11:08:02	11:08:05	3s	0	
 LogAnalyser	SUCCESS 	09:36:12	09:36:13	1s	0	
 peakup						
 ping	SUCCESS 	2014-02-10 08:41:23	08:41:24	1s	0	
 Receive File	FAILURE 	08:54:00	20:36:07	11h	1	
 running job	RUNNING 					
 17396		21:09:49		1h	0	
 scheduler_sosftp_import	SUCCESS 	20:36:06	20:41:15	5m 9s	0	
 Send File	FAILURE 	2014-02-11 17:26:41	08:39:30	15h	1	

De 1 à 10 Page 1 10 Lignes par page






Par défaut, on affiche des pages de 10 lignes et on navigue de pages en pages. Le système de filtre permet généralement de limiter les traitements concernés à quelques lignes pour ne travailler que sur un groupe de traitements ou en fonction d'un statut particulier.

Pour basculer sur une mode liste, il suffit de changer le nombre de lignes par page. Un ascenseur permet de se déplacer dans la liste.




Localisation	Statut	Heure de départ	Heure de fin	Durée	Exit	Prochain départ
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
LogAnalyser	SUCCESS	09:36:12	09:36:13	1s	0	
peakup						
ping	SUC	08:41:23	08:41:24	1s	0	
Receive File	FAI	08:54:00	20:36:07	11h	1	
running job	RUN					
17396		21:09:49		1h	0	
scheduler_sosftp_import	SUC	20:36:06	20:41:15	5m 9s	0	
Send File	FAI	11:26:41	08:39:30	15h	1	
sos						
ttstmr1						

La taille des fenêtres peut être ajustées en déplaçant les cloisons avec la souris. On peut ainsi agrandir la fenêtre de données au détriment de la zone de détail.

Le statut est indiqué de 3 manières : le nom, le symbole et le code couleur. Ceci permet de distinguer rapidement l'état du système.

Statut	Icône	Couleur	Explication
SUCCESS		Vert	L'exécution du traitement s'est bien déroulée.
FAILURE		Rouge clair	L'exécution s'est terminée avec une erreur, l'exit code est supérieur à 0.
RUNNING		Orange	Le traitement est en cours d'exécution.
STOPPED		Rouge foncé	Le traitement s'est terminé en erreur et un « Stop sur erreur » est actif. Une action est nécessaire pour le débloquent.
ACTIVATED		Bleu	L'ordre est défini et stocké en base de données mais n'a jamais été exécuté.

La zone de filtre indique les statuts en cours, en sélectionnant le statut FAILURE, on accède à l'ensemble des traitements en erreur.

Localisation	Statut	Heure de départ
<input type="text"/>	FAILURE <input type="text"/>	<input type="text"/>
scheduler	RUNNING 	2014-02-12 07:52:36
Receive File	FAILURE 	08:54:00
Send File	FAILURE 	2014-02-11 17:26:41

La colonne Exit correspond à l'exit code renvoyé par le shell. L'utilisation du filtre avec un opérateur permet de retrouver plus précisément les traitements en erreur par le code d'exit.

Les colonnes heure de départ, heure de fin et durée permettent de situer le traitement dans le temps.






Durée	Exit	Prochain départ
<input type="text"/>	<input type="text" value=">0"/>	<input type="text"/>
	11h	1
	15h	1

Le filtre permet de retrouver les traitements pour une date précise :

Localisation	Statut	Heure de départ	Heure de fin	Durée	Exit	Prochain départ	
<input type="text"/>	<input type="text" value="v"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="2014-02"/>	
scheduler	RUNNING	2014-02-12 07:52:36					
sos							
dailyschedule							
CreateDaysSchedule	SUCCESS	08:39:28	08:39:37	9s	0	2014-02-13 00:00:00	
update							
scheduler_check_updates	SUCCESS	2014-02-10 20:00:01	20:00:34	33s	0	2014-02-17 20:00:00	

Si le traitement est planifié, la colonne « Prochain départ » indique l'heure de lancement.

La dernière colonne indique le type de lancement du traitement :

Icone	Type	Explication
	Manuel	Le traitement a été mis en queue par une action manuelle. Le traitement est généralement un job indépendant.
	Ordonné	Le traitement est dans une chaîne, il sera exécuté par un ordre. Le traitement est généralement ordonné.
	Planifié	Les paramètres de date et d'heure sont associés au traitement.
	Cyclique	Le traitement est exécuté en fonction d'une fréquence définie.
	Instancié	Le traitement à un paramètre «tâches min. » qui indique le nombre de tâches concurrentes à exécuter systématiquement.

Informations

Un double clic suffit pour accéder aux informations du traitement. Les types d'informations sont identifiées par des onglets : Le détail de l'exécution, le journal et l'historique du traitement.

Le détail les paramètres et le dernier message du moteur concernant le traitement.

Localisation	Statut	Heure de départ	Heure de fin	Durée	Exit	Prochain départ	
<input type="text"/>	FAILURE	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
<div> scheduler</div> <div><div>RUNNING </div><div>2014-02-12 07:52:36</div></div>							
Receive File	FAILURE	08:54:00	20:36:07	11h	1		
Send File	FAILURE	2014-02-11 17:26:41	08:39:30	15h	1		
<div> De 1 à 3 Page 1 20 Lignes par page </div>							
Information							
Information		Log	Historique				
Job name		Receive File					
Statut		FAILURE					
SCHEDULER-140 Task logged error: ERROR [main] (SOSSFTP.java:778) - error occurred writing file [C:\Users\leric\AppData\Local\Temp\sos5455985043806183858.gz]: Unexpected end of sftp stream.							
Cause		queue_at		Task id		23415	
Heure de départ		2014-02-12 07:54:00		Exit code		1	
End Time		2014-02-12 19:36:07		Error		1	
Durée		11h		Steps		1	
Parameters							
ftp_host		arii.org					
ssh_auth_method		password					

Le journal permet d'afficher les informations du moteur et du traitement lors de l'exécution.

Information	Log	Historique		
Date	Type	Message	Code	
2014-02-12 08:54:00.272	I	state=starting (at=2014-02-12 08:53:59.803+0100)	918	
2014-02-12 08:54:01.295	I	[main] (SOSMsg.java:126) - LOG_I_0020: JobSchedulerLog4JAppender is configured as log4j-appender		
2014-02-12 08:54:01.297	I	[main] (JobSchedulerJobAdapter.java:117) - 1.5.4014 (2014-01-14T22:31:12.765+0100) Copyright 2003-2013 SOS GmbH Berlin		
2014-02-12 08:54:01.300	I	\$Id: JobSchedulerFTPSend.java 16869 2012-03-26 10:08:16Z oh \$		
2014-02-12 08:54:01.300	I	1.5.4014 (2014-01-14T22:31:12.765+0100) Copyright 2003-2013 SOS GmbH Berlin		
2014-02-12 08:54:01.319	I	[main] (SOSMsg.java:126) - JSJ_I_0010: JobSchedulerJobAdapter::getNodeName: Current Job is 'Receive File'.		
2014-02-12 08:54:01.445	I			
2014-02-12 08:54:01.445	I	**		
2014-02-12 08:54:01.445	I	**		
2014-02-12 08:54:01.445	I	* ----- *		
2014-02-12 08:54:01.445	I	* SOSFTP - Managed File Transfer Utility *		
2014-02-12 08:54:01.445	I	*****		
2014-02-12 08:54:01.445	I	*****		
2014-02-12 08:54:01.445	I	*****		

Pour une lecture rapide, on peut filtrer par type d'information pour extraire directement les avertissements ou les erreurs.

Information	Log	Historique		
Date	Type	Message	Code	
2014-02-12 18:54:34.066	E	[main] (SOSSFTP.java:778) - error occurred writing file [C:\Users\leric\AppData\Local\Temp\sos5455985043806183858.gz]: Unexpected er		
2014-02-12 18:54:34.068	E	[main] (SOSSFTP.java:778) - Error during putFile: com.sos.JSHelper.Exceptions.JobSchedulerException: error occurred writing file [C:\Use		

Le dernier type de message est l'historique des exécutions pour identifier la fréquence des erreurs.

Information	Log	Historique			
Heure de départ	Heure de fin	Durée	Error	Exit code	Message
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
2014-02-10 15:21:34	15:26:35	5m 1s	0	0	
2014-02-07 17:13:56	17:20:00	6m 4s	0	0	
2014-02-07 16:39:27	16:39:28	1s	1	1	SCHEDULER-140 Task logged error: error occurred connect to database failed: sos.connection.s
2014-02-07 16:36:28	16:36:29	1s	1	1	SCHEDULER-140 Task logged error: error occurred connect to database failed: sos.connection.s
2014-02-07 16:26:39	16:31:40	5m 1s	0	0	
2014-02-07 15:54:17	15:59:18	5m 1s	0	0	
2014-02-07 15:50:36	15:52:24	1m 48s	0	0	

Pour un diagnostic plus précis sur l'historique, il faut se référer à la partie « Historique des traitements ».

Barre d'outil

Export RSS

Le bouton RSS de la barre d'outil permet de générer le flux RSS des derniers traitements en erreurs.



S'abonner à ce flux en utilisant



Marque-pages dynamiques

☐ Toujours utiliser Marque-pages dynamiques pour s'abonner aux flux.

S'abonner maintenant

Job failures

Last jobs failed

2014-02-13 07:59:19 [scheduler]/running job (exit 99)

SCHEDULER-280 Process terminated with exit code 99 (0x63)

2014-02-13 07:39:55 [scheduler]/TSM/AF/FS/ACC/tlsqstm1 /ARC_G000_FS_SAR (exit 1)

SCHEDULER-280 Process terminated with exit code 1 (0x1)

2014-02-13 07:39:54 [scheduler]/TSM/AF/FS/ACC/tlsrhps9 /ARC_G000_FS_SAR (exit 1)

SCHEDULER-280 Process terminated with exit code 1 (0x1)

2014-02-13 07:39:54 [scheduler]/TSM/AF/FS/ACC/tlsrnaplx04 /ARC_G000_FS_SAR (exit 1)

SCHEDULER-280 Process terminated with exit code 1 (0x1)

2014-02-13 07:39:53 [scheduler]/TSM/AF/FS/DEV/tlsdlhs1

Menu contextuel

Un menu contextuel est associé à un type d'objet et à son statut, ce système limite la liste des actions pour n'afficher que celles qui sont utiles par rapport à une situation particulière. Le menu contextuel est accessible avec un clic droit après avoir sélectionné l'objet.

La communication entre l'interface et le moteur d'ordonnancement est toujours en deux temps :

- la commande est envoyée au moteur
- le moteur renvoie une réponse et exécute la commande


Pour les actions sur le moteur, il faut se reporter au paragraphe *Moteurs p66*.

Actions sur le traitement



















Tout comme pour le moteur, les actions sur le traitement dépendent de son statut et de son type. Le cas le plus courant est un traitement indépendant en attente d'exécution, un clic droit donnera droit à trois actions possibles :

- démarrer le traitement
- le bloquer avec un STOP
- voir l'historique d'exécution

Lancer un traitement

Pour démarrer un traitement il faut  sélectionner un traitement indépendant, accéder au menu par un clic droit et choisir l'action : démarrer la tâche.

Un traitement ordonné ne peut être lancé qu'à travers une chaîne de traitements.

Localisation	Statut	Heure de départ	Heure de fin
<input type="text"/>	<input type="text" value="v"/>	<input type="text"/>	<input type="text"/>
 scheduler	RUNNING 	2014-02-12 07:52:36	
 pickup			
 Jade	SUCCESS 	2014-02-11 11:08:02	11:08:05
 LogAnalyser	SUCCESS 	6:12	09:36:13
 ping	SUCCESS 	9:44	21:39:48
 Receive File	FAILURE 	4:00	20:36:07
 running job	RUNNING 		
 2808		23:32:14	
 scheduler_sosftp_import	SUCCESS 	20:36:06	20:41:15
 Send File	FAILURE 	2014-02-11 17:26:41	08:39:30

Si le traitement est en exécution, l'action de kill est ajoutée. On pourra éventuellement démarrer une nouvelle instance si sa définition le permet.

Arrêter un traitement en cours

Lorsque le traitement est en cours, une nouvelle action apparaît : Killer la tâche.

En sélectionnant cette tâche, une demande d'arrêt est envoyé au système.

Localisation	Statut	Heure de départ	Heure de fin
<input type="text"/>	<input type="text" value="v"/>	<input type="text"/>	<input type="text"/>
[-] scheduler	RUNNING	2014-02-12 07:52:36	
[-] pickup			
[-] Jade	SUCCESS	2014-02-11 11:08:02	11:08:05
[-] LogAnalyser	SUCCESS	09:36:12	09:36:13
[-] ping	SUCCESS	21:39:44	21:39:48
[-] Receive File	FAILURE	08:54:00	20:36:07
[-] running job	RUNNING		
[-] 2808		23:32:14	
[-] scheduler_sosftp_import	SUCCESS	20:36:06	20:41:15
[-] Send File	FAILURE	17:26:41	08:39:30
<div>Start task Kill task Stop Job history</div>			
De 1 à 10 Page 1 1			

Stopper un traitement

Un stop consiste à invalider un traitement, généralement pour indiquer qu'il est en erreur ou peut entraîner une erreur.

Localisation	Statut	Heure de départ	Heure de fin
<input type="text"/>	<input type="text" value="v"/>	<input type="text"/>	<input type="text"/>
[-] scheduler	RUNNING	2014-02-12 07:52:36	
[-] pickup			
[-] Jade	SUCCESS	2014-02-11 11:08:02	11:08:05
[-] LogAnalyser	SUCCESS	09:36:12	09:36:13
[-] ping	SUCCESS	21:39:44	21:39:48
[-] Receive File	FAILURE	08:54:00	20:36:07
[-] running job	RUNNING		
[-] 2808		23:32:14	
[-] scheduler_sosftp_import		20:36:06	20:41:15
[-] Send File		2014-02-11 17:26:41	08:39:30
<div>Stop Job history</div>			

Si on a stoppé le traitement, on pourra tout de même le démarrer ou le débloquent pour une prochaine exécution.

Localisation	Statut	Heure de départ	Heure de fin
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
scheduler	RUNNING	2014-02-12 07:52:36	
LogAnalyser	STOPPED	00:36:12	09:36:13
peakup			
Jade	SUCCESS	00:02	11:08:05
ping	SUCCESS	00:44	21:39:48
Receive File	FAILURE	08:54:00	20:36:07
running job	SUCCESS	23:32:14	23:34:43
scheduler_sosftp_import	SUCCESS	20:36:06	20:41:15
Send File	FAILURE	2014-02-11 17:26:41	08:39:30
sos			

Start task
Unstop
Job history

L'historique du traitement permet d'accéder à un écran dédié.

Diagnostic

Le diagnostic interroge le moteur sur l'état d'un job afin de connaître la raison d'un blocage.

