

Hinemos



**Hinemos ver3.2**  
**設定リファレンス 第1.0版**

# 目次

<b>1</b>	<b>はじめに</b>	<b>5</b>
<b>2</b>	<b>前提条件</b>	<b>6</b>
<b>3</b>	<b>Hinemosのメンテナンス</b>	<b>7</b>
3.1	Hinemosマネージャのメンテナンス	7
3.1.1	概要	7
3.1.2	メンテナンススクリプト一覧	7
3.1.3	データベースから不要なデータを削除する	7
3.1.4	データベースの不要領域をシステムが再利用可能にする	9
3.1.5	データベースのバックアップ	9
3.1.6	データベースのリストア	10
3.1.7	Hinemos内部データベースの一時テーブル内の情報を削除する	10
3.1.8	通知抑制情報の消去	11
3.2	ログの削除	11
3.2.1	Hinemosマネージャのログ	11
3.2.2	Hinemosエージェントのログ	11
3.2.3	Hinemosクライアントのログ	12
3.3	Hinemosマネージャのバックアップ・リカバリ	12
3.3.1	Hinemos マネージャのバックアップ	12
3.3.2	Hinemos マネージャのリカバリ	12
3.4	エージェントのバックアップ・リカバリ	13
3.4.1	Linux版エージェントのバックアップ	13
3.4.2	Linux版エージェントのリカバリ	13
3.4.3	Windows版エージェントのバックアップ	13
3.4.4	Windows版エージェントのリカバリ	14
3.5	Hinemosクライアントのバックアップ・リカバリ	14
<b>4</b>	<b>Hinemosの機能セットアップ(OS/環境)</b>	<b>15</b>
4.1	ファイルディスクリプタの設定	15
4.2	内部DBへの接続(データソース)設定	15
4.3	OSロケールと文字コード設定	15
4.3.1	Hinemos クライアント	15
4.3.2	Hinemos マネージャ	16
4.3.3	Hinemos エージェント(Linux版エージェント)	16
4.3.4	Hinemos エージェント(Windows版エージェント)	16
<b>5</b>	<b>Hinemosの機能セットアップ(共通機能)</b>	<b>18</b>
5.1	イベントの最大ダウンロード件数の設定	18
5.2	INTERNALイベントの出力先変更	18
5.3	Hinemos マネージャ死活検知	19
5.4	メール通知を有効にする	20
5.4.1	メールサーバの設定	20
5.4.2	送信元情報の設定	21
5.4.3	SMTP AUTHの設定	21
5.4.4	エンベロープFromの設定	22
5.5	ログエスカレーション通知の設定	23
5.5.1	埋め込みホスト名の設定方法	23

5.6	セルフチェック機能	23
5.6.1	通知される情報の一覧	24
5.6.2	セルフチェック機能の設定方法	26
<b>6</b>	<b>Hinemosの機能セットアップ(一括制御機能)</b>	<b>28</b>
6.1	一括制御機能を有効にする	28
6.1.1	エージェント側のセットアップ	28
6.1.2	マネージャでのリモートシエルの設定	30
6.1.3	FTPサーバの起動	31
<b>7</b>	<b>Hinemosの機能セットアップ(監視管理/性能管理機能)</b>	<b>32</b>
7.1	デバイス単位でのリソース情報を監視するための設定	32
7.2	SQL監視対象のRDBMS追加	32
7.3	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)	33
7.4	HTTPS監視の設定	34
7.4.1	証明書の準備	34
7.4.2	証明書のkeystoreへの登録	34
7.4.3	Java起動オプションによるkeystoreファイルの指定	34
7.5	syslog-ng監視の設定	35
7.5.1	エスケープ処理の無効化	35
7.5.2	ホスト名置換処理の設定	35
7.6	性能管理機能、監視管理機能を有効にする	36
7.6.1	Net-SNMPの設定	36
7.6.2	SNMP・WBEMの切り替え方法	37
<b>8</b>	<b>Hinemosの機能セットアップ(ジョブ管理機能)</b>	<b>38</b>
8.1	起動コマンドの動作変更	38
8.2	JBoss再起動時のジョブスケジューリング制御の設定	38
8.3	ファイル転送ジョブを有効にする	39
<b>9</b>	<b>セキュリティに関する設定</b>	<b>41</b>
9.1	データベースアクセスのパスワードを変更する	41
9.1.1	PostgreSQLの設定変更	41
9.1.2	Hinemosマネージャの設定変更	41
<b>10</b>	<b>Hinemosの動作ログに関する設定</b>	<b>45</b>
10.1	マネージャのログファイル一覧	45
10.2	JBossのログ出力を変更する	46
10.3	PostgreSQLのログ出力を変更する	46
10.4	SyslogForwardのログ出力を変更する	48
10.5	エージェントのログファイル一覧	49
10.6	ジョブエージェントのログ出力を変更する	50
10.7	ログ転送エージェントのログ出力を変更する	50
10.8	syslog-ngの動作・ログ保存を変更する	51
<b>11</b>	<b>Hinemosマネージャの設定一覧</b>	<b>53</b>
<b>12</b>	<b>Hinemosエージェントの設定一覧</b>	<b>56</b>
12.1	ジョブエージェントの設定一覧	56
12.2	ログ転送エージェントの設定一覧	58
<b>13</b>	<b>変更履歴</b>	<b>60</b>

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>

## 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必要があります。

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。

## 3 Hinemosのメンテナンス

### 3.1 Hinemosマネージャのメンテナンス

#### 3.1.1 概要

Hinemosを長期間利用するために、運用条件に合わせて定期的に以下の作業を行なって下さい。これらの処理を実施するためのスクリプトが提供されています。メンテナンススクリプトは、/opt/hinemos/sbin/mngディレクトリに配置されます。

- データベースから不要なデータを削除する

```
hinemos_delete.sh
```

- データベースの不要領域をシステムが再利用可能にする

```
hinemos_cluster_db.sh
```

- データベースのバックアップを行なう

```
hinemos_backup.sh
```

- データベースのリストアを行なう

```
hinemos_restore.sh
```

- 一時キューの消去を行う

```
hinemos_clear_tmp.sh
```

- 通知抑制情報の消去を行う

```
hinemos_clear_notify.sh
```

以降、各作業の手順を示します。

#### 3.1.2 メンテナンススクリプト一覧

メンテナンススクリプトの実行条件は以下の表に記載の通りとなります。

表 3-1 メンテナンススクリプト一覧

処理内容	スクリプト名	マネージャ起動中	JBoss停止時 (PostgreSQLは起動)
データベースから不要なデータを削除	hinemos_delete.sh	X	○
データベースの再構成	hinemos_cluster_db.sh	X	○
データベースのバックアップ	hinemos_backup.sh	○	○
データベースのリストア	hinemos_restore.sh	X	○
一時キュー (テンポラリ情報) の消去	hinemos_clear_tmp.sh	X	○
通知抑制情報の消去	hinemos_clear_notify.sh	○	○

#### 3.1.3 データベースから不要なデータを削除する

Hinemosマネージャのデータベースには、以下のデータが蓄積されています。

- イベント
- ジョブ実行履歴
- 一括制御実行履歴
- 性能管理機能実績収集データ

これらのデータは、メンテナンス機能や性能管理機能の設定により定期的にデータを削除することが可能です（設定方法の詳細は、ユーザマニュアルを参照下さい）。hinemos\_delete.shコマンドを用いると、これらのデータをコマンド実行により手動で削除することができます。

1. Hinemosマネージャをインストールしたサーバに、rootアカウントでログインします。
2. JBossを停止します。

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

3. メンテナンススクリプトがインストールされているディレクトリに移動します。

```
# cd /opt/hinemos/sbin/mng
```

4. 削除対象のデータに応じたオプションを指定してスクリプトを実行します。

本スクリプトは、-wオプションでデータベース（PostgreSQL）のパスワードを指定することができます。指定しない場合は、パスワード入力求められます。

- 確認済みの監視ログ情報(イベント)を削除する場合

```
# ./hinemos_delete.sh -e
```

- 監視ログ情報(イベント)を全て削除する場合

```
# ./hinemos_delete.sh -E
```

- 終了したジョブの履歴情報を削除する場合

```
# ./hinemos_delete.sh -j
```

- ジョブの実行履歴情報を全て削除する場合

```
# ./hinemos_delete.sh -J
```

- 終了した一括制御実行履歴情報を削除する場合

```
# ./hinemos_delete.sh -c
```

- 一括制御実行履歴情報を全て削除する場合

```
# ./hinemos_delete.sh -C
```

- 性能管理機能実績収集データを収集ID指定で削除する場合

```
# ./hinemos_delete.sh -p 収集ID
```

- 性能管理機能実績収集データを全て削除する場合

```
# ./hinemos_delete.sh -P
```

- 性能管理機能実績収集データを直近の指定期間分残した状態で、収集IDを指定して削除する場合

```
# ./hinemos_delete.sh -p 収集ID -r 保存期間（日）
```



- ・ 性能管理機能実績収集データを直近の指定期間残した状態で、全ての収集IDを対象に削除する場合

```
# ./hinemos_delete.sh -P -r 保存期間 (日)
```

### 3.1.4 データベースの不要領域をシステムが再利用可能にする

3.1.3 データベースから不要なデータを削除する の手順で不要なデータを削除しても、一度データベースに割り当てられたディスク領域は開放されません。システムから再利用可能にするために、hinemos\_cluster\_db.shコマンドでデータベースの再構築を実施する必要があります。

※ データベースの不要領域開放は、定期的実施するようにして下さい。6ヶ月に一度は実行することを推奨します。

パフォーマンスが低下した場合などは、必要に応じて実行してください。

※ 本処理を実行する際、/opt/hinemos/var/dataディレクトリ配下に、テーブルの一時コピーが作成されます。

同様に、テーブルの各インデックスの一時コピーも作成されます。したがって、/opt/hinemos/var/dataディレクトリを含むディスクには、スクリプト実行前の/opt/hinemos/var/data配下の合計サイズと同等の空き領域が必要です。

