Hinemos® ver.3.1

設定リファレンス

第 1.2 版

2009 年 8 月 24 日

株式会社　ＮＴＴデータ

変更履歴

版	変更日	変更内容
1.0	2009/04/13	Hinemos Ver. 3.1.0 設定リファレンス 初版リリース
1.1	2009/06/16	3.6.4.メール通知のエンベロープ From の設定方法を追記
1.2	2009/08/24	3.11.syslog-ng 監視の設定を追記 3.4.1 hinemos_setup_collectiverun.sh の説明を追記 3.14 イベントの最大ダウンロード件数の設定 の追記 6. Hinemos マネージャの設定一覧を追記 7. Hinemos エージェントの設定一覧を追記

目次

本ソフトウェアは独立行政法人情報処理推進機構(IPA)の2004年度下期オープンソースソフトウェア活用基盤整備事業の委託を受けて開発しました。

テーマ名は「分散ファシリティ統合マネージャの開発」です。

http://www.ipa.go.jp/software/open/2004/result.html

商標

Hinemosは、(株)NTTデータの登録商標です。
Linuxは、Linus Torvalds氏の米国およびその他の国における登録商標または商標です。
その他、本書に記載されている会社名、製品名は、各社の登録商標または商標です。
なお、本文中には TM、®マークは表記しておりません。

1. はじめに

本リファレンスでは、Hinemos 動作の設定変更方法について説明します。本リファレンスでの設定は一例であり、実際に使用される際はご利用の環境のセキュリティポリシーに沿って設定を変更して使用されることをお勧めします。本ソフトウェアの使用により生じたいかなる損害に対しても、弊社は一切の責任を負いません。

2. 前提条件

本リファレンスは Hinemos のインストールを行った後にセキュリティ確保、性能チューニング、動作変更を行うための設定ファイルと設定方法を示します。本リファレンスの内容を実施するためには、あらかじめ Hinemos マネージャ、エージェント、クライアントのインストールが行われていることが必要です。

また、設定の内容によっては、Hinemos の動作が不安定になることがあります。設定を行う前にあらかじめバックアップを取得しておくことをお勧めします。

2.1. Hinemos のバックアップ方法

Hinemos マネージャは、以下のデータベースを使用します。

- PostgreSQL
- OpenLDAP

hinemos_mng_backup.sh コマンドを用いると、これらのバックアップ(ダンプファイルの生成)が行なえます。詳細はユーザマニュアル「データベースのバックアップ・リストアを行なう」を参照ください。

3. Hinemos の機能セットアップに関する追加設定

Hinemos では、各パッケージのインストールを行うとすぐにお使いいただけますが、さらに効果的にお使いいただくために追加の設定が必要になる場合があります。

3.1. Hinemos マネージャをサービス化する

Hinemos マネージャは、インストール直後はスクリプトを実行することで起動を行います。Hinemos マネージャを OS 起動時から有効にするために以下の手順を実行します。

3.1.1. サービス起動スクリプト

Hinemos マネージャをサービス化するためのスクリプトが、Hinemos マネージャのパッケージに含まれています。

(Hinemos_Manager-3.1.0_rhel5_32/hinemos/sbin/service/hinemos_manager ファイル)

サービス化するには、root ユーザで以下のコマンドを実行し、サービス起動スクリプトを配置します。下記の例では、/tmp ディレクトリ配下に、Hinemos マネージャのパッケージを展開した場合の手順です。

```
# cd /tmp/Hinemos_Manager-3.1.0_rhel5_32/hinemos/sbin/service/
# cp hinemos_manager /etc/init.d/
```

3.1.2. サービスの起動

Hinemos マネージャをサービスとして起動します。サービス起動コマンドを実行すると、PostgreSQL, OpenLDAP, JBoss を、この順番に起動していきます。

root ユーザで以下のコマンドを実行してください。

```
# service hinemos_manager start
Starting hinemos_pg:                                       [  OK  ]
Starting hinemos_ldap:                                     [  OK  ]
Starting hinemos_jboss:                                    [  OK  ]
```

上記のように表示された後、JBoss が起動完了するまでには暫く時間を要します。起動完了しているかどうかは、JBoss が出力するログファイル(/opt/hinemos/var/log/jboss.log)から確認することができます。

以下に、コマンド例を示します。(日付、時刻については読み替えてください。)

```
# tail /opt/hinemos/var/log/jboss.log
 (中略)
2008-08-28  15:37:07,111  INFO     [org.jboss.system.server.Server]  JBoss  (MX
MicroKernel) [4.2.2.GA (build: SVNTag=JBoss_4_2_2_GA date=200710221139)] Started in
33s:594ms
```

3.1.3. 起動状態の確認

Hinemos マネージャの起動状態を確認します。JBoss, PostgreSQL, OpenLDAP がそれぞれ起動状態である場合、Hinemos マネージャが起動状態であることを表すメッセージ（Hinemos is running）が出力されます。加えて、JBoss, PostgreSQL, OpenLDAP のプロセス ID が表示されます。

root ユーザで以下のコマンドを実行してください。

```
# service hinemos_manager status
Hinemos is running...
------------------
JBoss pid        : 4393
PostgreSQL pid : 4330
OpenLDAP pid     : 4340
------------------
```

3.1.4. サービスの停止

Hinemos マネージャを停止します。サービス停止コマンドを実行すると、JBoss, OpenLDAP, PostgreSQL を、この順番で停止していきます。

root ユーザで以下のコマンドを実行してください。

```
# service hinemos_manager stop
Stopping hinemos_jboss: ...ok
                                                        [  OK  ]
Stopping hinemos_ldap:                                  [  OK  ]

Stopping hinemos_pg: server stopped
                                                        [  OK  ]
```

3.1.5. 自動起動の設定

OS 起動時に Hinemos マネージャを自動的に起動するように設定するには、root ユーザで以下のコマンドを実行してください。

```
# chkconfig --add hinemos_manager
```

3.2. Hinemos エージェントをサービス化する

Hinemos エージェントは、インストール直後はスクリプトを実行することで起動を行います。Hinemos エージェントを OS 起動時から有効にするために以下の手順を実行します。

3.2.1. ジョブエージェントとログ転送エージェントのサービス化

ジョブエージェントとログ転送エージェントをサービス化するためのスクリプトが、Hinemos エージェントのパッケージ(Hinemos_Agent-3.1.0_rhel5_32/service ディレクトリ）に含まれています。

サービス化するには、root ユーザで以下のコマンドを実行してください。

```
# cd /tmp/Hinemos_Agent-3.1.0_rhel5_32/service/
# cp hinemos_agent /etc/init.d/
# cp hinemos_log_agent /etc/init.d/
```

3.2.2. サービスの起動

サービスとして登録したジョブエージェントとログ転送エージェントを起動します。

root ユーザで以下のコマンドを実行してください。

```
# service hinemos_agent start
hinemos_agent を起動中:                                    [  OK  ]
# service hinemos_log_agent start
hinemos_log_agent を起動中:                                [  OK  ]
```

3.2.3. サービスの停止

サービスとして起動しているジョブエージェントとログ転送エージェントを停止させます。

root ユーザで以下のコマンドを実行してください。

```
# service hinemos_agent stop
hinemos_agent を停止中:                                    [  OK  ]
# service hinemos_log_agent stop
hinemos_log_agent を停止中:                                [  OK  ]
```

3.2.4. 自動起動の設定

OS 起動時にジョブエージェントとログ転送エージェントを自動的に起動するように設定するには、root ユーザで以下のコマンドを実行してください。

```
# chkconfig --add hinemos_agent
# chkconfig --add hinemos_log_agent
```

3.3. 性能管理機能、監視管理機能を有効にする

性能管理機能・監視管理機能は Hinemos インストール時からおおむね有効ですが、Hinemos エージェントをインストールしていないノードの監視を行う際などに SNMP などの設定が必要になる場合があります。

また、WBEM を利用して監視する場合は、対象ノードの CIM サーバ（tog-pegasus）と http で通信できる必要があります。

3.3.1. Net-SNMP の設定

性能管理機能、監視管理機能（リソース監視、プロセス監視、SNMP 監視）では、各ノードからの各種情報を、SNMP や WBEM（Linux ノードのみ）を利用して取得しています。Net-SNMP がインストールされている状態で、エージェントをインストールした場合、以下の設定が Net-SNMP の設定ファイルに追記されます。

/etc/snmp/snmpd.conf

```
view    systemview      included        .1.3.6.1
```

- Net-SNMP の再起動
  root ユーザで以下のコマンドを実行します。

```
# service snmpd restart
```

3.3.2. SNMP・WBEM の切り替え方法

性能管理機能、監視機能（リソース監視、プロセス監視）では、各ノードからの各種情報集を、SNMP や WBEM（Linux ノードのみ）を利用して取得しています。

性能管理機能、監視機能（リソース監視）では、カテゴリ（CPU、メモリ、ディスク、ネットワーク、ファイルシステム）単位で、SNMP と WBEM を切り替えることができます（変更する際には、DB のバックアップを取得し、実施してください）。

マネージャサーバにて、ユーザ hinemos で、以下のコマンドを実行します。その際にパスワード入力を求められますので、パスワードを入力します（初期パスワードは”hinemos”）。

```
$ su - hinemos
$ /opt/hinemos/postgresql-8.3.1/bin/psql -p 24001 -c "UPDATE
  cc_collector_category_collect_mst SET collect_method = '(変更したいプロトコル)' WHERE
  category_code = '(変更したいカテゴリ)' and platform_id = 'LINUX'"
$ Password:
```

「変更したいプロトコル」部分には、「SNMP」か「WBEM」を入力してください（初期値はSNMP）。
「変更したいカテゴリ」部分は、以下の 5 つの中から変更したいカテゴリを入力してください。

- C000_CPU　･･･ CPU 関連の情報
- C001_MEM ･･･ メモリ関連の情報
- C002_DSK　･･･ ディスク関連の情報
- C003_NET　･･･ ネットワーク関連の情報
- C004_FS　　･･･ ファイルシステム関連の情報

※ WBEM プロトコルでは、一部取得できない値が存在します。また、WBEM を利用して監視できるファイルシステムは、EXT3/EXT2 **のみとなります。**

