

Hinemos® Ver. 2.3  
拡張機能インストール手順書

株式会社 イトダコンピュータ

# — 目次 —

はじめに.....	1
1. Hinemos 拡張機能プログラムファイルのインストール.....	2
1.1. ダウンロードファイル「hnex.war.tar.gz」のコピー.....	2
1.2. Hinemos マネージャサーバにログインする.....	2
1.3. カレントディレクトリの移動.....	2
1.4. ダウンロードファイル「hnex.war.tar.gz」の解凍.....	2
1.5. ライブラリの入手・設定.....	6
2. 環境情報の設定.....	7
2.1. DB.ini ファイルの編集.....	7
3. データベースのセットアップ.....	8
3.1. Hinemos マネージャサーバにログインする.....	8
3.2. カレントディレクトリの移動.....	8
3.3. psql の起動.....	8
3.4. テーブルの作成.....	9
3.5. インデックスの作成.....	9
3.6. ランゲージの作成.....	9
3.7. ファンクションの作成.....	10
3.8. マスタデータの作成.....	10
4. インシデント自動登録処理の設定.....	11
4.1. Hinemos マネージャサーバにログインする.....	11
4.2. cron の設定.....	11
5. Hinemos 拡張機能の動作確認.....	12
5.1. JBOSS を再起動する.....	12
5.2. Hinemos 拡張機能メニュー表示.....	12

## はじめに

Hinemos 拡張機能システムは、Hinemos マネージャサーバに追加インストールするため、Hinemos マネージャのセットアップを完了しておく必要があります。

※ Hinemos マネージャのセットアップについては、Hinemos インストール手順書を参照してください。

## 1. Hinemos 拡張機能プログラムファイルのインストール

### 1.1. ダウンロードファイル「hnex.war.tar.gz」のコピー

コマンドプロンプトの **FTP** コマンドや、**FTP** ツール等を使用し、**Hinemos** マネージャの「/opt/hinemos/jboss-4.0.3SP1/server/default/deploy」へダウンロードした「hnex.war.tar.gz」ファイルをコピーします。

### 1.2. Hinemos マネージャサーバにログインする

Hinemos マネージャサーバに hinemos ユーザでログインします。  
hinemos ユーザ以外のユーザでログインしている場合は、以下のコマンドでユーザを切り替えます。

```
> su hinemos
```

### 1.3. カレントディレクトリの移動

以下のコマンドを実行して、カレントディレクトリを移動します。

```
> cd /opt/hinemos/jboss-4.0.3SP1/server/default/deploy
```

### 1.4. ダウンロードファイル「hnex.war.tar.gz」の解凍

以下のコマンドを実行して、ダウンロードファイル「hnex.war.tar.gz」を解凍します。

```
> tar -xzf hnex.war.tar.gz
```

解凍が終了すると「/opt/hinemos/jboss-4.0.3SP1/server/default/deploy」の配下に、「hnex.war」ディレクトリが作成されます。「hnex.war」のディレクトリ構成は以下のようになります。（■はディレクトリ、・はファイルの意味）

■hnex.war
■conf
・DB.ini
・FILEATTACH.ini
■css
・hnexCommon.css
■DB
■data
・mst_dockubun_dump_ins
・mst_hardsyubetu_dump_ins
・mst_hardwarefuzokusybetu_dump_ins
・mst_incidentbunrui_dump_ins
・mst_itakusyubetsu_dump_ins
・mst_jyuyodo_dump_ins
・mst_licencetani_dump_ins
・mst_syssyubetu_dump_ins
・mst_taioukekka_dump_ins
■function
・create_func_incidentautosave.sql