1. Hinemosマネージャをインストールしたサーバに、rootアカウントでログインします。
2. JBossを停止します。

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

3. メンテナンススクリプトがインストールされているディレクトリに移動します。

```
$ cd /opt/hinemos/sbin/mng
```

4. メンテナンス用のスクリプトを実行します。

```
$ ./hinemos_cluster_db.sh
```

5. JBossを起動します。

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

### 3.1.5 データベースのバックアップ

Hinemosマネージャサーバのノード障害や、ユーザ操作ミスによる設定情報の削除等に備え、定期的にデータベースのバックアップを取得することを推奨します。データベースのバックアップを取得するには、以下のコマンドを実行してください。

1. Hinemosマネージャをインストールしたサーバに、rootアカウントでログインします。
2. ダンプファイルを配置するディレクトリ(ここでは/tmpとします)に移動します。

```
# cd /tmp
```

3. メンテナンス用のスクリプトを実行します。

```
# /opt/hinemos/sbin/mng/hinemos_backup.sh
```

本スクリプトは、-wオプションでデータベース (PostgreSQL) のパスワードを指定することができます。指定しない場合は、パスワード入力が求められます。

4. ダンプが出力されたファイルを確認します。

```
# ls
```

```
hinemos_pgdump.YYYY-MM-DD_HH:mm:ss (PostgreSQLのダンプファイル)
```

※ データベースのバックアップはHinemosマネージャがオンラインの状態で行うことが可能ですが、その時点での状態を保持することになります。ジョブ実行中にダンプを取得すると、 リストア時に該当ジョブのステータスが実行中のデータが復元されますのでご注意ください。

### 3.1.6 データベースのリストア

取得済みのダンプファイルからデータベースのリストアを行なう場合は、以下のコマンドを実行してください。

1. Hinemosマネージャをインストールしたサーバに、rootアカウントでログインします。
2. JBossを停止します。

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

3. ダンプファイルからデータベースを再作成します。

```
$ //opt/hinemos/sbin/mng/hinemos_restore PostgreSQLのダンプファイル名
```

本スクリプトは、-wオプションでデータベース（PostgreSQL）のパスワードを指定することができます。指定しない場合は、パスワード入力求められます。

4. JBossを起動します。

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

### 3.1.7 Hinemos内部データベースの一時テーブル内の情報を削除する

一時テーブル内には、以下の情報が格納されます。

- 通知情報
- ジョブセッション状態遷移中の情報

JBossの強制終了等により、一時テーブルに不要な情報が残り、異常な状態となる場合があります。

hinemos\_clear\_tmp.shコマンドを用いると、これら不要な情報を削除することができます。

※ 上記スクリプトを実行すると、一時テーブル内のキューが全て削除されてしまうため、JBoss停止直前の以下の情報が削除されます。

- 通知処理中の情報（イベント・ステータス）
- ジョブセッションの状態遷移中の情報

一時テーブル内の情報を削除するには、以下のコマンドを実行してください。

1. Hinemosマネージャをインストールしたサーバに、rootアカウントでログインします。
2. JBossを停止します。

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

3. メンテナンススクリプトがインストールされているディレクトリに移動します。

```
$ cd /opt/hinemos/sbin/mng
```

4. 一時テーブル内の不要情報を削除します。

```
$ ./hinemos_clear_tmp.sh
```

本スクリプトは、-wオプションでデータベース（PostgreSQL）のパスワードを指定することができます。指定しない場合は、パスワード入力求められます。

5. JBossを起動します。

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

### 3.1.8 通知抑制情報の消去

通知機能には抑制の機能があります（機能の詳細はユーザマニュアルを参照ください）。抑制状態をクリアするには、以下のコマンドを実行してください。

※ 全監視設定の通知抑制情報がクリアされます。

1. Hinemosマネージャをインストールしたサーバに、rootアカウントでログインします。
2. メンテナンススクリプトがインストールされているディレクトリに移動します。

```
$ cd /opt/hinemos/sbin/mng
```

3. 通知抑制情報をクリアします。

```
$ ./hinemos_clear_notify.sh
```

本スクリプトは、-wオプションでデータベース（PostgreSQL）のパスワードを指定することができます。指定しない場合は、パスワード入力が求められます。

## 3.2 ログの削除

Hinemosマネージャ/エージェント/クライアントがそれぞれ出力するログを確認し、サイズが大きくなっている場合は必要に応じて削除してください。

### 3.2.1 Hinemosマネージャのログ

Hinemosマネージャでは、以下のディレクトリにログが出力されます。

表 3-2 Hinemosマネージャのログ

パス	内容
/opt/hinemos/var/log	Hinemosマネージャの各ログ

### 3.2.2 Hinemosエージェントのログ

Linux版エージェントでは、以下のディレクトリにログが出力されます。

表 3-3 Linux版エージェントのログ

パス	内容
/opt/hinemos_agent/var/log	Hinemosジョブ/ログ転送エージェントのログ

Windows版エージェントでは、以下のディレクトリにログが出力されます。

表 3-4 Windows版エージェントのログ

パス	内容
<Hinemosインストールフォルダ>\Hinemos\Agent3.2.0\var\log	Hinemosジョブ/ログ転送エージェントのログ

### 3.2.3 Hinemosクライアントのログ

Hinemosクライアントでは、以下のディレクトリにログが出力されます。

表 3-5 Hinemosクライアントのログ

OS	パス	内容
Windows XP, Windows Server 2008	<Hinemosインストールフォルダ>\Hinemos\Client3.2.0\workspace\.metadata	Hinemosクライアントのログ
Windows Vista, 7	C:\Users\【ユーザ名】\AppData\Local\VirtualStore\ProgramFiles\Hinemos\Client3.2.0\workspace\.metadata	Hinemosクライアントのログ

## 3.3 Hinemosマネージャのバックアップ・リカバリ

Hinemos マネージャのバックアップ・リカバリ方法を示します。

### 3.3.1 Hinemos マネージャのバックアップ

Hinemos マネージャは、内部でデータベースを使用します。具体的には、以下のプロダクトを使用したデータベースが存在します。

- PostgreSQL

Hinemos マネージャのバックアップには以下のものを取得します。

- データベースのバックアップ
- 設定等の各種ファイルのバックアップ

hinemos\_backup.sh スクリプトを用いると、データベースのバックアップ(ダンプファイルの生成)が行なえます。詳細は「3.1.5 データベースのバックアップ」を参照ください。

Hinemos マネージャは /opt/hinemos 配下にインストールされますが、それ以外にOS/他アプリケーションと共有するファイル類とも関連しています。Hinemos マネージャのバックアップ用途として重要なファイル一覧を示します。本一覧以外に修正を行うファイルが有る場合には各自でバックアップを取得して下さい。

◇Hinemosマネージャ設定ファイル

- /opt/hinemos/hinemos.cfg
- /opt/hinemos/etc 配下の全ファイル/ディレクトリ
- /opt/hinemos 配下で個別に修正したファイル

◇サービス化スクリプト(サービス化の場合)

- /etc/init.d/hinemos\_manager

◇OS/他アプリケーションと共有するファイル

- /etc/syslog-ng/syslog-ng.conf
- /etc/logrotate.d/syslog

### 3.3.2 Hinemos マネージャのリカバリ

Hinemos マネージャのリカバリ手順は以下の通りです。

#### 1. Hinemos マネージャの再インストール

Hinemos マネージャを新規にインストール、または一度アンインストールを行い再度インストールの何れかを行います。Hinemos マネージャのインストール・アンインストール方法についてはインストールマニュアルを参照して下さい。

## 2. データベースのバックアップデータのリストア

データベース (PostgreSQL) のバックアップデータをリストアします。詳細は「3.1.5 データベースのバックアップ」を参照ください。

## 3. 設定等の各種ファイルのバックアップの適用

ユーザが変更した設定等の各種ファイルのバックアップを適用します。ファイルの上書きか変更部分の適用の何れかは変更内容とリカバリ時の環境条件によりますので、各々の変更内容に合わせて実施して下さい。

# 3.4 エージェントのバックアップ・リカバリ

エージェントのバックアップ・リカバリ方法を示します。

## 3.4.1 Linux版エージェントのバックアップ

Hinemos エージェントは、特にデータベースを使用しません。そのため設定等の各種ファイルのバックアップのみが必要になります。

◇Hinemosエージェント設定ファイル

- /opt/hinemos\_agent/lib/job\_agent/Agent.properties
- /opt/hinemos\_agent/lib/job\_agent/log4j.properties
- /opt/hinemos\_agent/lib/log\_agent/Agent.properties
- /opt/hinemos\_agent/lib/log\_agent/log4j.properties

◇サービス化スクリプト(サービス化の場合)

- /etc/init.d/hinemos\_agent
- /etc/init.d/hinemos\_log\_agent

◇OS/他アプリケーションと共有するファイル

- /etc/syslog-ng/syslog-ng.conf
- /etc/snmp/snmpd.conf
- /etc/logrotate.d/syslog

## 3.4.2 Linux版エージェントのリカバリ

Linux版エージェントのリカバリ手順は以下の通りです。

1. Hinemos エージェントの再インストール
2. 設定等の各種ファイルのバックアップの適用

Hinemos エージェントのインストール・アンインストール方法についてはインストールマニュアルを参照して下さい。

## 3.4.3 Windows版エージェントのバックアップ

Hinemos エージェントは、特にデータベースを使用しません。設定等の各種ファイルのバックアップのみが必要になります。

◇Hinemosエージェント設定ファイル

- インストールディレクトリ\hinemos\_agent\lib\job\_agent\Agent.properties
- インストールディレクトリ\hinemos\_agent\lib\job\_agent\log4j.properties
- インストールディレクトリ\hinemos\_agent\lib\log\_agent\Agent.properties
- インストールディレクトリ\hinemos\_agent\lib\log\_agent\log4j.properties