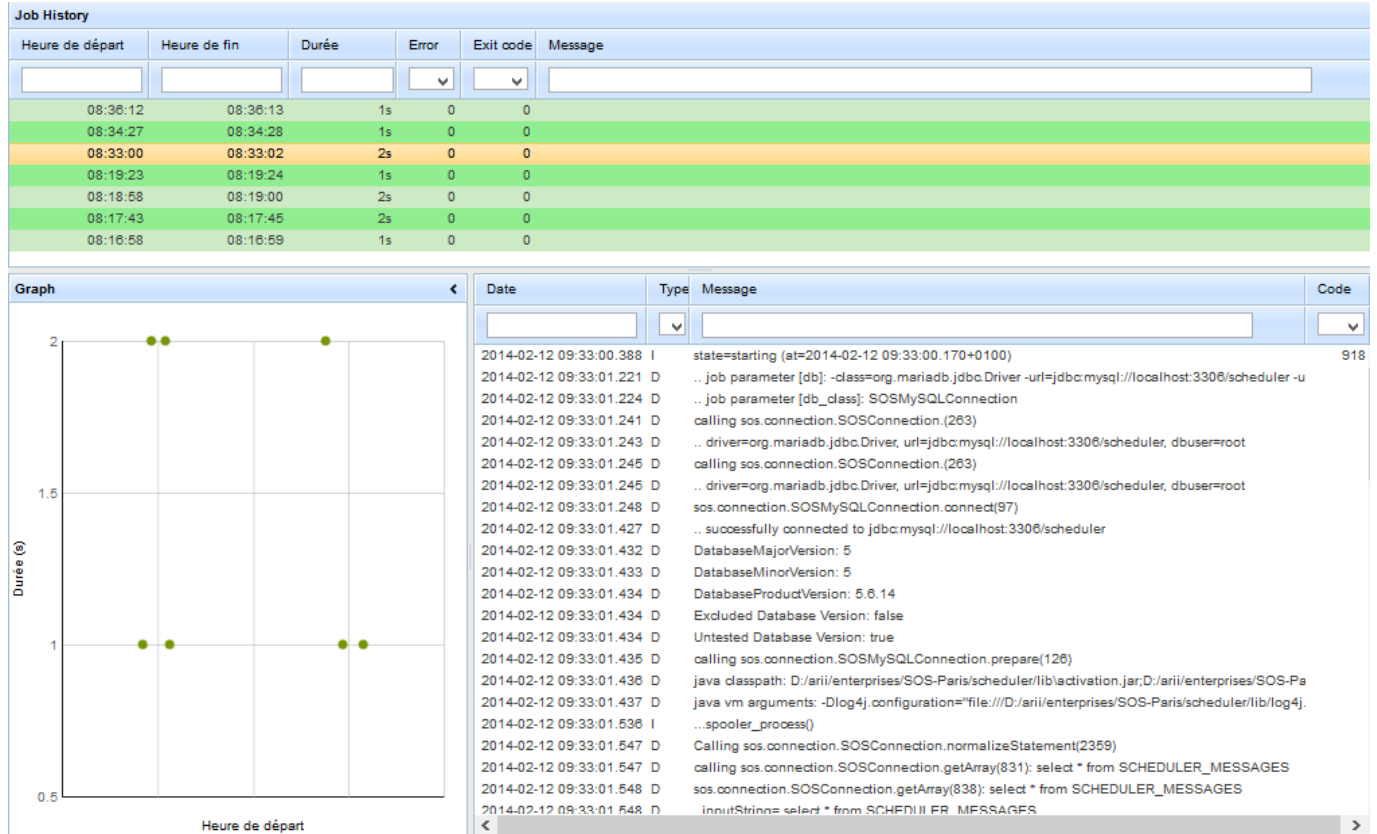
La cause du problème est indiquée sous les lignes rouges.

Dans l'exemple ci-contre, le job est dans une chaîne de traitements dont l'unique ordre est suspendu à l'étape 002. Pour débloquent cette situation, il sera nécessaire de réactiver l'ordre.

Diagnostic	
Abandon	
Nom	Valeur
job_chain_nodes	
job_chain_node	
job_chain	
path	/peakup/Job Chain
order_queue	
order	
1	
at	2038-01-19 03:14:07.000Z
suspended	true
id	ETAPE 2
length	1
order_state	002
if_order_is_ready	
job	Deuxieme

Historique des traitements

L'historique permet de situer un traitement dans le temps, par sa durée d'exécution ou par son statut en le visualisant directement sur dans un tableau filtré ou à travers un graphique. Les écrans sont synchronisés, un double clic sur le tableau ou un simple clic sur la partie graphique affiche directement le journal pour l'exécution concernée.

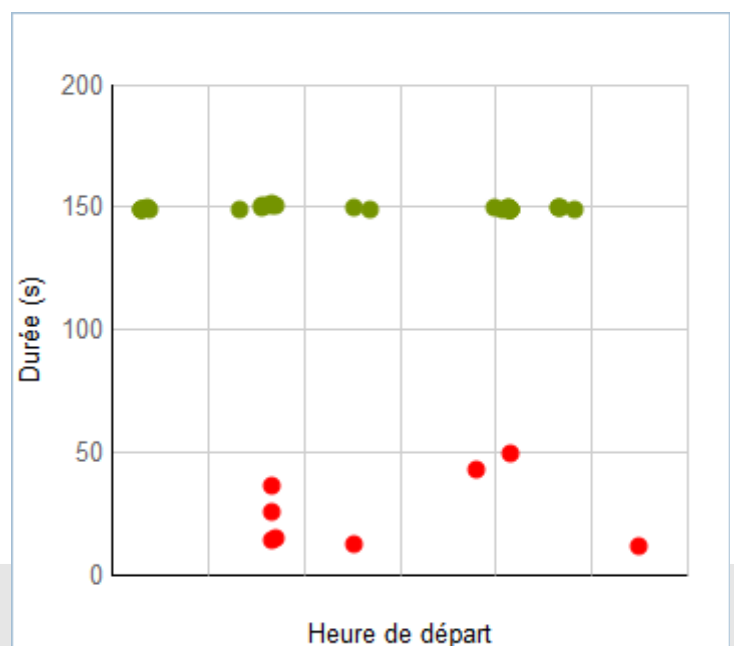


Le graphique indique l'heure en abscisse et la durée d'exécution en ordonnée. L'heure de départ indique le respect de la fréquence d'exécution, un écart horizontal entre deux points anormalement espacé signifie que le traitement ne suit pas un cycle, soit une exécution est manquante, soit un lancement a été forcé par un opérateur. Un écart vertical différent signifie que la durée du traitement est variable.

La couleur du point reprend le code couleur du statut : vert, rouge et orange.

Ordres

Le panneau des ordres permet de suivre les exécutions de chaînes de traitement. L'ergonomie est similaire au suivi des traitements mais permet de suivre un enchaînement complet. La notion de séquence induit aussi des notions de pilotage plus complexes en cas d'erreur car il faut gérer l'impact sur la suite des





traitements.

Localisation	Etat	Statut	Heure de départ	Heure de fin	Information	Next time
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
scheduler RUNNING 2014-02-12 07:52:36						
tistmt1						
TSM						
peakup						
Job Chain ACTIVE						
Test (toujours bien)	Fin	SUCCESS	2014-02-11 18:21:08	18:21:49		
Reprise sur erreur 1	000	ACTIVATED	2014-02-09 10:22:31			Jamais
154	Sync	SUCCESS	2014-02-11 18:21:19	18:21:49		
155	Sync	SUCCESS	2014-02-11 18:21:18	18:21:49		
Splitter						

L'organisation reprend les répertoires de chaque moteur dans lesquels on retrouve les chaînes de traitements et les ordres d'exécution.

On distingue différents types :




- L'ordre planifié  stocké en base de données et prêt à être exécuté. Si l'ordre a déjà été lancé, on peut consulter les étapes exécutées. Dans le cas contraire, l'ordre est considéré comme activé.
- L'ordre ajouté manuellement par un opérateur  et qui est exécuté immédiatement.





Dans le cas de l'ordre exécuté, planifié ou manuel, un clic permet d'ouvrir la chaîne pour afficher les étapes :

Localisation	Etat	Statut	Heure de départ	Heure de fin
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
peakup				
Job Chain ACTIVE				
Test (toujours bien)	Fin	SUCCESS	2014-02-11 18:21:08	18:21:49
1 peakup/Debut	000	SUCCESS	2014-02-11 17:21:09	2014-02-11 17:21:18
2 peakup/JobSolitter	Split	SUCCESS	2014-02-11 17:21:18	2014-02-11 17:21:18
3 peakup/Premiere	001	SUCCESS	2014-02-11 17:21:19	2014-02-11 17:21:28
4 peakup/JobSvnc	Sync	SUCCESS	2014-02-11 17:21:29	2014-02-11 17:21:30
5 peakup/JobSvnc	Sync	SUCCESS	2014-02-11 17:21:49	2014-02-11 17:21:49
6 peakup/Exit	004	SUCCESS	2014-02-11 17:21:49	2014-02-11 17:21:49
Reprise sur erreur 1	000	ACTIVATED	2014-02-09 10:22:31	

Dans cet exemple, six étapes ont été exécutées.

Le statut d'un ordre permet de retrouver l'état global de la chaîne de traitements :

Statut	Icône	Couleur	Explication
SUCCESS		Vert	L'ensemble de la chaîne s'est correctement déroulé.
FAILURE		Rouge clair	Le scénario s'est terminé par un chemin d'erreur.
WARNING		Jaune clair	Le scénario s'est terminé correctement mais en empruntant un chemin dégradé.

RUNNING		Orange	Le traitement est en cours d'exécution.
ACTIVATED		Bleu	L'ordre est programmé mais n'a jamais été utilisé.
SUSPENDED		Rouge ou Vert	L'ordre attend le déblocage par un opérateur.
SETBACK		Rouge clair	L'ordre est en reprise automatique.

Informations

Le panneau d'information reprend le système des onglets mais ils seront différents selon la nature de l'objet : ordre ou traitement.

Détail d'un ordre

Un double clic sur un ordre permet de donner les détails suivants :

Information	Order log	Historique	Graph
Test	Statut SUCCESS		
Séquence de traitements	peakup/Job Chain	Etat	Fin
Heure de départ	2014-02-11 17:21:08	Exit code	0
End Time	2014-02-11 17:21:49	Error	0
Durée	41s	Steps	1

Journal d'un ordre

Les informations ne concernant que l'ordre pour lequel on aura le journal et l'historique.

Information	Order log	Historique	Graph
Date	Task	Message	Code
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
2014-02-09 12:23:23.026	[info]	set_state 000, Job /peakup/Debut	
2014-02-11 17:26:29.646	[info]	Configuration file 'D:/arii/enterprises/SOS-Paris/scheduler/config/live/peakup/Job Chain,'	891
2014-02-11 17:26:31.967	[info]	Standing_order is 'active' now	893
2014-02-11 18:21:09.043	peakup/Debut:23394	Task is going to process Order peakup/Job Chain:Test "toujours bien", state=000, on Sc	842
2014-02-11 18:21:09.045	peakup/Debut:23394		
2014-02-11 18:21:09.045	peakup/Debut:23394	Task peakup/Debut:23394 - Protocol starts in D:/arii/enterprises/SOS-Paris/scheduler/lo	
2014-02-11 18:21:09.048	peakup/Debut:23394	state=starting (at=never)	918
2014-02-11 18:21:09.049	peakup/Debut:23394	Starting process: "C:\Users\eric\AppData\Local\Temp\sosA0A4F.cmd"	987

Historique de l'ordre

Concernant l'historique, il est à deux niveaux : un premier pour l'ordre lui-même et un deuxième niveau pour les différentes étapes.

Information		Order log	Historique		Graph		
Etat			Etat	Heure de départ	Heure de fin	Durée	Message
<div></div>			<div></div>	<div></div>	<div></div>	<div></div>	<div></div>
<div><div>Fin</div></div>			6	2014-02-11 17:21:08	2014-02-11 17:21:49	41s	(toujours bien)
<div><div><div>000</div></div></div>			1	2014-02-11 17:21:09	2014-02-11 17:21:18	9s	
<div><div><div>Split</div></div></div>			2	2014-02-11 17:21:18	2014-02-11 17:21:18		
<div><div><div>001</div></div></div>			3	2014-02-11 17:21:19	2014-02-11 17:21:28	9s	
<div><div><div>Sync</div></div></div>			4	2014-02-11 17:21:29	2014-02-11 17:21:30	1s	
<div><div><div>Sync</div></div></div>			5	2014-02-11 17:21:49	2014-02-11 17:21:49		
<div><div><div>004</div></div></div>			6	2014-02-11 17:21:49	2014-02-11 17:21:49		
<div><div>Fin</div></div>			6	2014-02-09 11:22:41	2014-02-09 11:23:22	41s	(toujours bien)

A partir de l'écran des données, un double clic sur une étape permet de retrouver les mêmes informations que pour les traitements.

Actions sur une chaîne de traitements



La chaîne de traitement peut être considérée comme un scénario constitué d'étapes, chaque étape peut déboucher sur deux nouvelles étapes en fonction de son statut :

- l'étape suivante si le statut est en succès
- l'étape d'erreur si le statut est en erreur

La chaîne peut être déclenchée à tout moment en ajoutant un ordre.

L'autre action possible est de bloquer la chaîne, dans ce cas aucun ordre ne pourra plus la déclencher.

Localisation	Etat	Statut
<input type="text"/>	<input type="text" value="v"/>	<input type="text" value="v"/>
scheduler		RUNNING
reprise		
Reprise automatique		SUCCESS
Reprise AUTO		
t1stmt1		



Ajouter un ordre

Dans le cas de l'ajout de l'ordre, une fenêtre permet d'indiquer les paramètres.

Add Order on JOB Chain


ID de l'Ordre


Titre


Start State

End State

none

 Add Order

 New Parameter

 Cancel

Nom	Valeur
Nouveau paramètre	Test

Ces paramètres sont les suivants :

- L'identifiant de l'ordre pour le retrouver plus simplement dans l'historique, si il n'est pas indiqué il sera automatiquement fourni par le moteur.
- Le titre permet à l'opérateur d'indiquer un commentaire.
- Le nœud de départ permet de démarrer le scénario à une étape précise, par défaut on démarre à la première étape
- Le nœud de fin indique la sortie du scénario

Un clic sur «Ajout de l'ordre » déclenche la chaîne.

Il est aussi possible d'indiquer des arguments en utilisant le bouton « Nouveau paramètre » et en indiquant directement le nom et sa valeur.

Bloquer une chaîne

Le blocage de la chaîne empêche tout lancement automatique mais permet tout de même les lancements forcés.

Lorsque la chaîne est bloquée, le menu contextuel propose le déblocage.

Localisation	Etat	Statut
<input type="text"/>	<input type="text" value="v"/>	STOPPED v
scheduler		RUNNING
pickup		
Job Chain		STOPPED
reprise	Ajouter un ordre Débloquer la chaîne de traitements	
t1stmt1		
TSM		

Actions sur un ordre

Un ordre est un déclencheur prédéfini, il permet d'indiquer la chaîne ou la portion de chaîne à exécuter, les paramètres et la planification horaire.

Un ordre peut avoir été créé à la volée, dans ce cas on le

distingue par un signe ⚡

dont le + indique l'ajout à posteriori.

Localisation	Etat	Statut
<input type="text"/>	<input type="text"/>	<input type="text"/>
[-] scheduler		RUNNING
[-] pickup		
[-] Job Chain		ACTIVE
+ ⚡ Test (toujours bien)	Fin	SUCCESS
+ ⚡ Reprise sur erreur 1	000	ACTIVATED
+ ⚡ 154		SUCCESS
+ ⚡ 155		SUCCESS
[-] reprise		
[-] Reprise au		SUCCESS

⚡ Déclencher l'ordre

⚡ Suspendre l'ordre

⚡ Historique des ordres

Déclencher un ordre

Dans le cas de l'ordre défini, il est possible de le déclencher par anticipation ou de le suspendre.

La fenêtre de déclenchement par un ordre est assez différente d'un ajout d'ordre :

- les paramètres sont fixés lors de la définition de l'ordre et ne sont donc pas modifiables
- les arguments pré-définis sont affichés et peuvent être modifiés

Start order 10099

Reprise AUTO Statut FAILURE

SCHEDULER-280 Process terminated with exit code 1 (0x1)

Séquence de traitements	reprise/Reprise automatique	Etat	IKO
Heure de départ	2014-02-22 18:29:43	Exit code	1
End Time	2014-02-22 18:34:56	Error	1
Durée	5m 13s	Steps	1

Arguments

HOST INCONNUE

⚡ Déclencher l'ordre now + Nouvel argument Abandon

Nom	Valeur
HOST	INCONNUE

Suspendre un ordre

Cette action permet de bloquer un ordre tant qu'un problème n'est pas résolu. Cette suspension peut intervenir à n'importe quelle étape du processus.