監視機能（プロセス監視）では、SNMP と WBEM を切り替えることができます。

マネージャサーバにて、ユーザ hinemos で、以下のコマンドを実行します。その際にパスワード入力を求められますので、パスワードを入力します（初期パスワードは"hinemos"）。

```
$ su - hinemos
$ /opt/hinemos/postgresql-8.3.1/bin/psql -p 24001 -c "UPDATE
  cc_monitor_process_method_mst SET collect_method = '(変更したいプロトコル)' WHERE
  platform_id = 'LINUX'"
$ Password:
```

「変更したいプロトコル」部分には、「SNMP」か「WBEM」を入力してください（初期値はSNMP）。

3.4. 一括制御機能を有効にする

一括制御機能は Hinemos インストール直後には利用可能になっていないので、一括制御を利用する場合にはセットアップを行う必要があります。

3.4.1. エージェント側のセットアップ

Hinemos で一括制御機能を利用するためには、ssh または rsh でマネージャから管理対象ノードにコマンド実行ができる必要があります。そのためのセットアップに関して説明します。

ssh を使用する場合

一括制御機能で使用するリモートシェルを ssh とする場合は、対象となる管理対象ノードで ssh の設定を行う必要があります（一括制御機能のリモートシェルとして rsh を利用する場合はこの設定は不要です)。

公開鍵の登録を行なうことで、マネージャサーバ（hinemos ユーザ）から、管理対象ノード（root ユーザ）へパスワードなしでコマンド実行が可能となる設定を行ないます。管理対象ノードを agent01（192.168.0.10）として説明します。

1. Hinemos マネージャがインストールされているマネージャサーバで、hinemos_setup_collectiverun.sh を実行します。実行は、JBoss 起動ユーザにスイッチして行います。
   以下、JBoss 起動ユーザが hinemos の場合において説明します。

注）hinemos_setup_collectiverun.sh を実行するには、expect(5.43.0-5.1 以降)のパッケージがマネージャにインストールされている必要があります。

```
# su - hinemos
$ cd /opt/hinemos/sbin/
$ ./hinemos_setup_collectiverun.sh
```

メニューが表示されます。

```
  ##########################################################################
  ###                                                                    ###
  ###              運用管理ソフトウェア Hinemos                          ###
  ###                  一括制御 セットアップスクリプト  Ver 3.1.0 ###
  ###                                                                    ###
```

```
###  Copyright (C) since 2006 NTT DATA Corporation.            ###
#########################################################################

一括制御機能のセットアップ

     1) hinemos ユーザの認証用の公開鍵の作成
     2) 監視対象ノードへの公開鍵登録
     9) 一括制御機能のセットアップ終了

===>1
```

2. プロンプトに“1”を入力します。
3. hinemos ユーザの認証用公開鍵の作成の確認メッセージが表示されますので、“Y”を入力します。

```
hinemos ユーザの認証用の公開鍵をパスフレーズなしで作成します。
よろしいですか? (Y/N default:Y)
Y
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hinemos/.ssh/id_rsa):
Created directory '/home/hinemos/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hinemos/.ssh/id_rsa.
Your public key has been saved in /home/hinemos/.ssh/id_rsa.pub.
The key fingerprint is:
**:**:**:**:**:**:**:**:**:**:**:**:**:**:**:** hinemos@manager

公開鍵を作成しました。


一括制御機能のセットアップ

     1) hinemos ユーザの認証用の公開鍵の作成
     2) 監視対象ノードへの公開鍵登録
     9) 一括制御機能のセットアップ終了

===>
```

4. プロンプトに“2”を入力します。
5. 管理対象ノードの root ユーザの authorized_keys ファイルに、マネージャサーバの hinemos ユーザの公開鍵を登録します。

```
===>2
```

```
監視対象ノードの root ユーザの authorized_keys ファイルに、hinemos ユーザの公開鍵を登
録します。

一括制御機能を利用する監視対象ノードの IP アドレスを入力して下さい。
終了する場合は、9 を入力して下さい。
192.168.0.10
```

6. 管理対象ノードの root ユーザのホームディレクトリを指定します。

```
監視対象ノードの root ユーザのホームディレクトリを指定して下さい。
/root/
/root/.ssh/authorized_keys に、hinemos ユーザの公開鍵を設定します。
よろしいですか? (Y/N default:Y)
Y
```

7. ホスト鍵の登録の確認メッセージが表示されますので、“yes”を入力します。続いて agent01 の root ユーザのパスワードの入力が求められますので入力します。

```
The authenticity of host '192.168.0.10 (192.168.0.10)' can't be established.
RSA key fingerprint is **:**:**:**:**:**:**:**:**:**:**:**:**:**:**:**.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.0.10' (RSA) to the list of known hosts.
root@192.168.0.10's password: (パスワードを入力)

192.168.0.10 への登録が完了しました。

一括制御機能を利用する監視対象ノードの IP アドレスを入力して下さい。
終了する場合は、9 を入力して下さい。
```

8. 終了する場合は“9”を 2 回入力してください。続いて登録する場合は管理対象ノードの IP アドレスを入力して下さい。

```
9

一括制御機能のセットアップ

    1) hinemos ユーザの認証用の公開鍵の作成
    2) 監視対象ノードへの公開鍵登録
    9) 一括制御機能のセットアップ終了

===> 9
```

rsh を使用する場合

一括制御機能で使用するリモートシェルを rsh とする場合は、対象となる管理対象ノードで rsh の設定を行う必要があります（一括制御機能のリモートシェルとして ssh を利用する場合はこの設定は不要です)。

2. /root ディレクトリ直下に、下記の内容の.rhosts ファイルを作成してください（既に.rhosts ファイルが存在する場合は、下記内容を追記してください)。

（*マネージャサーバの IP アドレス*） hinemos

例）/root/.rhosts

```
192.168.0.1  hinemos
```

3. /etc/securetty ファイルに、rsh を追加します。

例）/etc/securetty

```
Console
vc/1

(中略)

tty10
tty11
rsh
```

3.4.2. マネージャでのリモートシェルの設定

一括制御機能で利用するリモートシェルとして、ssh か rsh のどちらかを選択して利用することができます（デフォルトは ssh です)。

リモートシェルを rsh に変更する場合は以下のファイルを編集します。編集後、設定を有効にするために Hinemos マネージャを再起動してください。

/opt/hinemos/etc/hinemos.properties

```
##
## 一括制御 実行方法 設定
##
#collective.run.shell=rsh
collective.run.shell=ssh
```

以下のように変更します。

```
##
## 一括制御 実行方法 設定
##
collective.run.shell=rsh
#collective.run.shell=ssh
```

3.4.3. FTP サーバの起動

一括制御機能（RPM インストールとファイルのコピー）を利用する場合、管理対象ノードからアクセス可能な FTP サーバを起動させる必要があります。FTP サーバを起動し、インストール時に指定の FTP ユーザとパスワードで、管理対象ノードからアクセスできることを確認してください。

ここでは、FTP サーバとして Red Hat EL5 AS に含まれる vsftpd を利用した場合の起動方法を説明します。

1. vsftpd がインストールされていることを確認します。
   下記コマンドを実行します。vsftpd-（バージョン）が表示されることを確認してください。

```
# rpm -q vsftpd
```

2. vsftpd を起動します。
   root ユーザで下記コマンドを実行します。

```
# service vsftpd start
```

- Hinemos で利用する FTP サーバの設定変更

Hinemos で利用する FTP サーバは、マネージャサーバインストールの一括制御機能のインストール時に指定したもので設定されます。

インストール後に、一括制御で利用する FTP サーバの IP アドレス、ユーザ、パスワードの変更を行なうには、以下の 2 つのファイルを編集してください。

・/opt/hinemos/lib/cr/cp.sh
・/opt/hinemos/lib/cr/rpminstall.sh

以下のパラメータを編集してください。

```
FTP_HOST="（サーバ名）"
FTP_USER="（ユーザ名）"
FTP_PASSWD="（パスワード）"
```

3.5. SNMP トラップ監視機能を有効にする

Hinemos マネージャはデフォルトではユーザ hinemos で起動する設定となっています。この状態では、SNMP トラップ監視機能が使用できません。これは SNMP トラップの待ち受けポート（162/TCP）をバインドするには root 権限が必要となり、JBoss を root で起動する必要があるためです。セキュリティの観点から、デフォルトではユーザ hinemos で起動することとしています。

SNMP トラップ監視機能を有効とするには、以下の設定を行い、JBoss を root で起動します。

3.5.1. SNMP トラップ受信用のサービスを起動するように設定変更

以下の設定ファイルを JBoss の deploy ディレクトリにコピーします。

/opt/hinemos/contrib/snmptrap-service.xml

```
$ cd /opt/hinemos/contrib
$ cp snmptrap-service.xml /opt/hinemos/jboss-4.2.2.GA/server/default/deploy/
```

3.5.2. JBoss を root ユーザで起動

PostegreSQL および、OpenLDAP はユーザ hinemos で起動し、JBoss のみ root ユーザで起動します。そのために、以下のファイルを編集します。

・ /opt/hinemos/hinemos.cfg

```
export HINEMOS_JBOSS_USER=root（デフォルトは hinemos）
```

下記の手順で、それぞれを順次起動します。

```
$ su - hinemos
$ cd /opt/hinemos/bin/
$ ./pg_start.sh
waiting for postmaster to start....
 done
postmaster started
$ ./ldap_start.sh
$ su
Password:
# ./jboss_start.sh
```

停止の方法は、3.1.4 を参照ください。

注）一度 root ユーザで JBoss を起動すると、次からユーザ hinemos で JBoss を起動することができません（root ユーザが作成したファイルへ一般ユーザではアクセスできないためです)。

その場合は、一旦 JBoss を停止し、下記コマンドを実行してファイルのオーナーをユーザ hinemos に変更した後、再度 JBoss の起動を実行してください。起動ユーザを hinemos に戻した場合は、SNMP トラップ監視機能は利用できません。

```
# chown -R hinemos:hinemos /opt/hinemos/jboss-4.2.2.GA/
# chown -R hinemos:hinemos /opt/hinemos/var/log/
```

以下のファイルを編集します。

・/opt/hinemos/hinemos.cfg

```
export HINEMOS_JBOSS_USER=hinemos
```

3.6. メール通知を有効にする

3.6.1. メールサーバの設定

監視管理機能のメール通知機能で使用するメールサーバの設定を行います。

以下のファイルを編集します。編集後、設定を有効にするために Hinemos マネージャを再起動してください。

/opt/hinemos/etc/mail-service.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- $Id: mail-service.xml,v 1.4.2.2 2003/10/13 12:31:03 starksm Exp $ -->