**◇NTSyslogの設定**

設定のバックアップは取得できません（リカバリ時にGUIから再度入力して下さい）。

**◇SNMP Serviceの設定**

設定のバックアップは取得できません（リカバリ時にGUIから再度入力して下さい）。

**3.4.4 Windows版エージェントのリカバリ**

1. Hinemos エージェントの再インストール
2. 設定等の各種ファイルのバックアップの適用
3. NTSyslog, SNMP Serviceの再設定

Hinemos エージェントのインストール・アンインストール方法についてはインストールマニュアルを参照して下さい。

**3.5 Hinemosクライアントのバックアップ・リカバリ**

Hinemos クライアントでは特に設定変更を行うファイルやデータベースは存在しません。そのため、Hinemos クライアントのリカバリにはHinemos クライアントのアンインストール・インストールを行ってください。

## 4 Hinemosの機能セットアップ(OS/環境)

Hinemos を導入するサーバ環境に関する追加設定を示します。

### 4.1 ファイルディスクリプタの設定

1台のHinemosマネージャが管理するHinemosエージェントが多くなった場合、Hinemosマネージャのログファイル (/opt/hinemos/var/log/jboss.log) に、以下のログが出力されることがあります。

```
(Too many open files)
```

このログが出力された場合、ファイルディスクリプタ数が上限値に達しているため、変更が必要です。以下の手順により、ファイルディスクリプタ数の上限値を変更してください。

Hinemosマネージャがインストールされているサーバの、/opt/hinemos/hinemos.cfgに対して、以下の1行を追記します。(Red Hat Enterprise Linux 5のデフォルト値である1024から2048に変更する例を示します)

```
ulimit -n 2048
```

追記後、Hinemosマネージャを再起動してください。

### 4.2 内部DBへの接続(データソース)設定

Hinemos (JBoss上のアプリケーション) がプールするコネクション数の上限を超えて、PostgreSQLへの接続要求が出ている場合、Hinemosマネージャのログファイル (/opt/hinemos/var/log/jboss.log) に、以下のログが出力され、Hinemosの各処理が必要とする情報を内部データベースから取得できない状態となります。

```
ERROR [org.jboss.ejb.plugins.LogInterceptor] ...  
javax.ejb.EJBException: No ManagedConnections available within configured  
blocking timeout ( 30000 [ms] );
```

以下の手順により、Hinemosのデータソースの設定を変更します。(本設定値は、Hinemos ver. 3.1.4以降のデフォルト値となっています。)

PostgreSQLの最大接続数を変更します。/opt/hinemos/etc/postgresql.conf に定義されている最大接続数を以下のように変更します。

```
max_connections = 400 # (change requires restart)
```

Hinemosがプールする接続数の上限値を変更します。/opt/hinemos/etc/hinemos-ds.xml に定義されている最大接続数を以下のように変更します。

```
<max-pool-size>150</max-pool-size>
```

設定変更後、Hinemosマネージャを再起動してください。

### 4.3 OSロケールと文字コード設定

#### 4.3.1 Hinemos クライアント

Hinemos クライアントで日本語を扱う場合には、Hinemos クライアントをインストールするWindows OSの OSロケールが日本語である必要があります。文字コードは MS932 を前提として動作します。また、Hinemos マネージャも日本語環境としてセットアップする必要があります。



### 4.3.2 Hinemos マネージャ

Hinemos マネージャで日本語を扱う場合には、Hinemos マネージャサーバのOSの LANG 変数が ja\_JP.UTF-8である必要があります。LANG 変数は env コマンドで確認できます。

また、Hinemos マネージャをインストールする際に、インストールスクリプトを manager\_installer\_JP.sh にて実行している必要があります。manager\_uninstaller\_EN.sh よりインストールした場合は、アンインストールして再度インストールしてください。

Hinemos マネージャはOSロケールが ja\_JP.UTF-8 を前提に動作します。本設定は、/opt/hinemos/hinemos.cfg に設定されています。

```
export LANG=ja_JP.UTF-8
```

### 4.3.3 Hinemos エージェント(Linux版エージェント)

- ・ ジョブエージェント

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）の文字コードを指定することができます。Hinemos ジョブエージェントの Agent.properties の input.encoding パラメータにて変更します。デフォルトはOSシステムロケールとして設定した文字コードです。設定変更にはHinemos ジョブエージェントの再起動が必要です。

例えば、EUC-JPのLinux環境でジョブを実行する場合は、以下のように設定を変更します。

```
## character encoding of job's stdout and stderr
input.encoding=EUC-JP
```

- ・ ログ転送エージェント(ログ読み込み)

Linux環境ではログ転送するログファイルは、Hinemos エージェントサーバの syslog-ng 経由で Hinemos マネージャに syslog 送信されます。Hinemos ログ転送エージェントのログファイル読み込み時の文字コードを指定することができます。Hinemos ログ転送エージェントの Agent.properties の input.encoding パラメータにて変更します。デフォルトは UTF-8 です。設定変更にはHinemos ジョブエージェントの再起動が必要です。

```
## character encoding of log file
log.file.encoding=UTF-8
```

- ・ ログ転送エージェント(ログ送信)

Linux環境ではログ転送エージェントから送信される syslog メッセージの文字コードは OSシステムロケールに依存します。OSシステムロケールが ja\_JP.UTF-8 以外の場合は、次のように起動スクリプト log\_agent\_start.sh を変更して再起動してください。

```
## startup
JAVA_OPTS="${JAVA_OPTS} -Dfile.encoding=UTF-8"
...
```

- ・ OSシステムログの監視(syslog-ng)

Linux環境ではOSシステムログは、Hinemos エージェントサーバの syslog-ng よりHinemos マネージャに syslog 送信します。syslog-ng は文字コード変換の機能を持たないため、OSシステムロケールのままのメッセージを syslog としてHinemos マネージャに送信します。

Hinemos マネージャではUTF-8を前提に動作するため、文字化けを発生させないためには、Hinemos エージェントサーバのOSロケールを ja\_JP.UTF-8 とするか、マルチバイト文字を含まないASCII文字のみのログメッセージを扱うようにしてください。

### 4.3.4 Hinemos エージェント(Windows版エージェント)

- ・ ジョブエージェント

Linux版エージェントと同様です。



- ログ転送エージェント(ログ読み込み)

Windows環境ではログ転送するログファイルは、ログ転送エージェントより直接 Hinemos マネージャに syslog 送信されます。Hinemos ログ転送エージェントのログファイル読み込み時の文字コードを指定することができます。Hinemos ログ転送エージェントの Agent.properties の input.encoding パラメタにて変更します。デフォルトは UTF-8 です。設定変更にはHinemos ジョブエージェントの再起動が必要です。

MS932 の Windows 環境のログファイルを読み込む場合は、次のように変更します。

```
## character encoding of job's stdout and stderr
input.encoding=MS932
```

- ログ転送エージェント(ログ送信)

Windows環境ではログ転送から送信される syslog メッセージは UTF-8 です。

- Windows イベントの監視(NTSyslog)

Windows環境ではWindowsイベントは、NTSyslogによりHinemos マネージャに syslog 送信します。NTSyslog はWindows イベントを読み込み、UTF-8に変換してHinemos マネージャに syslog 送信します。

## 5 Hinemosの機能セットアップ(共通機能)

Hinemos 共通機能を使用する際の追加設定を示します。

### 5.1 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。

### 5.2 INTERNALイベントの出力先変更

Hinemos 内部で発生するイベント(INTERNALイベント)の出力先を変更することができます。INTERNALイベントの設定可能な出力先は3つです。`/opt/hinemos/etc/hinemos.properties` に設定することで、各々の出力先に出力するかを選択することができます。設定変更を反映させるにはHinemos マネージャの再起動が必要です。

- Syslog サーバ

`common.internal.syslog` パラメータによりsyslog送信の有効無効を設定します。送信先のSyslog サーバも合わせて本設定ファイルで指定します。

```
## Internal Log : Send Syslog  
common.internal.syslog=false  
common.internal.syslog.priority.critical=true  
common.internal.syslog.priority.warning=true  
common.internal.syslog.priority.info=true  
common.internal.syslog.priority.unknown=true  
  
common.internal.syslog.host=127.0.0.1  
common.internal.syslog.port=514  
common.internal.syslog.facility=daemon  
common.internal.syslog.severity=alert
```

- 監視[イベント]ビュー

`common.internal.event` パラメータにより監視[イベント]ビューへの通知の有効無効を設定します。

```
## Internal Log : Hinemos Event  
common.internal.event=true  
common.internal.event.priority.critical=true  
common.internal.event.priority.warning=true  
common.internal.event.priority.info=true  
common.internal.event.priority.unknown=true
```

- HinemosApl.log

common.internal.file パラメタによりHinemosApl.logファイルへの通知の有効無効を設定します。HinemosApl.logへのファイル出力は /opt/hinemos/etc/jboss-log4j.xml よりデフォルトで 無効のため、Hinemos AplLog FileAppenderのHINEMOS\_APL\_LOGのコメントアウトを外してください。

```
## Internal Log : Hinemos Log File (HinemosApl.log)
common.internal.file=false
common.internal.file.priority.critical=true
common.internal.file.priority.warning=true
common.internal.file.priority.info=true
common.internal.file.priority.unknown=true
```

```
<!-- Hinemos AplLog FileAppender -->
<!-- appender name="HINEMOS_APL_LOG" class="org.jboss.logging.appender.DailyRollingFileAppender">
  <errorHandler class="org.jboss.logging.util.OnlyOnceErrorHandler"/>
  <param name="File" value="/opt/hinemos/var/log/HinemosApl.log"/>
  <param name="Append" value="true"/>

  <param name="DatePattern" value=".'yyyy-MM-dd"/>

  <layout class="org.apache.log4j.PatternLayout">
    <param name="ConversionPattern" value="%m%n"/>
  </layout>
</appender -->
```