[-] scheduler		RUNNING	2014-02-22 20:39:58
[-] pickup			
[-] Job Chain		ACTIVE	
[-] Reprise sur erreur 1	Fin	SUSPENDED	21:11:25

Lorsque l'ordre est suspendu, le symbole 🛑 indique qu'une opération doit être effectuée pour résoudre le problème.

Débloquer un ordre

L'origine d'une suspension peut être manuelle ou programmée dans un scénario. Dans tous les cas, il n'existe que deux actions possibles : Résumer

- ⚡ Résumer l'ordre
- ⚡ Réinitialiser l'ordre
- ⚡ Historique des ordres

l'ordre ou le réinitialiser.

Dans le premier cas, l'exécution reprends à l'étape en erreur alors que la réinitialisation effectue une reprise à l'étape initiale. Le résumé et la réinitialisation reprenne les arguments fournis lors du premier lancement.

Actions sur une étape

Une étape représente un maillon de la chaîne de traitements, chaque maillon est lié à un traitement. Le statut de l'exécution du traitement donne l'étape suivante, si le traitement est en erreur, l'ordre est dirigé vers une étape d'erreur.

Localisation	Etat	Statut
<input type="text"/>	<input type="text"/>	<input type="text"/>
📁 scheduler		RUNNING
📁 pickup		
📁 Job Chain		ACTIVE
⚡ Reprise sur erreur 1	Fin	WARNING
1 ➡ pickup/Debut	000	SUCCESS
2 ➡ pickup/JobSplitter	Split	SUCCESS
3 ➡ pickup/Premiere	001	FAILURE
4 ➡ pickup/Reprise	00	
⚡ Test	00	
+ ⚡ 192	Erreur	SUCCESS

- Bloquer le noeud de la chaîne
- Sauter le noeud de la chaîne

Bloquer une étape

Bloquer une étape consiste à arrêter l'ordre au moment où il passe par l'étape concernée. Le blocage concernera tous les ordres pour une chaîne donnée, l'étape est donc indiquée en rouge au niveau de la chaîne de traitement. Le statut est STOPPED pour signifier le blocage nécessitant l'action de l'opérateur.

Lorsqu'un ordre est exécuté, il reste bloqué en mode RUNNING jusqu'à ce qu'on intervienne sur l'étape.

Localisation	Etat	Statut
<input type="text"/>	<input type="text"/>	<input type="text"/>
📁 scheduler		RUNNING
📁 pickup		
📁 Job Chain		ACTIVE
• Split		STOPPED
+ ⚡ Reprise sur erreur 1	Fin	SUCCESS
⚡ Test (toujours bien)	Split	RUNNING
1 ➡ pickup/Debut	000	SUCCESS
+ ⚡ 331	Sync	SUCCESS
+ ⚡ 330	Sync	SUCCESS
+ ⚡ 332	Sync	SUCCESS

Débloquer une étape

Le déblocage est l'opération inverse, elle permet à l'ordre de reprendre sa course. Un clic droit sur le nœud bloqué donne accès à l'option « débloquer le nœud ».

L'ordre continue à l'étape suivante.

Localisation	Etat	Statut
<input type="text"/>	<input type="text"/>	<input type="text"/>
scheduler		RUNNING
peakup		
Job Chain		ACTIVE
Split		Sauter le nœud de la chaîne Débloquer le nœud
Reprise sur erreur 1		
Test (toujours bien)		
1 peakup/Debut	000	SUCCESS
331	Sync	SUCCESS
330	Sync	SUCCESS
332	Sync	SUCCESS

Localisation	Etat	Statut
<input type="text"/>	<input type="text"/>	<input type="text"/>
scheduler		RUNNING
peakup		
Job Chain		ACTIVE
Reprise sur erreur 1	Fin	SUCCESS
Test (toujours bien)	Split	RUNNING
1 peakup/Debut	000	SUCCESS
2	Split	RUNNING
331	Sync	SUCCESS

Sauter une étape

Si on souhaite faire passer un ordre à travers une étape, il est possible de la désactiver sans pour autant bloquer la chaîne. Pour réaliser cette opération, on change le statut du nœud en SKIPPED, l'ordre considérera cette étape comme réussie sans que la commande ait été exécutée.

Comme pour le blocage, l'action est sur une étape du scénario, elle impacte l'ensemble des ordres liés à la chaîne. L'étape sautée est indiquée dans la chaîne et dans les ordres l'ayant utilisé.

Localisation	Etat	Statut
<input type="text"/>	<input type="text" value="v"/>	<input type="text" value="v"/>
[-] scheduler		RUNNING
[-] pickup		
[-] Job Chain		ACTIVE
▪ 000		SKIPPED
[-] ⚡ Reprise sur erreur 1	Fin	SUCCESS
1 → pickup/Debut	000	SKIPPED
2 → pickup/JobSplitter	Split	SUCCESS
3 → pickup/JobSync	Sync	SUCCESS
4 → pickup/JobSync	Sync	SUCCESS
5 → pickup/JobSync	Sync	SUCCESS

Réactiver une étape

Le nœud sauté peut être réactivé à tout moment.

Localisation	Etat	Statut
<input type="text"/>	<input type="text" value="v"/>	<input type="text" value="v"/>
[-] scheduler		RUNNING
[-] pickup		
[-] Job Chain		ACTIVE
▪ Split		SKIPPED
[-] ⚡ Reprise sur erreur 1		SUCCESS
1 → pickup/JobSoli		SKIPPED
2 → pickup/JobSync	Sync	SUCCESS
3 → pickup/JobSync	Sync	SUCCESS
4 → pickup/Exit	004	SUCCESS
⚡ Test (toujours bien)	Fin	SUCCESS

Bloquer le noeud de la chaîne

Réactiver le noeud

Plan

Le plan est le calcul et la vérification des prochaines planifications horaire, il est construit grâce à deux traitements :

- le CreateDaysSchedule qui génère le plan théorique
- le CheckDaysSchedule qui vérifie l'exécution à l'heure dite.

Un double clic affiche le détail de la planification.

Localisation	Statut	Exécuté	Délai	Début	Fin	Cycle
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
scheduler RUNNING 2014-02-22 20:39:58						
sos						
dailyschedule						
CreateDaysSchedule						
2014-02-23 23:00:01	LATE		3d			
2014-02-24 23:00:01	LATE		2d			
2014-02-25 23:00:01	LATE		1d			
2014-02-26 23:00:01	LATE		20h			
2014-02-27 23:00:01	WAITING		3h			
2014-02-28 23:00:01	WAITING		1d			
De 1 à 10 Page 1 10 Lignes par page						
Information						
Statut		WAITING				
Schedule Time		2014-02-27 23:00:01				
Heure de départ						
End Time						
Délai		3h				
Planned ID		12197				




Les statuts sont les suivants :

Statut	Description
SUCCESS	Le traitement planifié a bien été exécuté.
WAITING	Le traitement attend l'heure d'exécution.
LATE	L'heure d'exécution est dépassée.

Moteurs

Les moteurs sont automatiquement insérés dans la base de données par le job CreateDaysSchedule. On retrouve les informations nécessaires à la communication entre l'interface et l'ordonnanceur distant.

Moteur	Statut	Heure de départ	Hostname	Port
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
scheduler	RUNNING	2014-02-22 20:39:58	sosparis	4444

 Vérifier le moteur
 Pause
 Redémarrer le moteur
















Pour des environnements plus complexes avec des moteurs écoutant sur une adresse IP particulière, il est préférable d'utiliser le module d'administration qui offre une plus grande précision dans la définition des objets.




Actions

Le premier type d'objet est le moteur, les actions possibles sont :

- Vérifier l'état du moteur
- Le mettre en pause
- Le redémarrer

Ce menu est accessible dans les écrans de traitements et d'ordres.

Localisation	Statut	Heure de départ
<input type="text"/>	<input type="text"/>	<input type="text"/>
 scheduler	RUNNING	2014-02-12 07:52:36
 Jade		14-02-11 11:08:02
 LogAnalyser		09:36:12
 peakup		
 ping	SUCCESS 	2014-02-10 08:41:23
 Receive File	FAILURE 	08:54:00
 running job	RUNNING 	
 scheduler_sosftp_import	SUCCESS 	20:36:06
 Send File	FAILURE 	2014-02-11 17:26:41
 sos		

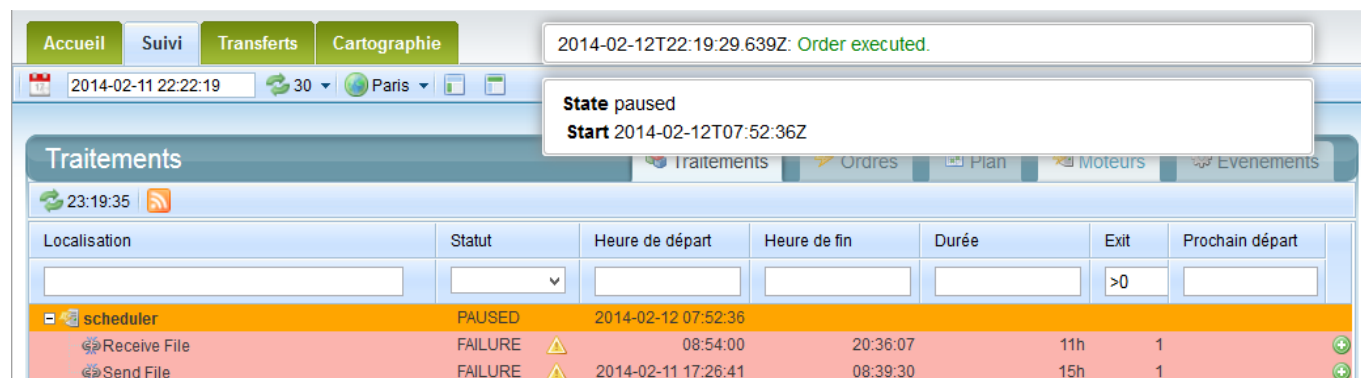
 Vérifier le moteur
 Mettre le moteur en pause
 Redémarrer le moteur

Vérifier le moteur

La vérification du moteur consiste à envoyer une demande simple et attendre le retour, si le moteur ne répond plus, son statut passe à l'état STOPPED.

Mise en pause

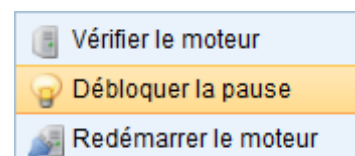
La mise en pause du moteur envoie une demande de mise en pause au moteur, la réponse est indiquée dans un deuxième pop up. Ces messages s'empilent et restent affichés à l'écran quelques secondes, un clic permet de les fermer immédiatement.



Localisation	Statut	Heure de départ	Heure de fin	Durée	Exit	Prochain départ
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	>0	<input type="text"/>
scheduler	PAUSED	2014-02-12 07:52:36				
Receive File	FAILURE	08:54:00	20:36:07	11h	1	
Send File	FAILURE	2014-02-11 17:26:41	08:39:30	15h	1	

Mise en pause

Le menu change en fonction de l'état de l'objet, si le moteur est en pause, l'action de pause est remplacée par le déblocage.



Vérifier le moteur

Débloquer la pause

Redémarrer le moteur

Redémarrer le moteur

Le redémarrage du moteur est une demande d'arrêt sans interruption brutale des traitements. Le moteur va attendre la fin des traitements mais ne prendra plus aucune demande, cet arrêt peut donc être assez long.

Attention ! Le redémarrage sur Unix est tributaire du délai de libération d'un port, il est généralement de 2 minutes pour Linux mais va jusqu'à 10 minutes sur AIX.

Événements

Les événements sont utilisés dans le cadre de l'orchestration pour définir des règles entre les chaînes de traitements d'un parc d'ordonnanceur.

Le principe est le suivant :

- on ajoute l'envoi d'événements aux traitements concernée
- on crée les règles de gestion des événements

Lorsqu'un traitement est terminé, l'événement est envoyé au moteur qui vérifie si la règle peut être exécutée.

Event	Moteur	Séquence de traitements	Ordre	Traitement	Exit	Creation
<input type="text"/>	<input type="text" value="v"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
PING						
Reprise	scheduler	Reprise automatique	Reprise AUTO	ping	0	2014-02-24 18:05:15
test						
De1 à 3 Page 1 10 Lignes par page						
Class	test		Creation	2014-02-25 13:08:00		
Id	2		Expires	2014-02-26 01:08:04		
Ordre	Test		Moteur	scheduler		
Séquence de traitements	Job Chain		Hôte	sosparis		
Traitement	Job Splitter		Port	4444		
Exit code	0					

Ce mécanisme est particulièrement efficace pour déclencher des traitements en fonction de règles complexes incluant des opérateurs ET et OU, à travers différents moteurs et/ou différents applications.

Messages

Les messages sont extraits du journal du serveur et stockés dans la base de données pour être supervisés par un outil de surveillance comme Nagios. Le principe du connecteur consiste à exécuter régulièrement une requête dans cette table pour calculer le statut du service dans la console de supervision.

L'extraction est réalisée par le traitement LogAnalyser.

Le contenu de cette table est publié afin d'identifier rapidement un problème. Il permet de retrouver une erreur ou un simple avertissement et de fournir les objets concernés.

Un double clic sur le message affiche le détail de l'erreur.

Log time	Moteur	Severity	Error	Message	Objects	Count
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
2014-02-27 04:10:49	scheduler	WARN	292	Missing from, to, subject or body. Suppressing email		0
2014-02-27 04:10:14	scheduler	WARN	292	Missing from, to, subject or body. Suppressing email		0
2014-02-27 04:10:06	scheduler	WARN	292	Missing from, to, subject or body. Suppressing email		0
2014-02-27 04:03:18	scheduler	WARN	6	ERRNO-2 No such file or directory [opendir] [files]	/scheduler_sosftp_history_file_order	0
2014-02-27 04:03:18	scheduler	WARN	292	Missing from, to, subject or body. Suppressing email		0
2014-02-27 04:03:16	scheduler	WARN	292	Missing from, to, subject or body. Suppressing email		0
2014-02-27 04:03:11	scheduler	WARN	399	Error in <include file="jobs/JobChainSplitter.xml"/>:	sos/jiti/JobChainSplitter	0
2014-02-27 04:03:10	scheduler	ERROR	161	There is no schedule '/reprise/jeudi_19h'	Reprise AUTO "Test de reprise"	0
2014-02-27 04:03:10	scheduler	ERROR	161	There is no schedule '/reprise/lundi_18h'	Tout va bien "Test de reprise"	0
2014-02-27 04:02:28	scheduler	WARN	289	TCP-port 4444 is blocked. Trying a further 120 seconds befo		0

De 1 à 14

Page 1

20 Lignes par page

Log time	2014-02-27 04:03:10	Message	3
Statut	ERROR	Count	0
Message	2014-02-27 04:03:10.055+0100 [ERROR] (Order Reprise AUTO "Test de reprise") SCHEDULER-161 There is no schedule '/reprise/jeudi_19h'		
Moteur	scheduler		
Séquence de traitements	Reprise AUTO "Test de reprise"		

Les filtres affichent directement l'information.

Log time	Moteur	Severity	Error	Message	Objects	Count
<input type="text"/>	<input type="text"/>	ERRO	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
2014-02-27 04:03:10	scheduler	ERROR	161	There is no schedule '/reprise/jeudi_19h'	Reprise AUTO "Test de reprise"	0
2014-02-27 04:03:10	scheduler	ERROR	161	There is no schedule '/reprise/lundi_18h'	Tout va bien "Test de reprise"	0

Focus

Focus est le module de synchronisation, son rôle principal est de mettre à jour les données des moteurs dans la base d'Ari'i afin d'avoir une vision globale des traitements.

