

Hinemos® ver.2.4

設定リファレンス

第 1.0 版

2008 年 3 月 31 日

株式会社 NTTデータ

設定リファレンス

変更履歴

版	変更日	変更内容
1.0	2008/3/31	Hinemos Ver2.4.0 向け初版

目次

1.	はじめに.....	6
2.	前提条件.....	6
2.1.	Hinemos のバックアップ方法.....	6
3.	Hinemos の機能セットアップに関する追加設定.....	7
3.1.	Hinemos マネージャをサービス化する.....	7
3.1.1.	JBoss, OpenLDAP, PostgreSQL のサービス化.....	7
3.1.2.	サービスの起動.....	7
3.1.3.	サービスの停止.....	8
3.1.4.	自動起動の設定.....	8
3.2.	Hinemos エージェントをサービス化する.....	8
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.....	8
3.2.2.	サービスの起動.....	9
3.2.3.	サービスの停止.....	9
3.2.4.	自動起動の設定.....	9
3.3.	性能管理機能、監視管理機能を有効にする.....	10
3.3.1.	Net-SNMP の設定.....	10
3.4.	SNMP ポーラの設定.....	10
3.4.1.	性能管理機能、監視管理機能（リソース監視）の設定変更.....	10
3.4.2.	監視管理機能（プロセス監視）の設定変更.....	12
3.5.	一括制御機能を有効にする.....	13
3.5.1.	エージェント側のセットアップ.....	13
3.5.2.	マネージャでのリモートシェルの設定.....	16
3.5.3.	FTP サーバの起動.....	17
3.6.	SNMP トラップ監視機能を有効にする.....	18
3.6.1.	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.....	18
3.6.2.	JBoss を root ユーザで起動.....	18
3.7.	メール通知を有効にする.....	19
3.7.1.	メールサーバの設定.....	19
3.7.2.	送信元情報の設定.....	20
3.8.	ファイル転送ジョブを有効にする.....	22
3.9.	ログ転送エージェントを有効にする.....	25
3.9.1.	エージェントの syslog-ng の設定.....	25
4.	セキュリティに関する設定.....	27
4.1.	データベースアクセスのパスワードを変更する.....	27

設定リファレンス

4.1.1.	PostgreSQL の設定変更.....	27
4.1.2.	Hinemos マネージャの設定変更.....	29
4.2.	LDAP アクセスのパスワードを変更する.....	31
4.2.1.	LDAP のパスワード変更.....	31
4.2.2.	Hinemos マネージャの設定変更.....	33
5.	Hinemos の動作ログに関する設定.....	34
5.1.	マネージャのログファイル一覧	34
5.2.	JBoss のログ出力を変更する	35
5.3.	PostgreSQL のログ出力を変更する	36
5.4.	OpenLDAP のログ出力を変更する	38
5.5.	SyslogForward のログ出力を変更する	38
5.6.	エージェントのログファイル一覧.....	39
5.7.	ジョブエージェントのログ出力を変更する	39
5.8.	ログ転送エージェントのログ出力を変更する.....	40
5.9.	syslog-ng の動作・ログ保存を変更する	41

設定リファレンス

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、®マークは表記していません。

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. JBoss, OpenLDAP, PostgreSQL のサービス化

Hinemos マネージャの JBoss, OpenLDAP, PostgreSQL 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-2.4.0_rhel4_32/hinemos/sbin/service ディレクトリ)

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_Manager-2.4.0_rhel4_32/hinemos/sbin/service/  
# cp hinemos_jboss /etc/init.d/  
# cp hinemos_ldap /etc/init.d/  
# cp hinemos_pg /etc/init.d/
```

3.1.2. サービスの起動

Hinemos マネージャのサービスとして登録した PostgreSQL, OpenLDAP, JBoss を順番に起動します (JBoss は最後に起動する必要があります)。

root ユーザで以下のコマンドを実行してください。

```
# service hinemos_pg start  
Starting PostgreSQL: ok  
# service hinemos_ldap start  
Starting OpenLDAP: ok  
# service hinemos_jboss start  
hinemos_jboss を起動中:
```

JBoss の起動には暫く時間がかかります。

3.1.3. サービスの停止

JBoss, OpenLDAP, PostgreSQL を順番に停止させます (JBoss を最初に停止してください)。

root ユーザで以下のコマンドを実行してください。

```
# service hinemos_jboss stop
hinemos_jboss を停止中: /opt/hinemos/jboss-4.0.3SP1
Shutdown message has been posted to the server.
Server shutdown may take a while - check logfiles for completion

# service hinemos_ldap stop
Stopping OpenLDAP: ok
# service hinemos_pg stop
Stopping PostgreSQL: postmaster stopped
ok
```

3.1.4. 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_jboss
# chkconfig --add hinemos_ldap
# chkconfig --add hinemos_pg
```

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-2.4.0_rhel4_32/service ディレクトリ)に含まれています。

設定リファレンス

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-2.4.0_rhel4_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。

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 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
```

3.4. SNMP ポーラの設定

Hinemos では、性能管理機能、監視管理機能（リソース監視、プロセス監視）で SNMP ポーリングの結果をもとに値の取得を行なっています。

以下のファイルを編集することで SNMP ポーリングのパラメータを変更することができます。

3.4.1. 性能管理機能、監視管理機能（リソース監視）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は Hinemos マネージャを再起動する必要があります。