<server>

  <!-- ===================================================================== -->
  <!-- Mail Connection Factory                                               -->
  <!-- ===================================================================== -->

  <mbean code="org.jboss.mail.MailService"
         name="jboss:service=Mail">
    <attribute name="JNDIName">java:/Mail</attribute>
    <attribute name="User">nobody</attribute>
    <attribute name="Password">password</attribute>
    <attribute name="Configuration">
       <!-- Test -->
       <configuration>
          <!-- Change to your mail server prototocol -->

 (中略)

          <!-- Change to the SMTP gateway server -->
          <property name="mail.smtp.host" value="smtp.nosuchhost.nosuchdomain.com"/>

          <!-- Change to the address mail will be from  -->
          <property name="mail.from" value="nobody@nosuchhost.nosuchdomain.com"/>

          <!-- Enable debugging output from the javamail classes -->
          <property name="mail.debug" value="false"/>
          <property name="mail.smtp.timeout" value="300000"/>
       </configuration>
    </attribute>
  </mbean>
```

```
<!-- Change to the SMTP gateway server -->
<property name="mail.smtp.host" value="（メールサーバの IP アドレス）"/>

<!-- Change to the address mail will be from  -->
<property name="mail.from" value="（メールの送信元として設定するメールアドレス）" >
```

3.6.2. 送信元情報の設定

監視管理機能のメール通知機能で送信されるメールの送信元情報の設定を行います。

以下のファイルを編集します。編集後、設定を有効にするために Hinemos マネージャの再

起動をしてください。

/opt/hinemos/etc/hinemos.properties

```
common.mail.from.address=admin@nosuchdomain.com
common.mail.from.personal.name=Hinemos Admin
common.mail.reply.to.address=admin@nosuchdomain.com
common.mail.reply.personal.name=Hinemos Admin
common.mail.errors.to.address=admin@nosuchdomain.com
```

以下のパラメータを設定してください。

common.mail.from.address=***送信元メールアドレス***
common.mail.from.personal.name=***送信先個人名***
common.mail.reply.to.address=***返信先メールアドレス***
common.mail.reply.personal.name=***返信先個人名***
common.mail.errors.to.address=***送信メールの Errors-To ヘッダに設定するメールアドレス***

3.6.3. SMTP AUTH の設定

監視管理機能のメール通知機能でメールを送信する際、SMTP AUTH を必要とするメールサーバに送信することができます。

以下のファイルを編集します。編集後、設定を有効にするために Hinemos マネージャの再起動をしてください。

/opt/hinemos/etc/mail-service.xml

```
        :

<mbean code="org.jboss.mail.MailService"
       name="jboss:service=Mail">
  <attribute name="JNDIName">java:/Mail</attribute>
  <attribute name="User">nobody</attribute>
  <attribute name="Password">password</attribute>
  <attribute name="Configuration">
    <!-- A test configuration -->
    <configuration>
      <!-- Change to your mail server prototocol -->

        :
      <!-- Use SMTP AUTH or not -->
      <property name="mail.smtp.auth" value="false"/>
        :

    </configuration>
  </attribute>
  <depends>jboss:service=Naming</depends>
</mbean>
```

<attribute name="User">*（ユーザ名）*</attribute>
<attribute name="Password">*（パスワード）*</attribute>

<!-- Use SMTP AUTH or not -->
<property name="mail.smtp.auth" value="*（SMTPAUTH を有効にする場合は true 無効にする場合は false）*"/>

認証方法が LOGIN、PLAIN、DIGEST-MD5 であればメールを送信することができます。また、複数の認証方式が有効になっているメールサーバに送信する際 LOGIN、PLAIN、DEGEST-MD5 の順で選択されます。

3.6.4. エンベロープ From の設定

メール通知において、SMTP の MAIL コマンドに渡される引数を設定することができます。

以下のファイルを編集します。編集後、設定を有効にするために Hinemos マネージャを再起動してください。

/opt/hinemos/etc/mail-service.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- $Id: mail-service.xml,v 1.4.2.2 2003/10/13 12:31:03 starksm Exp $ -->

<server>

  <!-- ===================================================================== -->
  <!-- Mail Connection Factory                                               -->
  <!-- ===================================================================== -->

  <mbean code="org.jboss.mail.MailService"
         name="jboss:service=Mail">
    <attribute name="JNDIName">java:/Mail</attribute>
    <attribute name="User">nobody</attribute>
    <attribute name="Password">password</attribute>
    <attribute name="Configuration">
       <!-- Test -->
       <configuration>
          <!-- Change to your mail server prototocol -->

 (中略)

          <property name="mail.smtp.timeout" value="300000"/>
       </configuration>
    </attribute>
  </mbean>

</server>
```

以下のパラメータを<configuration>タグ内に追記してください。

<property name=”mail.smtp.from” value=”エンベロープ From のアドレス”>

3.7. ファイル転送ジョブを有効にする

ファイル転送ジョブを使用する場合、下記の設定が必要となります。設定後、設定を有効にするために Hinemos エージェントの再起動をしてください。(一括制御機能でも設定できます。詳しくはユーザマニュアルの「ファイル転送ジョブの設定」の項を参照ください。)

・転送先の Agent.properties に、転送を実行するユーザの公開鍵を登録する。

・転送元の Agent.properties に、転送を実行するユーザの authorized_keys ファイル登録する

・ホスト鍵を登録する。

以下に、ファイル転送ジョブ設定の手順を示します。ここでは転送元ノードを agent01（192.168.0.10)、転送先ノードを agent02（192.168.0.11)、転送するユーザを hinemos として説明します。

※尚、転送元ノード上および転送先ノード上に、同一の転送を実施するユーザが存在するものとします。

1.転送先ノード（agent02）で転送するユーザ（hinemos）にスイッチユーザします。

```
[root@agent02 ~]# su - hinemos
[hinemos@agent02 ~]$
```

2.転送するユーザ（hinemos）の公開鍵を表示します。まだ、作成していない場合には転送するユーザ（hinemos）の認証用の公開鍵をパスフレーズなしで生成し表示します。

```
[hinemos@agent02 ~]$ cd .ssh/
[hinemos@agent02 .ssh]$ cat id_rsa.pub
ssh-rsa ****(中略)***** = hinemos@agent02
[hinemos@agent02 .ssh]$
```

3.root ユーザにスイッチユーザし、Agent.properties に上記で表示された公開鍵を登録します。

```
[hinemos@agent02 .ssh]$ su -
Password:
[root@agent02 ~]# vi /opt/hinemos_agent/lib/agent/Agent.properties

##
## サーバ接続設定
##
java.naming.factory.initial=org.jnp.interfaces.NamingContextFactory
（中略）

##scp(ssh)公開鍵
hinemos.public.key= ssh-rsa ****(中略)***** = hinemos@agent02
hinemos.authorized.keys.path=/home/hinemos/.ssh/authorized_keys
```

以下のパラメータを追加します（既にある場合には変更します）

（転送するユーザ）.public.key=　（上記で表示した公開鍵）

4.転送するユーザ（hinemos）で転送元のノード（agent01）にログインし、ホスト鍵を登録します。

```
[root@agent02 ~]# exit
[hinemos@agent02 ~]$ ssh 192.168.0.10
The authenticity of host '192.168.0.10 (192.168.0.10)' can't be established.
RSA key fingerprint is **:**:**:**:**:**:**:**:**:**:**:**:**:**:**:**.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.0.10' (RSA) to the list of known hosts.
hinemos@192.168.0.10's password:
[hinemos@agent01 ~]$
```

5.転送元ノード（agent01）に authorized_keys ファイルがなければ作成します。

```
[hinemos@agent01 ~]$ mkdir .ssh
[hinemos@agent01 ~]$ chmod 700 .ssh
[hinemos@agent01 ~]$ cd .ssh
[hinemos@agent01 .ssh]$ touch authorized_keys
[hinemos@agent01 .ssh]$ chmod 600 authorized_keys
```

6.root ユーザにスイッチし、Agent.properties に上記ファイルを設定します。

```
[hinemos@agent01 .ssh]$ su -
Password:
[root@agent01 ~]# vi /opt/hinemos_agent/lib/agent/Agent.properties

##
## サーバ接続設定
##
java.naming.factory.initial=org.jnp.interfaces.NamingContextFactory
（中略）

##scp(ssh)公開鍵
hinemos.authorized.keys.path=/home/hinemos/.ssh/authorized_keys
```

以下のパラメータを追加します（既にある場合には変更します）

（転送するユーザ）. authorized.keys.path =（上記で作成した authorized_keys ファイルのパス）

3.8. ログ転送エージェントを有効にする

3.8.1. エージェントの syslog-ng の設定

監視管理機能（syslog-ng 監視、アプリケーションログ監視）では、各ノードからのログを syslog-ng 経由でマネージャサーバに転送します。インストーラを用いてエージェントをインストールした場合、以下の設定が syslog-ng の設定ファイルに追記されます（注）。

/etc/syslog-ng/syslog-ng.conf

```
#add for Hinemos Agent 3.1.0
filter f_hinemos_log { facility(user);  };
source s_netudp { udp(ip(各ノードの IP アドレス) port(514)); };
log { source(s_netudp); filter(f_hinemos_log); destination(d_hinemos);};
destination d_hinemos { tcp("マネージャサーバの IP アドレス" port(514));};
log { source(s_local);filter(f_mesg);destination(d_hinemos);};
```

ログ転送機能を使用する場合は、syslog-ng の上記設定ファイルの use_dns を変更します。

```
options {
    sync (0);
    time_reopen (10);
    log_fifo_size (1000);
    long_hostnames (off);
    use_dns (yes);
    use_fqdn (no);
    create_dirs (no);
    keep_hostname (yes);
};
```

この場合マネージャサーバにて、監視対象のノード名から IP アドレスを解決する必要があるので、下記の対処のうちいずれかが必要になります。

- DNS サーバに登録
- hosts ファイルに記述

● syslog-ng の再起動

root ユーザで、以下のコマンドを実行します。

```
# service syslog-ng restart
```

注）マネージャサーバの syslog-ng.conf の設定が行われているノードにおいて、ログ転送エージェントを有効にする場合は、本章の設定を行う必要はありません。