L'interface native de JobScheduler renvoie les informations en temps réel lorsque l'utilisateur en fait la demande ou lors des rafraichissements automatique. Lorsqu'une équipe travaille avec le JobScheduler, le moteur renvoie les informations pour chacun utilisateur ce qui provoque une forte consommation des ressources au détriment des fonctions primaires de l'ordonnanceur.



Principe

Focus simule un utilisateur et communique avec le moteur pour récupérer un état global du système, cet état sera ensuite utilisé pour renvoyer les réponses aux différents utilisateurs. Quelque soit ce nombre d'utilisateur, une seule connexion à intervalle régulier est nécessaire pour satisfaire l'ensemble des besoins.

Etat des moteurs

La synchronisation passent par plusieurs canaux afin d'avoir toujours au moins une source active :

- L'Automate Arii récupère les informations en mode polling sur les automates distants
- Chaque automate renvoie ces propres informations vers le centralisateur
- L'utilisateur peut forcer une synchronisation à partir de l'interface
- Chaque action de l'utilisateur déclenche une synchronisation forcée.

Le but est d'avoir toujours un mécanisme de synchronisation actif. Pour éviter que plusieurs sources mettent à jour les mêmes données, le centralisateur rejette la demande si une synchronisation est déjà en cours ou si les données ont été rafraîchies dans les 5 dernières secondes.

Base de données

Les données du moteur ne concernent que la partie temps réel de l'automate, elles ne concernent donc que l'état présent et futur des traitements, c'est à dire ce qui est prête à être traité ou en cours de traitements.

Les informations concernant les traitements exécutés ou en cours d'exécution sont stockés dans la base de données. Cette base va permettre d'extraire deux types d'informations :

- les derniers statuts d'exécution
- la durée d'exécution moyenne

L'ensemble des informations permet de disposer d'une chronologie complète.

Visualisation

Les informations passé/présent/futur permettent d'obtenir des vues chronologiques complètes comme le Gantt.

Opérations

Le module Opérations est une version centralisée du « Job Opérations Center » de l'automate. Il repose entièrement sur les données synchronisées par le module Focus. Il est donc nécessaire d'avoir mis en place l'ensemble des traitements d'envoi et de récupération des données.

Le principe est de retrouver l'ergonomie de l'interface et d'apporter les évolutions suivantes :

- capacité à gérer un groupe d'automate
- amélioration des performances
- communication optimale avec les moteurs



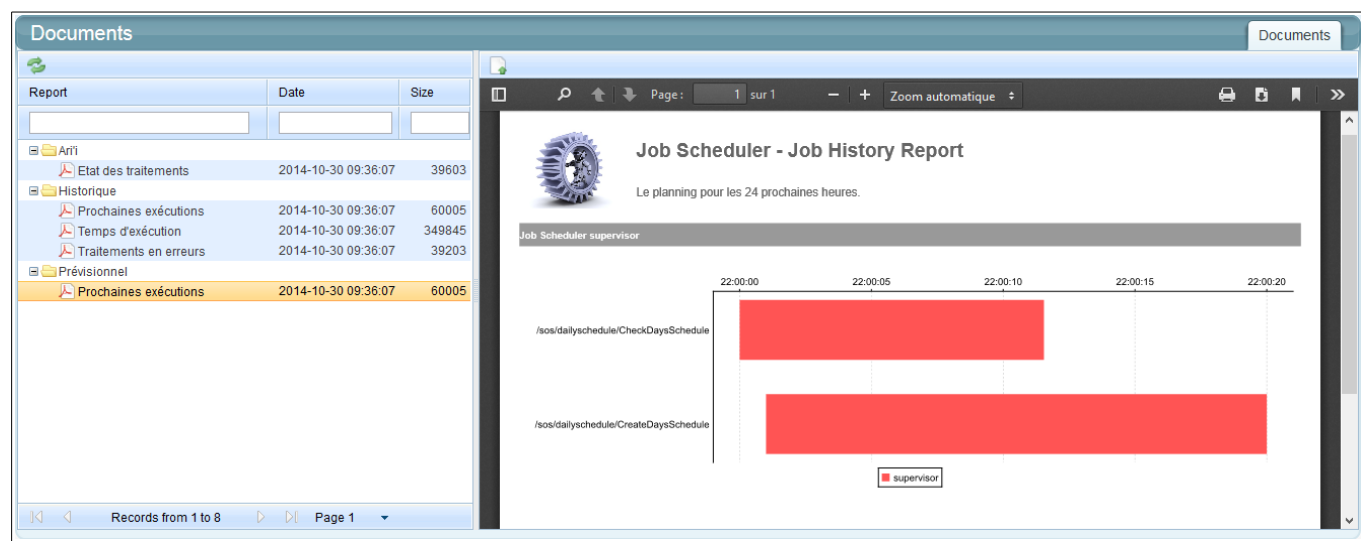
Configuration

La synchronisation est réalisée par des appels d'url exécutés par des traitements cycliques de JobScheduler.

Ergonomie

Rapport

Tout développeur PHP peut participer au développement des modules du portail, ceci sera d'autant plus simple si il connaît déjà le framework Symfony2.



Cartographie

Le module de cartographie offre une visualisation graphique des éléments contenus dans le répertoire live et remote de l'ordonnanceur.

Ce module est l'interface d'un script perl utilisable dans un shell unix ou une commande Windows. Ce qui permet d'automatiser la génération de graphiques sans passer par l'interface web.

Ce module utilise Graphviz qui est un outil de modélisation à partir d'un langage simple en mode texte, pour plus d'informations :

<http://graphviz.org>



Configuration

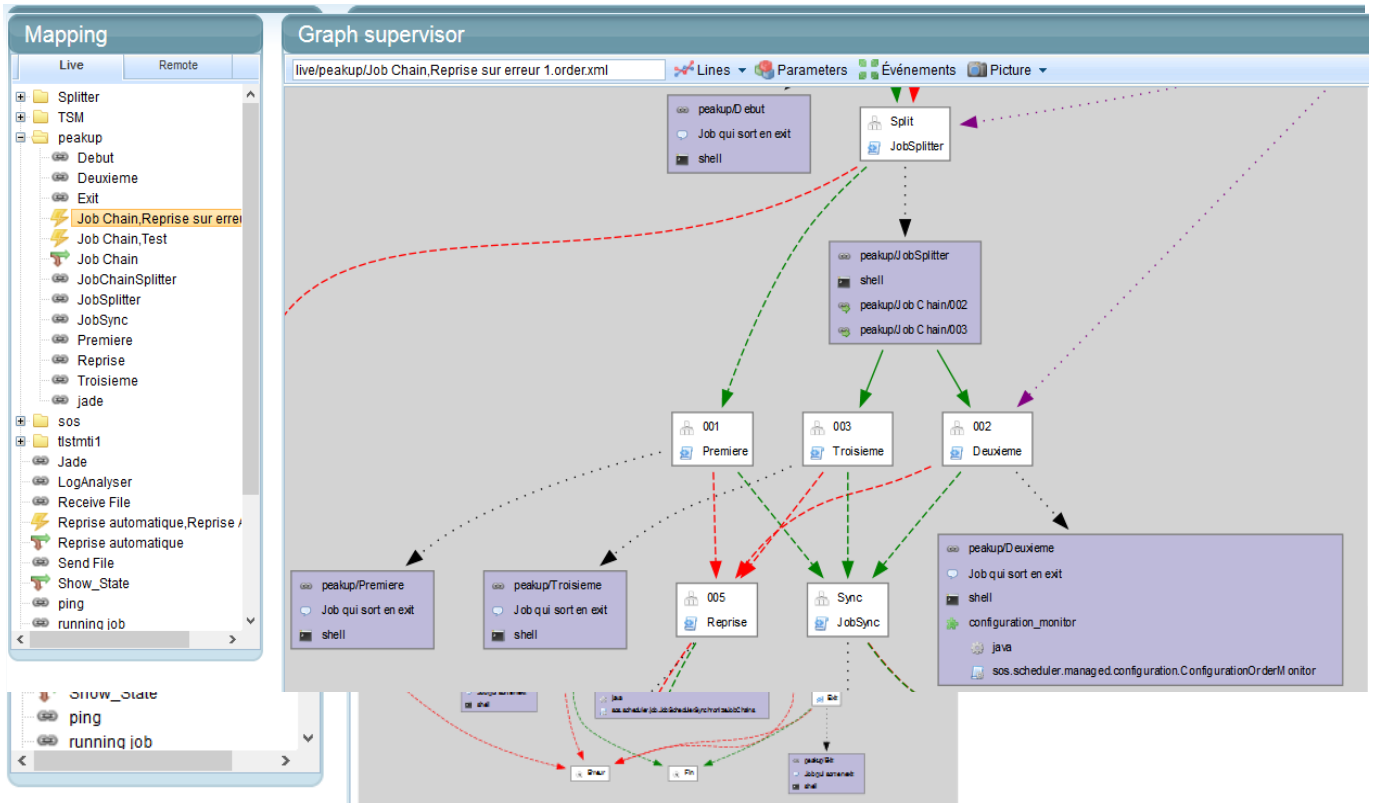
La configuration consiste à indiquer l'emplacement des différents éléments.

Paramètre	Valeur par défaut
perl	/usr/bin/perl
osjs_config	/home/osjs/jobscheduler/scheduler/config
graphviz_dot	/usr/bin/dot
graphviz_images	/usr/share/symfony/web/arii/images/silk

Utilisation

La manipulation des graphiques, que ce soit le déplacement sur l'écran ou le zoom, se fait à la souris.

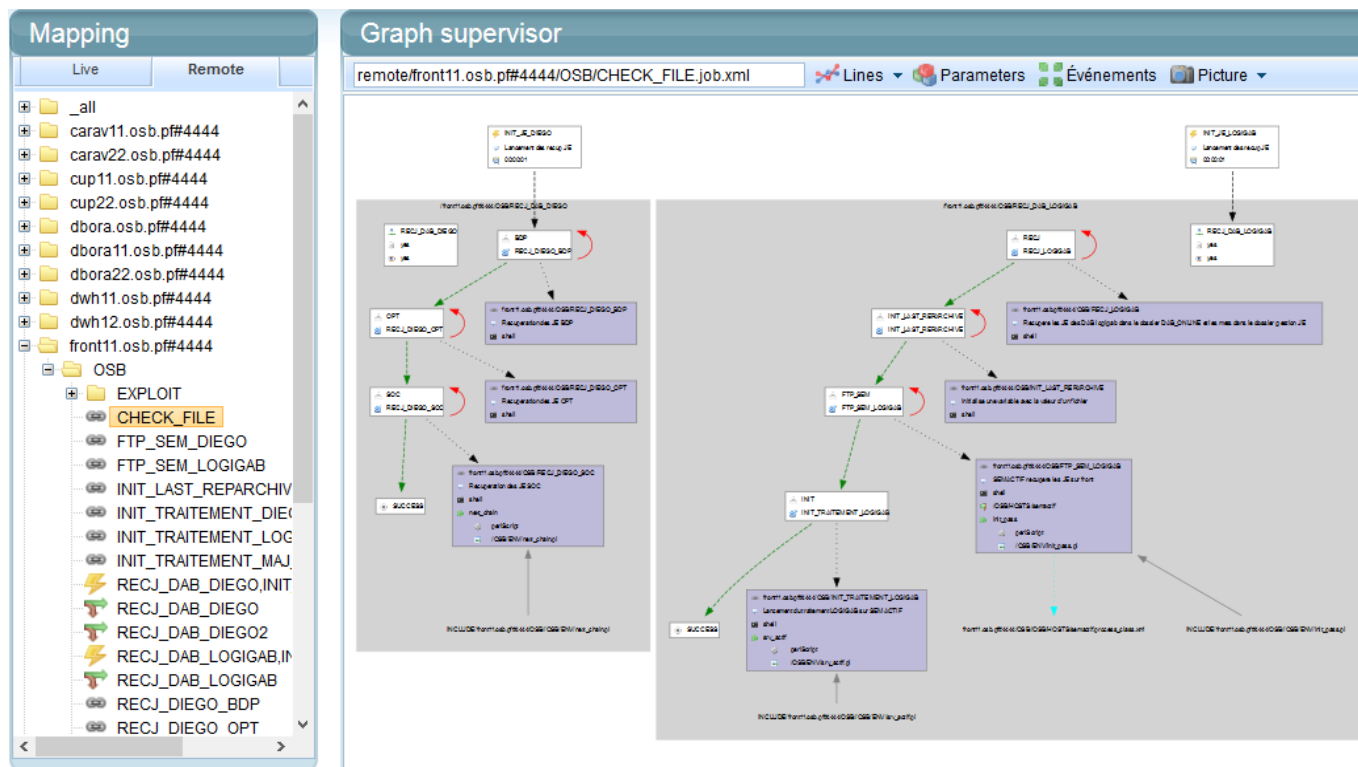
La molette de la souris permet de contrôler le zoom.



Répertoires

On considère le répertoire « live » et le répertoire « remote » présents dans la configuration de l'ordonnanceur, ces répertoires sont accessibles par les onglets dans le menu de gauche. Le répertoire live contient les objets pris en compte par le moteur local alors que remote contient les objets qui seront déployés sur les moteurs distants.

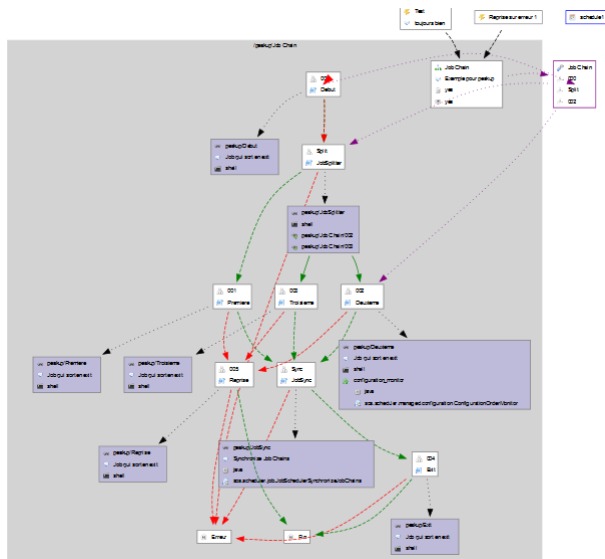
Pour visualiser l'ensemble des objets distribués sur le parc, il est nécessaire de connecter l'interface web au superviseur.



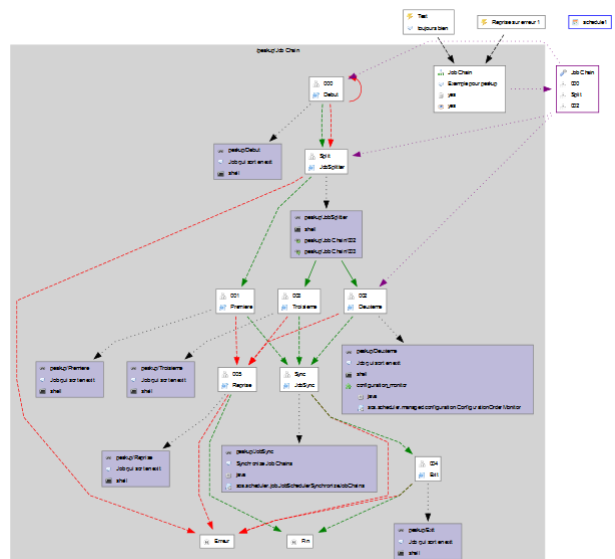
Lignes

Un clic sur le bouton  permet de choisir le type de ligne.

