

```

-----
      Atelier de Maintenance
Jate 13.8.1 - (c) philippe.billerot@gmail.com
-----

```

```

1 / PRESENTATION

2 / LANCEMENT DE L'ATELIER
3 / PARAMETRAGE DE L'ATELIER
  3.1 / Paramétrer l'atelier
  3.2 / Paramétrer les outils
  3.3 / Paramétrer les actions
  3.4 / Paramétrer les extensions
4 / LES COMMANDES DE L'ATELIER
  4.1 / COMMANDES DOS
  4.2 / COMMANDES SQL
5 / LA FENETRE RESULTAT
6 / MACROS DANS LES LIGNES DE COMMANDES
7 / HISTORIQUE DES MISES A JOUR

```

## 1 / PRESENTATION

```

-----
Jate est atelier pour maintenir un ensemble de fichiers texte.
Les noms de fichier sont présentés dans une liste (la partie gauche de l'atelier)
Le contenu de la liste est modifiable à souhait.

```

```

Les fichiers texte sont visualisables et modifiables dans l'éditeur de l'atelier
(la partie droite de l'atelier).
Il est possible d'ouvrir plusieurs fichiers.

```

```

La partie inférieure est un fenêtre où seront affichés les résultats des actions
réalisées dans l'éditeur ou des commandes du menu Outil.

```

```

-----
| liste      | éditeur      |
|            |              |
|            |              |
|            |              |
|            |              |
-----
| résultat   |              |
|            |              |
|            |              |
|            |              |
-----

```

## 2 / LANCEMENT DE L'ATELIER

```

-----
L'atelier a été développé en java avec jdk 1.3 de SUN.
Le runtime java 1.2 ou version suivante devra donc être installé préalablement.

```

```

java.exe -jar jate.jar [fichier de configuration]
ou
javaw.exe -jar jate.jar [fichier de configuration]

```

```

[fichier de configuration]
  paramètre optionnel pour préciser le nom du fichier de configuration
  où seront mémorisées les propriétés de l'atelier :
  - taille, position de la fenêtre
  - commandes du menu outil
  - actions possibles dans le menu contextuel de l'éditeur

```

par défaut le nom du fichier de configuration est jate.ate

Le répertoire de travail de l'atelier est celui donné au démarrage.  
Par exemple pour créer un raccourci sur le bureau :

```
Cible :
    u:\repertoireDuJar\jate.jar
Démarrer en :
    x:\repertoireAilleurs\jate.ate
```

### 3 / PARAMETRAGE DE L'ATELIER

-----  
Le menu Fichier propose 3 possibilités pour personnaliser l'atelier :

paramétrer l'atelier...

pour personnaliser les options générales de l'atelier

paramétrer les outils...

pour spécifier les commandes que l'on veut mettre à disposition dans le menu Outils

paramétrer les actions...

pour spécifier les commandes qui seront proposées dans le menu contextuel de l'éditeur

paramétrer les extensions...

pour associer une application à l'ouverture du fichier. Dans ce cas, le fichier ne sera pas ouvert dans l'éditeur de l'atelier.

#### 3.1 / paramétrer l'atelier

```
-----
aide.cmd=
    la commande dos pour afficher l'aide en ligne qui sera accessible dans le menu "?"
    {$edit;readme.txt}

aide.titre=Aide
    le texte qui sera affichée dans la ligne du menu d'aide

atelier.liste=jate.lst
    le nom du fichier texte qui mémorisera la liste des fichiers gérés dans l'atelier
    par défaut : jate.lst

atelier.titre=Atelier de Maintenance
    le titre qui sera affiché dans la barre de titre de la fenêtre de l'atelier

ftp.open=
    adresse ip ou nom DNS du serveur FTP (proposé par défaut à la connexion)
    Dès lors que ce paramètre est renseigné l'éditeur disposera des fonctions ftp

ftp.user=
    nom du logon sur le site FTP (proposé par défaut à la connexion)
    (le mot de passe sera demandé à la connexion)
```

#### 3.2 / paramétrer les outils

```
-----
Le titre du menu tout d'abord sera spécifié dans la variable :
```

```
outils.titre = Outils
```

Une ligne du menu sera définie dans 2 variables :

```

utils.X.cmd=
    la commande qui sera lancée
utils.X.titre=
    le titre de la commande dans le menu

    X démarrera à 1.

