

Hinemos

Hinemos ver4.0
管理者ガイド 第1.1版

目次

1	はじめに	6
2	前提条件	7
3	メンテナンス	8
3.1	Hinemosマネージャのメンテナンス	8
3.1.1	概要	8
3.1.2	メンテナンススクリプト一覧	8
3.1.3	データベース内の不要なデータの削除	9
3.1.4	データベースの再構成	10
3.1.5	データベースのバックアップ	10
3.1.6	データベースのリストア	11
3.1.7	一時キューに格納されている情報の消去	11
3.1.8	通知の抑制基準となる履歴情報の消去	12
3.1.9	環境サマリ情報の取得	12
3.2	ログファイルの削除	12
3.2.1	Hinemosマネージャのログファイル	13
3.2.2	Hinemosエージェントのログファイル	13
3.2.3	Hinemosクライアントのログファイル	13
3.3	Hinemosマネージャのバックアップ・リカバリ	13
3.3.1	Hinemosマネージャのバックアップ	13
3.3.2	Hinemosマネージャのリカバリ	14
3.4	Hinemosエージェントのバックアップ・リカバリ	14
3.4.1	Hinemosエージェントのバックアップ	14
3.4.2	Hinemosエージェントのリカバリ	15
3.5	Hinemosクライアントのバックアップ・リカバリ	16
4	OS/環境	17
4.1	ファイルディスクリプタの設定	17
4.2	データベースへの接続数の設定	17
4.3	Javaヒープメモリの設定	17
4.4	OSロケールと文字コード設定	18
4.4.1	Hinemosクライアント	18
4.4.2	Hinemosマネージャ	18
4.4.3	Hinemosエージェント	18
4.5	HinemosエージェントのファシリティIDの設定	19
5	通知機能	20
5.1	イベント通知	20
5.1.1	最大ダウンロード件数の設定	20
5.2	メール通知	20
5.2.1	メール通知を有効にする	20
5.2.2	SMTP AUTHの設定	21
5.2.3	SSH/TLSの設定	22
5.2.4	エンベロープFromの設定	23
5.2.5	ポート番号の指定	23
5.3	ログエスカレーション通知	24
5.3.1	埋め込みホスト名の設定方法	24

6	一括制御機能	25
6.1	一括制御機能を有効にする	25
6.1.1	管理対象ノードのセットアップ	25
6.1.2	Hinemosマネージャのリモートシェルの設定	27
6.1.3	FTPサーバの起動	28
7	監視設定/性能管理機能	29
7.1	SQL監視	29
7.1.1	監視対象のRDBMSの追加	29
7.2	プロセス監視	29
7.2.1	『値取得の失敗』の通知が発生する場合の対処	29
7.3	HTTPS監視	30
7.3.1	証明書の準備	31
7.3.2	証明書のkeystoreへの登録	31
7.3.3	Java起動オプションによるkeystoreファイルの指定	31
7.4	システムログ監視	31
7.4.1	syslogの送信に関する設定	31
7.4.2	syslogの受信に関する設定	31
7.4.3	制御文字に対するエスケープ処理の無効化	32
7.4.4	syslogのHEADER部に対するホスト名置換処理の設定	32
7.5	SNMPTRAP監視	33
7.5.1	SNMPTRAPの受信に関する設定	33
7.6	Windowsサービス監視	33
7.6.1	WinRMのインストール	33
7.6.2	リモートコンピュータ管理の許可	33
7.6.3	Basic認証の許可	34
7.6.4	非暗号通信の許可	34
7.6.5	HTTPSの設定	34
7.6.6	接続先OSユーザの準備	35
7.6.7	疎通確認	35
7.7	カスタム監視	35
7.7.1	コマンドの動作変更	35
7.7.2	標準出力からの読み込み最大量の設定	35
7.7.3	標準出力に含まれる改行コードの設定	36
7.7.4	コマンドの実行多重度の設定	36
7.8	ポーリングプロトコルの設定	36
7.8.1	Net-SNMPの設定	36
7.8.2	SNMP・WBEMの切り替え方法	37
7.9	数値監視の収集値の設定	37
8	ジョブ管理機能	38
8.1	起動コマンドの動作変更	38
8.2	Hinemosマネージャ再起動時のジョブスケジュール制御の設定	39
8.3	ファイル転送ジョブを有効にする	39
9	その他の機能	42
9.1	セルフチェック機能	42
9.1.1	通知される情報の一覧	43

9.1.2	セルフチェック機能の設定	44
9.2	INTERNALイベント	46
9.2.1	出力先の設定	46
9.3	Hinemosマネージャ死活検知	47
10	セキュリティ	49
10.1	データベースアクセスのパスワードを変更する	49
10.1.1	PostgreSQL（接続先）の設定変更	49
10.1.2	Hinemosマネージャ（接続元）の設定変更	50
11	動作ログ	52
11.1	Hinemosマネージャのログファイル一覧	52
11.2	JBossのログ出力を変更する	53
11.3	PostgreSQLのログ出力を変更する	54
11.4	操作ログのログ出力を変更する	54
11.5	Hinemosエージェントのログファイル一覧	54
11.6	Hinemosエージェントのログ出力を変更する	55
11.7	Hinemosクライアントのログファイル一覧	56
12	Hinemosマネージャの設定一覧	57
13	Hinemosエージェントの設定一覧	65
14	変更履歴	71

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

本マニュアルでは、Hinemosを運用する上でのメンテナンス方法、Hinemosの基本動作に関する設定方法に関して説明します。

ここでいう設定方法とは、主にセキュリティ・性能のチューニング・基本動作の変更に関連した設定ファイル等のことを示しており、運用中に使用するHinemosクライアントからの設定方法に関しては、Hinemosユーザマニュアルを参照してください。

本マニュアル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推奨します。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

本マニュアルは、Hinemosのメンテナンス方法、セキュリティ・性能のチューニング・基本動作の変更に関連した設定方法を説明します。

そのため、Hinemosインストールマニュアルに従い、事前にHinemosマネージャ、エージェント、クライアントが使用可能な状態となっていることを前提としています。

また、設定値の内容や組み合わせによってはHinemosの動作が不安定になる可能性もあるため、設定を変更する前にあらかじめ変更前の設定情報のバックアップを取得しておくことを推奨します。

3 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を長期間で使用するため、導入している環境の運用条件に合わせて、定期的にメンテナンス作業を実施していただく必要があります。Hinemosマネージャのメンテナンス作業用には、以下のスクリプトが提供されています。

- データベース内から不要なデータ(保持する必要のなくなったログ情報など)を削除する

```
hinemos_delete.sh
```

- データベースを再構成して不要領域(使用されていない確保領域)をシステムが再利用可能にする

```
hinemos_cluster_db.sh
```

- データベース内のデータ(設定・ログ情報など)をバックアップする

```
hinemos_backup.sh
```

- ダンプファイルからデータベース内にデータ(設定・履歴情報など)をリストアする

```
hinemos_restore.sh
```

- 一時キューに格納されている情報(テンポラリ情報)を消去する

```
hinemos_clear_tmp.sh
```

- 通知の抑制基準となる履歴情報を消去する(全ての通知情報に対する抑制をリセット)

```
hinemos_clear_notify.sh
```

- 環境サマリ情報を取得する

```
hinemos_manager_summary.sh
```

上記のメンテナンススクリプトは全て /opt/hinemos/sbin/mng ディレクトリ内に配置されています。

3.1.2 メンテナンススクリプト一覧

各メンテナンススクリプトを実行する上での前提条件は表3-1の通りです。

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	Hinemosマネージャ 起動中	PostgreSQL起動中 JBoss停止中
不要なデータの削除	hinemos_delete.sh	X	O
データベースの再構成	hinemos_cluster_db.sh	X	O
データベースのバックアップ	hinemos_backup.sh	O	O
データベースのリストア	hinemos_restore.sh	X	X
一時キューの消去	hinemos_clear_tmp.sh	X	O
通知抑制情報の消去	hinemos_clear_notify.sh	O	X
環境サマリ情報の取得	hinemos_manager_summary.sh	O	X

3.1.3 データベース内の不要なデータの削除

データベース内から不要なデータ(保持する必要のなくなったログ情報など)を削除するには、hinemos_delete.shを実行します。Hinemosマネージャのデータベース内には、以下のログ情報が蓄積されています。

- ・ 監視ログ情報(イベント)
- ・ ジョブの実行履歴
- ・ 一括制御の実行履歴
- ・ 数値監視の収集値

これらの履歴情報は、メンテナンス機能の履歴情報削除機能によって定期的に削除できます。(履歴情報削除機能の詳細は、Hinemosユーザマニュアルを参照してください)