## 5.3 Hinemos マネージャ死活検知

Hinemos クライアントより定期的にHinemos マネージャへ接続を試み、応答がない場合はHinemos マネージャの障害としてダイアログ表示をします。監視間隔はメニュー「設定」->「設定」より表示される「設定ダイアログ」より行います。

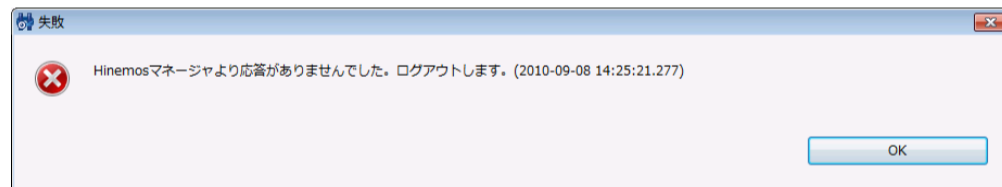


図 5-1 Hinemos マネージャ死活検知

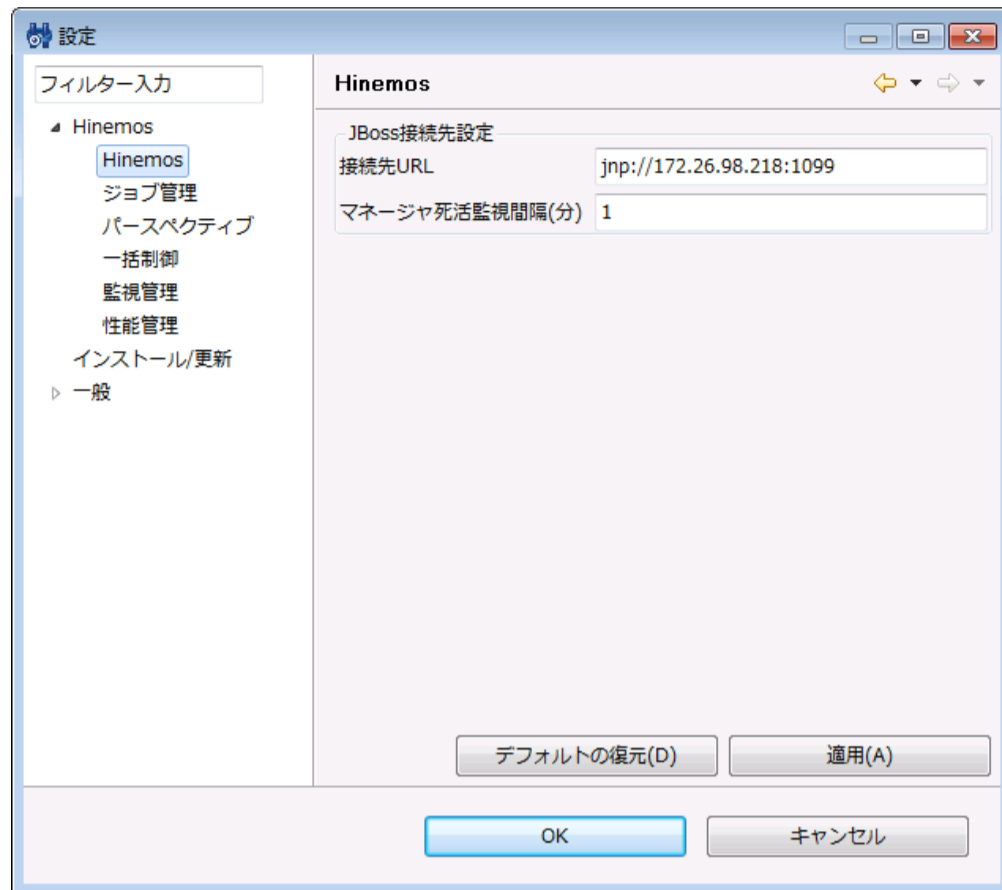


図 5-2 Hinemos マネージャ死活検知間隔

## 5.4 メール通知を有効にする

### 5.4.1 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 protocol -->

```

(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>

</server>
```

以下のパラメータを設定してください。

```
<!-- 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)"/>
```

## 5.4.2 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

これらのパラメータはRFC822に準拠したフォーマットで記述して下さい。

## 5.4.3 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 -->

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

  </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。

#### 5.4.4 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">
```

## 5.5 ログエスケレーション通知の設定

ログエスケレーション通知機能に関する設定項目を以下に記載します。

### 5.5.1 埋め込みホスト名の設定方法

以下の設定ファイル内のパラメータ `common.manager.hostname` を変更することにより、送信されるsyslog(RFC3164)のHeader部に埋め込むホスト名をより柔軟に指定することが可能です。

`/opt/hinemos/etc/hinemos.properties`