```

exemple :

```

utils.titre = Outils
utils.1.cmd=dos javadoc -version -author -d javadoc -public ja*.java
utils.1.titre=Générer la doc Javadoc
utils.2.cmd=dos createJar.bat
utils.2.titre=Créer le fichier jar
utils.3.cmd=dos {commande dos}
utils.3.titre=Commande DOS
utils.4.cmd=dos compilerAll.bat
utils.4.titre=Compiler tout

```

### 3.3 / paramétrer les actions

-----

même principe que pour les utils :

```

actions.1.cmd=sql %1 {paramètre SQL}
actions.1.titre=exécuter SQL
actions.2.cmd=dos %1
actions.2.titre=exécuter DOS
actions.3.cmd=dos javac {$fichier} -verbose -deprecation
    -classpath .;d\:\j sdk\lib\j sdk2.jar
actions.3.titre=Java Compiler...

```

### 3.4 / paramétrer les extensions

-----

```

extensions.txt=notepad %1
extensions.ini=notepad %1
extensions.bmp=pbrush %1

extension.exe=nul ou null
    pour spécifier à l'atelier de ne rien lancer lorsque le fichier
    aura l'extension indiquée.

```

## 4 / LES COMMANDES DE L'ATELIER

-----

2 types de commandes :

- des commandes DOS
- des commandes SQL

Le résultat des commandes sera affiché dans la fenêtre résultat.

### 4.1 / COMMANDES DOS

-----

Une commande DOS sera préfixée par dos ...  
suivie de la commande qui sera proposée au shell du système.

Exemple :

```
dos dir c:\temp
```

## 4.2 / COMMANDES SQL

-----

Une commande SQL sera préfixée par sql ...  
suivie de la commande qui sera au moteur du gestionnaire de la base de donnée.

Exemple :

sql select \* from tableCLIENTS

## 5 / LA FENETRE RESULTAT

-----

La fenêtre inférieure de l'atelier affiche le résultat des commande DOS ou SQL.

Elle permet aussi la saisie de commandes (dos ou sql) directement dedans.

La validation se fera alors par la touche F11 sur la ligne de commande.

## 6 / MACROS DANS LES LIGNES DE COMMANDES

-----

Une ligne de commande pourra comprendre une macro qui sera interprétée avant par l'atelier :

{nom paramètre}

le paramètre sera demandé avant l'envoi de la commande  
la valeur saisie remplacera la macro dans la commande

{nom paramètre, item 1, item 2, ...}

le paramètre sera présenté cette fois à travers une liste  
L'item choisi remplacera la macro dans la commande  
Si l'item est de la forme "(code) label", c'est le code seulement qui sera retournée.

{\${fichier}}

sera remplacée par le nom du fichier ouvert dans l'éditeur

{\${repertoireAtelier}}

sera remplacée par le répertoire de l'atelier

{\${ecouteSocket;né du port}}

l'atelier se met à l'écoute sur un socket avec le né du port donné.

{\${socket;adresse du serveur;né du port;message}}

envoie le message sur le socket distant avec l'adresse et le né du port donnés.

{\${mail;serveur smtp;email émetteur;email destinataire;email en copie;

sujet;nom du fichier HTML}

envoie un email.

{\${edit;fichier}}

ouvre le fichier dans une fenêtre d'édition.

{\${play;file://chemin du fichier son}}

ouvre la fenêtre du player et joue le fichier son

{\${getHttp:url de la requête}}

lance une requête sur un serveur HTTP.