履歴情報削除機能からではなく、手動で即時に削除したい場合に hinemos_delete.sh を使用します。

1. Hinemosマネージャを停止し、PostgreSQLのみ起動している状態とします。

```
(root) # /opt/hinemos/bin/hinemos_stop.sh
(root) # /opt/hinemos/bin/pg_start.sh
```

2. 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。-w オプションを指定しなかった場合、パスワードを対話式に入力します。

- ・ 確認済みの監視ログ情報(イベント)を削除する場合

```
(root) # /opt/hinemos/sbin/mng/hinemos_delete.sh -e
```

- ・ 全ての監視ログ情報(イベント)を削除する場合

```
(root) # /opt/hinemos/sbin/mng/hinemos_delete.sh -E
```

- ・ 終了したジョブの実行履歴情報を削除する場合

```
(root) # /opt/hinemos/sbin/mng/hinemos_delete.sh -j
```

- ・ 全てのジョブの実行履歴情報を削除する場合

```
(root) # /opt/hinemos/sbin/mng/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

```
(root) # /opt/hinemos/sbin/mng/hinemos_delete.sh -c
```

- ・ 全ての一括制御実行履歴情報を削除する場合

```
(root) # /opt/hinemos/sbin/mng/hinemos_delete.sh -C
```

- ・ 監視項目IDを指定して数値監視の収集値を削除する場合

```
(root) # /opt/hinemos/sbin/mng/hinemos_delete.sh -p [監視項目ID]
```

- ・ 全ての数値監視の収集値を削除する場合

```
(root) # /opt/hinemos/sbin/mng/hinemos_delete.sh -P
```

- ・ 監視項目IDと直近の保存期間(削除しない直近の日数)を指定して数値監視の収集値を削除する場合

```
(root) # /opt/hinemos/sbin/mng/hinemos_delete.sh -p [監視項目ID] -r [保存期間(日数指定)]
```

- 直近の保存期間(削除しない直近の日数)を指定して数値監視の収集値を削除する場合

```
(root) # /opt/hinemos/sbin/mng/hinemos_delete.sh -P -r [保存期間(日数指定)]
```

- Hinemosマネージャを起動します。

```
(root) # /opt/hinemos/bin/pg_stop.sh  
(root) # /opt/hinemos/bin/hinemos_start.sh
```

3.1.4 データベースの再構成

データベースを再構成して不要領域(使用されていない確保領域)をシステムが再利用可能にするには、hinemos_cluster_db.shを実行します。

データベース内の不要なデータの削除により不要なデータを削除しても、データベースに割り当てられているディスク領域は開放されないため、Hinemosのデータベースによって使用されているディスク領域のサイズは変わりません。

この領域を開放してシステムから再利用可能にするためには、データベースを再構成する必要があります。

データファイルのフラグメンテーション(特にインデックス)などによる性能低下を避けるため、定期的(少なくとも6か月に1回以上)に、データベースを再構成することを推奨します。

また、以下に該当する場合もデータベースの再構成を推奨します。

- 監視対象での障害発生等により大量の通知が発生した
- 不必要となったログ情報を大量に削除した
- Hinemosマネージャのパフォーマンスが低下した

なお、hinemos_cluster_db.shでは既存のデータファイルから新たなデータファイルにコピーしながら再構成するため、データベースファイル /opt/hinemos/var/data/ が格納されているファイルシステム上に、十分な空き領域が必要となります。作業領域として、現在のデータベースサイズ(/opt/hinemos/var/data/配下のファイルサイズ)の2倍以上の空き領域を確保しておくことを推奨します。

- Hinemosマネージャを停止し、PostgreSQLのみ起動している状態とします。

```
(root) # /opt/hinemos/bin/hinemos_stop.sh  
(root) # /opt/hinemos/bin/pg_start.sh
```

- メンテナンススクリプトを実行します。

```
(root) # /opt/hinemos/sbin/mng/hinemos_cluster_db.sh
```

- Hinemosマネージャを起動します。

```
(root) # /opt/hinemos/bin/pg_stop.sh  
(root) # /opt/hinemos/bin/hinemos_start.sh
```

3.1.5 データベースのバックアップ

データベース内のデータ(設定・ログ情報など)をバックアップするには、hinemos_backup.shを実行します。

マネージャサーバの予期せぬ障害発生や、ユーザのオペレーションミスによる設定情報の消失などに備えるため、定期的にデータベース内のデータ(設定、ログ情報など)をバックアップしておくことを推奨します。

- ダンプファイル(データベースのバックアップファイル)を出力するディレクトリ(/tmpを例とします)に移動します。

```
(root) # cd /tmp
```

- メンテナンススクリプトを実行します。

```
(root) # /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。-w オプションを指定しなかった場合、パスワードを対話式に入力します。

3. カレントディレクトリに出力されたダンプファイルを確認します。

```
(root) # ls  
hinemos_pgdump.YYYY-MM-DD_HHmms
```

なお、ダンプファイルはHinemosマネージャを起動した状態でも取得できますが、その時点の状態でバックアップされることにご注意ください。

例えば、ジョブの実行中に取得されたダンプファイルをリストアすると、そのバックアップ時点の状態に復元されて、該当のジョブは実行中のままとなります。リストアした直後に予期せずジョブが実行されることを避けるため、ジョブが実行されていない時間帯にバックアップすることを推奨します。

3.1.6 データベースのリストア

データベースのバックアップにより取得したダンプファイルからデータベース内にデータ(設定・履歴情報など)をリストアするには、hinemos_restore.sh を実行します。

1. Hinemosマネージャを停止します。

```
(root) # /opt/hinemos/bin/hinemos_stop.sh  
(root) # /opt/hinemos/bin/pg_start.sh
```

2. メンテナンススクリプトを実行します。

```
(root) # /opt/hinemos/sbin/mng/hinemos_restore.sh hinemos_pgdump.YYYY-MM-DD_HHmms
```

3. Hinemosマネージャを起動します。

```
(root) # /opt/hinemos/bin/pg_stop.sh  
(root) # /opt/hinemos/bin/hinemos_start.sh
```

3.1.7 一時キューに格納されている情報の消去

一時キューに格納されている情報(テンポラリ情報)を消去するには、hinemos_clear_tmp.sh を実行します。

hinemos_clear_tmp.shがデータベース内のテーブルから削除するのは、一時キューに格納された以下の情報です。

- 処理待ちの通知情報
- ジョブセッション状態遷移に関する情報
- (Hinemosマネージャの強制停止などにより)処理されなくなった不要な情報

なお、一時キュー内の情報が全て削除されるため、Hinemosマネージャ停止前に処理待ちとなっていた以下の情報も削除対象となります。

- 処理待ちの通知情報(イベント、ステータスなど)
- ジョブセッションの状態遷移に関する情報

1. Hinemosマネージャを停止し、PostgreSQLのみ起動している状態とします。

```
(root) # /opt/hinemos/bin/hinemos_stop.sh  
(root) # /opt/hinemos/bin/pg_start.sh
```

2. メンテナンススクリプトを実行します

```
(root) # /opt/hinemos/sbin/mng/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。-w オプションを指定しなかった場合、パスワードを対話式に入力します。

3. Hinemosマネージャを起動します。

```
(root) # /opt/hinemos/bin/pg_stop.sh  
(root) # /opt/hinemos/bin/hinemos_start.sh
```

3.1.8 通知の抑制基準となる履歴情報の消去

通知の抑制基準となる履歴情報を消去するには、hinemos_clear_notify.sh を実行します。

通知機能には重複する通知を抑制する機能が提供されています。(通知機能の詳細は、Hinemosユーザマニュアルを参照してください) 内部では、この通知の抑制のために、直近の通知履歴情報を保持しています。

hinemos_clear_notify.sh を実行すると、保持されている通知履歴情報がクリアされ、抑制されている状態を一時的にリセットすることができます。(全ての通知情報に対する抑制が解除されます)

1. メンテナンススクリプトを実行します

```
(root) # /opt/hinemos/sbin/mng/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。-w オプションを指定しなかった場合、パスワードを対話式に入力します。

3.1.9 環境サマリ情報の取得

マネージャの環境サマリ情報を取得するには、hinemos_manager_summary.shを実行します。

1. メンテナンススクリプトを実行します

- Hinemosマネージャのサマリ情報のみを取得する場合

```
(root) # /opt/hinemos/sbin/mng/hinemos_manager_summary.sh -v
```

- Hinemosマネージャのサマリ情報とOS情報を取得する場合

```
(root) # /opt/hinemos/sbin/mng/hinemos_manager_summary.sh -vv
```

- Hinemosマネージャのサマリ情報(詳細版)を取得する場合

```
(root) # /opt/hinemos/sbin/mng/hinemos_manager_summary.sh -vvv
```

- Hinemosマネージャのサマリ情報(詳細版)と動作ログを取得する場合

```
(root) # /opt/hinemos/sbin/mng/hinemos_manager_summary.sh -o
```