■ hnex. war
■ DB
■ index
• create_idx_dat_incident.sql
■ language
• create_lang_plpgsql.sql
■ table
• create_tbl_dat_hardwarefuzoku.sql
• create_tbl_dat_hardwarekousei.sql
• create_tbl_dat_incident.sql
• create_tbl_dat_incidenthokanbasyo.sql
• create_tbl_dat_itakugyosya.sql
• create_tbl_dat_shintyoku.sql
• create_tbl_dat_shintyokuhokan.sql
• create_tbl_dat_software.sql
• create_tbl_dat_syshokanbasyo.sql
• create_tbl_dat_sysinfo.sql
• create_tbl_dat_taiourireki.sql
• create_tbl_dat_taiourirekihokanbasyo.sql
• create_tbl_mst_bukyoku.sql
• create_tbl_mst_dockubun.sql
• create_tbl_mst_gyosya.sql
• create_tbl_mst_hardsyubetu.sql
• create_tbl_mst_hardwarefuzokusybetu.sql
• create_tbl_mst_incidentbunrui.sql
• create_tbl_mst_itakusyubetsu.sql
• create_tbl_mst_jyuyodo.sql
• create_tbl_mst_kashitsu.sql
• create_tbl_mst_licencetani.sql
• create_tbl_mst_milestone.sql
• create_tbl_mst_software.sql
• create_tbl_mst_syokuin.sql
• create_tbl_mst_syssyubetu.sql
• create_tbl_mst_taioukekka.sql
■ doc
■ ijikanri
■ incident
■ kouchiku
■ narejji
■ syssub
■ system
■ taiou
■ images
• (画像ファイル名記述省略)
■ log

■ hnex.war
■ scripts
■ calendar
• bk.gif
• blank.html
• body_cal.html
• calendar.css
• calendar.js
• head_cal.html
• index.html
• hnex0010.js
• hnex0020.js
• hnex0051.js
• hnex0052.js
• hnex0053.js
• hnex0054.js
• hnex0055.js
• hnex0056.js
• hnex0057.js
• hnex0058.js
• hnex0080.js
• hnex0090.js
• hnex0110.js
• hnex0120.js
• hnex0210.js
• hnex0220.js
• hnex0310.js
• hnex0320.js
• hnex0330.js
• hnex0340.js
• hnex0350.js
• hnex0410.js
• hnex0510.js
• hnex0610.js
• hnex0620.js
• hnex0710.js
• hnex0720.js
• hnex0730.js
• hnex0740.js
• hnex0750.js
• hnex0810.js
• hnex0820.js
• hnex9000.js
• hnex9010.js
• hnex9020.js
• hnex9030.js
• hnex9040.js
• hnex9050.js

■ hnex.war
■ scripts
• hnex9060.js
• hnex9070.js
• hnex9080.js
• hnex9090.js
• hnex9100.js
• hnex9110.js
• hnex9120.js
• hnexCommon.js
■ WEB-INF
■ classes
■ dbAccess
• DataBase.class
• DataBase.java
■ fileAttachment
• DownloadServlet.class
• DownloadServlet.java
• web.xml
• brows.sh
• hnex0010.jsp
• hnex0020.jsp
• hnex0051.jsp
• hnex0052.jsp
• hnex0053.jsp
• hnex0054.jsp
• hnex0055.jsp
• hnex0056.jsp
• hnex0057.jsp
• hnex0058.jsp
• hnex0080.jsp
• hnex0090.jsp
• hnex0110.jsp
• hnex0120.jsp
• hnex0210.jsp
• hnex0220.jsp
• hnex0310.jsp
• hnex0320.jsp
• hnex0330.jsp
• hnex0340.jsp
• hnex0350.jsp
• hnex0410.jsp
• hnex0510.jsp
• hnex0610.jsp
• hnex0620.jsp
• hnex0710.jsp
• hnex0720.jsp
• hnex0730.jsp
• hnex0740.jsp

■ hnex. war
• hnex0750. jsp
• hnex0810. jsp
• hnex0820. jsp
• hnex9000. jsp
• hnex9010. jsp
• hnex9020. jsp
• hnex9030. jsp
• hnex9040. jsp
• hnex9050. jsp
• hnex9060. jsp
• hnex9070. jsp
• hnex9080. jsp
• hnex9090. jsp
• hnex9100. jsp
• hnex9110. jsp
• hnex9120. jsp
• hnexIncidentAutoSave. jsp

### 1.5. ライブラリの入手・設定

拡張機能を動作させる上で必要なライブラリがあります。「commons-fileupload-1.0.jar」です。下記の URL より「commons-fileupload-1.0.jar」を入手します。