```
## syslog transfer : Hostname for syslog header
#common.manager.hostname=${NODE}
```

表5-1. syslogホスト名(`common.manager.hostname`)に指定できる値

hinemos.propertiesの設定値	ホスト名として埋め込まれる文字列	送信されるsyslogの内容
未定義(DEFAULT)あるいは空文字列	送信元となるマネージャサーバのノード名 <sup>1</sup> を埋め込む	<PRI> Mmm dd hh:mm:ss hostname message...
半角英数字の文字列(ex. XXX)	指定された文字列をホスト名として埋め込む	<PRI> Mmm dd hh:mm:ss XXX message...
<code>\${FACILITY_ID}</code>	(組み込みスコープに対する通知情報の場合) 送信元となるマネージャサーバのノード名を埋め込む(その他) 通知対象となっているファシリティのファシリティIDを埋め込む	<PRI> Mmm dd hh:mm:ss facilityid message...
<code>\${NODE}</code>	(ノード単位の通知情報の場合) 通知対象となっているノードのノード名を埋め込む(その他) 送信元となるマネージャサーバのノード名を埋め込む	<PRI> Mmm dd hh:mm:ss nodename message...

<sup>1</sup> hostnameコマンドの実行結果

## 5.6 セルフチェック機能

Hinemosマネージャの内部状態を定期的に確認して、その結果をユーザに通知するセルフチェック機能が提供されます。

確認対象となる内部状態は以下のとおりです。

- ・ アプリケーション障害
  - ・ データベースへのアクセス障害に伴う機能不全
  - ・ 内部スケジューラの異常に伴う機能不全
  - ・ 外部プログラムとの連携不能に伴う機能不全
- ・ リソース枯渇
  - ・ 設定追加や通知量に伴うメモリ領域(Java Virtual Machine内)の枯渇
  - ・ Hinemosマネージャが利用するファイルシステム空き容量の枯渇
- ・ パフォーマンス低下
  - ・ 設定追加や通知量増大に伴う一時情報格納領域の肥大化
  - ・ 履歴情報(イベント、性能情報、ジョブ履歴など)の肥大化
  - ・ Hinemosマネージャサーバのスワップアウト発生

異常が検知された場合、その概要をイベントとして格納したり、オリジナルメッセージをsyslogとして外部の機器へ送信することができます。

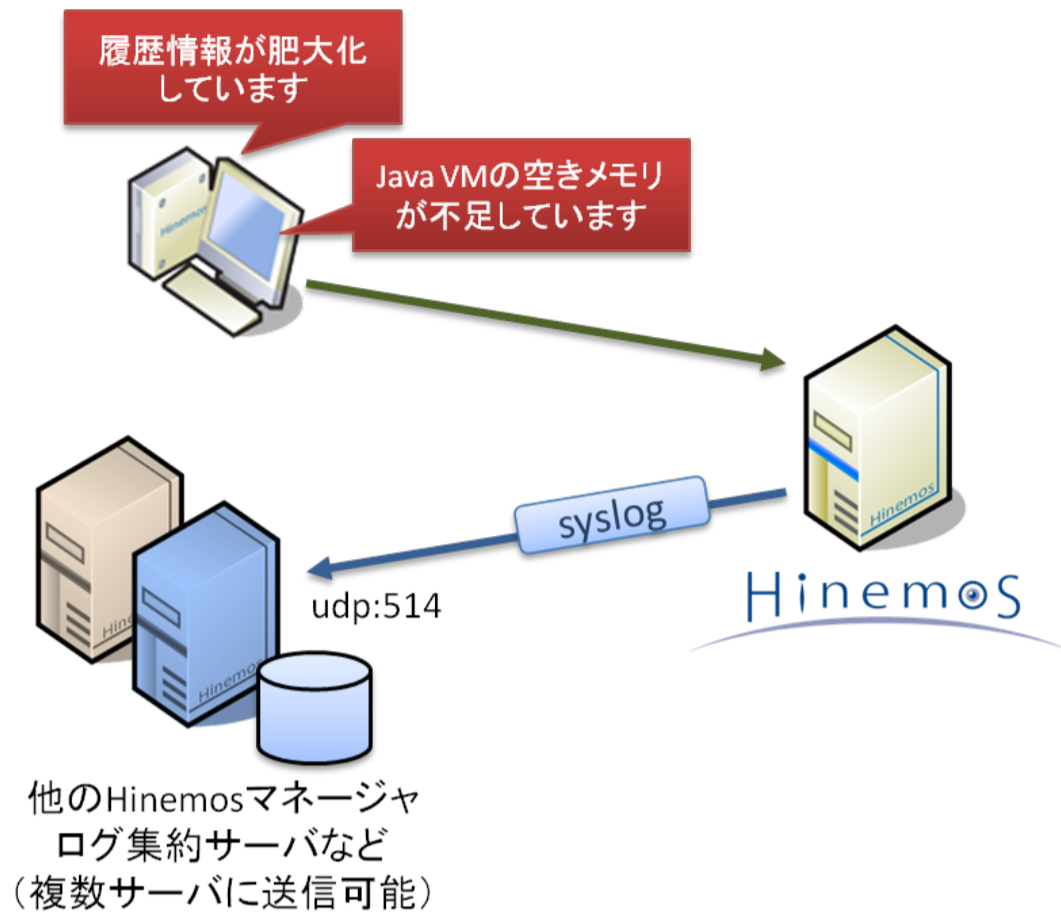


図5-3. セルフチェック機能の概要

## 5.6.1 通知される情報の一覧

表5-2. セルフチェック機能の通知情報

区分	異常内容	監視項目ID	メッセージID	重要度
アプリケーション障害	データベースにアクセス不能	SYS_DS_HinemosDS	001	警告
	スケジューラの動作遅延	SYS_SCHEDULE_スケジューラ名 <sup>2</sup>	001	警告
	外部プログラム(SyslogForward)との連携不能	SYS_EXT_SyslogForward_IPアドレス <sup>3</sup>	001	警告
リソース枯渇	Java VMのメモリ領域枯渇	SYS_JVM_HEAP	001	警告
	ファイルシステムの空き領域枯渇	SYS_FS_マウントポイント <sup>4</sup>	001	警告
パフォーマンス低下	一時格納情報の容量増大	SYS_JMS_識別子 <sup>5</sup>	001	警告
	履歴情報の容量増大	SYS_DBSIZE_テーブル名_閾値種別 <sup>6</sup>	001	警告
	スワップアウトの発生	SYS_SWAPOUT	001	警告

<sup>2</sup> スケジューラ名(Quartz, QuartzRAM)

<sup>3</sup> インストール時に指定したHinemosマネージャのIPアドレス

<sup>4</sup> /opt/hinemos/var/logおよび/opt/hinemos/var/dataが含まれるパーティションのマウントポイント



- 5 JMS Queueの識別子(clustercontrolCollectiveRunExecute, clustercontrolCollectiveRunStatus, clustercontrolJobManagementStatus, clustercontrolLogtransferLogTransferFileManage, clustercontrolNotifyCommand, clustercontrolNotifyControl, clustercontrolNotifyControlNonPersistence, clustercontrolNotifyEvent, clustercontrolNotifyJob, clustercontrolNotifyLogEscalation, clustercontrolNotifyMail, clustercontrolNotifyStatus, clustercontrolSnmpTrapSnmpTrapNotify, clustercontrolSyslogngDup, clustercontrolVmControlExecute, clustercontrolVmControlStatus), JMS TOPICの識別子(clustercontrolJobManagementExecute, clustercontrolLogtransferUpdate, clustercontrolRepositoryUpdate, clustercontrolSnmpTrapUpdate)
- 6 テーブル名(cc\_event\_log, cc\_calculated\_data, cc\_job\_session, cc\_crun\_session) および閾値種別(MByte : 1, レコード数 : 2)

**表5-3. セルフチェック機能の通知情報(メッセージ)**

監視項目ID	メッセージID	メッセージ
SYS_DS_HinemosDS	001	データソース (HinemosDS) が利用できません。定常的に出力される場合、同時に実行される処理量 (監視など) が過多であるため、設定量を削減してください。
SYS_SCHEDULE_スケジューラ名	001	スケジューラ (スケジューラ名) に遅延 (300 [sec]以上) が発生しています。連続して出力される場合、定期的に行われる各処理 (監視など) が正しく動作していない可能性があるため、Hinemosマネージャの再起動を推奨します。
SYS_EXT_SyslogForward_IPアドレス	001	SyslogForward (IPアドレス) から接続されていません。syslog-ngを再起動し、SyslogForwardを再接続させることを推奨します。
SYS_JVM_HEAP	001	利用可能なメモリ容量 (XXX [MByte]) が閾値 (32 [MByte]) を下回りました。定常的に出力される場合、Hinemosマネージャの再起動を推奨します。また、登録サーバ数や各処理の設定量 (監視など) が過剰なため、設定量を削減してください。
SYS_FS_マウントポイント	001	ファイルシステム (マウントポイント) の使用量 (XXX [%]) が閾値 (75 [%]) を超えました。メンテナンス運用 (ログファイル除去、メンテナンス機能およびスクリプト) を実施し、不必要となった情報を取り除いてください。
SYS_JMS_識別子	001	一時情報 ([Queue/Topic].識別子) が蓄積量 (XXX) が閾値 (5000) を超えました。定常的に出力される場合、Hinemosマネージャからの通知 (監視結果など) が過剰であるため、設定を変更してください。
SYS_DBSIZE_テーブル名_閾値種別	001	内部DBにログ情報 (テーブル名) が非常に多く蓄積 (XXX mbyte - YYY rows > 閾値 512 [mbyte/rows]) されています。パフォーマンス低下の要因となるため、メンテナンス運用 (メンテナンス機能およびスクリプト) を実施してください。
SYS_SWAPOUT	001	メモリのスワップアウト (XXX [Blocks]) が発生しました。サーバのリソース状況を確認し、Hinemosマネージャのパフォーマンスをご確認ください。

**表5-4. セルフチェック機能の通知情報(オリジナルメッセージ)**

監視項目ID	メッセージID	オリジナルメッセージ
SYS_DS_HinemosDS	001	datasource(HinemosDS) is not available. failure in query(SELECT 1).
SYS_SCHEDULE_スケジューラ名	001	scheduler (スケジューラ名) has not been running for 300 [sec] (since 最終動作日時).
SYS_EXT_SyslogForward_IPアドレス	001	connection from syslogforward(IPアドレス) is not registered.
SYS_JVM_HEAP	001	free heap of jvm (XXX [mbyte]) is not enough (threshold 32 [mbyte]).
SYS_FS_マウントポイント	001	usage of filesystem(マウントポイント) is too high (XXX [%] > threshold 75 [%]).
SYS_JMS_識別子	001	temporally data in [Queue/Topic] (識別子.属性名) is too many (XXX > threshold 5000).
SYS_DBSIZE_テーブル名_閾値種別	001	stored data (テーブル名) is too large (XXX mbyte - YYY rows > threshold 512 [mbyte/rows]).
SYS_SWAPOUT	001	ram swap-out(XXX [blocks]) occurred since 最終確認日時.

## 5.6.2 セルフチェック機能の設定方法

セルフチェック機能の設定ファイルは以下となります。

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

表5-5. セルフチェック機能の設定値

パラメータ名	設定内容	デフォルト値
IntervalSec	内部コンポーネントの状態確認間隔[sec]	150
AlertThreshold	通知するまでの異常の検知回数	3
Event	全ての通知情報をイベントとして出力する場合はtrue, 出力しない場合はfalseとする。	true
Syslog	全ての通知情報をsyslogで送信する場合はtrue, 送信しない場合はfalseとする。	false
SyslogHost	syslogの送信先とするIPアドレスを指定する。(半角カンマ区切りで複数指定可能)	127.0.0.1
SyslogPort	syslogの送信先とするポート番号(UDP)を指定する。	514
SyslogFacility	送信するsyslogのファシリティを指定する。	daemon
SyslogSeverity	送信するsyslogの重要度を指定する。	alert
SnmpCommunity	マネージャ自身(127.0.0.1)に対するSNMPポーリングで利用するコミュニティ名	public
SnmpPort	マネージャ自身(127.0.0.1)に対するSNMPポーリングで利用するポート番号	161
SnmpVersion	マネージャ自身(127.0.0.1)に対するSNMPポーリングで利用するバージョン(1/2c)	2c
SnmpTimeoutMSec	マネージャ自身(127.0.0.1)に対するSNMPポーリングで利用するタイムアウト[sec]	3000
SnmpRetries	マネージャ自身(127.0.0.1)に対するSNMPポーリングで利用するリトライ回数	3
JVMHeapMonitoring	SYS_JVM_HEAPを有効とする場合はtrue, 無効とする場合はfalseとする。	true
JVMHeapThresholdMByte	SYS_JVM_HEAPで利用する閾値[MByte]	32
FileSystemMonitoring	SYS_FS_マウントポイントを有効とする場合はtrue, 無効とする場合はfalseとする。	false
MountPointDatabase	/opt/hinemos/var/dataが含まれるパーティションのマウントポイント(/optなど)	環境に依存する
MountPointLog	/opt/hinemos/var/logが含まれるパーティションのマウントポイント(/optなど)	環境に依存する
FileSystemDBThresholdPer	/opt/hinemos/var/dataが含まれるパーティションのファイルシステム利用率閾値 [%]	75
FileSystemLogThresholdPer	/opt/hinemos/var/logが含まれるパーティションのファイルシステム利用率閾値 [%]	75
SwapOutMonitoring	SYS_SWAPOUTを有効とする場合はtrue, 無効とする場合はfalseとする。	false
DatasourceMonitoring	SYS_DS_HinemosDSを有効とする場合はtrue, 無効とする場合はfalseとする。	true
SchedulerMonitoring	SYS_SCHEDULE_スケジューラ名を有効とする場合はtrue, 無効とする場合はfalseとする。	true
SchedulerDelayThresholdSec	スケジューラの動作遅延を判定する最大遅延時間 [sec]	300
JMSMonitoring	SYS_JMS_識別子を有効とする場合はtrue, 無効とする場合はfalseとする。	true
JMSMessageCountThreshold	一時格納情報(JMS)の肥大化を判定する最大メッセージ数	5000
SyslogForwardMonitoring	SYS_EXT_SyslogForward_IPアドレスを有効とする場合はtrue, 無効とする場合はfalseとする。	true

SyslogForwardHost	外部プロセス(SyslogForward)が動作するホストのIPアドレスを指定する	環境に依存する
LogTableSizeMonitoring	SYS_DBSIZE_テーブル名_閾値種別を有効とする場合はtrue, 無効とする場合はfalseとする。	true
ThresholdEventLog	イベント情報の肥大化を判定する最大閾値 [mbyte or rows]	512
ThresholdTypeEventLog	イベント情報の肥大化を判定する閾値の種別 [mbyte : 1, rows : 2]	1
ThresholdCollectedResourceLog	収集された性能値情報の肥大化を判定する最大閾値 [mbyte or rows]	5120
ThresholdTypeCollectedResourceLog	収集された性能値情報の肥大化を判定する閾値の種別 [mbyte : 1, rows : 2]	1
ThresholdJobLog	ジョブ履歴情報の肥大化を判定する最大閾値 [mbyte or rows]	1000
ThresholdTypeJobLog	ジョブ履歴情報の肥大化を判定する閾値の種別 [mbyte : 1, rows : 2]	2
ThresholdCrunLog	一括制御履歴情報の肥大化を判定する最大閾値 [mbyte or rows]	1000
ThresholdTypeCrunLog	一括制御履歴情報の肥大化を判定する閾値の種別 [mbyte : 1, rows : 2]	2

変更した設定値を反映させるためには、JBossの再起動が必要となります。

## 6 Hinemosの機能セットアップ(一括制御機能)

Hinemos 一括制御機能を使用する際の追加設定を示します。

### 6.1 一括制御機能を有効にする

一括制御機能は使用するには、Hinemosインストール後にOS環境へのセットアップが必要です。

#### 6.1.1 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。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。

##### 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 (root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。JBoss起動ユーザはrootです。

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。

監視対象ノードのユーザrootへのセットアップを開始します。