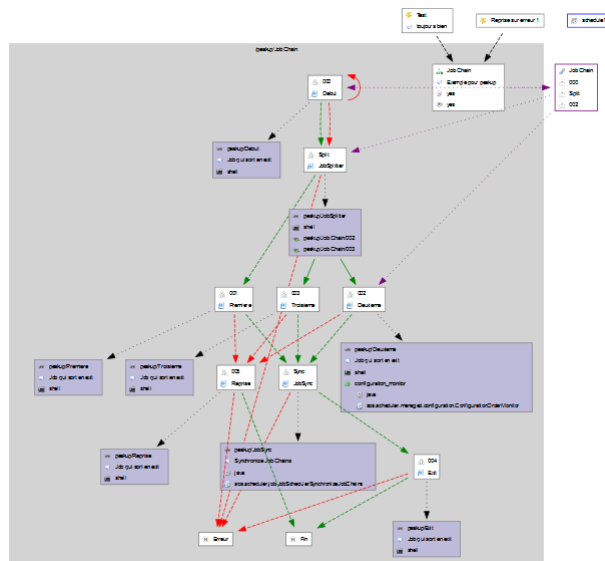
Courbes



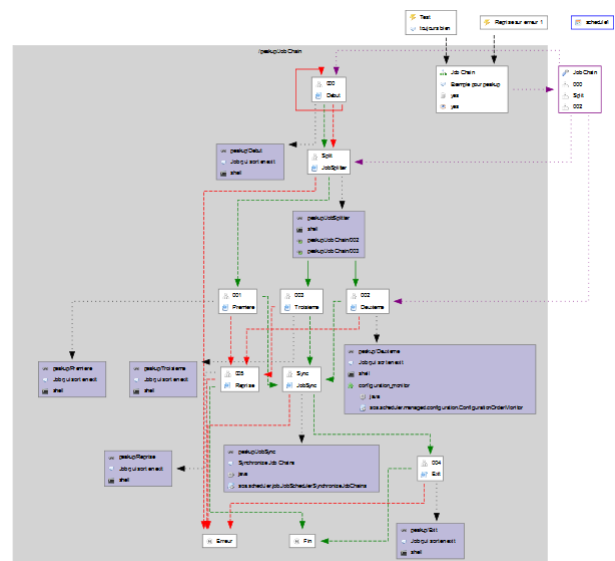
Lignes brisées



Lignes directes



Lignes perpendiculaires



Le dernier choix permet de supprimer les lignes, ce choix est pratique en phase de conception.

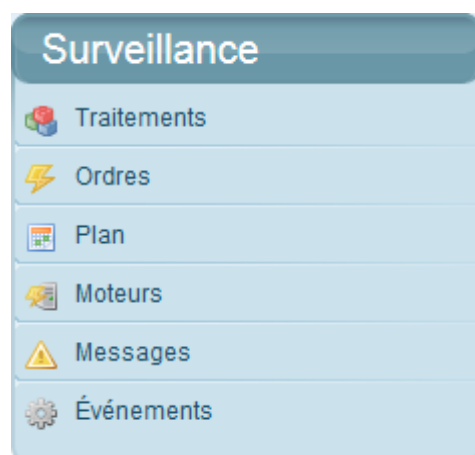
Suivi d'exploitation

Ce module permet de suivre les traitements passés et en cours d'exécution. Il est directement connecté à la base de données et ne sollicite le moteur qu'au moment de l'exécution des commandes évitant ainsi la consommation de ressources. Ce module est dédié au pilotage ou aux utilisateurs souhaitant visualiser l'état des traitements.

Le suivi reprend la configuration du portail pour se connecter sur la base de données et communiquer avec le moteur d'ordonnancement.



Ergonomie



Open Source JobScheduler stocke dans la base de données les historiques d'exécutions. Il stocke aussi les états afin de reprendre les processus en cas d'arrêt imprévu du système.

Le module de suivi reprend l'ensemble de ces données pour les afficher sous forme de graphiques, de tableaux ou bien encore de chronogrammes.

La vue calendrier permet de visualiser par rapport à la date de référence. Pour rappel, la date de référence est celle indiquée dans la barre d'outil du portail.

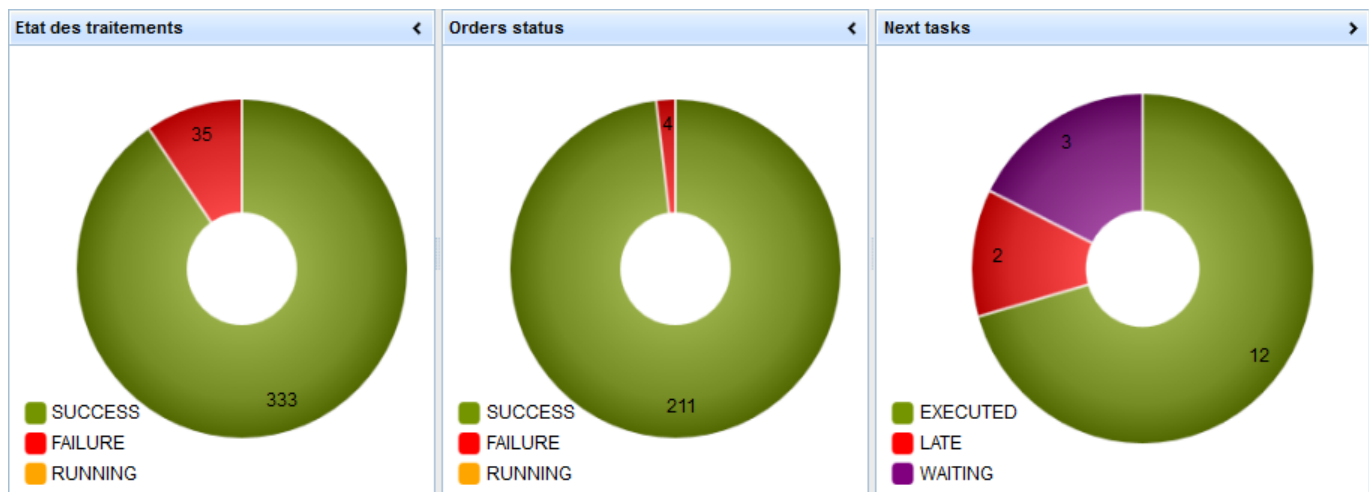
Dans l'exemple ci-contre, on ne prendra en compte que les données datant de 4 jours avant la date de référence jusqu'à 2 jours dans le futur.

La vue calendrier permet de spécifier la plage de dates à



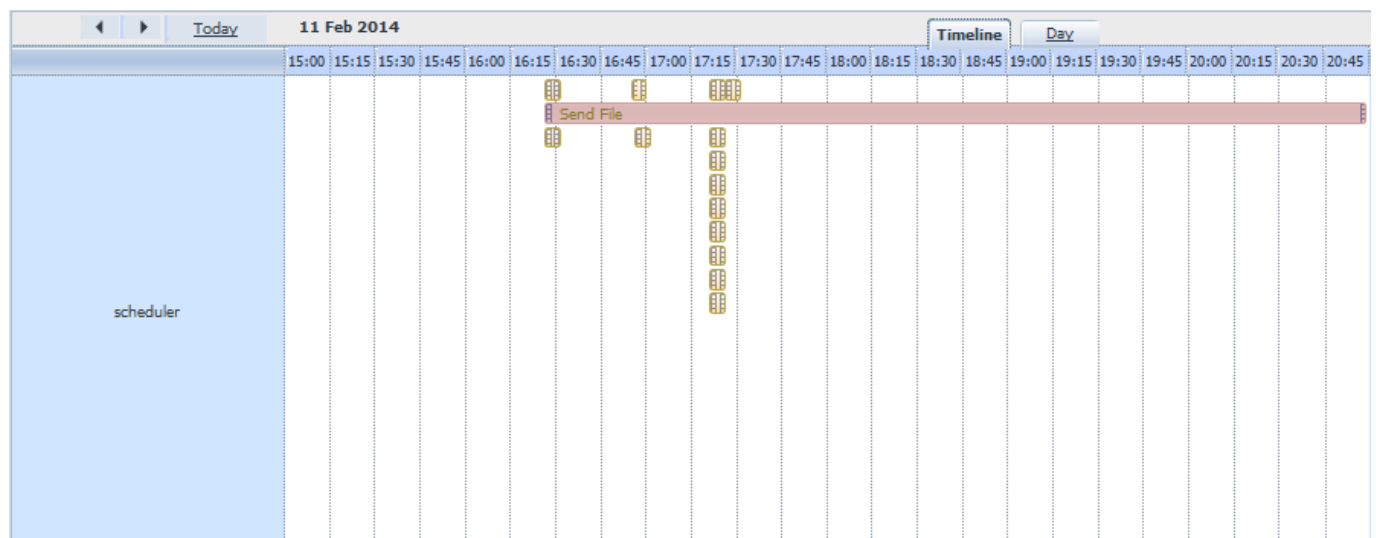
Vue globale

La vue globale permet d'avoir une vision graphique des traitements sur la période concernée. On affiche les 3 grands domaines : les traitements (indépendants ou ordonnés), les ordres et les prochaines planifications.



Attention ! Il s'agit d'un historique, il est donc possible que les traitements en erreur aient été corrigés au moment de la visualisation. Pour état en temps réel, il est nécessaire d'utiliser le module **Opérations**.

Ce graphique est complété par un chronogramme qui donne l'état des traitements sur une ligne de temps.




Traitements

La zone de visualisation des traitements est découpée en deux zones : une zone de données et une zone de détail qui donne accès aux informations du traitement courant, à son dernier journal d'exécution et à son historique.





Les traitements sont des tâches informatiques indépendantes ou utilisées dans des séquences. On les

distingue par l'icone utilisé, le symbole  signifie que le traitement est indépendant, alors qu'un

traitement utilisé dans une chaine sera symbolisé comme suit 

Chaque colonne peut être filtrée pour un accès rapide à l'information. Ces filtres sont de 3 types : texte, liste déroulante et nombre. Le premier cas permet de retrouver un traitement par une partie de son nom, la liste déroulante sert à la sélection d'un statut et le dernier filtre permet de sélectionner un exit code.


























L'organisation reprend l'arborescence des fichiers, on distingue trois niveaux : le moteur, les répertoires et

Localisation	Statut	Heure de départ	Heure de fin
<input type="text" value="ftp"/>	<input type="text" value="v"/>	<input type="text"/>	<input type="text"/>
 scheduler	RUNNING 	2014-02-12 07:52:36	
 scheduler_sosftp_import	SUCCESS 	20:36:06	20:41:15

les traitements.

Un quatrième niveau est ajouté lorsque le traitement est en cours, il est représenté par un suivi du numéro de processus sur la machine concernée. Dans l'exmple suivant, le traitement running job est en cours d'exécution avec le PID 17396.



Localisation	Statut	Heure de départ	Heure de fin	Durée	Exit	Prochain départ
<input type="text"/>	<input type="text" value="v"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
 scheduler	RUNNING 	2014-02-12 07:52:36				
 Jade	SUCCESS 	2014-02-11 11:08:02	11:08:05	3s	0	
 LogAnalyser	SUCCESS 	09:36:12	09:36:13	1s	0	
 peakup						
 ping	SUCCESS 	2014-02-10 08:41:23	08:41:24	1s	0	
 Receive File	FAILURE 	08:54:00	20:36:07	11h	1	
 running job	RUNNING 					
 17396		21:09:49		1h	0	
 scheduler_sosftp_import	SUCCESS 	20:36:06	20:41:15	5m 9s	0	
 Send File	FAILURE 	2014-02-11 17:26:41	08:39:30	15h	1	

De 1 à 10 Page 1 10 Lignes par page






Par défaut, on affiche des pages de 10 lignes et on navigue de pages en pages. Le système de filtre permet généralement de limiter les traitements concernés à quelques lignes pour ne travailler que sur un groupe de traitements ou en fonction d'un statut particulier.

Pour basculer sur une mode liste, il suffit de changer le nombre de lignes par page. Un ascenseur permet de se déplacer dans la liste.




Localisation	Statut	Heure de départ	Heure de fin	Durée	Exit	Prochain départ
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
LogAnalyser	SUCCESS	09:36:12	09:36:13	1s	0	
peakup						
ping	SUC	08:41:23	08:41:24	1s	0	
Receive File	FAI	08:54:00	20:36:07	11h	1	
running job	RUN					
17396		21:09:49		1h	0	
scheduler_sosftp_import	SUC	20:36:06	20:41:15	5m 9s	0	
Send File	FAI	11:26:41	08:39:30	15h	1	
sos						
ttstmr1						

La taille des fenêtres peut être ajustées en déplaçant les cloisons avec la souris. On peut ainsi agrandir la fenêtre de données au détriment de la zone de détail.

Le statut est indiqué de 3 manières : le nom, le symbole et le code couleur. Ceci permet de distinguer rapidement l'état du système.

Statut	Icône	Couleur	Explication
SUCCESS		Vert	L'exécution du traitement s'est bien déroulée.
FAILURE		Rouge clair	L'exécution s'est terminée avec une erreur, l'exit code est supérieur à 0.
RUNNING		Orange	Le traitement est en cours d'exécution.
STOPPED		Rouge foncé	Le traitement s'est terminé en erreur et un « Stop sur erreur » est actif. Une action est nécessaire pour le débloquent.
ACTIVATED		Bleu	L'ordre est défini et stocké en base de données mais n'a jamais été exécuté.

La zone de filtre indique les statuts en cours, en sélectionnant le statut FAILURE, on accède à l'ensemble des traitements en erreur.

Localisation	Statut	Heure de départ
<input type="text"/>	FAILURE <input type="text"/>	<input type="text"/>
scheduler	RUNNING 	2014-02-12 07:52:36
Receive File	FAILURE 	08:54:00
Send File	FAILURE 	2014-02-11 17:26:41

La colonne Exit correspond à l'exit code renvoyé par le shell. L'utilisation du filtre avec un opérateur permet de retrouver plus précisément les traitements en erreur par le code d'exit.

Les colonnes heure de départ, heure de fin et durée permettent de situer le traitement dans le temps.






Durée	Exit	Prochain départ
<input type="text"/>	<input type="text" value=">0"/>	<input type="text"/>
	11h	1
	15h	1

Le filtre permet de retrouver les traitements pour une date précise :

Localisation	Statut	Heure de départ	Heure de fin	Durée	Exit	Prochain départ	
<input type="text"/>	<input type="text" value="v"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	2014-02	
scheduler RUNNING 2014-02-12 07:52:36							
sos							
dailyschedule							
CreateDaysSchedule	SUCCESS	08:39:28	08:39:37	9s	0	2014-02-13 00:00:00	
update							
scheduler_check_updates	SUCCESS	2014-02-10 20:00:01	20:00:34	33s	0	2014-02-17 20:00:00	

Si le traitement est planifié, la colonne « Prochain départ » indique l'heure de lancement.

La dernière colonne indique le type de lancement du traitement :

Icone	Type	Explication
	Manuel	Le traitement a été mis en queue par une action manuelle. Le traitement est généralement un job indépendant.
	Ordonné	Le traitement est dans une chaîne, il sera exécuté par un ordre. Le traitement est généralement ordonné.
	Planifié	Les paramètres de date et d'heure sont associés au traitement.
	Cyclique	Le traitement est exécuté en fonction d'une fréquence définie.
	Instancié	Le traitement à un paramètre «tâches min. » qui indique le nombre de tâches concurrentes à exécuter systématiquement.

Informations

Un double clic suffit pour accéder aux informations du traitement. Les types d'informations sont identifiées par des onglets : Le détail de l'exécution, le journal et l'historique du traitement.

Le détail les paramètres et le dernier message du moteur concernant le traitement.