<http://archive.apache.org/dist/commons/fileupload/binaries/>

上記のページより、「commons-fileupload-1.0.zip」というパイパーリンクをクリックします。

上記のページより、「commons-fileupload-1.0.zip」というパイパーリンクをクリックします。ダウンロード確認が行われますので、保存ボタンをクリックします。

「commons-fileupload-1.0.zip」のダウンロードが終了しましたら、このファイルを解凍します。解凍後「commons-fileupload-1.0」というフォルダができますので、その配下の「commons-fileupload-1.0.jar」ファイルをHinemos マネージャサーバに配置します。配置する場所は「/opt/hinemos/jboss-4.0.3SP1/server/default/lib」の配下となります。

## 2. 環境情報の設定

### 2.1. DB.ini ファイルの編集

/opt/hinemos/jboss-4.0.3SP1/server/default/deploy/conf/ ディレクトリ配下の DB.ini ファイルをエディタなどで開き、db.server に Hinemos マネージャ機の IP アドレスを記述します。

< DB.ini >

```
db.driver=org.postgresql.Driver
db.server=xxx.xxx.xxx.xxx      ← Hinemos マネージャサーバのアドレスを記述
db.port=24001
db.name=hinemos
db.userid=hinemos
db.password=hinemos
```

### 3. データベースのセットアップ

#### 3.1. Hinemos マネージャサーバにログインする

Hinemos マネージャサーバに hinemos ユーザでログインします。  
hinemos ユーザ以外のユーザでログインしている場合は、以下のコマンドでユーザを切り替えます。

```
> su hinemos
```

#### 3.2. カレントディレクトリの移動

以下のコマンドを実行して、カレントディレクトリを移動します。

```
> cd /opt/hinemos/jboss-4.0.3SP1/server/default/deploy/hnex.war/DB
```

#### 3.3. psql の起動

以下のコマンドを実行して、psql を起動します。

```
> /opt/hinemos/postgresql-8.1.3/bin/psql -p 24001
```

パスワード入力を要求されますので「hinemos」と入力します。

```
> Password:hinemos ← 入力パスワードは画面上に表示されません
```

以下のようにhinemos=# のプロンプトが表示されれば psql の起動が行われたことになります。

```
[hinemos@hnms-mgr-ex root]$ /opt/hinemos/postgresql-8.1.3/bin/psql -p 24001
could not change directory to "/root"
Password:
Welcome to psql 8.1.3, the PostgreSQL interactive terminal.

Type:  ¥copyright for distribution terms
       ¥h for help with SQL commands
       ¥? for help with psql commands
       ¥g or terminate with semicolon to execute query
       ¥q to quit

hinemos=#
```

### 3.4. テーブルの作成

以下のコマンドを実行して、テーブルを作成します。

```
hinemos=# ¥i table/create_tbl_dat_hardwarefuzoku.sql
hinemos=# ¥i table/create_tbl_dat_hardwarekousei.sql
hinemos=# ¥i table/create_tbl_dat_incident.sql
hinemos=# ¥i table/create_tbl_dat_incidenthokanbasyo.sql
hinemos=# ¥i table/create_tbl_dat_itakugyosya.sql
hinemos=# ¥i table/create_tbl_dat_shintyoku.sql
hinemos=# ¥i table/create_tbl_dat_shintyokuhokan.sql
hinemos=# ¥i table/create_tbl_dat_software.sql
hinemos=# ¥i table/create_tbl_dat_syshokanbasyo.sql
hinemos=# ¥i table/create_tbl_dat_sysinfo.sql
hinemos=# ¥i table/create_tbl_dat_taiourireki.sql
hinemos=# ¥i table/create_tbl_dat_taiourirekihokanbasyo.sql
hinemos=# ¥i table/create_tbl_mst_bukyoku.sql
hinemos=# ¥i table/create_tbl_mst_dockubun.sql
hinemos=# ¥i table/create_tbl_mst_gyosya.sql
hinemos=# ¥i table/create_tbl_mst_hardsyubetu.sql
hinemos=# ¥i table/create_tbl_mst_hardwarefuzokusyubetu.sql
hinemos=# ¥i table/create_tbl_mst_incidentbunrui.sql
hinemos=# ¥i table/create_tbl_mst_itakusyubetsu.sql
hinemos=# ¥i table/create_tbl_mst_jyuyodo.sql
hinemos=# ¥i table/create_tbl_mst_kashitsu.sql
hinemos=# ¥i table/create_tbl_mst_licencetani.sql
hinemos=# ¥i table/create_tbl_mst_milestone.sql
hinemos=# ¥i table/create_tbl_mst_software.sql
hinemos=# ¥i table/create_tbl_mst_syokuin.sql
hinemos=# ¥i table/create_tbl_mst_syssyubetu.sql
hinemos=# ¥i table/create_tbl_mst_taioukekka.sql
```

