

Premiers Pas

Table of contents

1 Premier lancement.....	2
2 Perspective LDAP.....	4
3 Perspective JMX.....	6
4 Traitement d'un ticket.....	8
5 Edition de template.....	12

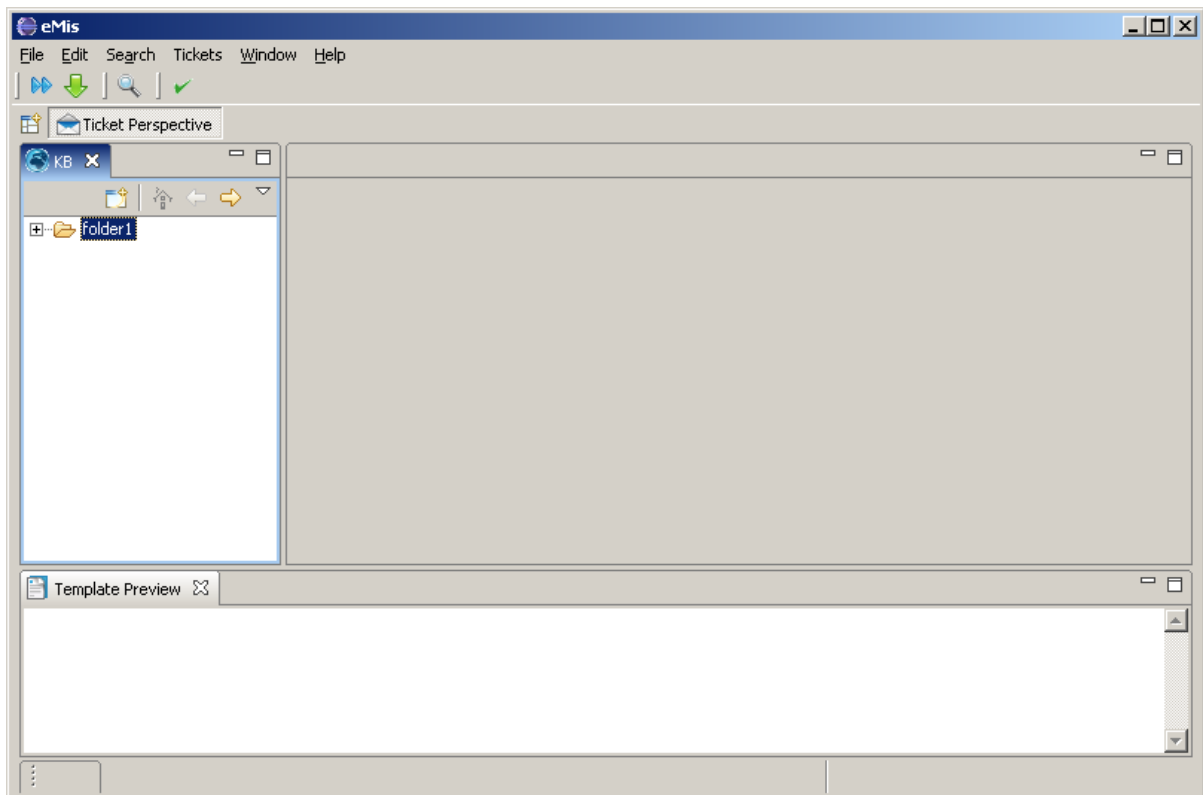
Ce document décrit une première utilisation de l'application en expliquant le fonctionnement.

Vous avez installé la partie serveur, déployé la partie d'exemple (vous avez paramétré un compte mail à utiliser), vous avez installé l'application cliente, le site d'update et vous avez lancé l'application en vous authentifiant avec le user de test 'user'.

Il est tant d'explorer cette interface.

1. Premier lancement

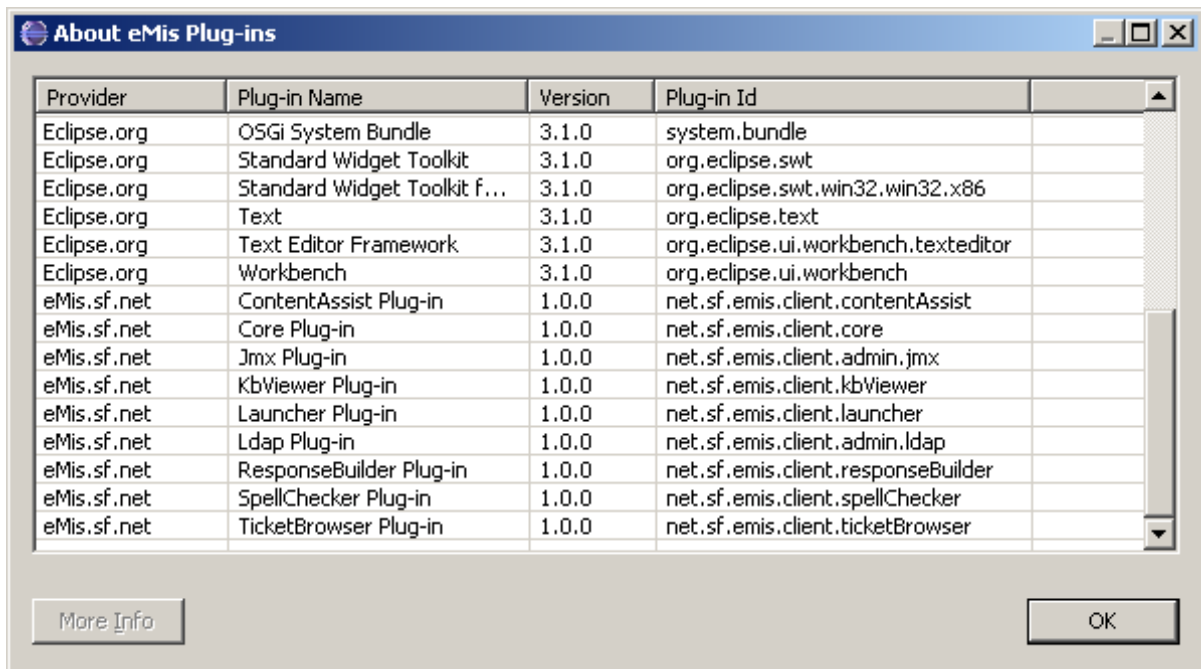
Vous devez avoir quelque chose qui ressemble à ça :