3.9. ジョブエージェント、ログ転送エージェントの文字コード設定

3.9.1. ジョブエージェントの文字コード設定

ジョブの実行結果として受け取ることのできるメッセージ（ジョブとして実行されたプロセスの標準出力・標準エラー）の文字コードを以下の方法で指定することができます。

例：MS932 のメッセージを受け取る場合

1. 設定ファイルを変更する
   インストールディレクトリ¥hinemos_agent¥lib¥agent¥Agent.properties を以下のように追記します。

```
input.encoding=MS932
```

図 1　ジョブエージェントの文字コード設定

2. ジョブエージェントを再起動する

3.9.2. ログ転送エージェントの文字コード設定

ログ転送エージェントのデフォルト設定では、扱うことのできる転送対象ログファイルの文字コードは、UTF-8 です。以下の手順で対象文字コードの変更が行えます。

例：MS932 のログファイルの監視を行う場合

1. 設定ファイルを変更する
   インストールディレクトリ¥hinemos_agent¥lib¥log_agent¥Agent.properties を以下のように編集します。

```
log.file.encoding=MS932
```

図 2　ログ転送エージェントの文字コード設定

2. ログ転送エージェントを再起動する

3.10. JBoss 再起動時のジョブスケジュール制御の設定

JBoss 停止中に実行予定時刻を過ぎてしまったジョブスケジュールに関して、JBoss 起動時の動作は下記のようになりますのでご注意ください。

また、データベースのバックアップデータをリストアし、JBoss を再起動する際も、データベース内の情報が過去のものとなり、JBoss 起動時刻との差分が発生するため、同様に注意が必要です（詳細については、Hinemos ユーザマニュアルの注意事項を参照ください）。

- **実行予定時刻からの経過時間が、起動失敗と判定する閾値（デフォルトでは 1 時間）以内の場合**

  JBoss 起動直後に、スケジュールされていたジョブが実行されます。

- **実行契機時刻からの経過時間が、起動失敗と判定する閾値（デフォルトでは 1 時間）以上経過している場合**

  スケジュールされていたジョブの実行は見送られ、次回実行予定時刻に実行されます。

起動失敗と判定するまでの時間の閾値は以下の方法で変更します。

以下のファイルを編集します。

/opt/hinemos/etc/quartz-service.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<server>

  <!--
  <classpath codebase="." archives="quartz.jar"/>
  -->
  <classpath codebase="." archives="hinemos-commons.jar"/>

  <!--
  <mbean code="org.quartz.ee.jmx.jboss.QuartzService"
      name="user:service=QuartzService,name=QuartzService">
  -->
  <mbean code="com.clustercontrol.commons.quartz.CustomQuartzService"
      name="user:service=QuartzService,name=QuartzService">

 (中略)

      org.quartz.dataSource.QuartzDS.maxConnections = 20

      org.quartz.jobStore.misfireThreshold = 3600000
    </attribute>

  </mbean>

</server>

  </local-tx-datasource>
```

以下のパラメータを変更します

org.quartz.jobStore.misfireThreshold = （閾値（ms））

注）本パラメータは、ジョブスケジュール実行だけでなく、監視の定期実行にも影響を与えるものであるため、デフォルトの 360000 より小さくすることはお勧めできません。

3.11. syslog-ng 監視の設定

Syslog-ng 監視のパターンマッチ表現では、Java の正規表現を使用することができます。ただし、Hinemos ver3.0 以降では別途設定が必要です。
手順は以下の通りです。

1. Hinemos マネージャサーバの/etc/syslog-ng/syslog-ng.conf を修正します。

/etc/syslog-ng/syslog-ng.conf

```
#add for Hinemos Manager 3.1.0

（中略）

destination d_hinemos { program("/opt/hinemos/jre1.5.0_15/bin/java -Xms16m -Xmx64m -cp /opt/hinemos/lib/syslogforward:/opt/hinemos/lib/NotifyEJB.jar:/opt/hinemos/lib/SyslogNGEJB.jar:/opt/hinemos/lib/commons-logging-1.1.jar:/opt/hinemos/lib/log4j-1.2.15.jar:/opt/hinemos/lib/hinemos-commons.jar:/opt/hinemos/lib/RepositoryEJB.jar:/opt/hinemos/lib/CalendarEJB.jar:/opt/hinemos/lib/clustercontrol.jar:/opt/hinemos/lib/javassist.jar:/opt/hinemos/lib/jboss-aop-jdk50.jar:/opt/hinemos/lib/jboss-messaging-client.jar:/opt/hinemos/lib/jboss-remoting.jar:/opt/hinemos/lib/jbossall-client.jar:/opt/hinemos/lib/trove.jar:/opt/hinemos/lib/syslogforward/syslogforward.jar: com.clustercontrol.syslogng.forward.LogForward /opt/hinemos/lib/syslogforward/LogForward.properties"  template("<$PRI>$DATE $HOST $MSG¥n") template_escape(no));};
```

2. syslog-ng を再起動します。

```
# service syslog-ng restart
```

3.12. HTTPS 監視の設定

HTTPS 監視では Hinemos マネージャが HTTPS クライアントとなり HTTPS サーバに接続して監視を行います。HTTPS サーバのサーバ証明書を Hinemos マネージャに登録することで HTTPS 監視が可能になります。
手順は以下の通りです。

1.証明書の準備
2.証明書の keystore への登録
3.JBoss 起動オプションによる keystore ファイルの指定
4. Hinemos マネージャの再起動

手順 1～3 の詳細については以下の通りです。

3.12.1. 証明書の取得

HTTPS サーバのサーバ証明書（[DER encoded binary X.509] または [Base-64 encoded X.509]の形式）を準備して下さい。

3.12.2. 証明書の keystore への登録

次に、Java のコマンド keytool にてサーバ証明書を/opt/hinemnos/.keystore ファイルに登録します。

keystore は keytool コマンドを初めて実行した際に作成されます。HTTPS 通信を行う対象サーバ分のサーバ証明書を.keystore に追加します。下記例ではサーバ証明書は Hinemos マネージャサーバの/tmp 配下にあるものとします。別のディレクトリで作業する場合は適宜読み替えてください。また、alias で指定する文字列は.keystore に対してユニークなもの（下記例では hinemos を指定）にして下さい。

```
$ su - hinemos
$ /opt/hinemos/jre1.5.0_15/bin/keytool -import -file /tmp/（サーバ証明書） -alias hinemos
キーストアのパスワードを入力してください: （デフォルトは changeit）
所有者: EMAILADDRESS=root@example.com, CN=172.19.188.60, OU=Testing, O=Test
Company, L=Raleigh, ST=North Carolina, C=JP
実行者: EMAILADDRESS=root@example.com, CN=172.19.188.60, OU=Testing, O=Test Company,
L=Raleigh, ST=North Carolina, C=JP
シリアル番号: 0
有効日: Mon Mar 09 16:03:54 JST 2009 有効期限: Tue Mar 09 16:03:54 JST 2010
```

```
証明書のフィンガープリント:
        MD5:  80:F9:93:D1:F9:A3:0B:77:FD:4B:50:32:A8:D5:E2:44
        SHA1: 08:B5:4B:20:51:98:35:29:B1:B8:77:C3:6F:C8:56:7B:80:A9:72:94
この証明書を信頼しますか? [no]:  yes
証明書がキーストアに追加されました。
```

※keytool コマンドの詳細は
http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/keytool.html
を参照して下さい。

3.12.3. Java 起動オプションによる keystore ファイルの指定

Hinemos マネージャ(JBoss)の起動オプションにて.keystore を参照するように設定を変更します。

/opt/hinemos/jboss-4.2.2.GA/bin/run.conf の javax.net.ssl.trustStore を設定する JAVA_OPT のコメントを外して下さい。

```
#### Hinemos : For Https Connection Keystore File (default : comment out)
JAVA_OPTS="$JAVA_OPTS
-Djavax.net.ssl.trustStore=${HINEMOS_HOME}/.keystore"
```

3.13. プロセス監視の設定（「値取得の失敗」の通知が発生する場合の対処）

プロセス監視は、

①SNMP ポーリングによるプロセス一覧情報を収集する

②収集されたプロセス一覧情報を参照して、プロセス数をカウントする

の２つの処理を非同期に実行することで実現しています。（図参照）

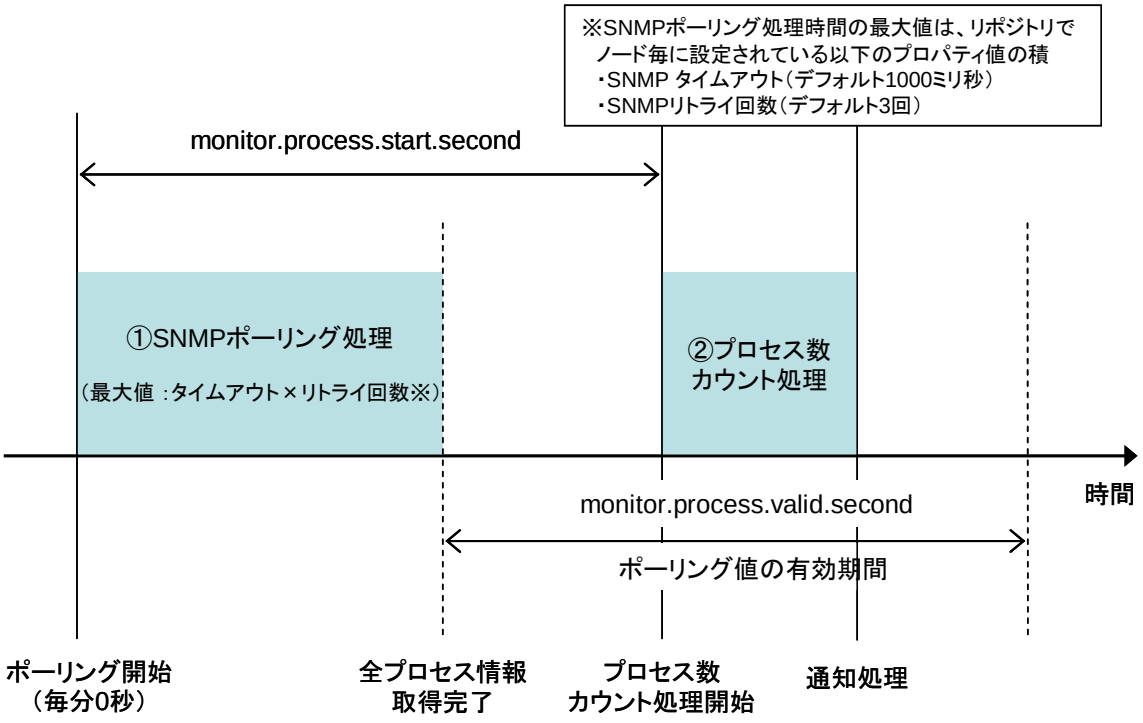

図 3　プロセス監視の処理

設定値 monitor.process.start.second は、①が開始されてから何秒後に②を実行するかを決定するパラメータです。このパラメータは 1～59 の整数値である必要があります。

設定値 monitor.process.valid.second は、①で収集した情報を何秒間有効なものとして扱うかを決定するパラメータです。有効期限が切れた後に②が実行された場合、「取得値が古いためチェックは行われませんでした」の通知が「値取得の失敗」の通知重要度で出力されます。このパラメータは、0 以上の整数値である必要があります。

①の SNMP ポーリングによるプロセス一覧情報の収集にかかる時間は、実行環境の SNMP ポーリングの応答速度に依存します。また、SNMP ポーリングにはタイムアウト時間とリトライ回数のパラメータが設定されているため、最大でも （タイムアウト時間×リトライ回数）［秒］を越えることはありません。この時間内にプロセス一覧情報を収集できなかった場合、「値を取得できませんでした」という情報が①の収集結果となります。

パラメータを変更するには、以下のファイルを編集します。編集後、設定を有効にするために Hinemos マネージャを再起動してください。

加えて、monitor.process.start.second の設定変更を反映するには、既存のプロセス監視の設定を一度「無効」とし、再度「有効」とする必要があります。

/opt/hinemos/etc/hinemos.properties