```
#####
###
###      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. 3.2.0
###
### Copyright (C) 2006 - 2009 NTT DATA Corporation.
#####

一括制御機能のセットアップ

1) root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

====>
```

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。

```

====> 1

root 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 (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
5e:92:49:03:e6:7b:e3:10:a9:29:46:fd:2f:4a:48:e6 root@sawait_dev_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) root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、 root 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、 root 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
```

以上でsshの設定は終了です。管理対象ノードのユーザをrootより変更したい場合は、hinemos\_setup\_collectiverun.shの引数にユーザ名を指定して上記手順を実行してください。

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

### 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。

1. /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
```

2. /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
```

以上でrshの設定は終了です。

## 6.1.2 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、hinemos.propertiesのcollective.run.shellを編集します。また監視対象サーバでコマンド実行するユーザをrootより変更する場合は、collective.run.userを編集します。ただし、collective.run.userをrootより変更すると、既に一括制御機能の登録済みのroot権限が必要なコマンドが実行できなくなります。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

```
##
## Collective Run function
##
```



```
## Collective Run : Run Method (ssh / rsh)
collective.run.shell=ssh
## Collective Run : Remote User
collective.run.user=root
```

### 6.1.3 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。

- FTPサーバの設定例

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)"
```

## 7 Hinemosの機能セットアップ(監視管理/性能管理機能)

Hinemos 監視管理/性能管理機能を使用する際の追加設定を示します。

### 7.1 デバイス単位でのリソース情報を監視するための設定

Hinemosでは、ディスク・ネットワークについて、デバイス単位の性能値を取得・監視することができます。デバイス単位の性能値を取得するためには、リポジトリ情報にノードのデバイス情報を登録しておく必要があります。ここでは、ディスク・ネットワーク情報を監視するために必要な手順について解説します。

- ディスク情報の登録方法

Windowsノードのディスク情報を監視するためには、論理ディスクの情報をリポジトリに登録する必要があります。これにより、論理ディスク単位で性能値を取得することができます。下記のようにして、論理ディスクの情報をリポジトリに登録してください。

1. デバイスINDEX

Aドライブの場合を1として、以降アルファベットの順番で加算した値を入力してください。(Cドライブの場合3、Dドライブの場合4となります。)

2. デバイス名

ドライブ文字を入力してください。(任意の名前を設定することが可能です。)

3. デバイス種別

"disk"と入力して下さい。

- NIC情報の登録方法

ユーザマニュアル「ノード情報の作成 NIC情報の登録方法」を参照して下さい。

※ Windows 2000 Advanced Server のディスク情報を監視する場合

Hinemosでは、Windowsノードのディスク情報を、パフォーマンスモニタの論理ディスクオブジェクトより取得しています。Windows 2000では、論理ディスクオブジェクトに関する情報取得はデフォルトでオフとなっているため、下記コマンドを実行してオンに変更しておく必要があります。

(参考：<http://support.microsoft.com/kb/253251/ja>)

論理ディスクオブジェクトをオンにするコマンド

```
# diskperf -YV
```

### 7.2 SQL監視対象のRDBMS追加

SQL監視対象のRDBMSを新規に追加する方法を記載します。SQL監視機能はHinemos マネージャのJBoss(Javaアプリケーション)よりJDBCアクセスすることで実現しています。Java SE Runtime Environment 5.0上で動作するJDBCドライバが必要です。

- 追加手順

1. JDBCドライバの配置

用意したJDBCドライバを /opt/hinemos/jboss-4.2.2.GA/server/default/lib/ に配置します。

```
$ cp add_jdbc.jar /opt/hinemos/jboss-4.2.2.GA/server/default/lib/
```



2. jdbc.propertiesの編集

追加するRDBMSの情報を /opt/hinemos/etc/jdbc.properties に追加します。まず、Hinemos で扱えるRDBMSの種類数 jdbc.driver の数をカウントアップし、 RDBMSの詳細情報を追加します。

```
# count of jdbc drivers
jdbc.driver = 4
```

```
jdbc.driver.name.4 = {RDBMS表示名}
jdbc.driver.classname.4 = {JDBCドライバクラス名}
jdbc.driver.logintimeout.4 = {JDBCログインタイムアウト設定}
jdbc.driver.properties.4 = {JDBC接続時パラメタ}
```

3. Hinemos マネージャの再起動

設定を反映するためHinemos マネージャを再起動します。再起動後にSQL[作成・変更]ダイアログより「接続先DB」のプルダウンメニューに追加されていることが 確認できれば終了です。

### 7.3 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、

1. SNMP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 3-1 参照）

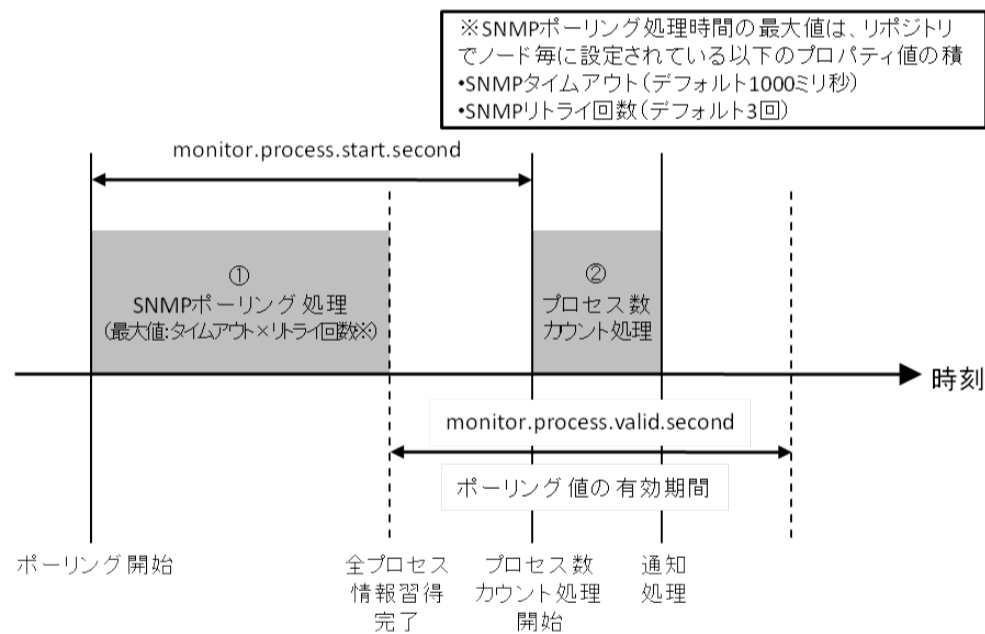


図 7-1 プロセス監視の処理

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
```

## 7.4 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。

### 7.4.1 証明書の準備

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。

### 7.4.2 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/hinemos/.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_22/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。

### 7.4.3 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"
```

## 7.5 syslog-ng監視の設定

### 7.5.1 エスケープ処理の無効化

Hinemos ver3.0以降では、監視対象のログの文字列中にエスケープ対象文字（'や"）が含まれている場合、該当文字の前にはが挿入されてエスケープされます。これにより、そのログを検知するためのパターンマッチ表現を単純なjavaのパターンマッチ表現で表現できない場合があります。

表 7-1 エスケープ対象文字

対象文字	ASCIIコード
'	0x27
"	0x22

表 7-2 エスケープされるログの例

ログに出力される文字列	HTTP/1.0" 50
syslog-ngのtemplateによってエスケープされた後の文字列	HTTP/1.0\" 50
Hinemosマネージャへ転送される文字列	HTTP/1.0\" 50
上記とマッチするパターンマッチ表現	.*HTTP/1.0\" 50.*

エスケープ処理を無効化するには以下の設定を変更します。

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 (template\_escape(no); を追記します。)

```
#add for Hinemos Manager 3.2.0
```

(中略)

```
destination d_hinemos_syslogforward { program("/opt/hinemos/jre1.5.0_22/bin/java -Xms16m
-Xmx64m -cp /opt/hinemos/lib/syslogforward:/opt/hinemos/lib/HinemosCommon.jar:/opt/hinem
os/lib/HinemosIF.jar:/opt/hinemos/lib/HinemosManager.jar:/opt/hinemos/lib/commons-loggin
g-1.1.jar:/opt/hinemos/lib/log4j-1.2.15.jar:/opt/hinemos/lib/javassist.jar:/opt/hinemos/
lib/jboss-aop-jdk50.jar:/opt/hinemos/lib/jboss-messaging-client.jar:/opt/hinemos/lib/jbo
ss-remoting.jar:/opt/hinemos/lib/jbossall-client.jar:/opt/hinemos/lib/trove.jar:/opt/hin
emos/lib/syslogforward/SyslogForward.jar: com.clustercontrol.syslogng.forward.LogForward
/opt/hinemos/lib/syslogforward/LogForward.properties /opt/hinemos/lib/syslogforward/log4
j.properties" template("<$PRI>$DATE $HOST $MSG\n") template_escape(no));};
```