Premier lancement

L'application est constitué d'un ensemble de plugins rassemblé en features. Pour afficher l'ensemble des plugins activés, veuillez utiliser le menu **'Help'** -> **'About eMis'** puis **'Plug-in Details'**

Vous verrez alors l'ensemble des plugins eclipse nécessaires au fonctionnement, et en bas de la liste tous les plugins eMis :

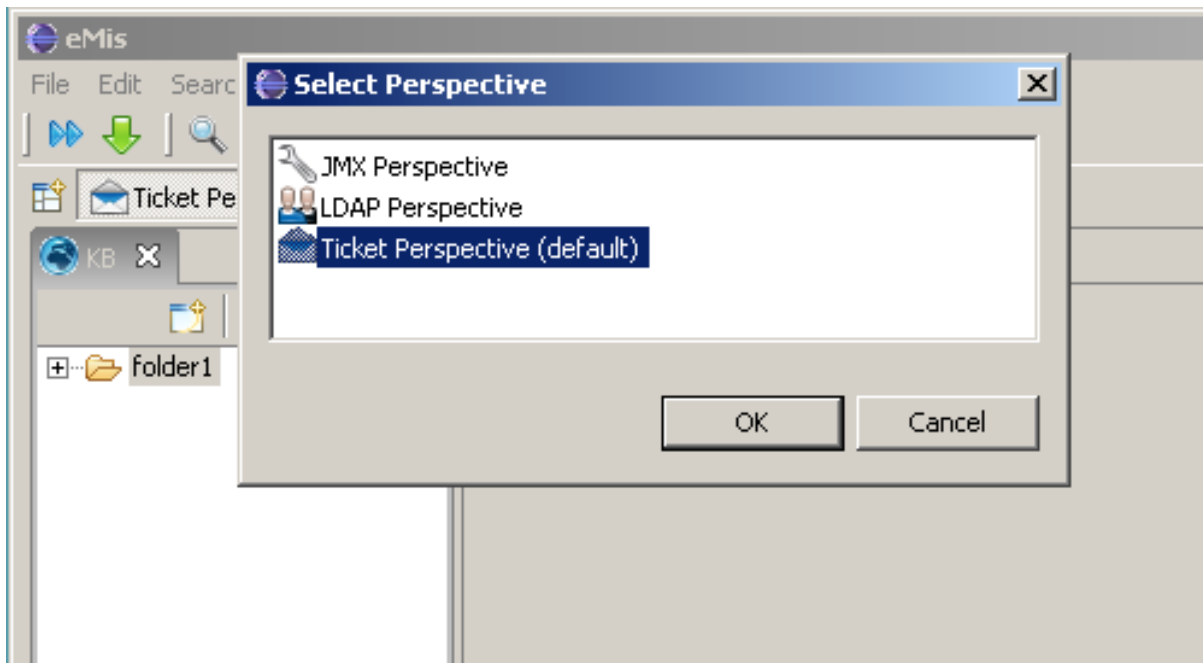


The screenshot shows a dialog box titled "About eMis Plug-ins" with a table listing installed plug-ins. The table has four columns: Provider, Plug-in Name, Version, and Plug-in Id. Below the table are two buttons: "More Info" and "OK".

Provider	Plug-in Name	Version	Plug-in Id
Eclipse.org	OSGi System Bundle	3.1.0	system.bundle
Eclipse.org	Standard Widget Toolkit	3.1.0	org.eclipse.swt
Eclipse.org	Standard Widget Toolkit f...	3.1.0	org.eclipse.swt.win32.win32.x86
Eclipse.org	Text	3.1.0	org.eclipse.text
Eclipse.org	Text Editor Framework	3.1.0	org.eclipse.ui.workbench.texteditor
Eclipse.org	Workbench	3.1.0	org.eclipse.ui.workbench
eMis.sf.net	ContentAssist Plug-in	1.0.0	net.sf.emis.client.contentAssist
eMis.sf.net	Core Plug-in	1.0.0	net.sf.emis.client.core
eMis.sf.net	Jmx Plug-in	1.0.0	net.sf.emis.client.admin.jmx
eMis.sf.net	KbViewer Plug-in	1.0.0	net.sf.emis.client.kbViewer
eMis.sf.net	Launcher Plug-in	1.0.0	net.sf.emis.client.launcher
eMis.sf.net	Ldap Plug-in	1.0.0	net.sf.emis.client.admin.ldap
eMis.sf.net	ResponseBuilder Plug-in	1.0.0	net.sf.emis.client.responseBuilder
eMis.sf.net	SpellChecker Plug-in	1.0.0	net.sf.emis.client.spellChecker
eMis.sf.net	TicketBrowser Plug-in	1.0.0	net.sf.emis.client.ticketBrowser

Les plug-ins

La perspective par défaut est nommée 'Ticket Perspective'. Disons qu'une perspective est une configuration donnée. Si vous cliquez sur le bouton 'Open Perspective', vous devriez voir apparaître ceci :