### 3.5. インデックスの作成

以下のコマンドを実行して、インデックスを作成します。

```
hinemos=# ¥i index/create_idx_dat_incident.sql
```

### 3.6. ランゲージの作成

以下のコマンドを実行して、ランゲージを作成します。

```
hinemos=# ¥i language/create_lang_plpgsql.sql
```

### 3.7. ファンクションの作成

以下のコマンドを実行して、ファンクションを作成します。

```
hinemos=# ¥i function/create_func_incidentautosave.sql
```

### 3.8. マスタデータの作成

以下のコマンドを実行して、マスタテーブルにデータをセットします。

```
hinemos=# ¥i data/mst_dockubun_dump_ins  
hinemos=# ¥i data/mst_hardsyubetu_dump_ins  
hinemos=# ¥i data/mst_hardwarefuzokusybetu_dump_ins  
hinemos=# ¥i data/mst_incidentbunrui_dump_ins  
hinemos=# ¥i data/mst_itakusyubetsu_dump_ins  
hinemos=# ¥i data/mst_jyuyodo_dump_ins  
hinemos=# ¥i data/mst_licencetani_dump_ins  
hinemos=# ¥i data/mst_syssyubetu_dump_ins  
hinemos=# ¥i data/mst_taioukekka_dump_ins
```

## 4. インシデント自動登録処理の設定

### 4.1. Hinemos マネージャサーバにログインする

Hinemos マネージャサーバに root ユーザでログインします。  
root ユーザ以外のユーザでログインしている場合は、以下のコマンドでユーザを切り替えます。

```
> su root
```

### 4.2. cron の設定

以下のコマンドを実行して、クローン設定ファイルを開きます。

```
> crontab -e
```

クローン設定ファイルに、以下の内容を記述します。  
\*/5”は5分間隔を意味しますので、10分がよければ\*/10と変更してください

```
*/5 * * * * /opt/hinemos/jboss-4.0.3SP1/server/default/deploy/hnex.war/brows.sh
```

最後に変更状態を保存し、ファイルを閉じます。  
ESC キーを押下し、以下のコマンドを入力してください。

```
:wq
```

## 5. Hinemos 拡張機能の動作確認

### 5.1. JBOSS を再起動する

今まで行ってきた内容を反映させるために、JBOSS の再起動を行わなければいけません。

root ユーザ以外のユーザでログインしている場合は、以下のコマンドでユーザを切り替えます。以下のコマンドを入力します。

```
> su root
```

JBOSS を停止します。以下のコマンドを入力します。

```
> /opt/hinemos/bin/jboss_stop.sh
```

JBOSS を起動します。以下のコマンドを入力します。開始するまで少々時間がかかります。

```
> /opt/hinemos/bin/jboss_start.sh
```

### 5.2. Hinemos 拡張機能メニュー表示

ブラウザから以下の URL を表示し、Hinemos 拡張機能システムメニューが表示される事を確認します。

```
http://xxx.xxx.xxx.xxx:8080/hnex/hnex0010.jsp
```

※ xxx.xxx.xxx.xxx には Hinemos マネージャ機の IP アドレスが入ります。

以下のようにメニュー画面が表示されますと、インストールが完了となります。