- Hinemosマネージャのスレッドダンプを取得する場合

```
(root) # /opt/hinemos/sbin/mng/hinemos_manager_summary.sh -t
```

3.2 ログファイルの削除

ログファイルの増大によるディスク領域の圧迫を避けるため、Hinemosマネージャ、エージェント、クライアントが出力するログファイルを確認して、不必要となったファイルを削除あるいは移動してください。

3.2.1 Hinemosマネージャのログファイル

Hinemosマネージャのログファイルは、表3-2のディレクトリに出力されます。

表 3-2 Hinemosマネージャのログディレクトリ

パス	内容
/opt/hinemos/var/log/	Hinemosマネージャが出力する各種ログファイル

なお、Hinemosマネージャ向けに 最終更新日から一定の期間(31日)経過したログファイルを削除するスクリプトを提供していません。

- /opt/hinemos/contrib/hinemos_manager

このスクリプトを日次で実行するためには、以下のように /etc/cron.daily/ に配置してください。

```
(root) # cp -p /opt/hinemos/contrib/hinemos_manager /etc/cron.daily/
```

3.2.2 Hinemosエージェントのログファイル

Linux版エージェントのログファイルは、表3-3のディレクトリに出力されます。

表 3-3 Linux版エージェントのログディレクトリ

パス	内容
/opt/hinemos_agent/var/log/	Hinemosエージェントが出力する各種ログファイル

Windows版エージェントのログファイルは、表3-4のディレクトリに出力されます。

表 3-4 Windows版エージェントのログディレクトリ

パス	内容
<Hinemosエージェントのインストールフォルダ>\var\log\	Hinemosエージェントが出力する各種ログファイル

3.2.3 Hinemosクライアントのログファイル

Hinemosクライアントのログファイルは、表3-5のディレクトリに出力されます。

表 3-5 Hinemosクライアントのログディレクトリ

パス	内容
<ユーザのホームディレクトリ>\AppData\Roaming\hinemos\	Hinemosクライアントのログファイル

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・リカバリ方法を説明します。

3.3.1 Hinemosマネージャのバックアップ

- データベースのバックアップ
データベースのバックアップにより、ダンプファイルを取得してください。

- 構成ファイル(設定ファイルなど)のバックアップ

一般的にバックアップ対象とするファイル一覧を以下に示します。(この一覧以外に修正したファイルがある場合、必要に応じてバックアップしてください)

- Hinemosマネージャの設定ファイル
 - /opt/hinemos/etc 配下の全ファイル
- サービススクリプト(Hinemosマネージャをサービス化した場合)
 - /etc/init.d/hinemos_manager
- ログファイルの削除スクリプト
 - /etc/cron.daily/hinemos_manager
- 同一のOS上の他プロセスが使用するファイル
 - /etc/rsyslog.conf
 - /etc/rsyslog_hinemos.conf
 - /etc/sysconfig/rsyslog

3.3.2 Hinemosマネージャのリカバリ

1. Hinemosマネージャの再インストール

Hinemosマネージャをインストールします。Hinemosマネージャが既にインストールされている場合、一度アンインストールした後でインストールします。

Hinemosマネージャのインストール・アンインストールの詳細は、Hinemos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

[データベースのバックアップ](#) で取得したダンプファイルを [データベースのリストア](#) に従ってリストアします。

3. バックアップ済みの構成ファイルの適用

バックアップ済みの構成ファイル(設定ファイルなど)を反映します。

バックアップしている構成ファイルの内容を確認して、反映すべきものがあれば、再インストールしたHinemosマネージャに反映してください。(反映方法は、ファイルの種類・設定ファイルのパラメータに依存します)

3.4 Hinemosエージェントのバックアップ・リカバリ

Hinemosエージェントのバックアップ・リカバリ方法を説明します。

3.4.1 Hinemosエージェントのバックアップ

Linux版エージェント

- 構成ファイル(設定ファイルなど)のバックアップ

一般的にバックアップ対象とするファイル一覧を以下に示します。(この一覧以外に修正しているファイルがある場合、各自でバックアップしてください)

- Hinemosエージェント設定ファイル
 - /opt/hinemos_agent/conf/ 配下の全ファイル
- サービス化スクリプト(サービス化の場合)
 - /etc/init.d/hinemos_agent
- 同一のOS上の他プロセスが使用するファイル
 - /etc/rsyslog.conf
 - /etc/snmp/snmpd.conf
 - /etc/sysconfig/snmpd

Windows版エージェント

- 構成ファイル(設定ファイルなど)のバックアップ

一般的にバックアップ対象とするファイル一覧を以下に示します。(この一覧以外に修正しているファイルがある場合、各自でバックアップしてください)

- Hinemosエージェント設定ファイル
 - [Hinemosエージェントのインストールディレクトリ]\conf\ 配下の全ファイル

- NTSyslogの設定情報

NTSyslogの設定情報 (Syslogデーモンやイベントログの転送設定等) は下記のレジストリに格納されます。

- (32bit) HKEY_LOCAL_MACHINE\SOFTWARE\SaberNet
- (64bit) HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\SaberNet

NTSyslogの設定情報のバックアップとして、上記のレジストリをエクスポートします。

- SNMP Serviceの設定

設定情報のバックアップは取得できません。(リカバリの際はGUIから再入力してください)

3.4.2 Hinemosエージェントのリカバリ

プラットフォーム共通

1. Hinemosエージェントの再インストール

Hinemosエージェントをインストールします。(Hinemosエージェントが既にインストールされている場合、一度アンインストールした後、インストールします)

Hinemosエージェントのインストール・アンインストールの詳細は、Hinemosインストールマニュアルを参照して下さい。

2. バックアップ済みの構成ファイル(設定ファイルなど)の適用

バックアップ済みの構成ファイル(設定ファイルなど)を反映します。

バックアップしている構成ファイルの内容を確認して、反映すべきものがあれば、再インストールしたHinemosエージェントに反映してください。(反映方法は、ファイルの種類・設定ファイルのパラメータに依存します)

Windows版エージェント

1. NTSyslogの設定情報

エクスポートしたレジストリをインポートします。設定変更を反映させるには、NTSyslogの再起動が必要です。

2. SNMP Serviceの再設定

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、バックアップ対象となるファイルがありません。Hinemosクライアントの再インストールにより、リカバリを行ってください。

Hinemosクライアントのインストール・アンインストールの詳細は、Hinemosインストールマニュアルを参照して下さい。

4 OS/環境

Hinemosを導入するサーバ環境に関する追加設定について説明します。

4.1 ファイルディスクリプタの設定

1台のHinemosマネージャに接続するHinemosエージェントの数が非常に多い場合、Hinemosマネージャのログファイル(/opt/hinemos/var/log/jboss.log)に、以下のメッセージが出力されることがあります。

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

このメッセージは、JBossプロセスが扱うファイルディスクリプタ数が上限値に達しており、新たなソケットを生成できない、あるいはファイルを開けない状態であることを示します。

この場合、ファイルディスクリプタ数の上限値の変更が必要となります。Hinemosマネージャの設定ファイル(hinemos.cfg)内に定義されている JBOSS_FD_MAXNUMの設定値を変更させてください。(JBOSS_FD_MAXNUMは、JBossプロセスに付与するulimit値として使用されます)

- /opt/hinemos/hinemos.cfg

```
export JBOSS_FD_MAXNUM=4098
```

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。

4.2 データベースへの接続数の設定

JBossが保持するコネクションプールの上限を超えて、PostgreSQLへのコネクションが不足すると、Hinemosマネージャのログファイル(/opt/hinemos/var/log/jboss.log)に以下のメッセージが出力されます。

```
javax.resource.ResourceException:  
No ManagedConnections available within configured blocking timeout ( 30000 [ms] )
```

このメッセージは、コネクション不足により一部の処理が異常終了していることを示します。この場合、以下の設定値を変更して、プールされるPostgreSQLのコネクション数を変更する必要があります。(Hinemos ver.4.0.0のデフォルト値を例として記載します)

1. PostgreSQLが受け付ける最大接続数の変更

/opt/hinemos/etc/postgresql/postgresql.conf に定義されている max_connections の値を変更します。

```
max_connections = 200 # (change requires restart)
```

2. コネクションプール数の変更

/opt/hinemos/etc/jboss/hinemos-ds.xml, /opt/hinemos/etc/jboss/scheduler-ds.xml に定義されている min-pool-size, max-pool-size の値を変更します。

```
<max-pool-size>60</max-pool-size>
```