```
## プロセス監視 値取得開始時間（秒）設定（1-59）
monitor.process.start.second=30

## SNMP ポーラー 収集許容時間（秒）設定
monitor.process.valid.second=50
```

3.14. イベントの最大ダウンロード件数の設定

イベントの最大ダウンロード件数はデフォルトで 1000 件に設定されています。この設定を変更するには、/opt/hinemos/etc/hinemos.properties の以下のパラメータを変更してください。

```
## イベントの最大ダウンロード数
monitor.common.report.event. count = 1000
```

上記の設定変更を反映させるには、JBoss を再起動してください。

4. セキュリティに関する設定

4.1. データベースアクセスのパスワードを変更する

4.1.1. PostgreSQL の設定変更

- 以下の手順でパスワードを変更します。

3. ユーザ hinemos で、以下のコマンドを実行します。その際にパスワード入力を求められますので、初期パスワードである"hinemos"を入力します（変更する際には、DB のバックアップを取得し、実施してください)。

```
$ su - hinemos
$ /opt/hinemos/postgresql-8.3.1/bin/psql -p 24001
Password:
Welcome to psql 8.3.1, the PostgreSQL interactive terminal.
```

4. psql が起動しますので、以下のコマンドを実行します。

```
hinemos= # ALTER USER hinemos PASSWORD '(パスワード)';
```

5. psql を終了します。

```
hinemos= # ¥q
```

- 以下の設定ファイルを編集し、PostgreSQL のアクセス権限を設定します。

/opt/hinemos/var/data/pg_hba.conf

```
# PostgreSQL Client Authentication Configuration File
# ===================================================

(中略)

# TYPE  DATABASE    USER        CIDR-ADDRESS          METHOD

# "local" is for Unix domain socket connections only
local   postgres,hinemos            hinemos            md5
# IPv4 local connections:
host    hinemos         hinemos     0.0.0.0/0          md5
# IPv6 local connections:
#host    all         all         ::1/128               trust
```

“# IPv4 local connections:”の箇所を編集してください。

注）上記の設定は一例です。ご利用の環境のセキュリティポリシーに沿って接続の設定を変更することをお勧めします。

4.1.2. Hinemos マネージャの設定変更

以下の２つのファイルを編集します。編集後、設定を有効にするためには Hinemos マネージャを再起動する必要があります。

- hinemos-ds.xml
- postgres-ds.xml
- quartz-service.xml

1. hinemos-ds.xml の編集

以下のファイルを編集します。

/opt/hinemos/etc/hinemos-ds.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- ===================================================================== -->
<!--                                                                       -->
<!--  JBoss Server Configuration                                           -->
<!--                                                                       -->
<!-- ===================================================================== -->

<!-- $Id: postgres-ds.xml,v 1.3 2004/09/15 14:37:40 loubyansky Exp $ -->
<!-- ===================================================================== -->
<!--  Datasource config for Postgres                                       -->
<!-- ===================================================================== -->


<datasources>
  <local-tx-datasource>
    <jndi-name>HinemosDS</jndi-name>
    <connection-url>jdbc:postgresql://127.0.0.1:24001/hinemos</connection-url>
    <driver-class>org.postgresql.Driver</driver-class>
    <user-name>hinemos</user-name>
    <password>hinemos</password>

 (中略)

  </local-tx-datasource>
</datasources>
```

以下のパラメータに4.1.1 PostgreSQLの設定変更 の手順2で登録したパスワードを設定してください。

<password>*(パスワード)*</password>

2. postgres-ds.xml の編集

以下のファイルを編集します。

/opt/hinemos/etc/postgres-ds.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- ===================================================================== -->
<!--                                                                       -->
<!--  JBoss Server Configuration                                           -->
<!--                                                                       -->
<!-- ===================================================================== -->

<!-- $Id: postgres-ds.xml,v 1.3 2004/09/15 14:37:40 loubyansky Exp $ -->
<!-- ===================================================================== -->
<!--  Datasource config for Postgres                                       -->
<!-- ===================================================================== -->


<datasources>
  <local-tx-datasource>
    <jndi-name>DefaultDS</jndi-name>
    <connection-url>jdbc:postgresql://127.0.0.1:24001/hinemos</connection-url>
    <driver-class>org.postgresql.Driver</driver-class>
    <user-name>hinemos</user-name>
    <password>hinemos</password>

 (中略)

  </local-tx-datasource>
</datasources>
```

以下のパラメータに4.1.1 PostgreSQLの設定変更 の手順2で登録したパスワードを設定してください。

<password>*(パスワード)*</password>

3. quartz-service.xml の編集

以下のファイルを編集します。
/opt/hinemos/etc/quartz-service.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<server>

  <classpath codebase="." archives="quartz.jar"/>

  <mbean code="org.quartz.ee.jmx.jboss.QuartzService"
      name="user:service=QuartzService,name=QuartzService">

    <!--
        Wait until the DataSources  deployed. This option
         ensures correct  deployment order at JBoss startup.
         Change the 'QuartzDS' to your datasource name.
         Important!==> this is NOT the JNDI name of the datasource.
         (JNDI name for it is set in a separate xxx-service.xml file).
     -->

 (中略)

      org.quartz.dataSource.QuartzDS.driver = org.postgresql.Driver
      org.quartz.dataSource.QuartzDS.URL = jdbc:postgresql://127.0.0.1:24001/hinemos
      org.quartz.dataSource.QuartzDS.user = hinemos
      org.quartz.dataSource.QuartzDS.password = hinemos
      org.quartz.dataSource.QuartzDS.maxConnections = 20

      org.quartz.jobStore.misfireThreshold = 120000
    </attribute>


  </mbean>

</server>
```

以下のパラメータに4.1.1 PostgreSQLの設定変更 の手順2で登録したパスワードを設定してください。

org.quartz.dataSource.QuartzDS.password = *(パスワード)*

4.2. LDAP アクセスのパスワードを変更する

4.2.1. LDAP のパスワード変更

1. LDAP ユーザ用のパスワードを生成します。

以下のコマンドを実行します。

```
$ . /opt/Hinemos/Hinemos.cfg
$ /opt/hinemos/openldap-2.3.39/sbin/slappasswd -h {MD5}
```

パスワードの入力を求められますので入力します。

出力された文字列（パスワードのハッシュ）を保存しておきます（2.で設定ファイル slapd.conf に設定します)。

```
例）
  $ slappasswd -h {MD5}
  New password: (パスワード)
  Re-enter new password: (パスワード)
  {MD5}XO3MO1qnZdYdgyfeulLPmQ==  ⇐ パスワードのハッシュ
```

2. パスワードを設定します。

以下のファイルを編集します。

/opt/hinemos/etc/slapd.conf

以下のパラメータを設定してください。

rootpw　　*(slappasswd コマンドで出力された文字列)*

例）/opt/hinemos/etc/slapd.conf

```
#
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
#
include     /opt/hinemos/openldap-2.3.20/etc/openldap/schema/core.schema
include     /opt/hinemos/openldap-2.3.20/etc/openldap/schema/corba.schema

 （中略）

database        bdb
suffix          "dc=hinemos,dc=com"
rootdn          "cn=Manager,dc=hinemos,dc=com"
# Cleartext passwords, especially for the rootdn, should
# be avoid.  See slappasswd(8) and slapd.conf(5) for details.
# Use of strong authentication encouraged.
rootpw          {MD5}X03MO1qnZdYdgyfeuILPmQ==
# The database directory MUST exist prior to running slapd AND
# should only be accessible by the slapd and slap tools.
# Mode 700 recommended.
directory /opt/hinemos/var/openldap-data
# Indices to maintain
index   objectClass eq
index   cn,mail,sn,givenName eq,sub,approx
index   ccFacilityId eq
index   entryCSN,entryUUID eq

#loglevel 256
```

4.2.2. Hinemos マネージャの設定変更

以下のファイルを編集します。編集後、設定を有効にするために Hinemos マネージャの再起動をしてください。

/opt/hinemos/etc/ldap-service.xml

```
<server>
    <!-- ==================================================================== -->
    <!-- LDAP Connection Factory                                              -->
    <!-- ==================================================================== -->

    <!-- Bind a remote LDAP server -->
    <mbean code="org.jboss.naming.ExternalContext"
name="jboss.jndi:service=ExternalContext,jndiName=external/hinemos/ldap/provider">
        <attribute name="JndiName">external/hinemos/ldap/provider</attribute>
        <attribute name="Properties">
            java.naming.factory.initial=com.sun.jndi.ldap.LdapCtxFactory
            java.naming.provider.url=ldap://127.0.0.1:24000/dc=hinemos,dc=com
            java.naming.security.principal=cn=Manager,dc=hinemos,dc=com
            java.naming.security.authentication=simple
            java.naming.security.credentials=hinemos
        </attribute>
        <attribute name="InitialContext">javax.naming.ldap.InitialLdapContext</attribute>
        <attribute name="RemoteAccess">true</attribute>
        <attribute name="CacheContext">false</attribute>
    </mbean>

    <!-- Bind a remote LDAP server -->
    <mbean code="org.jboss.naming.ExternalContext"
name="jboss.jndi:service=ExternalContext,jndiName=external/hinemos/ldap/consumer">
        <attribute name="JndiName">external/hinemos/ldap/consumer</attribute>
        <attribute name="Properties">
            java.naming.factory.initial=com.sun.jndi.ldap.LdapCtxFactory
            java.naming.provider.url=ldap://127.0.0.1:24000/dc=hinemos,dc=com
            java.naming.security.principal=cn=Manager,dc=hinemos,dc=com
            java.naming.security.authentication=simple
            java.naming.security.credentials=hinemos
        </attribute>
        <attribute name="InitialContext">javax.naming.ldap.InitialLdapContext</attribute>
        <attribute name="RemoteAccess">true</attribute>
        <attribute name="CacheContext">false</attribute>
    </mbean>
</server>
```