Localisation	Statut	Heure de départ	Heure de fin	Durée	Exit	Prochain départ	
<input type="text"/>	FAILURE	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
scheduler RUNNING 2014-02-12 07:52:36							
Receive File	FAILURE	08:54:00	20:36:07	11h	1		
Send File	FAILURE	2014-02-11 17:26:41	08:39:30	15h	1		
De 1 à 3 Page 1 20 Lignes par page							
Information	Log	Historique					
Job name	Receive File						
Statut	FAILURE						
SCHEDULER-140 Task logged error: ERROR [main] (SOSSFTP.java:778) - error occurred writing file [C:\Users\leric\AppData\Local\Temp\sos5455985043806183858.gz]: Unexpected end of sftp stream.							
Cause	queue_at			Task id	23415		
Heure de départ	2014-02-12 07:54:00			Exit code	1		
End Time	2014-02-12 19:36:07			Error	1		
Durée	11h			Steps	1		
Parameters							
ftp_host	arii.org						
ssh_auth_method	password						

Le journal permet d'afficher les informations du moteur et du traitement lors de l'exécution.

Information	Log	Historique		
Date	Type	Message	Code	
2014-02-12 08:54:00.272	I	state=starting (at=2014-02-12 08:53:59.803+0100)	918	
2014-02-12 08:54:01.295	I	[main] (SOSMsg.java:126) - LOG_I_0020: JobSchedulerLog4JAppender is configured as log4j-appender		
2014-02-12 08:54:01.297	I	[main] (JobSchedulerJobAdapter.java:117) - 1.5.4014 (2014-01-14T22:31:12.765+0100) Copyright 2003-2013 SOS GmbH Berlin		
2014-02-12 08:54:01.300	I	\$Id: JobSchedulerFTPSend.java 16869 2012-03-26 10:08:16Z oh \$		
2014-02-12 08:54:01.300	I	1.5.4014 (2014-01-14T22:31:12.765+0100) Copyright 2003-2013 SOS GmbH Berlin		
2014-02-12 08:54:01.319	I	[main] (SOSMsg.java:126) - JSJ_I_0010: JobSchedulerJobAdapter::getNodeName: Current Job is 'Receive File'.		
2014-02-12 08:54:01.445	I			
2014-02-12 08:54:01.445	I	**		
2014-02-12 08:54:01.445	I	**		
2014-02-12 08:54:01.445	I	* ----- *		
2014-02-12 08:54:01.445	I	* SOSFTP - Managed File Transfer Utility *		
2014-02-12 08:54:01.445	I	*****		
2014-02-12 08:54:01.445	I	*****		
2014-02-12 08:54:01.445	I	*****		

Pour une lecture rapide, on peut filtrer par type d'information pour extraire directement les avertissements ou les erreurs.

Information	Log	Historique		
Date	Type	Message	Code	
2014-02-12 18:54:34.066	E	[main] (SOSSFTP.java:778) - error occurred writing file [C:\Users\leric\AppData\Local\Temp\sos5455985043806183858.gz]: Unexpected er		
2014-02-12 18:54:34.068	E	[main] (SOSSFTP.java:778) - Error during putFile: com.sos.JSHelper.Exceptions.JobSchedulerException: error occurred writing file [C:\Use		

Le dernier type de message est l'historique des exécutions pour identifier la fréquence des erreurs.

Information	Log	Historique			
Heure de départ	Heure de fin	Durée	Error	Exit code	Message
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="v"/>	<input type="text" value="v"/>	<input type="text"/>
2014-02-10 15:21:34	15:26:35	5m 1s	0	0	
2014-02-07 17:13:56	17:20:00	6m 4s	0	0	
2014-02-07 16:39:27	16:39:28	1s	1	1	SCHEDULER-140 Task logged error: error occurred connect to database failed: sos.connection.s
2014-02-07 16:36:28	16:36:29	1s	1	1	SCHEDULER-140 Task logged error: error occurred connect to database failed: sos.connection.s
2014-02-07 16:26:39	16:31:40	5m 1s	0	0	
2014-02-07 15:54:17	15:59:18	5m 1s	0	0	
2014-02-07 15:50:36	15:52:24	1m 48s	0	0	

Pour un diagnostic plus précis sur l'historique, il faut se référer à la partie « Historique des traitements ».

Barre d'outil

Export RSS

Le bouton RSS de la barre d'outil permet de générer le flux RSS des derniers traitements en erreurs.



S'abonner à ce flux en utilisant



Marque-pages dynamiques

☐ Toujours utiliser Marque-pages dynamiques pour s'abonner aux flux.

S'abonner maintenant

Job failures

Last jobs failed

2014-02-13 07:59:19 [scheduler]/running job (exit 99)

SCHEDULER-280 Process terminated with exit code 99 (0x63)

2014-02-13 07:39:55 [scheduler]/TSM/AF/FS/ACC/tlsqstm1 /ARC_G000_FS_SAR (exit 1)

SCHEDULER-280 Process terminated with exit code 1 (0x1)

2014-02-13 07:39:54 [scheduler]/TSM/AF/FS/ACC/tlsrhps9 /ARC_G000_FS_SAR (exit 1)

SCHEDULER-280 Process terminated with exit code 1 (0x1)

2014-02-13 07:39:54 [scheduler]/TSM/AF/FS/ACC/tlsrnaplx04 /ARC_G000_FS_SAR (exit 1)

SCHEDULER-280 Process terminated with exit code 1 (0x1)

2014-02-13 07:39:53 [scheduler]/TSM/AF/FS/DEV/tlsdlhs1

Menu contextuel

Un menu contextuel est associé à un type d'objet et à son statut, ce système limite la liste des actions pour n'afficher que celles qui sont utiles par rapport à une situation particulière. Le menu contextuel est accessible avec un clic droit après avoir sélectionné l'objet.

La communication entre l'interface et le moteur d'ordonnancement est toujours en deux temps :

- la commande est envoyée au moteur
- le moteur renvoie une réponse et exécute la commande


Pour les actions sur le moteur, il faut se reporter au paragraphe *Moteurs p66*.

Actions sur le traitement



















Tout comme pour le moteur, les actions sur le traitement dépendent de son statut et de son type. Le cas le plus courant est un traitement indépendant en attente d'exécution, un clic droit donnera droit à trois actions possibles :

- démarrer le traitement
- le bloquer avec un STOP
- voir l'historique d'exécution

Lancer un traitement

Pour démarrer un traitement il faut  sélectionner un traitement indépendant, accéder au menu par un clic droit et choisir l'action : démarrer la tâche.

Un traitement ordonné ne peut être lancé qu'à travers une chaîne de traitements.

Localisation	Statut	Heure de départ	Heure de fin
<input type="text"/>	<input type="text" value="v"/>	<input type="text"/>	<input type="text"/>
 scheduler	RUNNING 	2014-02-12 07:52:36	
 pickup			
 Jade	SUCCESS 	2014-02-11 11:08:02	11:08:05
 LogAnalyser	SUCCESS 	6:12	09:36:13
 ping	SUCCESS 	9:44	21:39:48
 Receive File	FAILURE 	4:00	20:36:07
 running job	RUNNING 		
 2808		23:32:14	
 scheduler_sosftp_import	SUCCESS 	20:36:06	20:41:15
 Send File	FAILURE 	2014-02-11 17:26:41	08:39:30

Si le traitement est en exécution, l'action de kill est ajouté. On pourra éventuellement démarrer une nouvelle instance si sa définition le permet.

Arrêter un traitement en cours

Lorsque le traitement est en cours, une nouvelle action apparaît : Killer la tâche.

En sélectionnant cette tâche, une demande d'arrêt est envoyé au système.

Localisation	Statut	Heure de départ	Heure de fin
<input type="text"/>	<input type="text" value="v"/>	<input type="text"/>	<input type="text"/>
[-] scheduler	RUNNING	2014-02-12 07:52:36	
[-] pickup			
[-] Jade	SUCCESS	2014-02-11 11:08:02	11:08:05
[-] LogAnalyser	SUCCESS	09:36:12	09:36:13
[-] ping	SUCCESS	21:39:44	21:39:48
[-] Receive File	FAILURE	08:54:00	20:36:07
[-] running job	RUNNING		
[-] 2808		23:32:14	
[-] scheduler_sosftp_import	SUCCESS	20:36:06	20:41:15
[-] Send File	FAILURE	17:26:41	08:39:30
<div>Start task Kill task Stop Job history</div>			
De 1 à 10 Page 1 1			

Stopper un traitement

Un stop consiste à invalider un traitement, généralement pour indiquer qu'il est en erreur ou peut entraîner une erreur.

Localisation	Statut	Heure de départ	Heure de fin
<input type="text"/>	<input type="text" value="v"/>	<input type="text"/>	<input type="text"/>
[-] scheduler	RUNNING	2014-02-12 07:52:36	
[-] pickup			
[-] Jade	SUCCESS	2014-02-11 11:08:02	11:08:05
[-] LogAnalyser	SUCCESS	09:36:12	09:36:13
[-] ping	SUCCESS	21:39:44	21:39:48
[-] Receive File	FAILURE	08:54:00	20:36:07
[-] running job	RUNNING		
[-] 2808		23:32:14	
[-] scheduler_sosftp_import		20:36:06	20:41:15
[-] Send File		2014-02-11 17:26:41	08:39:30
<div>Stop Job history</div>			

Si on a stoppé le traitement, on pourra tout de même le démarrer ou le débloquent pour une prochaine exécution.

Localisation	Statut	Heure de départ	Heure de fin
<input type="text"/>	<input type="text" value="v"/>	<input type="text"/>	<input type="text"/>
scheduler	RUNNING	2014-02-12 07:52:36	
LogAnalyser	STOPPED	00:36:12	09:36:13
peakup			
Jade	SUCCESS	00:02	11:08:05
ping	SUCCESS	00:44	21:39:48
Receive File	FAILURE	08:54:00	20:36:07
running job	SUCCESS	23:32:14	23:34:43
scheduler_sosftp_import	SUCCESS	20:36:06	20:41:15
Send File	FAILURE	2014-02-11 17:26:41	08:39:30
sos			

Start task

 Unstop

 Job history

L'historique du traitement permet d'accéder à un écran dédié.

Diagnostic

Le diagnostic interroge le moteur sur l'état d'un job afin de connaître la raison d'un blocage.

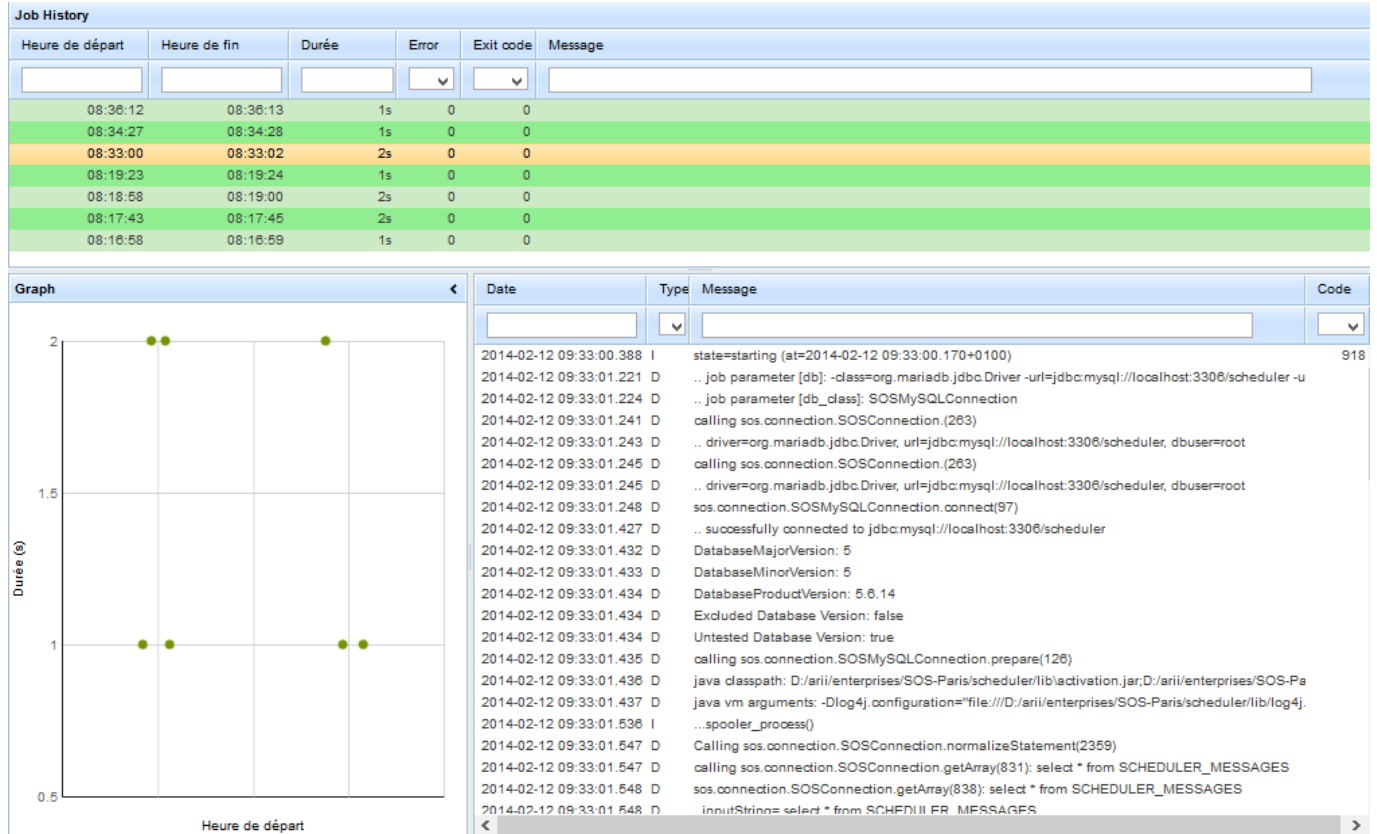
La cause du problème est indiquée sous les lignes rouges.

Dans l'exemple ci-contre, le job est dans une chaîne de traitements dont l'unique ordre est suspendu à l'étape 002. Pour débloquent cette situation, il sera nécessaire de réactiver l'ordre.

Diagnostic	
	Abandon
Nom	Valeur
job_chain_nodes	
job_chain_node	
job_chain	
path	/peakup/Job Chain
order_queue	
order	
1	
at	2038-01-19 03:14:07.000Z
suspended	true
id	ETAPE 2
length	1
order_state	002
if_order_is_ready	
job	Deuxieme

Historique des traitements

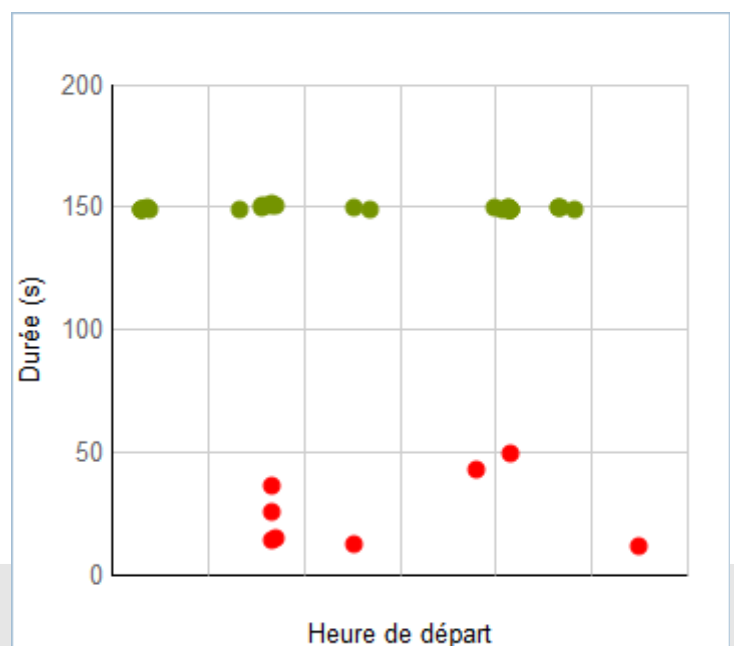
L'historique permet de situer un traitement dans le temps, par sa durée d'exécution ou par son statut en le visualisant directement sur dans un tableau filtré ou à travers un graphique. Les écrans sont synchronisés, un double clic sur le tableau ou un simple clic sur la partie graphique affiche directement le journal pour l'exécution concernée.