```
/opt/hinemos/jboss-4.0.3SP1/server/default/deploy/snmpolling-perf-service.xml
```

設定リファレンス

```
<?xml version="1.0" encoding="UTF-8"?>
<server>
  <classpath codebase="." archives="SnmpSharedTable.jar"/>
  <mbean code="com.clustercontrol.snmpoller.jmx.SnmpPollerService"
        name="user:service=SnmpPollerService,name=SnmpPerf">
    ( 中略 )
    <attribute name="DefaultPort">-1</attribute>
    <attribute name="DefaultVersion">1</attribute>
    <attribute name="DefaultCommunity">public</attribute>
    <attribute name="DefaultRetries">-1</attribute>
    <attribute name="DefaultTimeout">1000</attribute>
  ( 省略 )

```

下記の箇所を表 3-1 SNMP ポーリングのパラメータを参考に編集してください。

```
<attribute name="パラメータ名">値</attribute>
```

表 3-1 SNMP ポーリングのパラメータ

パラメータ名	内容
DefaultPort	ポーリングを実行する TCP ポート (-1:デフォルト (161))
DefaultVersion	SNMP のバージョン (0:SNMPv1 1:SNMPv2)
DefaultCommunity	コミュニティ名
DefaultRetries	リトライ回数 (-1:デフォルト (3 回))
DefaultTimeout	タイムアウト (ミリ秒)

3.4.2. 監視管理機能 (プロセス監視) 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は Hinemos マネージャを再起動する必要があります。

/opt/hinemos/jboss-4.0.3SP1/server/default/deploy/snmpolling-proc-service.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<server>
  <classpath codebase="." archives="SnmpSharedTable.jar"/>
  <mbean code="com.clustercontrol.snmpoller.jmx.SnmpPollerService"
    name="user:service=SnmpPollerService,name=SnmpProc">
    (中略)
    <attribute name="DefaultPort">-1</attribute>
    <attribute name="DefaultVersion">1</attribute>
    <attribute name="DefaultCommunity">public</attribute>
    <attribute name="DefaultRetries">-1</attribute>
    <attribute name="DefaultTimeout">1000</attribute>
    (省略)
```

下記の箇所を表 3-1 SNMP ポーリングのパラメータを参考に編集してください。

```
<attribute name="パラメータ名">値</attribute>
```

3.5. 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にはセットアップを行う必要があります。

3.5.1. 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。

```
# 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 2.4.0 ###
###                                     ###
### Copyright (C) 2006-2008 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
```


設定リファレンス

6. 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:
```

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。

7. 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
```

3.5.2. 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 マネージャの再起動をしてください。

/opt/hinemos/jboss-4.0.3SP1/server/default/conf/collectiverun.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
```

3.5.3. 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 EL4 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)"
```

3.6. 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、SNMP トラップ監視機能が無効の状態となっています。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、デフォルトでは一般ユーザ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。

3.6.1. 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-4.0.3SP1/server/default/deploy/
```

3.6.2. JBoss を root ユーザで起動

PostgreSQL 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。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
```

設定リファレンス

```
postmaster started
$ ./ldap_start.sh
$ su
Password:
# ./jboss_start.sh
```

停止の方法は、3.1.3 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。

```
# chown R hinemos:hinemos /opt/hinemos/
```

3.7. メール通知を有効にする

3.7.1. 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 マネージャの再起動をしてください。

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

```

3.7.2. 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/jboss-4.0.3SP1/server/default/conf/mail.properties
```

設定リファレンス

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

以下のパラメータを設定してください。

```
from.address=送信元メールアドレス
from.personal.name=送信先個人名
reply.to.address=返信先メールアドレス
reply.personal.name=返信先個人名
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
```

3.8. 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。また、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:
```


設定リファレンス

```
[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)

3.9. ログ転送エージェントを有効にする

3.9.1. 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 2.4.0
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ため、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。

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。

1. 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。

```
$ su - hinemos
$ /opt/hinemos/postgresql-8.1.3/bin/psql -p 24001
Password:
Welcome to psql 8.1.3, the PostgreSQL interactive terminal.
```

2. 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)';
```

3. 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
```

設定リファレンス

“# 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
- 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/jboss-4.0.3SP1/server/default/deploy/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 loubiansky 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. quartz-service.xml の編集

以下のファイルを編集します。

/opt/hinemos/jboss-4.0.3SP1/server/default/deploy/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/openldap-2.3.20/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}X03M01qnZdYdgyfeuILPmQ==   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/openldap-2.3.20/etc/openldap/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/openldap-2.3.20/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 )

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}X03M01qnZdYdgyfeuILPmQ==
# 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/jboss-4.0.3SP1/server/default/deploy/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/jboss-4.0.3SP1/server/default/conf/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/jboss-4.0.3SP1/server/default/conf/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-4.0.3SP1/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

設定リファレンス

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/jboss-4.0.3SP1/server/default/conf/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/var/data/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/openldap-2.3.20/etc/openldap/slapd.conf
出力レベル	loglevel 64
ローテーション	Daily(無期限)
内容	OpenLDAP 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/jboss-4.0.3SP1/server/default/conf/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/pg813doc/html/runtime-config-logging.html> を参照してください。

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

設定リファレンス

```
# -----
# 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-2.3.20/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-2.3.20/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
```

設定リファレンス

```
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
```

設定リファレンス

```
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));};
```