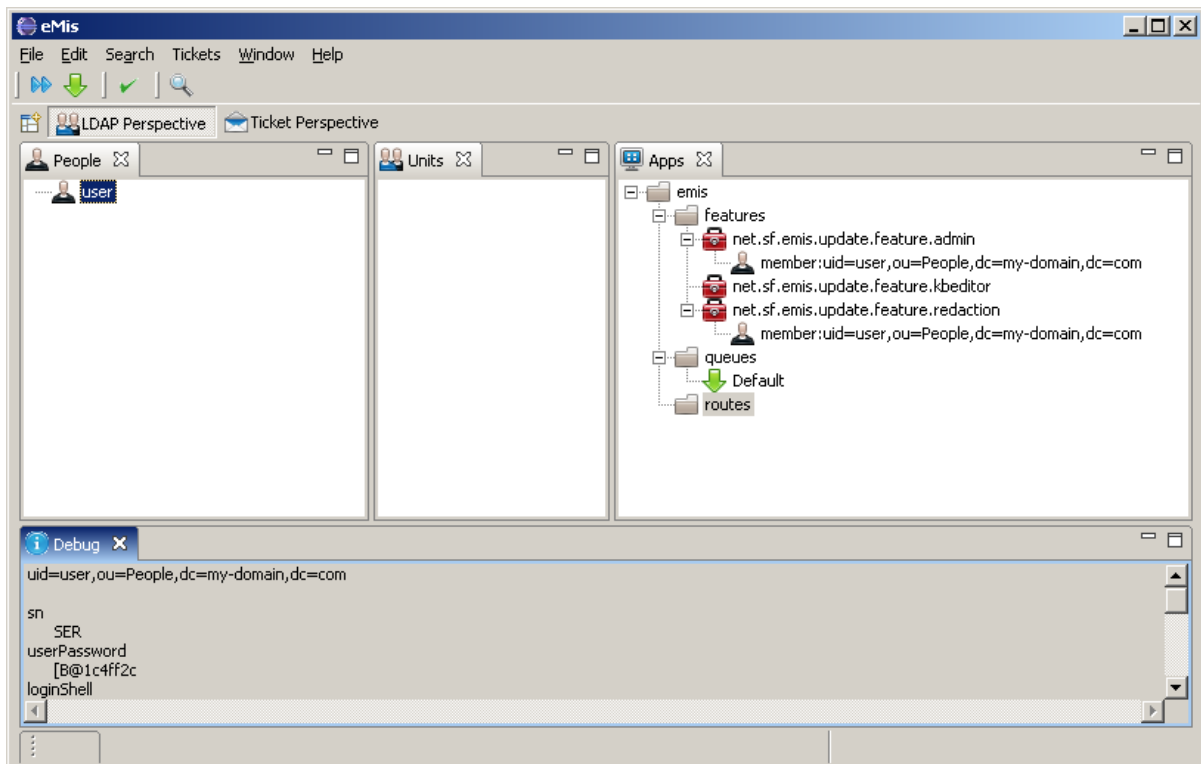


Choix des perspectives

- **Ticket Perspective** est celle utilisée pour le traitement des emails, ou encore l'édition de la KB.
- **LDAP Perspective** est utilisée pour gérer les droits LDAP.
- **JMX perspective** permet de gérer la partie middleware de l'application (server J2EE).

2. Perspective LDAP

Ouvrons la perspective LDAP. Vous devriez avoir quelque chose qui ressemble à ça :



La perspective LDAP

On voit un découpage en 3 zones :

- a gauche, la zone nommée **People**, vous y trouverez les utilisateurs.
- au milieu, la zone nommée **Units**, vous y trouverez les groupes.
- à droite, la zone **Apps**, ou on retrouve les entrées liées à l'application.

Dans la zone Apps on distingue :

- la branche **features** : y sont référencées toutes les features de l'application. On voit que notre user est membre de la feature 'admin' et 'redaction'.
- la branche **queues** : on y trouve les files d'attente workflow, il n'y en a qu'une, la 'Default', et visiblement personne n'a de droit dessus
- la branche **routes** : on y trouve les droits de routage. Nous y reviendrons.

Cette articulation nous permet de gérer les droits de manière groupale. Je m'explique : Pour attribuer des permissions sur une ressource (feature, queue, route) nous avons plusieurs solutions :

- nous pouvons rendre un utilisateur 'member' de la ressource. Il aura alors accès à cette ressource. C'est le cas actuellement pour les features et l'utilisateur 'user'. Mais si nous

avons un nombre important d'utilisateurs, nous préférons gérer des 'skills'

- nous pouvons rendre un groupe membre de la ressource : alors tous les utilisateurs membres de ce groupe auront accès à cette ressource
- nous pouvons rendre une OU membre de ce la ressource. Tous les membres des groupes fils de cette OU auront accès à cette ressource. De cette manière, les permissions sont héritées.

Ainsi, pour savoir si un utilisateur à accès à une ressource, l'application vérifie si il est membre de cette ressource, ou s'il est membre d'un groupe lui-même membre de cette ressource, ou encore s'il est membre d'un groupe descendant d'un objet (OU ou groupe) membre de cette ressource.