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。

4.3 Javaヒープメモリの設定

Hinemosマネージャが利用するJavaヒープメモリのサイズについては、下記の設定ファイルで変更できます。

/opt/hinemos/etc/jboss/run.conf

```
#### Hinemos : For Tune Heap Memory (default : Xmx=512m, Large : Xmx=1024m)  
#JAVA_OPTS="${JAVA_OPTS} -Xms1024m -Xmx1024m -XX:NewSize=320m -XX:MaxNewSize=320m -XX:MaxPermSize=256m"
```

```
JAVA_OPTS="${JAVA_OPTS} -Xms512m -Xmx512m -XX:NewSize=160m -XX:MaxNewSize=160m -XX:MaxPermSize=192m"
JAVA_OPTS="${JAVA_OPTS} -XX:CMSInitiatingOccupancyFraction=75 -XX:+UseCMSInitiatingOccupancyOnly"
```

上記のコメントアウトの付け外しで設定を切り替えることができます。管理対象ノードが100台未満の場合はデフォルト値（512m）を、管理対象ノードが100台以上の場合は1024mを指定することを推奨します。

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。

4.4 OSロケールと文字コード設定

4.4.1 Hinemosクライアント

Hinemosクライアントで日本語を表示する場合、HinemosクライアントをインストールするWindows OSのOSロケールを日本語とする必要があります。Hinemosクライアントは文字コードがMS932であることを前提として動作します。また、接続するHinemosマネージャが日本語環境としてセットアップされている必要があります。

4.4.2 Hinemosマネージャ

Hinemosマネージャで日本語を扱う場合には、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マネージャのインストール時に使用するインストールスクリプトをmanager_installer_JP.shとする必要があります。manager_uninstaller_EN.shを利用してインストールした場合、一度アンインストールして、manager_installer_JP.shを利用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であることを前提に動作します。本設定は、/opt/hinemos/hinemos.cfgに記載されています。

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

4.4.3 Hinemosエージェント

各種プラットフォーム共通

- ・ ジョブの標準出力・標準出力エラーの文字コード

ジョブの実行結果には、ジョブの起動コマンドとして実行されたプロセスの標準出力・標準出力エラーが含まれます。Hinemosエージェントでは、この標準出力・標準出力エラーの文字コードを指定できます。なお、ジョブ単位ではなく、Hinemosエージェントプロセス単位での指定となります。

文字コードは、Agent.propertiesのinput.encodingパラメータにて指定されます。選択可能なパラメータはUTF-8, EUC-JP, MS932です。input.encodingパラメータが定義されていない（デフォルト）の場合、OSシステムロケールに従います。

指定された文字コードとしてデコード可能なバイトシーケンス以外が読み込まれた場合、特定のコードポイント（uFFFF）に置換されます。

標準出力・標準出力エラーの文字コードをEUC-JPとする設定例を以下に示します。

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

設定を変更した場合、Hinemosエージェントの再起動が必要となります。

- ・ 監視対象となるログファイルの文字コード

ログファイル監視機能の対象となるログファイルの文字コードを指定できます。指定可能なパラメータは UTF-8, EUC-JP, MS932 です。（ログファイル単位ではなく、Hinemosエージェントプロセス単位での指定となります）

文字コードは、Agent.properties の log.file.encoding パラメータにて定義可能です。log.file.encoding が定義されていない場合（デフォルト）の場合、OSシステムロケールに従います。

指定された文字コードとしてデコード可能なバイトシーケンス以外が読み込まれた場合、特定のコードポイント（uFFFD）に置換されます。

標準出力・標準出力エラーの文字コードをMS932とする設定例を以下に示します。

```
## character encoding of log file
log.file.encoding=MS932
```

設定を変更した場合、Hinemosエージェントの再起動が必要となります。

- ・ 監視対象となるOSシステムログの文字コード

Hinemosマネージャは、受信するsyslogの文字コードがUTF-8であることを前提として動作します。このため、管理対象のノードからHinemosマネージャに対してsyslogを送信する場合、ASCII文字のみ、あるいはUTF-8（マルチバイト文字を含む場合）として送信する必要があります。

Windows版エージェントのみ

- ・ Windows イベントの監視結果の文字コード

管理対象ノード上で出力されたWindowsイベントは、NTSyslogによってsyslogに変換され、Hinemosマネージャに送信されます。このとき、Windowsイベントに含まれるメッセージはUTF-8に変換されます。

4.5 HinemosエージェントのファシリティIDの設定

HinemosエージェントはIPアドレスとホスト名から、自身のファシリティIDを特定します。しかし、NATなどを利用してIPアドレスが変換されてしまう環境等では、この機能は正常に動作しません。

そのような場合は、Hinemosエージェントの設定ファイル(Agent.properties)に下記の行を追加する事により、ファシリティIDを設定する事が可能です。下記はファシリティIDを『node01』に設定する例となります。

```
facilityId=node01
```

設定変更を反映させるには、Hinemosエージェントの再起動が必要です。

5 通知機能

Hinemos 通知機能の追加設定について説明します。

5.1 イベント通知

5.1.1 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2000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ます。

```
## イベントの最大ダウンロード数
monitor.common.report.event.count = 2000
```

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。

ただし、`monitor.common.report.event.count` パラメータの値を大きくした場合、Hinemosマネージャのメモリが不足する可能性があります。このため、基本的にはデフォルト値から変更しないことを推奨します。

5.2 メール通知

5.2.1 メール通知を有効にする

メール通知を利用するには、メールサーバと送信元情報について設定していただく必要があります。

メールサーバの設定

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。

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

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- $Id: mail-service.xml 62349 2007-04-15 16:48:15Z dimitris@jboss.org $ -->
<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">
      <!-- A test configuration -->
      <configuration>

        (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"/>

        (中略)

      </configuration>
    </attribute>
  </mbean>
</server>
```

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

</server>
```

以下のパラメータを設定します。

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

<!-- Change to the address mail will be from -->
<property name="mail.from" value="(送信元メールアドレス)"/>
```

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。

送信元情報の設定

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。

/opt/hinemos/etc/hinemos.properties

```
## Notify (Mail) : Mail Header
common.mail.from.address=admin@hinemos.com
common.mail.from.personal.name=Hinemos Admin
common.mail.reply.to.address=admin@hinemos.com
common.mail.reply.personal.name=Hinemos Admin
common.mail.errors.to.address=admin@hinemos.com

## Notify (Mail) : Mail Resending
common.mail.retry.count=0
common.mail.retry.sleep=10000
```

以下のパラメータを設定します。

- 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
- common.mail.retry.count=メール送信に失敗した場合の再送回数
- common.mail.retry.sleep=メール送信に失敗した後に、再送するまでに待機する時間(ミリ秒)

これらのパラメータについては、RFC822に準拠したフォーマットで指定して下さい。

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。

5.2.2 SMTP AUTHの設定

メール通知において、SMTP AUTHを必要とするSMTPサーバを利用するには、以下のファイルを編集します。

/opt/hinemos/etc/jboss/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>
```

```
(中略)

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

(中略)

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

以下のパラメータを設定します。

- <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) "/>

認証方法としては、LOGIN、PLAIN、DIGEST-MD5に対応しています。また、複数の認証方法が有効になっているSMTPサーバに送信する場合、LOGIN、PLAIN、DEGEST-MD5の順で選択されます。

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。

5.2.3 SSH/TLSの設定

メール通知において、SSH/TLS(STARTTLS)を必要とするSMTPサーバを利用するには、以下のファイルを編集します。

/opt/hinemos/etc/jboss/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>

(中略)

    <property name="mail.smtp.starttls.enable" value="true"/>

(中略)

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

以下のパラメータを設定します。

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

以下のパラメータを<configuration>タグ内に追記します。

- <property name="mail.smtp.starttls.enable" value=" (STARTTLSを有効にする場合はtrue/無効にする場合はfalse) "/>

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。

5.2.4 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には、以下のファイルを編集します。

/opt/hinemos/etc/jboss/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>

(中略)

    <property name="mail.smtp.from" value="admin@hinemos.com">

(中略)

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

以下のパラメータを<configuration>タグ内に追記します。

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

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。

5.2.5 ポート番号の指定

メール通知において、使用するSMTPサーバのポート番号（デフォルトは25）を変更するには、以下のファイルを編集します。

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

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- $Id: mail-service.xml 62349 2007-04-15 16:48:15Z dimitris@jboss.org $ -->
<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">
    <!-- A test configuration -->
    <configuration>

(中略)

    <!-- The mail server port -->
    <property name="mail.smtp.port" value="25"/>

(中略)
```



```

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

</server>

```

以下のパラメータを設定します。

```

<!-- The mail server port -->
<property name="mail.smtp.port" value="(SMTPサーバのポート番号)"/>

```

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。

5.3 ログエスケレーション通知

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。

5.3.1 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、Hinemosマネージャが送信するsyslog(RFC3164)のHEADER部に埋め込むホスト名を、柔軟に指定できます。

`/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 ¹ 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 ² 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 (その他の場合) 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	<PRI> Mmm dd hh:mm:ss nodename message...