2. syslog-ngを再起動します。

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

### 7.5.2 ホスト名置換処理の設定

syslogパケットのHEADER部には日付とホスト名またはIPアドレス、または空白が入ります。

例)syslogパケットのHEADER部(TIMESTAMP HOSTNAME)

```
Feb 25 14:09:07 webserver
```

syslog-ngは、設定ファイル (/etc/syslog-ng/syslog-ng.conf) の use\_dnsとkeep\_hostnameの設定により、syslogメッセージ受信時にHOSTNAMEを解釈し変更します。

表 7-3 syslog-ngのオプション

オプション	yes指定時の動作
use_dns	HOSTNAMEが空白の場合、名前解決を試みる
keep_hostname	HOSTNAMEを変更しない

これらのオプション指定時の動作は以下となります。

**受信syslogメッセージのHEADERにホスト名が設定されている場合：**

keep\_hostname , use\_dns の設定に関係なく  
HOSTNAME=HEADERのホスト名 (変更なし)

**受信syslogメッセージのHEADERにIPアドレスが設定されている場合：**

- use\_dns (no);  
HOSTNAME=HEADERのIPアドレス (変更なし)
- keep\_hostname(yes);  
HOSTNAME=HEADERのIPアドレス (変更なし)
- keep\_hostname (no); かつ use\_dns (yes);  
HOSTNAME=HEADERのIPアドレスを名前解決(/etc/hosts等)

**受信syslogメッセージのHEADERにホスト名、IPアドレスが設定されていない(空白)場合：**

- use\_dns (yes);  
HOSTNAME=送信元のIPアドレス、名前解決が可能ならホスト名
- use\_dns (no);  
HOSTNAME=送信元のIPアドレス

## 7.6 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で通信する必要があります。

### 7.6.1 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
```

## 7.6.2 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）。

## 8 Hinemosの機能セットアップ(ジョブ管理機能)

Hinemos ジョブ管理機能を使用する際の追加設定を示します。

### 8.1 起動コマンドの動作変更

ジョブは実行時にOS プラットフォームを自動的に識別します。このOS プラットフォームの識別結果により「起動コマンド」の動作が異なります。/opt/hinemos\_agent/lib/job\_agent/Agent.properties の command.create.mode パラメタを変更することでこの動作を変更できます。設定変更にはジョブエージェントの再起動が必要です。

表 8-1 起動コマンドの動作変更

設定値	説明
auto	プラットフォームの自動識別(デフォルト)
windows	Windowsプラットフォーム用にコマンド生成
unix	UNIX/Linuxプラットフォーム用にコマンド生成
compatible	Hinemos ver3.1/ver3.0互換モード

### 8.2 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>
</datasources>
```

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。

## 8.3 ファイル転送ジョブを有効にする

Windows版エージェントではWindows OSがデフォルトでsshデーモン、sshクライアントを提供していないため、ファイル転送は利用できません。（詳しくはユーザマニュアルの「Windows版エージェントの制限事項とTIPS」の項を参照ください。）

Linux版エージェントで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）



## 9 セキュリティに関する設定

### 9.1 データベースアクセスのパスワードを変更する

#### 9.1.1 PostgreSQLの設定変更

- 以下の手順でパスワードを変更します。

- 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.
```

- 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) ';
```

- 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。

#### 9.1.2 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 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. 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 loubiansky 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)
```

## 10 Hinemosの動作ログに関する設定

### 10.1 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 10-1に示すログファイルに出力されます。

表 10-1 Hinemosマネージャのログファイル一覧

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	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
----	----------------------------------

## 10.2 JBossのログ出力を変更する

- 変更ファイル

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。

- 変更適用方法

Hinemos マネージャの再起動、または10分間隔での自動設定読み込み機構によりログ出力のレベルが反映されます。

## 10.3 PostgreSQLのログ出力を変更する

- 変更ファイル

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
                                     # debug1
                                     # info
                                     # notice
                                     # warning
                                     # error
                                     # log
                                     # fatal
                                     # panic
```

```

#log_error_verbosity = default      # terse, default, or verbose messages

#log_min_error_statement = panic    # Values in order of increasing severity:
# debug5
# debug4
# debug3
# debug2
# debug1
# info
# notice
# warning
# error
# panic(off)

#log_min_duration_statement = -1    # -1 is disabled, 0 logs all statements
# and their durations, in milliseconds.

#silent_mode = off                  # DO NOT USE without syslog or
# redirect_stderr

# - What to Log -

#debug_print_parse = off
#debug_print_rewritten = off
#debug_print_plan = off
#debug_pretty_print = off
#log_connections = off
#log_disconnections = off
#log_duration = off
log_line_prefix = '%t'              # Special values:
#log_line_prefix = "                # Special values:
# %u = user name
# %d = database name
# %r = remote host and port
# %h = remote host
# %p = PID
# %t = timestamp (no milliseconds)
# %m = timestamp with milliseconds
# %i = command tag
# %c = session id
# %l = session line number
# %s = session start timestamp
# %x = transaction id
# %q = stop here in non-session
#      processes
# %% = '%'
# e.g. '<%u%%%d> '

#log_statement = 'none'             # none, mod, ddl, all
#log_hostname = off

```

- 変更適用方法

Hinemos マネージャの再起動によりログ出力のレベルが反映されます。

## 10.4 SyslogForwardのログ出力を変更する



- 変更ファイル

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 [%t] [%c] %m%n

#log4j.rootLogger=info, file
log4j.logger.com.clustercontrol.syslogng.forward=info, file
```

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

- 変更適用方法

syslog-ngサービスの再起動、または10分間隔での自動設定読み込み機構によりログ出力のレベルが反映されます。

## 10.5 エージェントのログファイル一覧

Linux版エージェントのログは表 10-2に示すログファイルに出力されます。

表 10-2 Linux版エージェントのログファイル一覧

ファイル名	job_agent.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/job_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	log_agent.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

Windows版エージェントのログは表 5-3に示すログファイルに出力されます。

表 10-3 Windows版エージェントのログファイル一覧

ファイル名	job_agent.log*
格納ディレクトリ	C:\Program Files\Hinemos\Agent3.2.0\var\log\
ログ出力設定ファイル	C:\Program Files\Hinemos\Agent3.2.0\lib\job_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	log_agent.log*
格納ディレクトリ	C:\Program Files\Hinemos\Agent3.2.0\var\log\

ログ出力設定ファイル	C:\Program Files\Hinemos\Agent3.2.0\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

## 10.6 ジョブエージェントのログ出力を変更する

- 変更ファイル

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。

- [Linux版エージェントの場合]

/opt/hinemos\_agent/lib/job\_agent/log4j.properties

- [Windows版エージェントの場合]

C:\Program Files\Hinemos\Agent3.2.0\lib\job\_agent\log4j.properties

```
log4j.appender.file=org.apache.log4j.RollingFileAppender
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 [%t] [%c] %m%n

#log4j.rootLogger=info, file
log4j.logger.com.clustercontrol.agent=info, file
log4j.appender.file.File=/opt/hinemos_agent/var/log/job_agent.log
```

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。

- 変更適用方法

ジョブエージェントの再起動、または10分間隔での自動設定読み込み機構によりログ出力のレベルが反映されます。

## 10.7 ログ転送エージェントのログ出力を変更する

- 変更ファイル

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。

- [Linux版エージェントの場合]

/opt/hinemos\_agent/lib/log\_agent/log4j.properties

- [Windows版エージェントの場合]

C:\Program Files\Hinemos\Agent3.2.0\lib\log\_agent\log4j.properties

```
### direct messages to file mylog.log ###
log4j.appender.file=org.apache.log4j.RollingFileAppender
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 [%t] [%c] %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.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

log4j.appender.file.File=/opt/hinemos_agent/var/log/log_agent.log
log4j.appender.syslog.SyslogHost={エージェントサーバのホスト名}
```

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。

- 変更適用方法

ログ転送エージェントの再起動、または10分間隔での自動設定読み込み機構によりログ出力のレベルが反映されます。

## 10.8 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)); };
```

## 11 Hinemosマネージャの設定一覧

Hinemosマネージャの設定は、`/opt/hinemos/etc/hinemos.properties` に記述されています。設定の反映にはHinemosマネージャの再起動が必要です。また、`properties`ファイルにマルチバイト文字を記述する場合は`properties`エディタが必要です。

- ・ 共通設定の設定

### パラメタ[`common.mail.from.address`]

プロパティ	<code>common.mail.from.address</code>
プロパティ名	(通知)メール通知 FROM アドレス
説明	メール通知によるメール送信時の FROM アドレス
データ型	文字列
デフォルト値	<code>admin@hinemos.com</code>

### パラメタ[`common.mail.from.personal.name`]

プロパティ	<code>common.mail.from.personal.name</code>
プロパティ名	(通知)メール通知 送信者名
説明	メール通知によるメール送信時の送信者名
データ型	文字列
デフォルト値	Hinemos Admin

### パラメタ[`common.mail.reply.to.address`]

プロパティ	<code>common.mail.reply.to.address</code>
プロパティ名	(通知)メール通知 REPLY アドレス
説明	メール通知によるメールの REPLY アドレス
データ型	文字列
デフォルト値	<code>admin@hinemos.com</code>

### パラメタ[`common.mail.reply.personal.name`]

プロパティ	<code>common.mail.reply.personal.name</code>
プロパティ名	(通知)メール通知 返信先名
説明	メール通知によるメールの 返信先名
データ型	文字列
デフォルト値	Hinemos Admin

### パラメタ[`common.mail.errors.to.address`]

プロパティ	<code>common.mail.errors.to.address</code>
プロパティ名	(通知)メール通知 ERROR アドレス
説明	メール通知によるメールの ERROR アドレス
データ型	文字列
デフォルト値	<code>admin@hinemos.com</code>