3. Perspective JMX

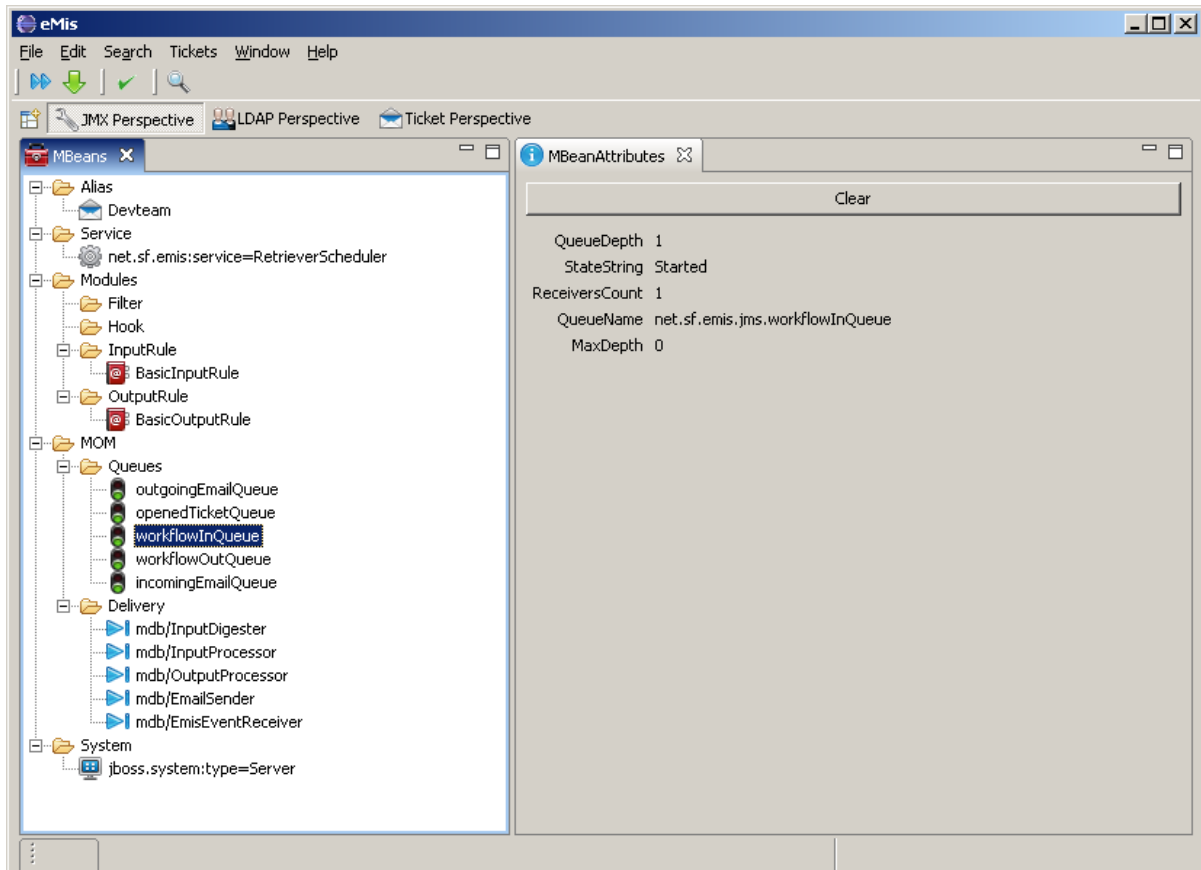
Revenons à la perspective par défaut.

Notre application est censée nous aider à traiter des emails, donc le mieux de regarder cette fonctionnalité de plus près.

Envoyons un eMail vers le compte d'exemple que nous avons paramétré lors du build.

Si vous avez une vue sur les logs jboss, ca devrait beaucoup parlé.

Allons maintenant explorer la Perspective JMX. Nous y trouvons les éléments nous permettant d'administrer et de monitorer l'application.



Perspective JMX

Voici quelques explications :

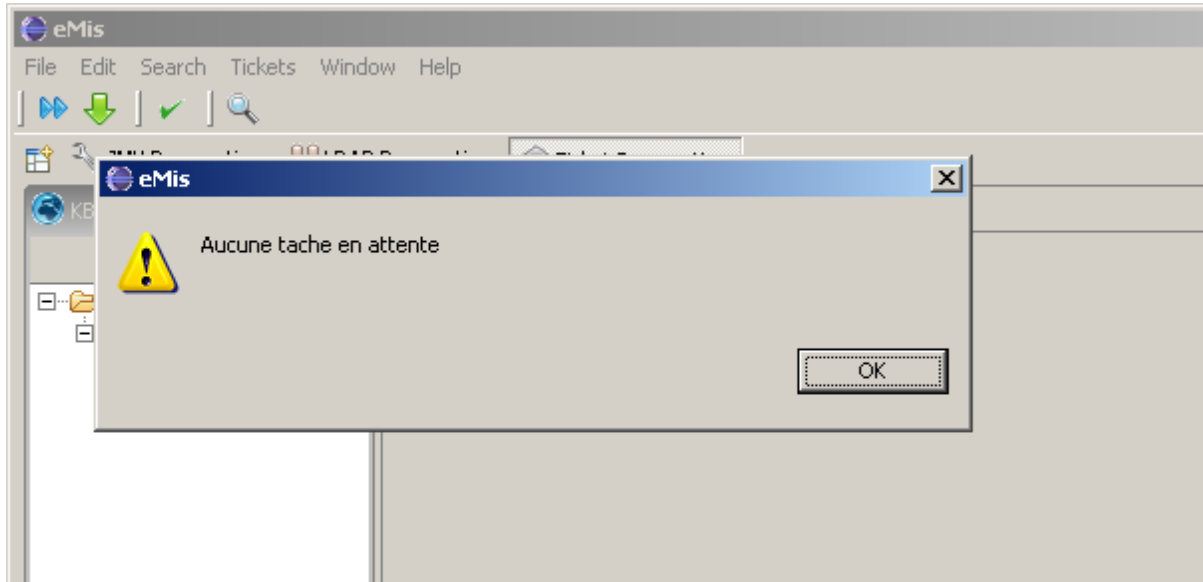
- **Alias** : nous y trouverons les comptes mails que nous gérons via eMis.
- **Service** : nous y trouvons le Scheduler du retriever.
- **Modules** : les modules sont les plugins côtés server, les EJB de customisation.
- **MOM**, la partie messaging de l'application J2EE
 - **Queues** : les différentes queues. On peut monitorer leur état. 'QueueDepth' est le nombre de messages en attente sur la queues.
 - **Delivery** : les connexions entre les queues et les receivers. Nous pouvons gérer ici les flux en tout point du système.
- **System** : nous permet de monitorer le server JBoss.