¹ 登録ノードすべて、未登録ノード、Hinemos内部スコープ

² hostnameコマンドの実行結果

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。

6 一括制御機能

Hinemos 一括制御機能の追加設定について説明します。

6.1 一括制御機能を有効にする

一括制御機能は利用するには、Hinemosインストール後、マネージャサーバと管理対象ノードのOS環境のセットアップを実施する必要があります。

6.1.1 管理対象ノードのセットアップ

Hinemosで一括制御機能を利用するためには、マネージャサーバからsshまたはrshで管理対象ノードに対してコマンド実行できる必要があります。以下で一括制御機能のセットアップ方法について説明します。

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、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、本設定は不要です。

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）に対して、パスワードなしでコマンド実行できるように設定します。以下で、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。

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

```
(root) # cd /opt/hinemos/sbin/  
(root) # ./hinemos_setup_collectiverun.sh
```

メニューが表示されますので、プロンプトに"1"を入力します。

```
(root) # /opt/hinemos/sbin/hinemos_setup_collectiverun.sh  
The setup of the node for the watch to user root begins.  
  
-----  
Welcome to the Setup of the collective run function  
Hinemos Version 4.0.x  
  
Copyright (C) since 2006 NTT DATA Corporation  
-----  
  
the Setup of the collective run function  
  
1) Making of public key for attestation of root user  
2) Public key registration to node for monitoring  
9) Setup end of collective run function  
  
====> 1
```

2. root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。

```
The public key for the attestation of the root user is made without the passphrase.
Is it good? (Y/N default:Y)

Y
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
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:
xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx root@agent01
The key's randomart image is:
+--[ RSA 2048]-----+
|          oo o       |
|        (中略)      |
|          +         |
+-----+

The public key was made.
```

3. プロンプトに"2"を入力します。

```
the Setup of the collective run function

1) Making of public key for attestation of root user
2) Public key registration to node for monitoring
9) Setup end of collective run function

===> 2
```

4. 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管理対象ノードのIPアドレスを入力します。

```
root user's public key is registered in the root user's with node for the authorized_keys file.

Please input IP address of the node that uses the collective run function for the monitoring.
Please input 9 when ending.
192.168.0.10
```

5. 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。

```
Please specify the home directory of the root user with the node for the monitoring.
/root/
```

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、"Y"を入力します。

```
root user's public key is set to /root/.ssh/authorized_keys.
Is it good? (Y/N default:Y)
Y
```

7. 管理対象ノードの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。

```
root@192.168.0.10's password: (パスワードを入力)

Registration to 192.168.0.10 was completed.
```

8. 終了する場合は"9"を入力します。続けて登録する場合は管理対象ノードのIPアドレスを入力します。

```
Please input IP address of the node that uses the collective run function for the monitoring.
Please input 9 when ending.
```

```
9
```

```
the Setup of the collective run function
```

- 1) Making of public key for attestation of root user
- 2) Public key registration to node for monitoring
- 9) Setup end of collective run function

```
====> 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から変更したい場合は、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ます。

```
(root) # cd /opt/hinemos/sbin/
(root) # ./hinemos_setup_collectiverun.sh hoge
```

rshを使用する場合

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、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、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 ファイルを作成します。既に.rhostsファイルが存在する場合は、下記の内容を追記します。

```
/root/.rhosts
```

```
(マネージャサーバのIPアドレス) hinemos
```

例)

```
192.168.0.1 hinemos
```

2. /etc/securettyファイルに、rshを追加します。

```
/etc/securetty
```

例)

```
console
vc/1
```

(中略)

```
tty10
tty11
rsh
```

以上でrshの設定は終了です。

6.1.2 Hinemosマネージャ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、管理対象ノードでコマンド実行するユーザをrootから変更する場合、collective.run.userを編集します。ただし、collective.run.userをrootから変更すると、一括制御機能でroot権限が必要なコマンドを実行できなくなりますのでご注意ください。

```
/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
```

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。

6.1.3 FTPサーバの起動

一括制御機能の『RPMインストール』や『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サーバを準備する必要があります。事前に、FTPサーバを起動し、インストール時に指定した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。

以下にて、FTPサーバとしてRed Hat Enterprise Linux 6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とkrb5-workstation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)、krb5-workstation- (バージョン) が表示されることを確認します。

```
(root) # rpm -q vsftpd
vsftpd-2.2.2-6.el6.i686
(root) # rpm -q krb5-workstation
krb5-workstation-1.9-22.el6.i686
```

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。

```
(root) # service vsftpd start
```

・ファイル転送元、転送先の設定

krb5-workstation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。krb5-workstation- (バージョン) が表示されることを確認します。

```
(root) # rpm -q krb5-workstation
krb5-workstation-1.9-22.el6.i686
```

・Hinemosで利用するFTPサーバの設定変更

Hinemosで利用するFTPサーバには、Hinemosマネージャのインストール時に指定したものが設定されます。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、パスワードを変更するには、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=" (IPアドレス) "
FTP_USER=" (ユーザ名) "
FTP_PASSWD=" (パスワード) "
```

7 監視設定/性能管理機能

Hinemos 監視設定/性能管理機能の追加設定について説明します。

7.1 SQL監視

7.1.1 監視対象のRDBMSの追加

SQL監視の対象RDBMSを追加する方法を説明します。

HinemosマネージャからJDBC Driver経由でRDBMSに接続します。このため、SQL監視の対象RDBMSを追加するには、そのRDBMSに対応したJDBC Driverを用意する必要があります。（Java Runtime Environment 6.0で動作するJDBC Driverとなります）

- ・ 追加手順

1. JDBC Driverの配置

用意したJDBC Driverを /opt/hinemos/jboss-as/server/hinemos/lib/ に配置します。

```
$ cp new_jdbc.jar /opt/hinemos/jboss-as/server/hinemos/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のSQL監視で利用できるRDBMSの種類の数 jdbc.driver の数を増やします。（追加するRDBMSを含めた、jdbc.driver.name.Xの数となります）

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

そして、新たなRDBMSおよびJDBC Driverに関する情報を追加します。

```
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接続時パラメータ}
```

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。

監視設定[一覧]ビューからSQL[作成・変更]ダイアログを開き、追加したRDBMSが『接続先DB』のプルダウンメニューに表示されることを確認します。

7.2 プロセス監視

7.2.1 『値取得の失敗』の通知が発生する場合の対処

プロセス監視では、下記の2つの処理が非同期に実行されます。（図 7-1 参照）

1. SNMPポーリングにより、監視対象ノードのプロセス一覧情報を収集する

2. プロセス一覧情報から監視対象となっているプロセスのプロセス数をカウントする

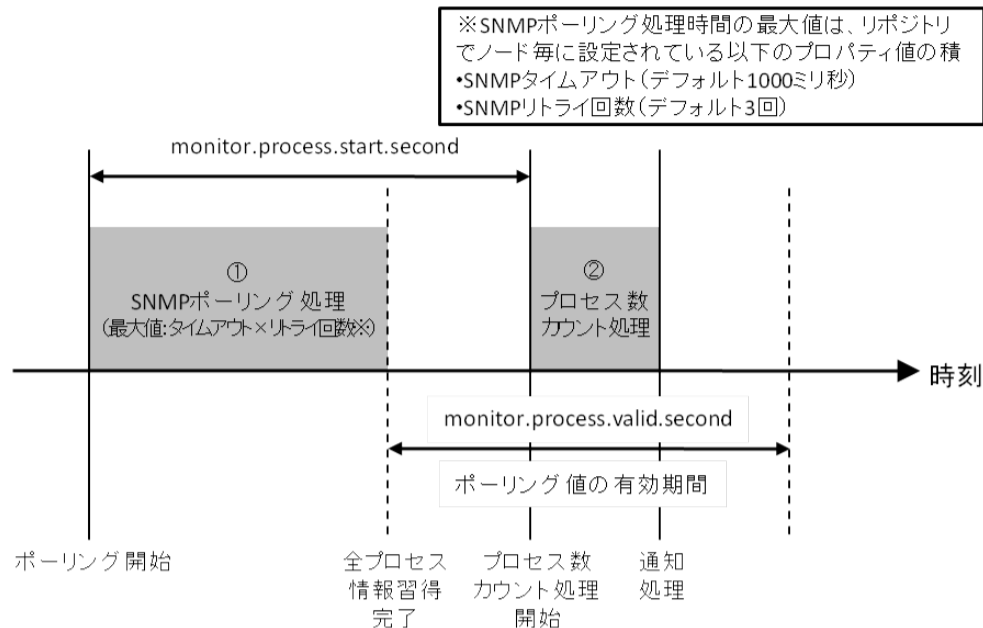


図 7-1 プロセス監視の処理

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で収集したプロセス一覧を有効な情報として扱う秒数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要する時間は、実行環境のSNMPポーリングの応答速度に依存します。SNMPポーリングの所要時間は最大でタイムアウト時間×リトライ回数[秒]となります。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、①の収集結果は『タイムアウトしました』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。