以下のパラメータに 4.2.1 LDAPのパスワード変更 の手順 1 で登録したパスワードを設定してください（2 箇所あります）。

java.naming.security.credentials= *（パスワード）*

5. Hinemos の動作ログに関する設定

5.1. マネージャのログファイル一覧

Hinemosマネージャのログは 表 5-1 に示すログファイルに出力されます。

表 5-1 Hinemos マネージャのログファイル一覧

ファイル名	HinemosApl.log*
格納ディレクトリ	/opt/hinemos/var/log/
ログ出力設定ファイル	/opt/hinemos/etc/jboss-log4j.xml
出力レベル	priority INFO
ローテーション	Daily(無期限)
内容	Hinemos マネージャの内部イベントログ(内部エラーなど)

ファイル名	Hinemos.log*
格納ディレクトリ	/opt/hinemos/var/log/
ログ出力設定ファイル	/opt/hinemos/etc/jboss-log4j.xml
出力レベル	Priority INFO
ローテーション	Daily(無期限)
内容	JBoss 上で動作する Hinemos アプリケーション部分のログ

ファイル名	boot.log*
格納ディレクトリ	/opt/hinemos/jboss-4.2.2.GA/server/default/log/
ログ出力設定ファイル	―
出力レベル	―
ローテーション	再起動のたびにローテーション
内容	JBoss の boot 時のログ

ファイル名	jboss_stdout.log
格納ディレクトリ	/opt/hinemos/var/log/
ログ出力設定ファイル	/opt/hinemos/bin/jboss_start.sh
出力レベル	―
ローテーション	再起動のたびに上書き
内容	JBoss からの標準出力

ファイル名	jboss.log*
格納ディレクトリ	/opt/hinemos/var/log/
ログ出力設定ファイル	/opt/hinemos/etc/jboss-log4j.xml
出力レベル	priority INFO
ローテーション	Daily(無期限)
内容	JBoss の動作ログ

ファイル名	Postgresql.log*
格納ディレクトリ	/opt/hinemos/var/log/
ログ出力設定ファイル	/opt/hinemos/etc/postgresql.conf
出力レベル	―
ローテーション	Daily(無期限)
内容	PostgreSQL の動作ログ

ファイル名	openldap.log*
格納ディレクトリ	/opt/hinemos/var/log/
ログ出力設定ファイル	/opt/hinemos/etc/slapd.conf
出力レベル	loglevel 64
ローテーション	Daily(無期限)
内容	OpneLDAP の動作ログ

ファイル名	syslogforward.log*
格納ディレクトリ	/opt/hinemos/var/log/
ログ出力設定ファイル	/opt/hinemos/lib/syslogforward/log4j.properties
出力レベル	priority INFO
ローテーション	Daily(無期限)
内容	ログ監視機能で使われるモジュール SyslogForward のログ

5.2. JBoss のログ出力を変更する

Hinemos で利用する JBoss のログ出力のレベルを変更するためには、以下のファイルを編集します。

/opt/hinemos/etc/jboss-log4j.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">

 (中略)


   <category name="hinemos.apllog">
     <priority value="INFO" />
     <appender-ref ref="HINEMOS_APL_LOG"/>
   </category>

   <!-- Hinemos Log category -->
   <category name="com.clustercontrol">
     <priority value="INFO" />
     <appender-ref ref="HINEMOS_LOG"/>
   </category>

 (中略)


   <root>
      <!-- <appender-ref ref="CONSOLE"/> -->
      <appender-ref ref="FILE"/>
   </root>
```

本ファイルでHinemosApl.log、Hinemos.log、jboss.logの出力レベルを変更することが出来ます。Log4jの設定例については、http://docs.jboss.org/process-guide/en/html/logging.html を参照してください。

5.3. PostgreSQL のログ出力を変更する

Hinemos で利用する PostgreSQL のログ出力のレベルを変更するためには、以下のファイルを編集します。設定例については、
http://www.postgresql.jp/document/pg831doc/html/runtime-config-logging.htmlを参照してください。

/opt/hinemos/var/data/postgresql.conf

```
# -----------------------------
# PostgreSQL configuration file
# -----------------------------

(中略)

#------------------------------------------------------------------------------
# ERROR REPORTING AND LOGGING
#------------------------------------------------------------------------------

# - Where to Log -

#log_destination = 'stderr'             # Valid values are combinations of
                                        # stderr, syslog and eventlog,
                                        # depending on platform.

# This is used when logging to stderr:
#redirect_stderr = off                  # Enable capturing of stderr into log
redirect_stderr = on                    # Enable capturing of stderr into log
                                        # files

# These are only used if redirect_stderr is on:
log_directory = '/opt/hinemos/var/log'  # Directory where log files are written
#log_directory = 'pg_log'               # Directory where log files are written
                                        # Can be absolute or relative to PGDATA
log_filename = 'postgresql.log-%Y-%m-%d' # Log file name pattern.
#log_filename = 'postgresql-%Y-%m-%d_%H%M%S.log' # Log file name pattern.
                                        # Can include strftime() escapes
#log_truncate_on_rotation = off # If on, any existing log file of the same
                                        # name as the new log file will be
                                        # truncated rather than appended to. But
                                        # such truncation only occurs on
                                        # time-driven rotation, not on restarts
                                        # or size-driven rotation. Default is
                                        # off, meaning append to existing files
                                        # in all cases.
#log_rotation_age = 1440                # Automatic rotation of logfiles will
                                        # happen after so many minutes.  0 to
                                        # disable.
#log_rotation_size = 10240              # Automatic rotation of logfiles will
                                        # happen after so many kilobytes of log
                                        # output.  0 to disable.

# These are relevant when logging to syslog:
#syslog_facility = 'LOCAL0'
#syslog_ident = 'postgres'


# - When to Log -

#client_min_messages = notice           # Values, in order of decreasing detail:
                                        #   debug5
                                        #   debug4
                                        #   debug3
                                        #   debug2
                                        #   debug1
                                        #   log
                                        #   notice
                                        #   warning
                                        #   error

#log_min_messages = notice              # Values, in order of decreasing detail:
                                        #   debug5
                                        #   debug4
                                        #   debug3
                                        #   debug2
```

5.4. OpenLDAP のログ出力を変更する

Hinemos で利用する OpenLDAP のログ出力のレベルを変更するためには、以下のファイルを編集します。
/opt/hinemos/openldap-2.3.39/etc/openldap/slapd.conf

```
#
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
#
include         /opt/hinemos/openldap-2.3.20/etc/openldap/schema/core.schema
include         /opt/hinemos/openldap-2.3.20/etc/openldap/schema/corba.schema

 （中略）

#loglevel 256
#loglevel 320
loglevel 64

index   ccTreeType eq
```

上記の loglevel を変更します。設定値に関しては以下のコマンドを実行して参照してください。

```
$man /opt/hinemos/openldap-2.3.39/man/man5/slapd.conf.5
```

5.5. SyslogForward のログ出力を変更する

Syslogforward ログの出力レベルを変更するためには以下のファイルを編集します。
/opt/hinemos/lib/syslogforward/log4j.properties

```
### direct messages to file mylog.log ###
log4j.appender.file=org.apache.log4j.DailyRollingFileAppender
log4j.appender.file.File=/opt/hinemos/var/log/syslogforward.log
log4j.appender.file.Append=true
log4j.appender.file.layout=org.apache.log4j.PatternLayout
```

```
log4j.appender.file.layout.ConversionPattern=%d %5p %c{1} - %m%n


log4j.rootLogger=INFO, file
```

設定の詳細については、
http://www.jajakarta.org/log4j/jakarta-log4j-1.1.3/docs-ja/manual.htmlを参照ください。

5.6. エージェントのログファイル一覧

Hinemosエージェントのログは 表 5-2 に示すログファイルに出力されます。

表 5-2 Hinemos エージェントのログファイル一覧

ファイル名	agents.log*
格納ディレクトリ	/opt/hinemos_agent/var/log/
ログ出力設定ファイル	/opt/hinemos_agent/lib/agent/log4j.properties
出力レベル	priority INFO
ローテーション	ファイルサイズ 20MB (4 回)
内容	Hinemos ジョブエージェントのログ

ファイル名	logagent.log*
格納ディレクトリ	/opt/hinemos_agent/var/log/
ログ出力設定ファイル	/opt/hinemos_agent/lib/log_agent/log4j.properties
出力レベル	priority INFO
ローテーション	ファイルサイズ 20MB (4 回)
内容	Hinemos ログ転送エージェントのログ

5.7. ジョブエージェントのログ出力を変更する

ジョブエージェントの出力レベルを変更するには、以下のファイルを編集します。
/opt/hinemos_agent/lib/agent/log4j.properties

```
### direct messages to file mylog.log ###

log4j.appender.file=org.apache.log4j.RollingFileAppender
log4j.appender.file.File=/opt/hinemos_agent/var/log/agents.log
log4j.appender.file.MaxFileSize = 20MB
log4j.appender.file.MaxBackupIndex = 4
log4j.appender.file.Append=true
log4j.appender.file.layout=org.apache.log4j.PatternLayout
log4j.appender.file.layout.ConversionPattern=%d %5p %c{1} - %m%n



log4j.rootLogger=info, file
```

設定の詳細については、
http://www.jajakarta.org/log4j/jakarta-log4j-1.1.3/docs-ja/manual.htmlを参照ください。

5.8. ログ転送エージェントのログ出力を変更する

ログ転送エージェントの出力レベルを変更するには、以下のファイルを編集します。
/opt/hinemos_agent/lib/log_agent/log4j.properties