A ce stade, si vous regardez le nombre de messages pour la queue workflowInQueue, vous devriez avoir 1. Il s'agit du mail que nous avons envoyé, enfin plutôt du ticket qui a été ouvert à la réception de cet eMail.

Revenons à la perspective par défaut.

4. Traitement d'un ticket

Afin de traiter ce ticket, utilisez le menu '**Ticket**' -> '**Pull Task**'. Vous devriez avoir le message suivant :



Pas de task en attente

Que se passe t'il ?

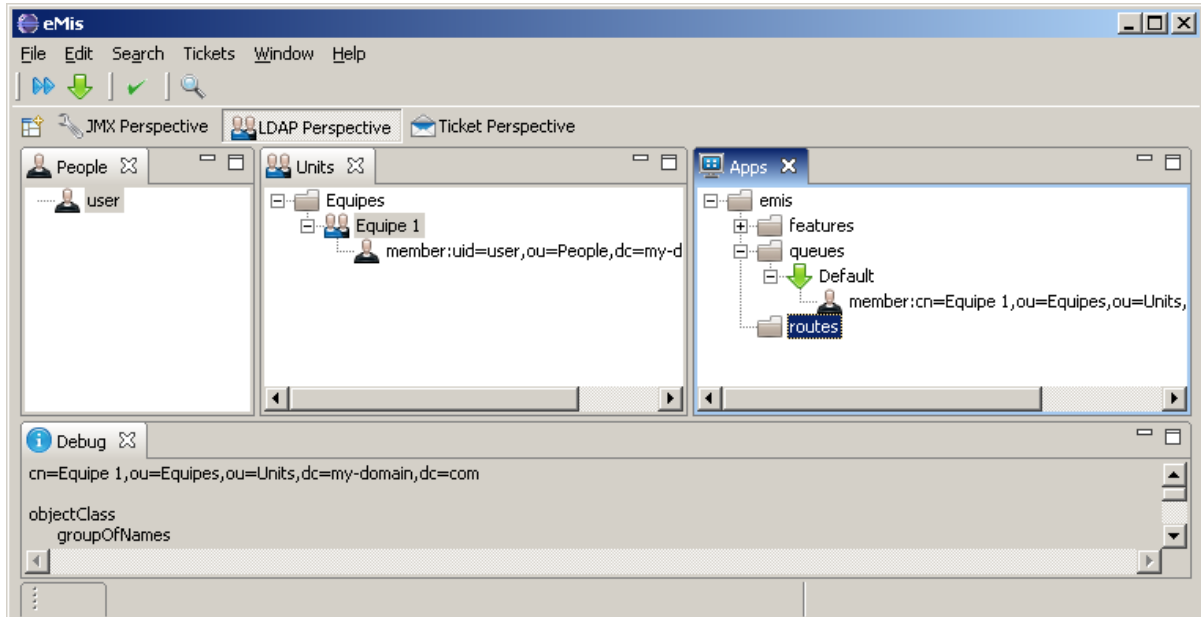
La InputRule par défaut (BasicInputRule) assigne tout ticket entrant à la file 'Default'. Or nous avons vu dans la perspective LDAP que personne n'avais de droits sur la cette file. Nous allons remédier à ça de suite.

Réouvrez la perspective LDAP.

Nous allons créer un OU dans la zone du milieu (Units) :

1. Click droit dans la zone 'Units' puis '**Create OU**'
2. Nommons cette OU 'Equipes'. Nous allons référencer nos équipes dans cette OU.
3. Créons un group : clique droit sur 'Equipes' -> '**Create Group**'
4. nous nommons ce groupe 'Equipe 1'
5. Nous rendons l'utilisateur 'user' membre de 'Equipe 1'. Pour cela, déplacez l'objet 'user' vers 'Equipe 1' par drag'n'drop. Confirmez.
6. Pour finir, nous allons rendre 'Equipe 1' membre de la queue 'Default' dans la zone 'Apps' de la même manière.

Finalement, vous devez avoir quelque chose qui ressemble à ça :



Nouvelles permissions

Pour vérifier, si nos droits sont corrects, nous allons lister toutes les permissions de user : Cliquez droit sur 'user', menu '**Show Permissions**'. Dans la fenêtre 'Debug' vous devez voir quelque chose qui ressemble à ceci :

```
Unités :
- cn=Equipe 1, ou=Equipes, ou=Units, dc=my-domain, dc=com
Queues accessibles :
- Default
- uid=user, ou=People, dc=my-domain, dc=com
Modules accessibles :
- net.sf.emis.update.feature.admin
- net.sf.emis.update.feature.redaction
Membre de :
-
- cn=net.sf.emis.update.feature.admin, ou=features, ou=emiss, ou=Apps, dc=my-domain, dc=com
-
- cn=net.sf.emis.update.feature.redaction, ou=features, ou=emiss, ou=Apps, dc=my-domain, dc=com
- cn=Equipe 1, ou=Equipes, ou=Units, dc=my-domain, dc=com
```

Cela signifie que 'user' :

- est membre du groupe d'unité 'Equipe 1'
- a accès aux files suivantes :
 - Default (de part son appartenance à 'Equipe 1')