Le graphique indique l'heure en abscisse et la durée d'exécution en ordonnée. L'heure de départ indique le respect de la fréquence d'exécution, un écart horizontal entre deux points anormalement espacé signifie que le traitement ne suit pas un cycle, soit une exécution est manquante, soit un lancement a été forcé par un opérateur. Un écart vertical différent signifie que la durée du traitement est variable.

La couleur du point reprend le code couleur du statut : vert, rouge et orange.

Ordres





Le panneau des ordres permet de suivre les exécutions de chaînes de traitement. L'ergonomie est similaire au suivi des traitements mais permet de suivre un enchaînement complet. La notion de séquence induit aussi des notions de pilotage plus complexes en cas d'erreur car il faut gérer l'impact sur la suite des

traitements.

Localisation	Etat	Statut	Heure de départ	Heure de fin	Information	Next time
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
scheduler RUNNING 2014-02-12 07:52:36						
tistmt1						
TSM						
peakup						
Job Chain ACTIVE						
Test (toujours bien)	Fin	SUCCESS	2014-02-11 18:21:08	18:21:49		
Reprise sur erreur 1	000	ACTIVATED	2014-02-09 10:22:31			Jamais
154	Sync	SUCCESS	2014-02-11 18:21:19	18:21:49		
155	Sync	SUCCESS	2014-02-11 18:21:18	18:21:49		
Splitter						

L'organisation reprend les répertoires de chaque moteur dans lesquels on retrouve les chaînes de traitements et les ordres d'exécution.

On distingue différents types :




- L'ordre planifié  stocké en base de données et prêt à être exécuté. Si l'ordre a déjà été lancé, on peut consulter les étapes exécutées. Dans le cas contraire, l'ordre est considéré comme activé.
- L'ordre ajouté manuellement par un opérateur  et qui est exécuté immédiatement.





Dans le cas de l'ordre exécuté, planifié ou manuel, un clic permet d'ouvrir la chaîne pour afficher les étapes :

Localisation	Etat	Statut	Heure de départ	Heure de fin
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
peakup				
Job Chain ACTIVE				
Test (toujours bien)	Fin	SUCCESS	2014-02-11 18:21:08	18:21:49
1 peakup/Debut	000	SUCCESS	2014-02-11 17:21:09	2014-02-11 17:21:18
2 peakup/JobSolitter	Split	SUCCESS	2014-02-11 17:21:18	2014-02-11 17:21:18
3 peakup/Premiere	001	SUCCESS	2014-02-11 17:21:19	2014-02-11 17:21:28
4 peakup/JobSvnc	Sync	SUCCESS	2014-02-11 17:21:29	2014-02-11 17:21:30
5 peakup/JobSvnc	Sync	SUCCESS	2014-02-11 17:21:49	2014-02-11 17:21:49
6 peakup/Exit	004	SUCCESS	2014-02-11 17:21:49	2014-02-11 17:21:49
Reprise sur erreur 1	000	ACTIVATED	2014-02-09 10:22:31	

Dans cet exemple, six étapes ont été exécutées.

Le statut d'un ordre permet de retrouver l'état global de la chaîne de traitements :

Statut	Icône	Couleur	Explication
SUCCESS		Vert	L'ensemble de la chaîne s'est correctement déroulé.
FAILURE		Rouge clair	Le scénario s'est terminé par un chemin d'erreur.
WARNING		Jaune clair	Le scénario s'est terminé correctement mais en empruntant un chemin dégradé.

RUNNING		Orange	Le traitement est en cours d'exécution.
ACTIVATED		Bleu	L'ordre est programmé mais n'a jamais été utilisé.
SUSPENDED		Rouge ou Vert	L'ordre attend le déblocage par un opérateur.
SETBACK		Rouge clair	L'ordre est en reprise automatique.

Informations

Le panneau d'information reprend le système des onglets mais ils seront différents selon la nature de l'objet : ordre ou traitement.

Détail d'un ordre

Un double clic sur un ordre permet de donner les détails suivants :

Information	Order log	Historique	Graph
Test		Statut	SUCCESS
Séquence de traitements	peakup/Job Chain	Etat	Fin
Heure de départ	2014-02-11 17:21:08	Exit code	0
End Time	2014-02-11 17:21:49	Error	0
Durée	41s	Steps	1

Journal d'un ordre

Les informations ne concernant que l'ordre pour lequel on aura le journal et l'historique.

Information	Order log	Historique	Graph
Date	Task	Message	Code
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
2014-02-09 12:23:23.026	[info]	set_state 000, Job /peakup/Debut	
2014-02-11 17:26:29.646	[info]	Configuration file 'D:/arii/enterprises/SOS-Paris/scheduler/config/live/peakup/Job Chain,'	891
2014-02-11 17:26:31.967	[info]	Standing_order is 'active' now	893
2014-02-11 18:21:09.043	peakup/Debut:23394	Task is going to process Order peakup/Job Chain:Test "toujours bien", state=000, on Sc	842
2014-02-11 18:21:09.045	peakup/Debut:23394		
2014-02-11 18:21:09.045	peakup/Debut:23394	Task peakup/Debut:23394 - Protocol starts in D:/arii/enterprises/SOS-Paris/scheduler/lo	
2014-02-11 18:21:09.048	peakup/Debut:23394	state=starting (at=never)	918
2014-02-11 18:21:09.049	peakup/Debut:23394	Starting process: "C:\Users\eric\AppData\Local\Temp\sosA0A4F.cmd"	987

Historique de l'ordre

Concernant l'historique, il est à deux niveaux : un premier pour l'ordre lui-même et un deuxième niveau pour les différentes étapes.

Information	Order log	Historique	Graph			
Etat		Etat	Heure de départ	Heure de fin	Durée	Message
<div></div>		<div></div>	<div></div>	<div></div>	<div></div>	<div></div>
<div><div>Fin</div></div>		6	2014-02-11 17:21:08	2014-02-11 17:21:49	41s	(toujours bien)
<div><div><div>000</div></div></div>		1	2014-02-11 17:21:09	2014-02-11 17:21:18	9s	
<div><div><div>Split</div></div></div>		2	2014-02-11 17:21:18	2014-02-11 17:21:18		
<div><div><div>001</div></div></div>		3	2014-02-11 17:21:19	2014-02-11 17:21:28	9s	
<div><div><div>Sync</div></div></div>		4	2014-02-11 17:21:29	2014-02-11 17:21:30	1s	
<div><div><div>Sync</div></div></div>		5	2014-02-11 17:21:49	2014-02-11 17:21:49		
<div><div><div>004</div></div></div>		6	2014-02-11 17:21:49	2014-02-11 17:21:49		
<div><div><div>Fin</div></div></div>		6	2014-02-09 11:22:41	2014-02-09 11:23:22	41s	(toujours bien)

A partir de l'écran des données, un double clic sur une étape permet de retrouver les mêmes informations que pour les traitements.

Actions sur une chaîne de traitements



La chaîne de traitement peut être considérée comme un scénario constitué d'étapes, chaque étape peut déboucher sur deux nouvelles étapes en fonction de son statut :

- l'étape suivante si le statut est en succès
- l'étape d'erreur si le statut est en erreur

La chaîne peut être déclenchée à tout moment en ajoutant un ordre.

L'autre action possible est de bloquer la chaîne, dans ce cas aucun ordre ne pourra plus la déclencher.

Localisation	Etat	Statut
<input type="text"/>	<input type="text" value="v"/>	<input type="text" value="v"/>
scheduler		RUNNING
reprise		
Reprise automatique		SUCCESS
Reprise AUTO		
t1stmt1		

 Ajouter un ordre
 Bloquer la chaîne de traitements

Ajouter un ordre

Dans le cas de l'ajout de l'ordre, une fenêtre permet d'indiquer les paramètres.




Add Order on JOB Chain — □ ×

ID de l'Ordre

Titre

Start State

End State

 Add Order  New Parameter  Cancel

Nom	Valeur
Nouveau paramètre	Test

Ces paramètres sont les suivants :

- L'identifiant de l'ordre pour le retrouver plus simplement dans l'historique, si il n'est pas indiqué il sera automatiquement fourni par le moteur.
- Le titre permet à l'opérateur d'indiquer un commentaire.
- Le nœud de départ permet de démarrer le scénario à une étape précise, par défaut on démarre à la première étape
- Le nœud de fin indique la sortie du scénario

Un clic sur «Ajout de l'ordre » déclenche la chaîne.

Il est aussi possible d'indiquer des arguments en utilisant le bouton « Nouveau paramètre » et en indiquant directement le nom et sa valeur.

Bloquer une chaîne

Le blocage de la chaîne empêche tout lancement automatique mais permet tout de même les lancements forcés.

Lorsque la chaîne est bloquée, le menu contextuel propose le déblocage.

Localisation	Etat	Statut
<input type="text"/>	<input type="text" value="v"/>	STOPPED v
scheduler		RUNNING
pickup		
Job Chain		STOPPED
reprise	<div>Ajouter un ordre</div> <div>Débloquer la chaîne de traitements</div>	
t1stmt1		
TSM		

Actions sur un ordre

Un ordre est un déclencheur prédéfini, il permet d'indiquer la chaîne ou la portion de chaîne à exécuter, les paramètres et la planification horaire.

Un ordre peut avoir été créé à la volée, dans ce cas on le

distingue par un signe ⚡

dont le + indique l'ajout à posteriori.

Localisation	Etat	Statut
<input type="text"/>	<input type="text"/>	<input type="text"/>
[-] scheduler		RUNNING
[-] pickup		
[-] Job Chain		ACTIVE
+ ⚡ Test (toujours bien)	Fin	SUCCESS
+ ⚡ Reprise sur erreur 1	000	ACTIVATED
+ ⚡ 154		SUCCESS
+ ⚡ 155		SUCCESS
[-] reprise		
[-] Reprise au		SUCCESS

⚡ Déclencher l'ordre

⚡ Suspendre l'ordre

⚡ Historique des ordres

Déclencher un ordre

Dans le cas de l'ordre défini, il est possible de le déclencher par anticipation ou de le suspendre.

La fenêtre de déclenchement par un ordre est assez différente d'un ajout d'ordre :

- les paramètres sont fixés lors de la définition de l'ordre et ne sont donc pas modifiables
- les arguments pré-définis sont affichés et peuvent être modifiés

Start order 10099

Reprise AUTO Statut FAILURE

SCHEDULER-280 Process terminated with exit code 1 (0x1)

Séquence de traitements	reprise/Reprise automatique	Etat	IKO
Heure de départ	2014-02-22 18:29:43	Exit code	1
End Time	2014-02-22 18:34:56	Error	1
Durée	5m 13s	Steps	1

Arguments

HOST INCONNUE

⚡ Déclencher l'ordre now + Nouvel argument Abandon

Nom	Valeur
HOST	INCONNUE

Suspendre un ordre

Cette action permet de bloquer un ordre tant qu'un problème n'est pas résolu. Cette suspension peut intervenir à n'importe quelle étape du processus.

[-] scheduler		RUNNING	2014-02-22 20:39:58
[-] pickup			
[-] Job Chain		ACTIVE	
[-] Reprise sur erreur 1	Fin	SUSPENDED	21:11:25

Lorsque l'ordre est suspendu, le symbole 🛑 indique qu'une opération doit être effectuée pour résoudre le problème.

Débloquer un ordre

L'origine d'une suspension peut être manuelle ou programmée dans un scénario. Dans tous les cas, il n'existe que deux actions possibles : Résumer

- ⚡ Résumer l'ordre
- ⚡ Réinitialiser l'ordre
- ⚡ Historique des ordres



l'ordre ou le réinitialiser.

Dans le premier cas, l'exécution reprends à l'étape en erreur alors que la réinitialisation effectue une reprise à l'étape initiale. Le résumé et la réinitialisation reprenne les arguments fournis lors du premier lancement.

Actions sur une étape

Une étape représente un maillon de la chaîne de traitements, chaque maillon est lié à un traitement. Le statut de l'exécution du traitement donne l'étape suivante, si le traitement est en erreur, l'ordre est dirigé vers une étape d'erreur.

Localisation	Etat	Statut
<input type="text"/>	<input type="text" value="v"/>	<input type="text" value="v"/>
📁 scheduler		RUNNING ▶
📁 pickup		
📁 Job Chain		ACTIVE ✓
⚡ Reprise sur erreur 1	Fin	WARNING 🔔
1 ➡ pickup/Debut	000	SUCCESS ✓
2 ➡ pickup/JobSplitter	Split	SUCCESS ✓
3 ➡ pickup/Premiere	001	FAILURE ⚠
4 ➡ pickup/Reprise	00	
⚡ Test	00	
+ ⚡ 192	Erreur	SUCCESS ✓

 Bloquer le noeud de la chaîne
 Sauter le noeud de la chaîne

Bloquer une étape

Bloquer une étape consiste à arrêter l'ordre au moment où il passe par l'étape concernée. Le blocage concernera tous les ordres pour une chaîne donnée, l'étape est donc indiquée en rouge au niveau de la chaîne de traitement. Le statut est STOPPED pour signifier le blocage nécessitant l'action de l'opérateur.

Lorsqu'un ordre est exécuté, il reste bloqué en mode RUNNING jusqu'à ce qu'on intervienne sur l'étape.

Localisation	Etat	Statut
<input type="text"/>	<input type="text" value="v"/>	<input type="text" value="v"/>
📁 scheduler		RUNNING ▶
📁 pickup		
📁 Job Chain		ACTIVE ✓
• Split		STOPPED ⚠
+ ⚡ Reprise sur erreur 1	Fin	SUCCESS ✓
⚡ Test (toujours bien)	Split	RUNNING ▶
1 ➡ pickup/Debut	000	SUCCESS ✓
+ ⚡ 331	Sync	SUCCESS ✓
+ ⚡ 330	Sync	SUCCESS ✓
+ ⚡ 332	Sync	SUCCESS ✓

Débloquer une étape

Le déblocage est l'opération inverse, elle permet à l'ordre de reprendre sa course. Un clic droit sur le nœud bloqué donne accès à l'option « débloquer le nœud ».

L'ordre continue à l'étape suivante.

Localisation	Etat	Statut
<input type="text"/>	<input type="text"/>	<input type="text"/>
scheduler		RUNNING
peakup		
Job Chain		ACTIVE
Split		
Reprise sur erreur 1		
Test (toujours bien)		
1 peakup/Debut	000	SUCCESS
331	Sync	SUCCESS
330	Sync	SUCCESS
332	Sync	SUCCESS

Localisation	Etat	Statut
<input type="text"/>	<input type="text"/>	<input type="text"/>
scheduler		RUNNING
peakup		
Job Chain		ACTIVE
Reprise sur erreur 1	Fin	SUCCESS
Test (toujours bien)	Split	RUNNING
1 peakup/Debut	000	SUCCESS
2	Split	RUNNING
331	Sync	SUCCESS

Sauter une étape

Si on souhaite faire passer un ordre à travers une étape, il est possible de la désactiver sans pour autant bloquer la chaîne. Pour réaliser cette opération, on change le statut du nœud en SKIPPED, l'ordre considérera cette étape comme réussie sans que la commande ait été exécutée.

Comme pour le blocage, l'action est sur une étape du scénario, elle impacte l'ensemble des ordres liés à la chaîne. L'étape sautée est indiquée dans la chaîne et dans les ordres l'ayant utilisé.

Localisation	Etat	Statut
<input type="text"/>	<input type="text" value="v"/>	<input type="text" value="v"/>
[-] scheduler		RUNNING
[-] pickup		
[-] Job Chain		ACTIVE
000		SKIPPED
[-] Reprise sur erreur 1	Fin	SUCCESS
1 → pickup/Debut	000	SKIPPED
2 → pickup/JobSplitter	Split	SUCCESS
3 → pickup/JobSync	Sync	SUCCESS
4 → pickup/JobSync	Sync	SUCCESS
5 → pickup/JobSync	Sync	SUCCESS

Réactiver une étape

Le nœud sauté peut être réactivé à tout moment.