## LIGNES DE COMMANDE DANS L'EDITEUR

Le contenu de la fenêtre de l'éditeur peut être interprété puis exécutée en plaçant %1 derrière le nom de l'interpréteur :

```

dos %1
    le contenu de la fenêtre de l'éditeur sera envoyée dans l'entrée standard du shell
    de commande

sql %1
    le contenu de la fenêtre de l'éditeur sera envoyé dans l'entrée standard
    du moteur SQL.
    Les commandes seront terminées par un point virgule.

```

Exemple:

```

/* ceci est un commentaire
   sur plusieurs lignes    */
CONNECT ARBRES,SA;                // nom du driver ODBC user mdp

DROP TABLE ARBLIEUX;
CREATE TABLE ARBLIEUX
(
    idLieu          varchar(9) NOT NULL,      // né du lieux
    pays            varchar(30) NULL,
    departement     varchar(9) NULL,
    commune         varchar(30) NULL,
    CONSTRAINT cleARBIEUX PRIMARY KEY (idLieu)
);

// commentaire sur une ligne
INSERT INTO ARBLIEUX VALUES ('1', 'FRANCE', '79', 'ST MAIXENT L ECOLE');
INSERT INTO ARBLIEUX VALUES ('2', 'FRANCE', '79', 'AIEFFRES');
INSERT INTO ARBLIEUX VALUES ('3', 'FRANCE', '79', 'NIORT');
SELECT * FROM ARBLIEUX;

```

```

dos commande %0
    le paramètre %0 sera remplacé par le nom du fichier en édition

```

```

dos commande %2
    le paramètre %2 sera remplacé par le contenu de la ligne courante de l'éditeur

```

## 7 / HISTORIQUE DES MISES A JOUR

-----

version 3.0 du 25 décembre 2011

- adaptation lde l'atelier au monde linux (debian)
- en particulier l'accès au fichier commençant par /

version 1.90 du 10 novembre 2005

- bug gestion des paramètres avec JRE 1.4
- impression pdf sans description du chemin d'accès du reader

version 1.80 du 8 juillet 2004

- utilisation des api de mail.jar

version 1.70 du 22 mai 2003

- correction traitement des actions - bug suite traitement du %2

version 1.69 du 6 mai 2003

- correction détection changement du paramétrage des outils

version 1.68 du 9 avril 2003

- modif de la macro \$mail
- body = nom d'un fichier

- correction dialog des outils

version 1.67 du 18 février 2003

- ajout du paramètre %0 et %2 dans la ligne de commande

version 1.66 du 3 février 2003

- correction suppression d'un fichier dans la liste de gauche

version 1.64 du 2 janvier 2003

- correction lecture des outils si liste vide

version 1.61 de octobre 2002

- ajout paramètre de type liste {param,item 1,item2,...}
- nouvelle dialog de saisie des outils et actions
- modif technique du projet développé désormais dans Eclipse.  
Les version précédentes étaient développé dans JBuilder 4 qui ne permettait plus d'évoluer sauf en prenant une version payante.

version 1.60 de mars 2002

- ajout gestion des extensions
- version 1.51 du 1er décembre 2001
- ajout user et mot de passe à la commande SQL connect
- connect nomBase user motDePasse;
- suppression paramètre print.cmd print.titre

version 1.5 de octobre 2001

- ajout fonctions FTP sur l'éditeur avec soumission de job sur le MVS

version 1.3 de juin 2001

- ajout de la macro \$play
- ajout de la macro \$getHttp

version 1.1 de février 2001

- ajout de la macro \$ecouteSocket
- ajout de la macro \$socket
- ajout de la macro \$mail
- ajout de la macro \$edit

novembre 2000

- la première version 1.0

Vos remarques et suggestions seront les bienvenues.

philippe.billerot@laposte.net