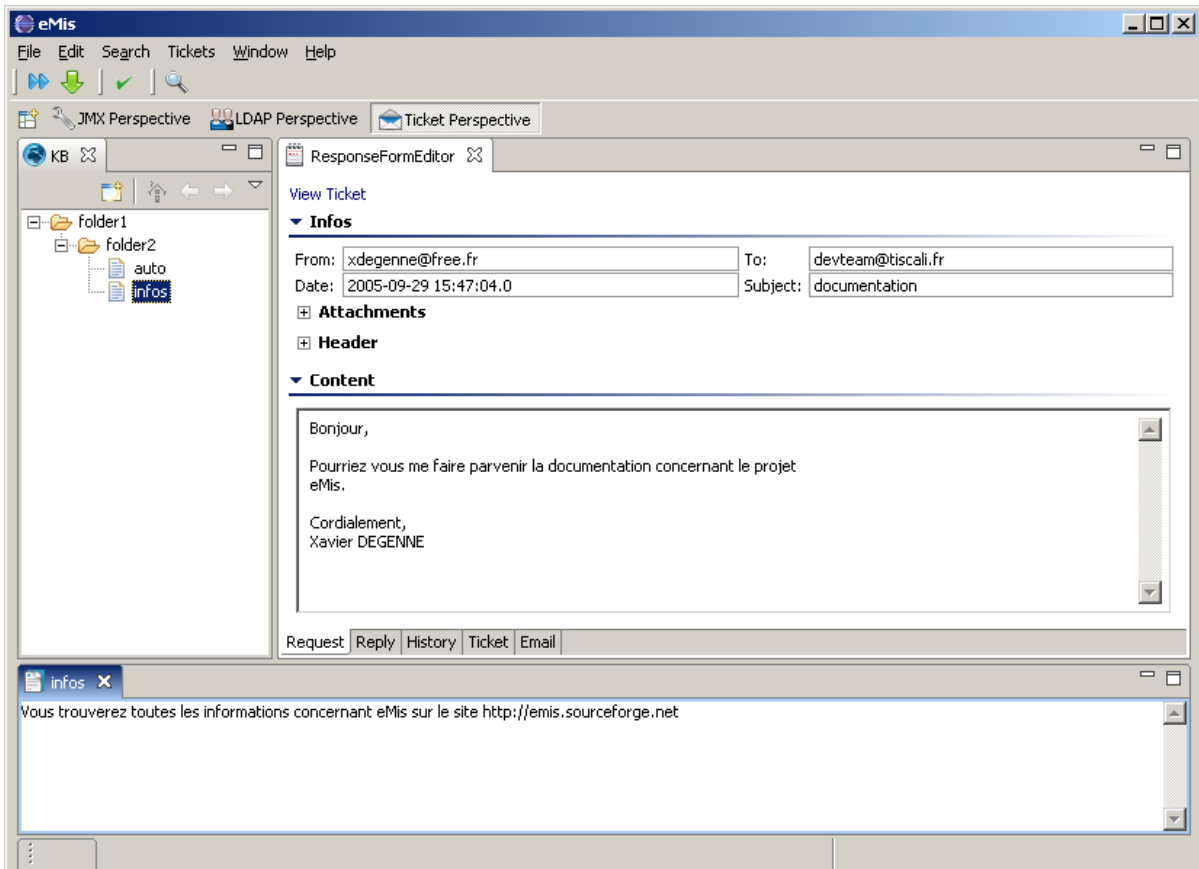
- uid=user,ou=People,dc=my-domain,dc=com (chaque user à une file personnelle).
- a accès aux features admin et redaction
- globalement, est membres des groupes 'admin', 'redaction' et 'Equipe 1'

A l'inverse, si nous regardons tous les ayants droits sur la file 'Default' : Cliquez droit sur la file 'Default' dans la partie de droite, puis menu '**Show Authorized Users**', nous devrions voir :

```
Users ayant accès à cette ressource :
uid=user , ou=People , dc=my-domain , dc=com
```

Donc nous nous sommes donné la permission de récupérer un ticket dans la file 'Default'

Revenons à la Perspective Ticket et activons à nouveau le menu '**Ticket**' -> '**Pull Task**'. Vous devez obtenir quelque chose comme ceci :



Une vue sur l'email

Nous allons traiter ce ticket, c'est à dire répondre à la demande.