```
/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
```

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。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。

7.3 HTTPS監視

HTTPS監視では、HinemosマネージャがHTTPSクライアントとして監視対象のHTTPSサーバに接続します。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。

7.3.1 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) を準備します。

7.3.2 証明書のkeystoreへの登録

次に、Javaのkeytoolコマンド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監視の対象サーバのサーバ証明書を.keystoreに追加します。

サーバ証明書をマネージャサーバの/tmp配下に配置した例を以下に示します。また、aliasには、サーバ証明書ごとに指定する文字列はユニークな文字列（下記例ではhinemos）を指定します。

```
(root) # su - hinemos
(hinemos) $ /usr/bin/keytool -import -file /tmp/ (サーバ証明書) -alias hinemos -keystore /opt/hinemos/.keystore
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/javase/ja/6/docs/ja/technotes/tools/solaris/keytool.html> を参照して下さい。

7.3.3 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-as/bin/run.confのjavax.net.ssl.trustStoreを設定するJAVA_OPTのコメントを外します。

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

7.4 システムログ監視

7.4.1 syslogの送信に関する設定

管理対象ノードでrsyslogではなくsyslogdを利用する場合、/etc/syslog.confに以下の設定を追加します。

```
*.info;mail.none;authpriv.none;cron.none @【マネージャサーバのIPアドレス】
```

修正を反映するには、syslogサービスの再起動が必要です。

```
(root) # service syslog restart
```

7.4.2 syslogの受信に関する設定

外部機器から送信されたsyslogを受信する際に使用する マネージャサーバのポート番号については、以下の設定ファイルにて変更できます。

/etc/rsyslog_hinemos.conf


```
$UDPServerRun 514
$InputTCPServerRun 514
```

マネージャサーバ内部で、rsyslogからHinemosマネージャ(JBoss)にsyslogを転送する際に使用する待ち受けアドレス、ポート番号、システムログ監視のフィルタリング処理の最大スレッド数について、以下の設定ファイルにて変更できます。

/opt/hinemos/etc/jboss/syslog-service.xml

```
<server>
  <classpath codebase="." archives="HinemosManager.jar"/>

  <mbean code="com.clustercontrol.systemlog.jmx.SyslogReceiveService" name="user:service=SyslogReceiveService,name=SyslogReceiveService">

    (中略)

    <attribute name="Port">24514</attribute>
    <attribute name="Host">0.0.0.0</attribute>

    (中略)

    <attribute name="ThreadPoolSize">1</attribute>

    (中略)

  </mbean>
</server>
```

7.4.3 制御文字に対するエスケープ処理の無効化

マネージャサーバに到達したsyslogは、rsyslogによって受信された後、Hinemosマネージャ(JBoss)に転送されます。

このとき、syslogに制御文字(BELなど)が含まれている場合、rsyslogの機能により、制御コード部分が3桁の8進数(#007など)に置換されます。エスケープされる制御コードの例を表7-1に、エスケープされるログの例を表7-2に示します。

表 7-1 エスケープされる制御コードの例

制御文字	置換後の文字列
NUL	#000
BEL	#007

表 7-2 エスケープされるログの例(制御文字NULを[NUL]として表記する)

rsyslogが受信する文字列	HTTP/1.0[NUL] 50
rsyslogからHinemosマネージャに転送される文字列	HTTP/1.0#000 50
上記とマッチするパターンマッチ表現	.*HTTP/1.0#000 50.*

エスケープ処理を無効化するためには、以下の設定を行います。

1. Hinemosマネージャサーバの/etc/rsyslog_hinemos.confを編集します。
/etc/rsyslog_hinemos.conf に以下の設定を追加します。

```
$EscapeControlCharactersOnReceive off
```

2. rsyslogを再起動します。

```
(root) # service rsyslog restart
```

7.4.4 syslogのHEADER部に対するホスト名置換処理の設定

syslogはRFC3164に従い、PRI・HEADER・MSGで構成されます。

syslogのHEADER部はTIMESTAMP、HOSTNAMEから構成されますが、TIMESTAMPには日付、HOSTNAMEにはホスト名・IPアドレス・空白が格納されます。

syslogパケットのHEADER部(TIMESTAMP HOSTNAME)の例を以下に示します。

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

rsyslogは受信したsyslogのHOSTNAMEが未定義（空白）の場合、IPアドレスを元にホスト名の逆引きを行い、syslogのHEADER部に埋め込みます。

rsyslogによるsyslogのホスト名の置換処理を無効化したい場合、以下の設定を行います。

1. Hinemosマネージャサーバ上の/etc/sysconfig/rsyslogを編集します。
/etc/sysconfig/rsyslog の該当箇所に"-x"オプションを指定します。

```
SYSLOGD_OPTIONS="-c 4 -x"
```

2. rsyslogを再起動します。

```
(root) # service rsyslog restart
```

7.5 SNMPTRAP監視

7.5.1 SNMPTRAPの受信に関する設定

外部機器から送信されたSNMPTRAPを受信する際に使用する マネージャサーバの待ち受けアドレス、ポート番号については、以下の設定ファイルにて変更できます。

/opt/hinemos/etc/jboss/snmptrap-service.xml

```
<server>
  <classpath codebase="." archives="HinemosManager.jar"/>

  <mbean code="com.clustercontrol.snmptrap.jmx.SnmpTrapService" name="user:service=SnmpTrapService,name=SnmpTrap">
    (中略)

    <attribute name="Port">162</attribute>
    <attribute name="Host">0.0.0.0</attribute>

  </mbean>
</server>
```

7.6 Windowsサービス監視

7.6.1 WinRMのインストール

Windowsサービス監視の対象ノードには、Windows管理フレームワーク WinRM 1.1 または WinRM 2.0がインストールされている必要があります。なお、監視対象ノードのOSがWindows Server 2008 R2の場合、WinRM 2.0がデフォルトでインストールされていますので、明示的にインストールする必要はありません。

WinRMの詳細情報やダウンロード、インストール方法については、Microsoftサポートサイトをご参照ください。

7.6.2 リモートコンピュータ管理の許可

監視対象ノードのコマンドプロンプトから下記コマンドを実行し、リモートコンピュータ管理を許可します。

```
> winrm quickconfig
WinRM は、管理用にこのコンピュータへのリモート アクセスを許可するように設定されていません。
次の変更を行う必要があります。
```

(中略)

```
変更しますか [y / n]? y
```

```
WinRM はリモート管理用に更新されました。
```

下記コマンドを実行し、WinRMのHTTP/HTTPSのポートを確認します。なお、WinRM 1.1のデフォルトでは、HTTP 80ポート、HTTPS 443ポートです。また、WinRM 2.0のデフォルトでは、HTTP 5985ポート、HTTPS 5986ポートです。

```
> winrm get winrm/config
```

下記コマンドを実行し、WinRMのHTTP/HTTPSのポートをLISTENしていることを確認します。

```
> netstat -an
```

7.6.3 Basic認証の許可

下記のコマンドを実行し、Basic認証を許可します。

```
> winrm set winrm/config/service/auth @{Basic="true"}
Auth
  Basic = true
```

7.6.4 非暗号通信の許可

下記コマンドを実行し、HTTPで非暗号通信を許可します。

```
> winrm set winrm/config/service @{AllowUnencrypted="true"}
Service
  AllowUnencrypted = true
```

7.6.5 HTTPSの設定

Windowsサービス監視でHTTPSを利用する場合、下記の手順が必要です。

1. 証明書の準備

WinRMのHTTPS接続で利用する証明書を用意します。

2. WinRMへの証明書の登録

以下のコマンドを実行します。

```
> winrm create winrm/config/Listener?Address=*+Transport=HTTPS @{Hostname="【IPアドレス】";
CertificateThumbprint="【証明書の捺印(16進数)"]}
```

3. Hinemosマネージャのkeystoreへの証明書の登録

[証明書のkeystoreへの登録](#)、[Java起動オプションによるkeystoreファイルの指定](#)を参照し、Hinemosマネージャのkeystoreに証明書を登録します。

4. Hinemosマネージャの再起動

7.6.6 接続先OSユーザの準備

WinRM へのリモートアクセスには、Windows ServerのOSユーザーを利用します。このため、監視対象ノードに、Hinemosによるリモートアクセス用のOSユーザーを用意する必要があります。