```
### direct messages to file mylog.log ###
log4j.appender.file=org.apache.log4j.RollingFileAppender
log4j.appender.file.File=/opt/hinemos_agent/var/log/logagent.log
log4j.appender.file.MaxFileSize = 20MB
log4j.appender.file.MaxBackupIndex = 4
log4j.appender.file.Append=true
log4j.appender.file.layout=org.apache.log4j.PatternLayout
log4j.appender.file.layout.ConversionPattern=%d %5p %c{1} - %m%n



### direct messages to file mylog.log ###
log4j.appender.syslog=org.apache.log4j.net.SyslogAppender
#log4j.appender.syslog.Facility=local6
log4j.appender.syslog.Facility=user
```

```
log4j.appender.syslog.SyslogHost=host1
log4j.appender.syslog.FacilityPrinting=false
log4j.appender.syslog.layout=org.apache.log4j.PatternLayout
log4j.appender.syslog.layout.ConversionPattern=%m%n

#log4j.rootLogger=info, file
log4j.logger.hinemos.syslog.transfer=debug, syslog
log4j.logger.com.clustercontrol.logagent=info, file
```

設定の詳細については、
http://www.jajakarta.org/log4j/jakarta-log4j-1.1.3/docs-ja/manual.htmlを参照ください。

注）org.apache.log4j.net.SyslogAppender で定義される出力は、ログ転送エージェントの機能自身であるので、org.apache.log4j.net.SyslogAppender に関連する設定を変更した場合に、ログ転送エージェントが正しく動作しなくなる可能性があります。

5.9. syslog-ng の動作・ログ保存を変更する

監視管理機能では、各ノードからのログを syslog-ng 経由で受け取ります。

マネージャサーバのインストーラを用いて Hinemos をインストールした場合、以下の設定が syslog-ng の設定ファイルに追記されます。

/etc/syslog-ng/syslog-ng.conf