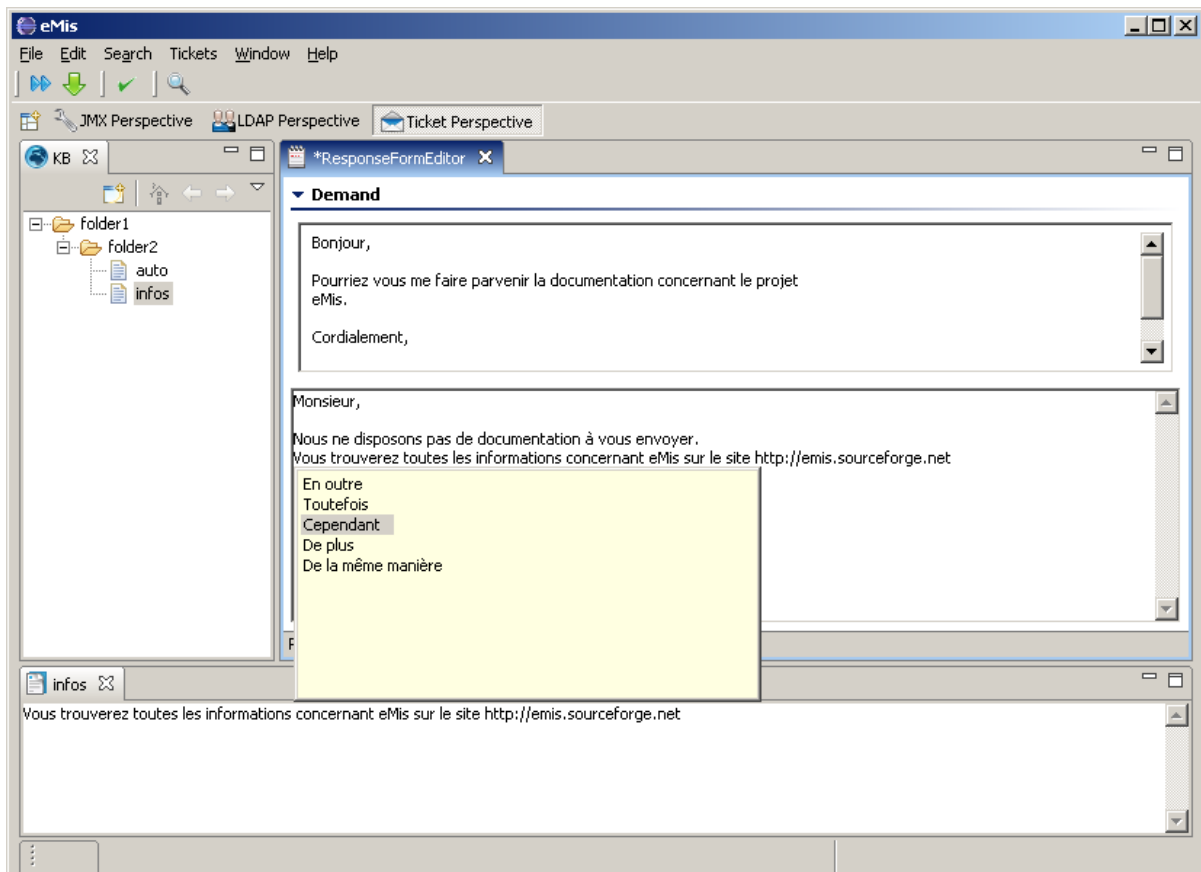
La partie de traitement est divisée en onglets :

- **Request** : le mail d'origine
- **Reply** : la zone de réponse
- **History** : la liste des tickets pour ce client.
- **Ticket** : une vue sur le ticket.
- **Email** : pour visualiser les emails de ce ticket.

Positionnez vous sur la zone 'Reply'.

Pour répondre à ce mail, vous pouvez utiliser la KB, le contexte, vos doigts, ou bien encore une liste de phrases type :

- insérez un article par drag'n'drop
- insérez un article par click droit puis '**Use in Response**'
- insérez une variable contextuelle par Drag'nDrap depuis l'arborescence de 'Contexte'
- dans la zone de réponse, si vous utilisez simultanément les touches 'Ctrl' et 'Espace' vous verrez un pop-up vous proposant une liste de mots de liaisons.



Le Content Assist

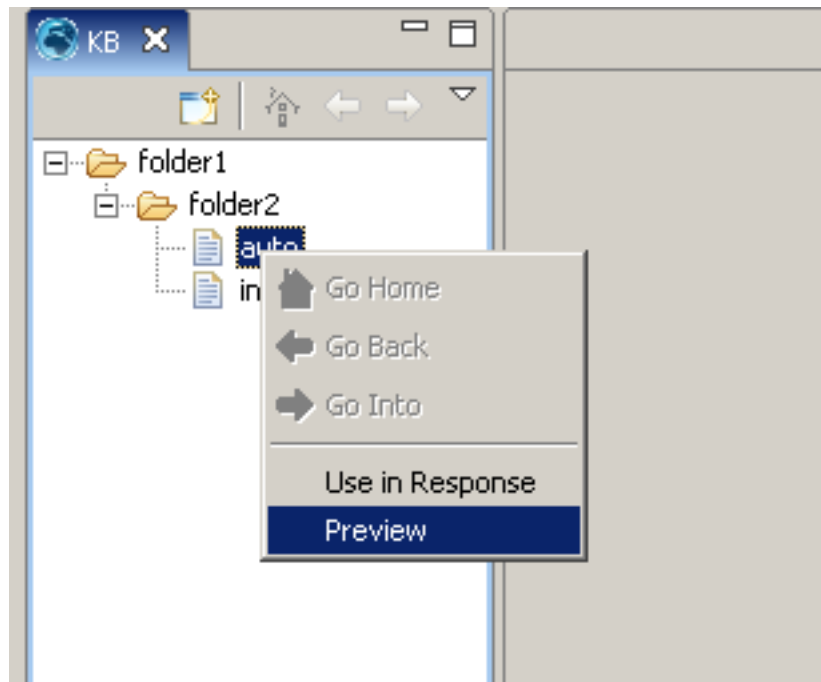
Quand vous avez terminé, faites '**File**' -> '**Save**' ou bien 'Ctrl + S' pour envoyer la réponse.

Vous pouvez aller vérifier le résultat sur votre boîte mail.

Pour voir le ticket complet, utilisez la fonction de recherche par '**Search**' -> '**Find Ticket**'. Entrez '1' dans ticket # et lancez la recherche. Pour afficher le ticket, double-cliquez sur la première colonne.