※このユーザーをAdministratorsグループに所属させる必要があります。

該当ユーザのWinRMのアクセス権限を設定するには、以下のコマンドを実行します。

- WinRM 1.1の場合

```
> winrm configSDDL
```

- WinRM 2.0の場合

```
> winrm configSDDL default
```

上記コマンドを実行すると、アクセス許可設定ダイアログが表示されます。用意したOSユーザーを選択し、アクセス権限を設定します。なお、Windowsサービス監視では読み取り権限が必要となります。

7.6.7 疎通確認

マネージャサーバにて下記コマンドを実行し、監視対象ノードと疎通できることを確認します。

```
(root) # wsmn -u 【接続先OSユーザー名】 -p 【パスワード】 -y basic -h 【IPアドレス】 -P 【HTTP/HTTPSのデフォルトポート】  
-d 6 enumerate http://schemas.microsoft.com/wbem/wsmn/1/wmi/root/cimv2/Win32_Service  
(root) # wsmn -u 【接続先OSユーザー名】 -p 【パスワード】 -y basic -h 【IPアドレス】 -P 【HTTP/HTTPSのデフォルトポート】  
-d 6 get http://schemas.microsoft.com/wbem/wsmn/1/wmi/root/cimv2/Win32_Service?Name=wudfsvc
```

7.7 カスタム監視

7.7.1 コマンドの動作変更

カスタム監視で設定したコマンドは、Hinemosエージェントによって実行されます。

Hinemosエージェントは、OSプラットフォームを自動的に識別し、OSプラットフォームに合わせて、コマンドの動作を切り替えます。

OSプラットフォームの識別方法は、下記の設定ファイルのcollector.command.modeパラメータで変更できます。なお、collector.command.modeパラメータのデフォルト値は"auto"となっております。

- /opt/hinemos_agent/conf/Agent.properties (Linux版エージェント)
- [Hinemosエージェントのインストールディレクトリ]\conf\Agent.properties (Windows版エージェント)

```
collector.command.mode=auto
```

カスタム監視のcollector.command.modeパラメータは、Hinemosエージェントのcommand.create.modeパラメータと同様に動作します。collector.command.modeパラメータに設定可能な値や、OSプラットフォームによる動作差分については、Hinemosエージェントの [起動コマンドの動作変更](#) を参照してください。

設定変更を反映させるには、Hinemosエージェントの再起動が必要です。

7.7.2 標準出力からの読み込み最大量の設定

カスタム監視で取得する値は、コマンドの標準出力から抽出されます。

ただし、コマンドから予期せずに大量の情報が標準出力に出力された場合、Hinemosエージェントのメモリ不足になり、機能不全が引き起こされる可能性があります。

このような機能不全を防ぐために、Hinemosエージェントでは標準出力から読み込む最大サイズが定義されています。以下の設定値により、この読み込み最大サイズ(デフォルト値は512[byte])を変更できます。

- /opt/hinemos_agent/conf/Agent.properties (Linux版エージェント)
- [Hinemosエージェントのインストールディレクトリ]\conf\Agent.properties (Windows版エージェント)

```
collector.command.buffer=512
```

設定変更を反映させるには、Hinemosエージェントの再起動が必要です。

7.7.3 標準出力に含まれる改行コードの設定

カスタム監視は、コマンドの標準出力の1行を単位として、1つの監視の対象とする値を抽出します。

以下の設定値により、行を識別する上で利用される改行コードを変更できます。 デフォルトでは、Linux版エージェントではLF、Windows版エージェントではCRLFとなっております。

- /opt/hinemos_agent/conf/Agent.properties (Linux版エージェント)
- [Hinemosエージェントのインストールディレクトリ]\conf\Agent.properties (Windows版エージェント)

```
collector.command.returncode=LF
```

設定を変更した後、Hinemosエージェントを再起動してください。

7.7.4 コマンドの実行多重度の設定

カスタム監視によるコマンド実行用のスレッドプールがHinemosエージェント内に設けられています。

コマンドを実行する度、スレッドプール内から利用されていないスレッドが割り当てられて、コマンドの実行終了あるいはタイムアウトした後、利用したスレッドをスレッドプールに返却します。

以下の設定値により、スレッドプールに用意するスレッド数（デフォルトは8スレッド）を定義することが可能です。

- /opt/hinemos_agent/conf/Agent.properties (Linux版エージェント)
- [Hinemosエージェントのインストールディレクトリ]\conf\Agent.properties (Windows版エージェント)

```
collector.command.workerthreads=8
```

同一の管理対象に対して大量のカスタム監視設定が割り当てられており、コマンドの実行タイミングが遅延している場合、この設定値を調整してください。

7.8 ポーリングプロトコルの設定

性能管理機能や性能管理機能（リソース監視、プロセス監視、SNMP監視）では、Hinemosマネージャから管理対象に対してSNMPやWBEMによるポーリングを行い、各機能が必要とする情報を取得しています。

そのため、上記の機能を利用する場合は、管理対象ノード側でHinemosマネージャからのSNMPやWBEMのポーリングに 応答可能に設定されている必要があります。

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。

管理対象ノードとする各機器の仕様に従って、Hinemosマネージャからのポーリングに 応答可能なように設定してください。

7.8.1 Net-SNMPの設定

Linux版エージェントが導入されている管理対象では、Hinemosエージェントのインストーラの実行中に、以下の設定がsnmpd.confに追記されます。

/etc/snmp/snmpd.conf

```
view systemview included .1.3.6.1
```

7.8.2 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）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）単位でポーリング手段(SNMPとWBEM)を切り替えられます。なお、変更する際には [データベースのバックアップ](#) に従って、バックアップを取得しておくことを推奨します。

Hinemosマネージャを停止し、rootユーザで以下のコマンドを実行します。パスワード入力を求められますので、PostgreSQLのログインパスワード(初期パスワードは"hinemos")を入力します。

```
(root) # /opt/hinemos/bin/pg_start.sh

(root) # /opt/hinemos/postgresql/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 for user hinemos:

(root) # /opt/hinemos/bin/pg_stop.sh
```

「変更したいプロトコル」部分には、「SNMP」または「WBEM」を入力してください（デフォルト値はSNMPとなっています）。「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

(注) WBEMでは一部取得できない値が存在します。また、WBEMで監視可能な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を切り替えられます。

Hinemosマネージャを停止し、以下のコマンドを実行します。パスワード入力を求められますので、PostgreSQLのログインパスワード(初期パスワードは"hinemos")を入力します。

```
(root) # /opt/hinemos/bin/pg_start.sh

(root) # /opt/hinemos/postgresql/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 for user hinemos:

(root) # /opt/hinemos/bin/pg_stop.sh
```

「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）。

7.9 数値監視の収集値の設定

監視設定の削除時の動作として、数値監視に対応する収集値を削除するかどうかを定義できます。必要に応じて、以下の設定ファイルを変更してください。

onとした場合、監視設定の削除と同時に収集値も削除されます。offとした場合、監視設定を削除しても収集値は削除されません。

- /opt/hinemos/etc/hinemos.properties

```
monitor.common.delete.cascade.perfdata=off
```

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。

数値監視の収集値を削除するには、メンテナンス機能の履歴情報削除機能を使用します。なお、数値監視の収集値は監視項目IDに紐づいて管理されます。このため、過去に存在した監視設定と同一の監視項目IDで新規の監視設定を作成し、数値監視の収集値をグラフ表示・ダウンロードすると、意図せず過去の監視設定で蓄積したデータがグラフ表示・ダウンロードされる場合がありますのでご注意ください。

8 ジョブ管理機能

Hinemosジョブ管理機能の追加設定について説明します。

8.1 起動コマンドの動作変更

Hinemosエージェントは、OSプラットフォームを自動的に識別し、OSプラットフォームに合わせて、ジョブの起動コマンドの動作を切り替えます。

OSプラットフォームの識別方法は、下記の設定ファイルのcommand.create.modeパラメータで変更できます。なお、command.create.modeパラメータのデフォルト値は"auto"となっております。

- /opt/hinemos_agent/conf/Agent.properties (Linux版エージェント)
- [Hinemosエージェントのインストールディレクトリ]\conf\Agent.properties (Windows版エージェント)

```
command.create.mode=auto
```

command.create.modeパラメータに指定できる値の一覧を表8-1に示します。

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	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

起動コマンドを echo XXX とした場合の動作例を以下に示します。

- OSプラットフォームがWindowsと識別された場合

```
Hinemosエージェントの起動ユーザと実効ユーザが同じ場合 :
コマンド : CMD
第1引数 : /C
第2引数 : echo XXX
```

```
Hinemosエージェントの起動ユーザと実効ユーザが異なる場合 :
実行できません
```