### パラメタ[`common.mail.retry.count`]

プロパティ	<code>common.mail.retry.count</code>
プロパティ名	(通知)メール通知 再送回数

説明	メール通知によるSMTPサーバへの送信失敗時の再送回数
データ型	整数
デフォルト値	0

#### パラメタ[common.manager.hostname]

プロパティ	common.manager.hostname
プロパティ名	(通知)ロスエスカレーション通知 送信元ホスト名
説明	ロスエスカレーション通知によるsyslog送信時のHOSTNAME部に指定する文字列
データ型	文字列
デフォルト値	\${NODE}

- 一括制御機能の設定

#### パラメタ[collective.run.shell]

プロパティ	collective.run.shell
プロパティ名	一括制御実行シェル
説明	一括制御機能を実行する際のシェル指定
データ型	文字列(ssh / rsh)
デフォルト値	ssh

#### パラメタ[collective.run.user]

プロパティ	collective.run.user
プロパティ名	一括制御実行シェル
説明	一括制御機能を実行する際にリモートログインするユーザ名
データ型	文字列
デフォルト値	root

- 監視管理機能の設定

#### パラメタ[monitor.common.report.event.count]

プロパティ	monitor.common.report.event.count
プロパティ名	Hinemos イベントダウンロード時件数
説明	監視[イベント]ビューのダウンロード機能の1回当たりのダウンロード最大件数
データ型	整数
デフォルト値	1000

#### パラメタ[monitor.common.thread.pool]

プロパティ	monitor.common.thread.pool
プロパティ名	監視同時実行スレッド最大数
説明	監視処理を行うスレッドの最大数
データ型	整数
デフォルト値	50

#### パラメタ[monitor.process.start.second]

プロパティ	monitor.process.start.second
-------	------------------------------

プロパティ名	(プロセス)プロセス数カウント処理開始間隔
説明	プロセス情報取得ポーリングが開始されてから結果を確認するまでの間隔
データ型	整数
デフォルト値	30

#### パラメタ[monitor.process.valid.second]

プロパティ	monitor.process.valid.second
プロパティ名	(プロセス)プロセス情報有効間隔
説明	プロセス情報取得ポーリングが開始されてから有効と判断する間隔
データ型	整数
デフォルト値	50

#### パラメタ[monitor.snmp.valid.second]

プロパティ	monitor.snmp.valid.second
プロパティ名	(SNMP)SNMP情報有効間隔
説明	SNMP監視ポーリングが開始されてから有効と判断する間隔
データ型	整数
デフォルト値	10

- ・ ジョブ管理機能の設定

#### パラメタ[job.message.retry]

プロパティ	job.message.retry
プロパティ名	ジョブメッセージ送信リトライ回数
説明	Hinemos エージェントへのジョブメッセージ送信失敗時のリトライ回数
データ型	整数
デフォルト値	10

#### パラメタ[job.message.timeout]

プロパティ	job.message.timeout
プロパティ名	ジョブメッセージ送信タイムアウト
説明	Hinemos エージェントへのジョブメッセージ送信タイムアウト
データ型	整数
デフォルト値	60



## 12 Hinemosエージェントの設定一覧

### 12.1 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/hinemos\_agent/lib/job\_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。また、propertiesファイルにマルチバイト文字を記述する場合はpropertiesエディタが必要です。

#### パラメタ[manager.polling.interval]

プロパティ	manager.polling.interval
プロパティ名	Hinemos マネージャの死活確認間隔(秒)
説明	Hinemos マネージャとの通信を確認するためのパケット送受信を行う間隔
データ型	整数
デフォルト値	300

#### パラメタ[facility.update.interval]

プロパティ	facility.update.interval
プロパティ名	リポジトリ情報の更新間隔(秒)
説明	Hinemos エージェント内で保持するリポジトリ情報の更新間隔
データ型	整数
デフォルト値	300

#### パラメタ[end.message.resend.interval]

プロパティ	end.message.resend.interval
プロパティ名	ジョブ終了メッセージ送信間隔(秒)
説明	ジョブ終了メッセージ送信失敗時の再送間隔
データ型	整数
デフォルト値	30

#### パラメタ[sendqueue.reconnection.interval]

プロパティ	sendqueue.reconnection.interval
プロパティ名	メッセージ送信接続切断時の再接続間隔(秒)
説明	Hinemos マネージャへメッセージ送信を行う接続切断時の再接続間隔
データ型	整数
デフォルト値	10

#### パラメタ[job.topic.reconnection.interval]

プロパティ	job.topic.reconnection.interval
プロパティ名	ジョブメッセージ受信接続切断時の再接続間隔(秒)
説明	ジョブ実行のメッセージ受信を行う接続切断時の再接続間隔
データ型	整数
デフォルト値	10

#### パラメタ[repository.topic.reconnection.interval]

プロパティ	repository.topic.reconnection.interval
プロパティ名	リポジトリメッセージ受信接続切断時の再接続間隔(秒)
説明	リポジトリ情報のメッセージ受信を行う接続切断時の再接続間隔
データ型	整数
デフォルト値	10

#### パラメタ[runhistory.clear.delay]

プロパティ	runhistory.clear.delay
プロパティ名	ジョブセッション情報の保持期間(秒)
説明	ジョブセッション情報の保持期間(Hinemos マネージャ異常時用)
データ型	整数
デフォルト値	604800

#### パラメタ[input.encoding]

プロパティ	input.encoding
プロパティ名	ジョブ標準入出力、標準エラー処理用文字コード
説明	ジョブ実行時の標準入出力、標準エラー処理用文字コード
データ型	文字列
デフォルト値	コメントアウト(OSロケール)

#### パラメタ[root.public.key]

プロパティ	root.public.key
プロパティ名	ファイル転送ジョブ用ssh公開鍵
説明	ファイル転送ジョブでssh(scp)使用時の公開鍵
データ型	文字列
デフォルト値	空文字列

#### パラメタ[root.authorized.keys.path]

プロパティ	root.authorized.keys.path
プロパティ名	rootユーザ authorized_keys パス
説明	ファイル転送ジョブで使用する authorized_keys ファイルパス
データ型	文字列
デフォルト値	/root/.ssh/authorized_keys

#### パラメタ[command.create.mode]

プロパティ	command.create.mode
プロパティ名	起動コマンド動作モード
説明	ジョブ実行時のOS プラットフォーム識別(& 互換モード)
データ型	文字列(auto/windows/unix/compatible)
デフォルト値	auto

#### パラメタ[java.naming.provider.url]

プロパティ	java.naming.provider.url
-------	--------------------------

プロパティ名	Hinemos マネージャ接続URL
説明	Hinemos マネージャへの接続URL文字列
データ型	文字列
デフォルト値	jnp://{インストール時指定}:1099

## 12.2 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/hinemos\_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。また、propertiesファイルにマルチバイト文字を記述する場合はpropertiesエディタが必要です。

### パラメタ[manager.polling.interval]

プロパティ	manager.polling.interval
プロパティ名	Hinemos マネージャの死活確認間隔(秒)
説明	Hinemos マネージャとの通信を確認するためのパケット送受信を行う間隔
データ型	整数
デフォルト値	300

### パラメタ[facility.update.interval]

プロパティ	facility.update.interval
プロパティ名	リポジトリ情報の更新間隔(秒)
説明	Hinemos エージェント内で保持するリポジトリ情報の更新間隔
データ型	整数
デフォルト値	300

### パラメタ[sendqueue.reconnection.interval]

プロパティ	sendqueue.reconnection.interval
プロパティ名	メッセージ送信接続切断時の再接続間隔(秒)
説明	Hinemos マネージャへメッセージ送信を行う接続切断時の再接続間隔
データ型	整数
デフォルト値	10

### パラメタ[log.topic.reconnection.interval]

プロパティ	log.topic.reconnection.interval
プロパティ名	ログメッセージ受信接続切断時の再接続間隔(秒)
説明	ログ転送設定のメッセージ受信を行う接続切断時の再接続間隔
データ型	整数
デフォルト値	10

### パラメタ[reporitory.topic.reconnection.interval]

プロパティ	reporitory.topic.reconnection.interval
プロパティ名	リポジトリメッセージ受信接続切断時の再接続間隔(秒)
説明	リポジトリ情報のメッセージ受信を行う接続切断時の再接続間隔
データ型	整数

デフォルト値	10
--------	----

#### パラメタ[unchanged.stats.period]

プロパティ	unchanged.stats.period
プロパティ名	ファイル名の変更チェック期間設定(秒)
説明	ファイルサイズが指定秒間変化がない場合のファイル切り替わりチェック間隔
データ型	整数
デフォルト値	5

#### パラメタ[file.max.size]

プロパティ	file.max.size
プロパティ名	読み込みファイル最大サイズ(byte)
説明	ログ転送対象のファイルサイズ最大値の閾値(警告イベント発生)
データ型	整数
デフォルト値	2147483648

#### パラメタ[log.file.encoding]

プロパティ	log.file.encoding
プロパティ名	ログファイル読み込みエンコーディング
説明	ログファイル読み込み時のエンコーディング指定(エージェントで1種類固定)
データ型	文字列
デフォルト値	UTF-8
参考	<a href="http://java.sun.com/j2se/1.5.0/ja/docs/ja/guide/intl/encoding.doc.html">http://java.sun.com/j2se/1.5.0/ja/docs/ja/guide/intl/encoding.doc.html</a>

#### パラメタ[log.msg.program]

プロパティ	log.msg.program
プロパティ名	ログメッセージ転送プログラム名
説明	ログメッセージのsyslog転送時のプログラム名
データ型	文字列
デフォルト値	hinemos_log_agent

#### パラメタ[java.naming.provider.url]

プロパティ	java.naming.provider.url
プロパティ名	Hinemos マネージャ接続URL
説明	Hinemos マネージャへの接続URL文字列
データ型	文字列
デフォルト値	jnp://{インストール時指定}:1099

## 13 変更履歴

### 変更履歴

版	変更日	変更内容
第1版	2010/9/30	初版発行

Hinemos ver3.2 設定リファレンス

非売品

禁無断複製- 無断転載- 無断再配布

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、Rマークは表記しておりません。