5. Edition de template

Nous souhaitons modifier un article. Regarder le menu contextuel sur un article, cela doit ressembler à ceci :



Sans la feature kbeditor

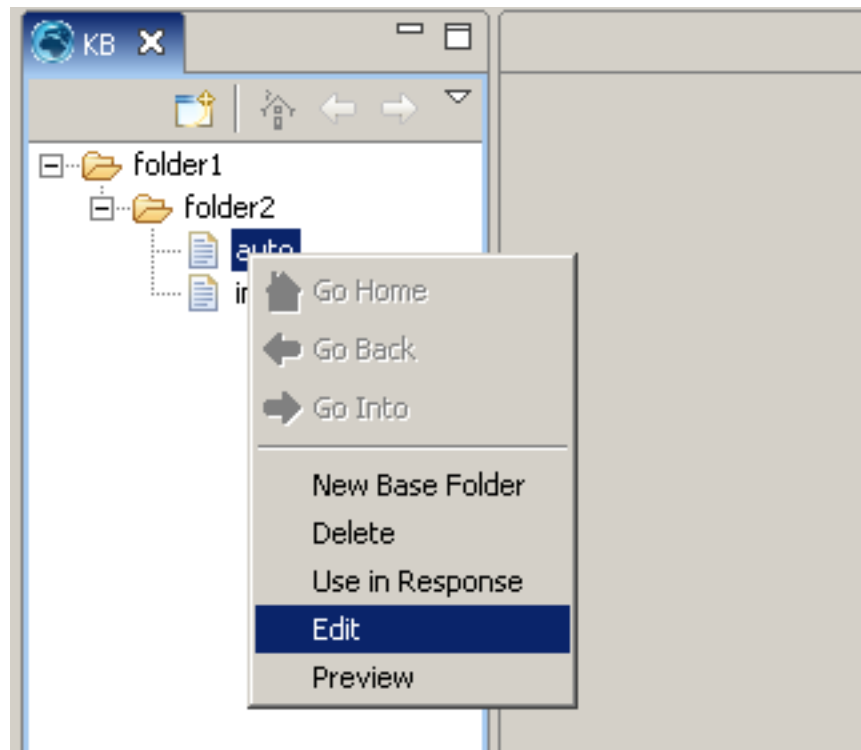
Nous n'avons pas les droits sur la feature kbeditor. Octroyons nous ce droit via la perspective LDAP.

Relançons l'application.

Note:

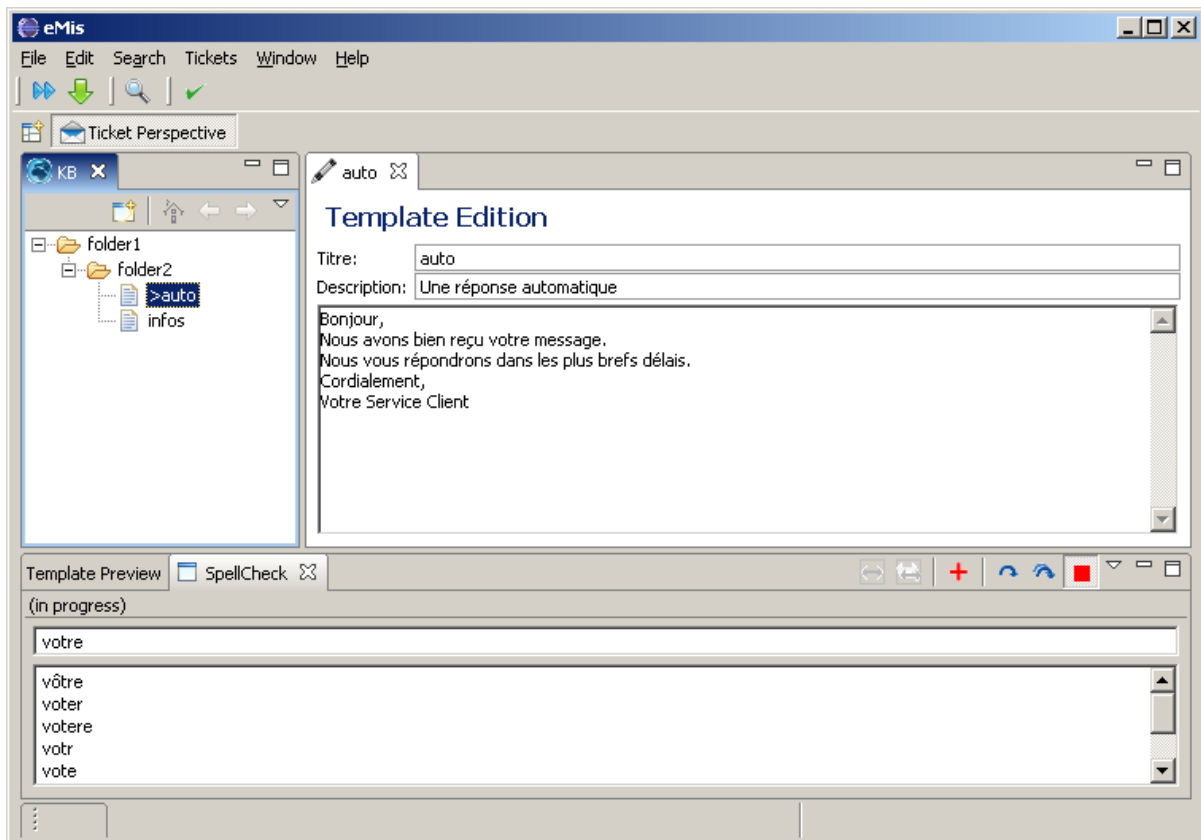
les droits sur les features nécessitent un redémarrage de l'application car celles-ci sont chargées au démarrage

Maintenant, le menu contextuel doit ressembler à ceci :



Avec la feature kbeditor

Si vous choisissez '**Edit**' vous verrez cela :



L'éditeur de KB

Pour modifier l'article, vous pouvez taper votre texte dans la zone d'édition ou bien encore utiliser une variable de contexte qui sera renseignée lors de l'utilisation de l'article.

Quand vous aurez effectué les modifications sur l'article, sauvegardez. Vous pouvez voir que le nom de l'article dans la KB est précédé du signe >. Cela signifie que la KB a été modifiée, elle n'est plus synchro avec le serveur, vous devez la sauvegarder (bouton save dans la zone KB).