```
#add for Hinemos Manager
source s_net { tcp(ip(0.0.0.0) port(514) max-connections(70)); udp(ip(0.0.0.0) port(514)); };
log { source(s_local); filter(f_mesg); destination(d_hinemos);};
log { source(s_net); filter(f_mesg); destination(d_hinemos);};
destination d_hinemos { program("/opt/hinemos/jre1.5.0_09/bin/java -cp
/opt/hinemos/lib/syslogforward:/opt/hinemos/lib/MonitorEJB.jar:/opt/hinemos/lib/SyslogNGEJB.jar
:/opt/hinemos/lib/commons-logging.jar:/opt/hinemos/lib/log4j.jar:/opt/hinemos/lib/hinemos-commo
ns.jar:/opt/hinemos/lib/RepositoryEJB.jar:/opt/hinemos/lib/CalendarEJB.jar:/opt/hinemos/lib/clu
stercontrol.jar:/opt/hinemos/lib/jbossall-client.jar:/opt/hinemos/lib/syslogforward/syslogforwa
rd.jar: com.clustercontrol.syslogng.forward.LogForward
/opt/hinemos/lib/syslogforward/LogForward.properties" );};
```

初期設定では、管理対象ノードからマネージャへの syslog-ng の接続数は最大 70 となっています。最大接続数を変更するには、下記の部分を編集してください。

source s_net { tcp(ip(0.0.0.0) port(514) max-connections(*最大接続数*)); udp(ip(0.0.0.0) port(514)); };

6. Hinemos マネージャの設定一覧

Hinemos マネージャの設定は、/opt/hienmos/etc/hinemos.properties に記述されています。設定の反映には JBoss の再起動が必要です。

- 共通設定

プロパティ	common.data.source
説明	Hinemos マネージャが使用するデータソースの識別子
データ型	java.lang.String
デフォルト値	java:/HinemosDS
ドメイン	固定値

プロパティ	common.ldap.timeout
説明	LDAP へアクセスする際のタイムアウト設定
データ型	long
デフォルト値	15000
ドメイン	

- リポジトリ管理機能

プロパティ	common.ldap.timekeeper.thread.count
説明	LDAP アクセス用にプールする最大スレッド数
データ型	int
デフォルト値	50
ドメイン	

プロパティ	common.mail.from.address
説明	メール通知の際、送信元（From）に設定されるメールアドレス
データ型	java.lang.String
デフォルト値	admin@hinemos.com
ドメイン	メールアドレスとして有効な文字列

プロパティ	common.mail.from.personal.name
説明	メール通知の際、送信元（From）に設定される名前
データ型	java.lang.String
デフォルト値	Hinemos Admin
ドメイン	任意の文字列

プロパティ	common.mail.reply.to.address
説明	メール通知の際、返信先（Reply-To）に設定されるメールアドレス
データ型	java.lang.String
デフォルト値	admin@hinemos.com
ドメイン	メールアドレスとして有効な文字列

プロパティ	common.mail.reply.personal.name
説明	メール通知の際、返信先（Reply-To）に設定される名前
データ型	java.lang.String
デフォルト値	Hinemos Admin
ドメイン	任意の文字列

プロパティ	common.mail.errors.to.address
説明	メール送信に失敗した際にエラーメールを送信する宛先
データ型	java.lang.String
デフォルト値	admin@hinemos.com
ドメイン	メールアドレスとして有効な文字列

プロパティ	common.mail.retry.count
説明	メール再送回数
データ型	int
デフォルト値	0
ドメイン	

プロパティ	common.repository.snmp.port
説明	Find By SNMP 実行時に SNMP ポーリング対象となるポート番号
データ型	int
デフォルト値	161
ドメイン	変更不要（デバッグ用のため使用不可）

プロパティ	common.repository.snmp.community
説明	Find By SNMP 実行時の SNMP ポーリングに設定されるコミュニティ名
データ型	java.lang.String
デフォルト値	public
ドメイン	変更不要（デバッグ用のため使用不可）

プロパティ	common.repository.snmp.timeout
説明	Find By SNMP 実行時の SNMP ポーリングに設定されるタイムアウト（ミリ秒）
データ型	int
デフォルト値	500
ドメイン	

プロパティ	common.repository.snmp.retry
説明	Find By SNMP 実行時の SNMP ポーリングに設定されるポーリング試行回数
データ型	int
デフォルト値	1
ドメイン	

プロパティ	common.repository.snmp.address
説明	Find By SNMP で情報を取得する対象ノードの IP アドレス
データ型	java.net.InetAddress
デフォルト値	127.0.0.1
ドメイン	変更不要（デバッグ用のため使用不可）

プロパティ	common.repository.snmp.version
説明	Find By SNMP 実行時の SNMP ポーリングに使用される SNMP バージョン
データ型	int
デフォルト値	0
ドメイン	変更不要（デバッグ用のため使用不可）

プロパティ	common.repository.snmp.oid.name
説明	Find By SNMP で「ホスト名」を取得するために使用する OID
データ型	java.lang.String
デフォルト値	.1.3.6.1.2.1.1.5.0
ドメイン	固定値

プロパティ	common.repository.snmp.oid.descr
説明	Find By SNMP で「説明」を取得するために使用する OID
データ型	java.lang.String
デフォルト値	.1.3.6.1.2.1.1.1.0
ドメイン	固定値

プロパティ	common.repository.snmp.oid.contact
説明	Find By SNMP で「連絡先」を取得するために使用する OID
データ型	java.lang.String
デフォルト値	.1.3.6.1.2.1.1.4.0
ドメイン	固定値

プロパティ	common.repository.snmp.oid.dev.name
説明	Find By SNMP で「デバイス名」を取得するために使用する OID
データ型	java.lang.String
デフォルト値	.1.3.6.1.2.1.25.3.2.1.3
ドメイン	固定値

プロパティ	common.repository.snmp.oid.dev.index
説明	Find By SNMP で「デバイス INDEX」を取得するために使用する OID
データ型	java.lang.String
デフォルト値	.1.3.6.1.2.1.25.3.2.1.1
ドメイン	固定値

プロパティ	common.repository.snmp.oid.dev.type
説明	Find By SNMP で「デバイス種別」を取得するために使用する OID
データ型	java.lang.String
デフォルト値	1.3.6.1.2.1.25.3.2.1.2

ドメイン	固定値

プロパティ	common.repository.snmp.oid.disk.index
説明	Find By SNMP でデバイス種別：disk の「デバイス INDEX」を取得するために使用する OID
データ型	java.lang.String
デフォルト値	.1.3.6.1.4.1.2021.13.15.1.1.1
ドメイン	固定値

プロパティ	common.repository.snmp.oid.disk.name
説明	Find By SNMP でデバイス種別：disk の「デバイス名」を取得するために使用する OID
データ型	java.lang.String
デフォルト値	.1.3.6.1.4.1.2021.13.15.1.1.2
ドメイン	固定値

プロパティ	common.repository.snmp.oid.nic.index
説明	Find By SNMP でデバイス種別：nic の「デバイス INDEX」を取得するために使用する OID
データ型	java.lang.String
デフォルト値	.1.3.6.1.2.1.2.2.1.1
ドメイン	固定値

プロパティ	common.repository.snmp.oid.nic.name
説明	Find By SNMP でデバイス種別：nic の「デバイス名」を取得するために使用する OID
データ型	java.lang.String
デフォルト値	.1.3.6.1.2.1.2.2.1.2
ドメイン	固定値

プロパティ	common.repository.snmp.oid.filesystem.index
説明	Find By SNMP で「ファイルシステム INDEX」を取得するために使用する OID
データ型	java.lang.String
デフォルト値	.1.3.6.1.2.1.25.2.3.1.1

ドメイン	固定値

プロパティ	common.repository.snmp.oid.filesystem.type
説明	Find By SNMP で「ファイルシステム種別」を取得するために使用する OID
データ型	java.lang.String
デフォルト値	.1.3.6.1.2.1.25.2.3.1.2
ドメイン	固定値

プロパティ	common.repository.snmp.oid.filesystem.name
説明	Find By SNMP で「ファイルシステム名」を取得するために使用する OID
データ型	java.lang.String
デフォルト値	.1.3.6.1.2.1.25.2.3.1.3
ドメイン	固定値

プロパティ	common.repository.cache.facility.ipaddress
説明	登録されているノード情報のうち、IP アドレスをキャッシュ化するかどうか
データ型	int
デフォルト値	1
ドメイン	0 または 1（0:無効, 1:有効）

プロパティ	common.repository.cache.facility.nodename
説明	登録されているノード情報のうち、ノード名をキャッシュ化するかどうか
データ型	int
デフォルト値	1
ドメイン	0 または 1（0:無効, 1:有効）

プロパティ	common.repository.cache.facilitytree.findall
説明	
データ型	int
デフォルト値	1
ドメイン	0 または 1（0:無効, 1:有効）

プロパティ	common.repository.cache.facilitytree.facilityid
説明	登録されているスコープ情報のうち、ファシリティ ID 情報をキャッシュ化するかどうか
データ型	int
デフォルト値	1
ドメイン	0 または 1（0:無効, 1:有効）

プロパティ	common.repository.cache.device.facilityid
説明	登録されているノード情報のうち、デバイス情報をキャッシュ化するかどうか
データ型	int
デフォルト値	1
ドメイン	0 または 1（0:無効, 1:有効）

プロパティ	common.repository.cache.filesystem.facilityid
説明	登録されているノード情報のうち、ファイルシステム情報をキャッシュ化するかどうか
データ型	int
デフォルト値	1
ドメイン	0 または 1（0:無効, 1:有効）

- 一括制御機能の設定

プロパティ	Collective.run.shell
説明	一括制御機能で使用するシェル
データ型	java.lang.String
デフォルト値	ssh
ドメイン	ssh または rsh

- 監視管理機能の設定

プロパティ	monitor.common.report.event.count

説明	イベントの最大ダウンロード件数
データ型	int
デフォルト値	1000
ドメイン	-1 以上の整数（-1 の場合すべてのイベントをダウンロード）

プロパティ	monitor.common.retry.interval
説明	監視結果の確認間隔（ミリ秒）
データ型	int
デフォルト値	1000
ドメイン	

プロパティ	monitor.common.thread.pool
説明	監視の同時実行スレッドプール数
データ型	int
デフォルト値	50
ドメイン	

プロパティ	monitor.ping.fping.enable
説明	ping 監視にて fping モジュールの使用有無を設定するフラグ
データ型	boolean
デフォルト値	true
ドメイン	true または false

プロパティ	monitor.ping.fping.path
説明	fping モジュールへの絶対パス
データ型	java.util.String
デフォルト値	/opt/hinemos/sbin/fping
ドメイン	ファイルパスとして有効な文字列（半角英数）

プロパティ	monitor.ping.fping6.path
説明	fping6 モジュールへの絶対パス
データ型	java.util.String
デフォルト値	/opt/hinemos/sbin/fping6
ドメイン	ファイルパスとして有効な文字列（半角英数）

プロパティ	monitor.ping.fping.count
説明	fping のデフォルト設定（ping 実行回数）
データ型	int
デフォルト値	1
ドメイン	

プロパティ	monitor.ping.fping.interval
説明	fping のデフォルト設定（ping 実行間隔）
データ型	int
デフォルト値	1000
ドメイン	

プロパティ	monitor.ping.fping.timeout
説明	fping のデフォルト設定（ping タイムアウト）
データ型	int
デフォルト値	1000
ドメイン	

プロパティ	monitor.process.start.second
説明	プロセス監視の値取得開始時間（秒）設定
データ型	int
デフォルト値	30
ドメイン	1-59

プロパティ	monitor.process.valid.second
説明	プロセス監視用 SNMP ポーラーの収集許容時間（秒）設定
データ型	int
デフォルト値	50
ドメイン	

プロパティ	monitor.process.details.display
説明	プロセス監視にて、条件にマッチしたプロセスの詳細を、オリジナルメッセージに表示させるかどうかのフラグ
データ型	boolean
デフォルト値	false

ドメイン	true または false

プロパティ	monitor.snmp.valid.second
説明	SNMP 監視用 SNMP ポーラーの収集許容時間（秒）設定
データ型	int
デフォルト値	10
ドメイン	

- ジョブ管理機能の設定

プロパティ	job.message.retry
説明	ジョブ実行メッセージ送信リトライ回数
データ型	int
デフォルト値	10
ドメイン	

プロパティ	job.message.timeout
説明	ジョブ実行メッセージ送信タイムアウト（秒）
データ型	int
デフォルト値	60
ドメイン	

プロパティ	job.open.forward.file.job
説明	ファイル転送ジョブ表示方法設定
データ型	boolean
デフォルト値	false
ドメイン	true または false

7. Hinemos エージェントの設定一覧

7.1. ジョブエージェントの設定一覧

ジョブエージェントの設定は、/opt/hienmos_agent/lib/agent/Agent.properties に記述されています。設定の反映にはジョブエージェントの再起動が必要です。

プロパティ	java.naming.factory.initial
説明	初期コンテキスト
データ型	java.lang.String
デフォルト値	org.jnp.interfaces.NamingContextFactory
ドメイン	固定値

プロパティ	user.name
説明	Hinemos マネージャにログインするためのユーザ名
データ型	java.lang.String
デフォルト値	HINEMOS_AGENT
ドメイン	固定値

プロパティ	user.password
説明	Hinemos マネージャにログインするためのパスワード
データ型	java.lang.String
デフォルト値	HINEMOS_AGENT
ドメイン	固定値

プロパティ	queue.user.name
説明	メッセージ Queue に接続するためのユーザ名
データ型	java.lang.String
デフォルト値	guest
ドメイン	固定値

プロパティ	queue.user.password
説明	メッセージ Queue に接続するためのパスワード
データ型	java.lang.String
デフォルト値	guest
ドメイン	固定値

プロパティ	topic.user.name
説明	メッセージ Topic に接続するためのユーザ名
データ型	java.lang.String
デフォルト値	guest

ドメイン	固定値

プロパティ	topic.user.password
説明	メッセージ Topic に接続するためのパスワード
データ型	java.lang.String
デフォルト値	guest
ドメイン	固定値

プロパティ	facility.update.interval
説明	ファシリティ ID の再取得間隔（ミリ秒）
データ型	long
デフォルト値	300000
ドメイン	

プロパティ	end.message.resend.interval
説明	終了メッセージ再送間隔（秒）
データ型	int
デフォルト値	10
ドメイン	

プロパティ	sendqueue.reconnection.interval
説明	メッセージ送信再接続処理間隔（秒）
データ型	int
デフォルト値	10
ドメイン	

プロパティ	receivetopic.reconnection.interval
説明	メッセージ受信再接続処理間隔（秒）
データ型	int
デフォルト値	30
ドメイン	

プロパティ	runhistory.clear.delay
説明	ジョブ履歴削除実行時間（秒）
データ型	int

デフォルト値	1
ドメイン	

プロパティ	limit.size.orgmsg
説明	オリジナルメッセージのサイズ上限
データ型	int
デフォルト値	8192
ドメイン	

プロパティ	input.encoding
説明	ジョブ実行結果メッセージ（実行したプロセスの標準出力、標準入力）の文字コード指定
データ型	java.lang.String
デフォルト値	MS932
ドメイン	Java がサポートする文字エンコーディングの識別子

プロパティ	root.public.key
説明	ファイル転送ジョブで使用する、scp(ssh)公開鍵
データ型	java.lang.String
デフォルト値	
ドメイン	

プロパティ	root.authorized.keys.path
説明	ファイル転送ジョブで使用する、root ユーザの authorized_keys ファイルへのパス
データ型	java.lang.String
デフォルト値	/root/.ssh/authorized_keys
ドメイン	

プロパティ	java.naming.provider.url
説明	接続先となる Hinemos マネージャ
データ型	java.lang.String
デフォルト値	jnp://【エージェントインストール時に指定した IP アドレス】:1099
ドメイン	エージェントインストール時に自動的に設定される

7.2. ログ転送エージェントの設定一覧

ログ転送エージェントの設定は、/opt/hienmos_agent/lib/log_agent/Agent.properties に記述されています。設定の反映にはログ転送エージェントの再起動が必要です。

プロパティ	java.naming.factory.initial
説明	初期コンテキスト
データ型	java.lang.String
デフォルト値	org.jnp.interfaces.NamingContextFactory
ドメイン	固定値

プロパティ	user.name
説明	Hinemos マネージャにログインするためのユーザ名
データ型	java.lang.String
デフォルト値	HINEMOS_AGENT
ドメイン	固定値

プロパティ	user.password
説明	Hinemos マネージャにログインするためのパスワード
データ型	java.lang.String
デフォルト値	HINEMOS_AGENT
ドメイン	固定値

プロパティ	queue.user.name
説明	メッセージ Queue に接続するためのユーザ名
データ型	java.lang.String
デフォルト値	guest
ドメイン	固定値

プロパティ	queue.user.password
説明	メッセージ Queue に接続するためのパスワード
データ型	java.lang.String
デフォルト値	guest

ドメイン	固定値

プロパティ	topic.user.name
説明	メッセージ Topic に接続するためのユーザ名
データ型	java.lang.String
デフォルト値	guest
ドメイン	固定値

プロパティ	topic.user.password
説明	メッセージ Topic に接続するためのパスワード
データ型	java.lang.String
デフォルト値	guest
ドメイン	固定値

プロパティ	manager.polling.interval
説明	マネージャへのポーリング周期（秒）
データ型	int
デフォルト値	300
ドメイン	

プロパティ	sendqueue.reconnection.interval
説明	メッセージ送信再接続処理間隔（秒）
データ型	int
デフォルト値	10
ドメイン	

プロパティ	receivetopic.reconnection.interval
説明	メッセージ受信再接続処理間隔（秒）
データ型	int
デフォルト値	30
ドメイン	

プロパティ	unchanged.stats.period
説明	ファイル変更チェック期間設定（秒）。ファイルサイズが n 秒間変わらなかったら、ファイルの切り替わりをチェックする。

データ型	int
デフォルト値	5
ドメイン	

プロパティ	file.max.size
説明	ファイルサイズ設定（byte）
データ型	long
デフォルト値	2147483648
ドメイン	

プロパティ	syslog.message.priority
説明	syslog 転送の重要度（log4j のログレベルで指定）
データ型	java.lang.String
デフォルト値	info
ドメイン	

プロパティ	log.file.encoding
説明	ログファイルの文字コードを指定
データ型	java.lang.String
デフォルト値	UTF-8
ドメイン	Java がサポートする文字エンコーディングの識別子

プロパティ	java.naming.provider.url
説明	接続先となる Hinemos マネージャ
データ型	java.lang.String
デフォルト値	jnp://【エージェントインストール時に指定した IP アドレス】:1099
ドメイン	エージェントインストール時に自動的に設定される