Localisation	Etat	Statut
<input type="text"/>	<input type="text" value="v"/>	<input type="text" value="v"/>
[-] scheduler		RUNNING
[-] pickup		
[-] Job Chain		ACTIVE
Split		SKIPPED
[-] Reprise sur erreur 1		SUCCESS
1 → pickup/JobSoli		SKIPPED
2 → pickup/JobSync	Sync	SUCCESS
3 → pickup/JobSync	Sync	SUCCESS
4 → pickup/Exit	004	SUCCESS
[-] Test (toujours bien)	Fin	SUCCESS

Bloquer le noeud de la chaîne

Réactiver le noeud

Plan

Le plan est le calcul et la vérification des prochaines planifications horaire, il est construit grâce à deux traitements :

- le CreateDaysSchedule qui génère le plan théorique
- le CheckDaysSchedule qui vérifie l'exécution à l'heure dite.

Un double clic affiche le détail de la planification.

Localisation	Statut	Exécuté	Délai	Début	Fin	Cycle
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
scheduler RUNNING 2014-02-22 20:39:58						
sos						
dailyschedule						
CreateDaysSchedule						
2014-02-23 23:00:01	LATE		3d			
2014-02-24 23:00:01	LATE		2d			
2014-02-25 23:00:01	LATE		1d			
2014-02-26 23:00:01	LATE		20h			
2014-02-27 23:00:01	WAITING		3h			
2014-02-28 23:00:01	WAITING		1d			
De 1 à 10 Page 1 10 Lignes par page						
Information						
Statut	WAITING					
Schedule Time	2014-02-27 23:00:01					
Heure de départ						
End Time						
Délai	3h					
Planned ID	12197					




Les statuts sont les suivants :

Statut	Description
SUCCESS	Le traitement planifié a bien été exécuté.
WAITING	Le traitement attend l'heure d'exécution.
LATE	L'heure d'exécution est dépassée.

Moteurs

Les moteurs sont automatiquement insérés dans la base de données par le job CreateDaysSchedule. On retrouve les informations nécessaires à la communication entre l'interface et l'ordonnanceur distant.

Moteur	Statut	Heure de départ	Hostname	Port
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
scheduler	RUNNING	2014-02-22 20:39:58	sosparis	4444

 Vérifier le moteur
 Pause
 Redémarrer le moteur
















Pour des environnements plus complexes avec des moteurs écoutant sur une adresse IP particulière, il est préférable d'utiliser le module d'administration qui offre une plus grande précision dans la définition des objets.




Actions

Le premier type d'objet est le moteur, les actions possibles sont :

- Vérifier l'état du moteur
- Le mettre en pause
- Le redémarrer

Ce menu est accessible dans les écrans de traitements et d'ordres.

Localisation	Statut	Heure de départ
<input type="text"/>	<input type="text"/>	<input type="text"/>
 scheduler	RUNNING	2014-02-12 07:52:36
 Jade		14-02-11 11:08:02
 LogAnalyser		09:36:12
 pickup		
 ping	SUCCESS 	2014-02-10 08:41:23
 Receive File	FAILURE 	08:54:00
 running job	RUNNING 	
 scheduler_sosftp_import	SUCCESS 	20:36:06
 Send File	FAILURE 	2014-02-11 17:26:41
 sos		

 Vérifier le moteur
 Mettre le moteur en pause
 Redémarrer le moteur

Vérifier le moteur

La vérification du moteur consiste à envoyer une demande simple et attendre le retour, si le moteur ne répond plus, son statut passe à l'état STOPPED.

Mise en pause

La mise en pause du moteur envoie une demande de mise en pause au moteur, la réponse est indiquée dans un deuxième pop up. Ces messages s'empilent et restent affichés à l'écran quelques secondes, un clic permet de les fermer immédiatement.

Localisation	Statut	Heure de départ	Heure de fin	Durée	Exit	Prochain départ
					>0	
scheduler	PAUSED	2014-02-12 07:52:36				
Receive File	FAILURE	08:54:00	20:36:07	11h	1	
Send File	FAILURE	2014-02-11 17:26:41	08:39:30	15h	1	

Mise en pause

Le menu change en fonction de l'état de l'objet, si le moteur est en pause, l'action de pause est remplacée par le déblocage.

Redémarrer le moteur

Le redémarrage du moteur est une demande d'arrêt sans interruption brutale des traitements. Le moteur va attendre la fin des traitements mais ne prendra plus aucune demande, cet arrêt peut donc être assez long.

Attention ! Le redémarrage sur Unix est tributaire du délai de libération d'un port, il est généralement de 2 minutes pour Linux mais va jusqu'à 10 minutes sur AIX.

Événements

Les événements sont utilisés dans le cadre de l'orchestration pour définir des règles entre les chaînes de traitements d'un parc d'ordonnanceur.

Le principe est le suivant :

- on ajoute l'envoi d'événements aux traitements concernée
- on crée les règles de gestion des événements

Lorsqu'un traitement est terminé, l'événement est envoyé au moteur qui vérifie si la règle peut être exécutée.

Event	Moteur	Séquence de traitements	Ordre	Traitement	Exit	Creation
<input type="text"/>	<input type="text" value="v"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
PING						
Reprise	scheduler	Reprise automatique	Reprise AUTO	ping	0	2014-02-24 18:05:15
test						
De 1 à 3 Page 1 10 Lignes par page						
Class	test		Creation	2014-02-25 13:08:00		
Id	2		Expires	2014-02-26 01:08:04		
Ordre	Test		Moteur	scheduler		
Séquence de traitements	Job Chain		Hôte	sosparis		
Traitement	Job Splitter		Port	4444		
Exit code	0					

Ce mécanisme est particulièrement efficace pour déclencher des traitements en fonction de règles complexes incluant des opérateurs ET et OU, à travers différents moteurs et/ou différentes applications.

Messages

Les messages sont extraits du journal du serveur et stockés dans la base de données pour être supervisés par un outil de surveillance comme Nagios. Le principe du connecteur consiste à exécuter régulièrement une requête dans cette table pour calculer le statut du service dans la console de supervision.

L'extraction est réalisée par le traitement LogAnalyser.

Le contenu de cette table est publié afin d'identifier rapidement un problème. Il permet de retrouver une erreur ou un simple avertissement et de fournir les objets concernés.

Un double clic sur le message affiche le détail de l'erreur.

Log time	Moteur	Severity	Error	Message	Objects	Count
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
2014-02-27 04:10:49	scheduler	WARN	292	Missing from, to, subject or body. Suppressing email		0
2014-02-27 04:10:14	scheduler	WARN	292	Missing from, to, subject or body. Suppressing email		0
2014-02-27 04:10:06	scheduler	WARN	292	Missing from, to, subject or body. Suppressing email		0
2014-02-27 04:03:18	scheduler	WARN	6	ERRNO-2 No such file or directory [opendir] [files]	/scheduler_sosftp_history_file_order	0
2014-02-27 04:03:18	scheduler	WARN	292	Missing from, to, subject or body. Suppressing email		0
2014-02-27 04:03:16	scheduler	WARN	292	Missing from, to, subject or body. Suppressing email		0
2014-02-27 04:03:11	scheduler	WARN	399	Error in <include file="jobs/JobChainSplitter.xml"/>:	sos/jiti/JobChainSplitter	0
2014-02-27 04:03:10	scheduler	ERROR	161	There is no schedule '/reprise/jeudi_19h'	Reprise AUTO "Test de reprise"	0
2014-02-27 04:03:10	scheduler	ERROR	161	There is no schedule '/reprise/lundi_18h'	Tout va bien "Test de reprise"	0
2014-02-27 04:02:28	scheduler	WARN	289	TCP-port 4444 is blocked. Trying a further 120 seconds befo		0
De 1 à 14 Page 1 20 Lignes par page						
Log time	2014-02-27 04:03:10			Message	3	
Statut	ERROR			Count	0	
Message	2014-02-27 04:03:10.055+0100 [ERROR] (Order Reprise AUTO "Test de reprise") SCHEDULER-161 There is no schedule '/reprise/jeudi_19h'					
Moteur	scheduler					
Séquence de traitements	Reprise AUTO "Test de reprise"					

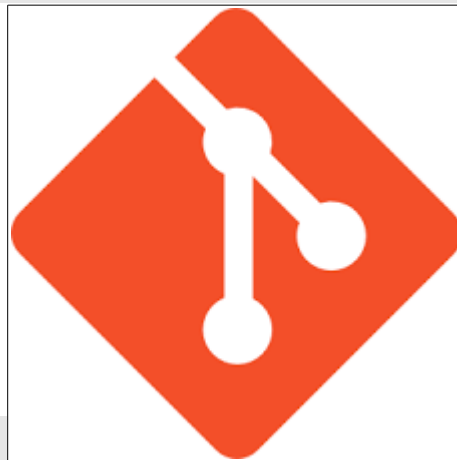
Les filtres affichent directement l'information.

Log time	Moteur	Severity	Error	Message	Objects	Count
<input type="text"/>	<input type="text"/>	ERRO	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
2014-02-27 04:03:10	scheduler	ERROR	161	There is no schedule '/reprise/jeudi_19h'	Reprise AUTO "Test de reprise"	0
2014-02-27 04:03:10	scheduler	ERROR	161	There is no schedule '/reprise/lundi_18h'	Tout va bien "Test de reprise"	0

GIT

Git un est un gestionnaire de versions libre et gratuit. Le module Git offre une interface simple et principalement dédié à la gestion des objets d'ordonnancement.

Ce module offre une intégration dans le portail Ari'i et propose les options les plus couramment utilisé. Si vous souhaitez utiliser des fonctions avancées, il est préférable de se diriger vers des produits comme GitWeb, GitLab, Gitorious et bien d'autres.



Navigation

L'espace centralisé est maintenu par le compte git.

Nouveau référentiel

Si vous souhaitez créer un nouveau référentiel pour un nouvel ordonnanceur, ou une autre utilisation, et que vous souhaitez ensuite publier les informations vers le référentiel centralisé pour une publication dans Ari'i, il faudra tout d'abord cloner un répertoire vide à partir du nouvel espace.

L'exemple suivant consiste à créer un nouveau référentiel pour l'automate arii. On ne va conserver les changements que pour les fichiers du répertoire config.

```
ari@ari64:~/jobscheduler/ari$ git clone git@ari64:ari
Cloning into 'ari'...
warning: You appear to have cloned an empty repository.
```

On substitue le répertoire qu'on vient de créer avec le répertoire de configuration. Le principe est de remplir le nouveau répertoire avec le contenu actuel puis de le renommer en config.

```
ari@ari64:~/jobscheduler/ari$ cp -rp config/* ari
ari@ari64:~/jobscheduler/ari$ mv config config.init
ari@ari64:~/jobscheduler/ari$ mv ari config
```

A partir de cet étape, vous disposez d'un référentiel prêt à être utilisé. On se positionne dans le répertoire config pour utiliser les commandes Git.

```
ari@ari64:~/jobscheduler/ari$ cd config
```

Ajout de fichiers dans le référentiel :

```
ari@ari64:~/jobscheduler/ari$ git add *
```

Confirmation des changements :

```
ari@ari64:~/jobscheduler/ari/config$ git commit -m 'Init'
```

Envoi des changements vers le référentiel centralisé :

```
ari@ari64:~/jobscheduler/ari/config$ git push origin master
Counting objects: 281, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (270/270), done.
Writing objects: 100% (281/281), 1.42 MiB, done.
Total 281 (delta 86), reused 0 (delta 0)
To git@ari64:ari
* [new branch]      master -> master
```

Timecode

Timecode est une norme de calcul de dates complexes. Ce module gère les calendriers de jours fériés et les calculs de planification.



Développement

Tout développeur PHP peut participer au développement des modules du portail, ceci sera d'autant plus simple si il connaît déjà le framework Symfony2.

Nouveau Module

Pour créer un nouveau module :

```
D:\www\arii>\xampp\php\php app/console generate:bundle
```

```
Welcome to the Symfony2 bundle generator
```

```
Your application code must be written in bundles. This command helps  
you generate them easily.
```

```
Each bundle is hosted under a namespace (like  
Acme/Bundle/BlogBundle).
```

```
The namespace should begin with a "vendor" name like your company  
name, your
```

```
project name, or your client name, followed by one or more optional  
category
```

```
sub-namespaces, and it should end with the bundle name itself  
(which must have Bundle as a suffix).
```

```
See
```

```
http://symfony.com/doc/current/cookbook/bundles/best\_practices.html#index-1
```

```
for more
```

```
details on bundle naming conventions.
```

```
Use / instead of \ for the namespace delimiter to avoid any problem.
```

```
Bundle namespace: Aarii/Bundle/GitBundle
```

```
In your code, a bundle is often referenced by its name. It can be the  
concatenation of all namespace parts but it's really up to you to  
come
```

```
up with a unique name (a good practice is to start with the vendor
```

name).

Based on the namespace, we suggest AriiGitBundle.

Bundle name [AriiGitBundle]:

The bundle can be generated anywhere. The suggested default directory uses the standard conventions.

Target directory [D:/www/arii/src]:

Determine the format to use for the generated configuration.

Configuration format (yaml, xml, php, or annotation): yaml

To help you get started faster, the command can generate some code snippets for you.

Do you want to generate the whole directory structure [no]? yes

Summary before generation

You are going to generate a "Arii\Bundle\GitBundle\AriiGitBundle" bundle in "D:/www/arii/src/" using the "yaml" format.

Do you confirm generation [yes]? yes

Bundle generation

Generating the bundle code: OK

Checking that the bundle is autoloading: OK

Confirm automatic update of your Kernel [yes]?

Enabling the bundle inside the Kernel: OK

Confirm automatic update of the Routing [yes]?

Importing the bundle routing resource: OK

You can now start using the generated code!

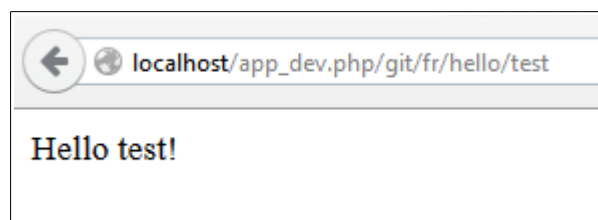
```

        new Aii\GitBundle\AiiGitBundle(),
arii_git:
    resource: "@AiiGitBundle/Resources/config/routing.yml"
    prefix: /

# =====
# Git
# -----

arii_Git:
    resource: "@AiiGitBundle/Resources/config/routing.yml"
    prefix: /git/{_locale}
    requirements:
        _locale: en|fr|es|de|cn|ar|ru|jp

```



Intégration Ari'i

Routage

```

54. arii_git_homepage:
55.     pattern: /hello/{name}
56.     defaults: { _controller: AiiGitBundle:Default:index }

```

Contrôleur

```

57. <?php
58.
59. namespace Aii\GitBundle\Controller;

```

```
60.
61. use Symfony\Bundle\FrameworkBundle\Controller\Controller;
62.
63. class DefaultController extends Controller
64. {
65.     public function indexAction($name)
66.     {
67.         return $this-
        >render('AriiGitBundle:Default:index.html.twig', array('name' =>
        $name));
68.     }
69. }
```

Vue

```
70. Hello {{ name }}!
```

Pour plus d'informations

Solutions Open Source Paris

93, avenue de Choisy

75013 Paris

<http://sos-paris.com>

Communauté

Support

Vous pouvez obtenir un support technique de SOS aux heures et aux jours ouvrés en allant sur le site dédié <http://Support.SOS-Paris.com> ou en envoyant un mail à support@sos-paris.com