- OSプラットフォームがLinuxと識別された場合

```
Hinemosエージェントの起動ユーザと実効ユーザが同じ場合 :
コマンド : sh
第1引数 : -c
第2引数 : echo XXX
```

```
Hinemosエージェントの起動ユーザと実効ユーザが異なる場合 :
sudo -u [実効ユーザ] sh -c [起動コマンド]
コマンド : sudo
第1引数 : -u
第2引数 : [実効ユーザ]
第3引数 : sh
第4引数 : -c
第5引数 : echo XXX
```


- 起動コマンドをcompatibleモードで実行する場合

Hinemosエージェントの起動ユーザと実効ユーザが同じ場合：
コマンド：echo
第1引数：XXX
(半角スペースを区切り文字として、コマンドや引数が分割されます)

Hinemosエージェントの起動ユーザと実効ユーザが異なる場合：
コマンド：su
第1引数：[実効ユーザ]
第2引数：-c
第3引数：echo XXX

設定を変更した後、Hinemosエージェントの再起動が必要です。

8.2 Hinemosマネージャ再起動時のジョブスケジュール制御の設定

Hinemosマネージャを起動した際、Hinemosマネージャを停止している間に実行予定時刻を過ぎたジョブスケジュールは、以下の動作となります。

- 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 Hinemosマネージャ起動直後に、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て、次回実行予定時刻に実行されます。

[データベースのリストア](#) によって過去のバックアップからリストアした場合でも、そのバックアップ時点からHinemosマネージャが停止していたと扱われて、同様の動作となります。

詳細は、Hinemosユーザマニュアルを参照してください。

なお、起動失敗と判定するまでの時間の閾値は、以下のパラメータで設定できます。 msecを単位として定義してください。

- /opt/hinemos/etc/jboss/schedule-dbms-service.xml

```
org.quartz.jobStore.misfireThreshold = 3600000
```

なお、上記のパラメータは、ジョブスケジュール実行だけでなく、監視の定期実行スケジュールにも影響を与えるため、デフォルト値の360000より小さくすることは推奨されていません。

8.3 ファイル転送ジョブを有効にする

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。（一括制御機能からも設定可能です。詳しくはHinemosユーザマニュアルを参照してください）

なお、ファイル転送ジョブでは内部でsshを利用しておりますが、Windows OSではsshの機能を提供されていないため、Windows版エージェントではファイル転送ジョブは利用できません。（詳しくは、Hinemosユーザマニュアルを参照してください）

一括制御機能を利用しない、または、一般ユーザでファイル転送ジョブを実行したい場合は、以下の手順通りに設定を行ってください。

- 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。

- 転送元ノード：ファイルの転送元となるノード(scpcmdによりログインされるサーバ)
- 転送先ノード：ファイルの転送先となるノード(scpcmdが実行されるサーバ)
- 転送するユーザ：ファイル転送ジョブの実効ユーザ(scpcmdを実行するユーザ)

※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) $ cat id_rsa.pub  
ssh-rsa ****(中略)***** = hinemos@agent02  
(hinemos@agent02) $
```

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) $ su -  
Password:  
(root@agent02) # vi /opt/hinemos_agent/lib/job_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 (既に存在す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)

(転送するユーザ) .authorized.keys.path= (転送元ノードのauthorized_keysファイルのパス)

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) $ touch authorized_keys  
(hinemos@agent01) $ 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) $ su -  
Password:  
(root@agent01) # vi /opt/hinemos_agent/lib/job_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（既に存在す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）

設定した後、転送先となるHinemosエージェントを再起動します。

9 その他の機能

Hinemos のその他の機能の追加設定について説明します。

9.1 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
 - ・ 実行中のジョブセッション数の増大
 - ・ 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、デフォルトで監視[イベント]ビューのHinemos内部スコープ(INTERNAL)に通知されます。また、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。なお、Hinemos 内部スコープ(INTERNAL)に通知される情報は以下の2種類があります。

- ・ INTERNAL イベント
- ・ セルフチェック機能からの通知情報

(INTERNAL イベントについての詳細は「Hinemos ユーザマニュアル 5.6.6 INTERNAL イベント」をご参照ください。)

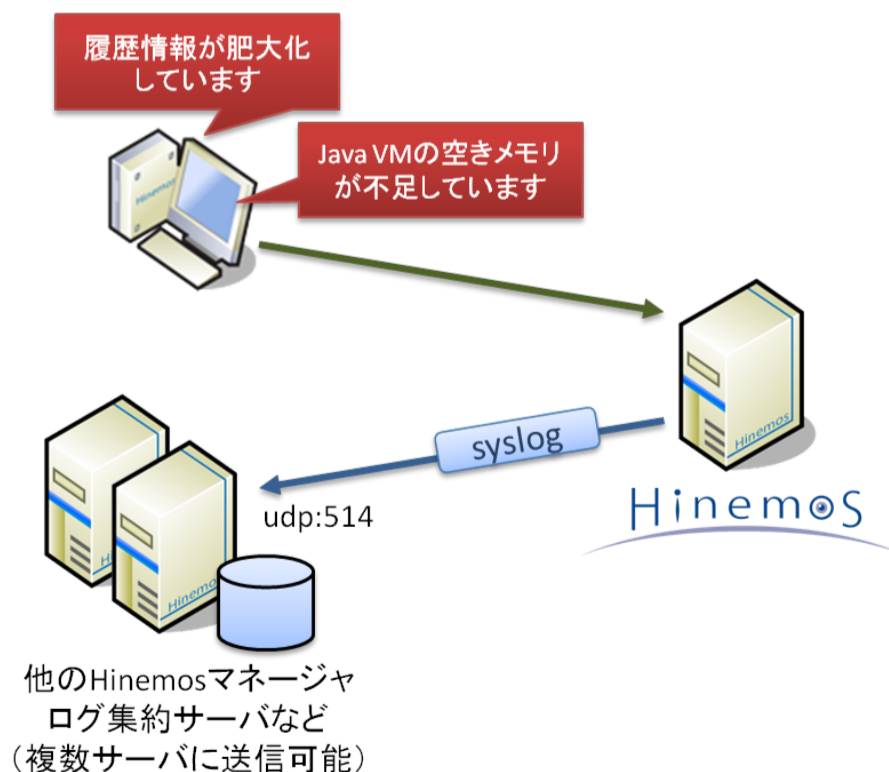


図9-1. セルフチェック機能の概要

9.1.1 通知される情報の一覧

表9-1. 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	データベースへのアクセス不能	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 ³	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 ⁴	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 ⁵	001	警告
	履歴情報の容量増大	SYS_DBSIZE_テーブル名_閾値種別 ⁶	001	警告
	実行中のジョブセッション数の増大	SYS_JOB_SESSION	001	警告
	スワップアウトの発生	SYS_SWAPOUT	001	警告

³ スケジューラ名(Quartz, QuartzRAM)

⁴ /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, clustercontrolNotifyEvent, clustercontrolNotifyJob, clustercontrolNotifyLogEscalation, clustercontrolNotifyMail, clustercontrolNotifyStatus, clustercontrolVmControlExecute, clustercontrolVmControlStatus), JMS TOPICの識別子(clustercontrolJobManagementExecute, clustercontrolLogtransferUpdate, clustercontrolRepositoryUpdate)

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9-2. 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、定期的に実行される各処理 (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_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_JOB_SESSION	001	実行中のジョブセッション数が多く({0} > 閾値 {1})存在します。 パフォーマンス低下の要因となるため、不要に実行中のままとなっている ジョブセッションを停止(「変更済み」または「終了」)させてください。
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9-3. 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_スケジュール名	001	scheduler (スケジュール名) has not been running for 300 [sec] (since 最終動作日時).
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_JOB_SESSION	001	job run session count is too large (XXX > threshold YYY).
SYS_SWAPOUT	001	ram swap-out(XXX [blocks]) occurred since 最終確認日時.

9.1.2 セルフチェック機能の設定

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/jboss/selfcheck-service.xml

表9-4. 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	下限値	上限値
IntervalSec	内部コンポーネントの状態確認間隔[sec]	150	1	2147483647
AlertThreshold	通知するまでの異常の検知回数	3	1	2147483647
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	—	—
Mail	メールとして通知する場合はtrue, 通知しない場合はfalseとする。 (SMTPサーバをmail-service.xmlにて指定済であることを前提とする)	false	—	—
MailAddress	送信先とするメールアドレスを入力する。		—	—
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	1	2147483647
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	1	2147483647
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	0 (注)	2147483647 (注)
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)	環境依存	—	—
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)	環境依存	—	—
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	0	100
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	0	100
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	0 (注)	2147483647 (注)
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	0 (注)	2147483647 (注)
	なお、この値を変更する場合は、 /opt/hinemos/etc/jboss/destinations-service.xml の設定値 MaxSizeを この最大メッセージ数よりも大きな値とする必要がある。			
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]	5120	0	9223372036 854775807
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	0	9223372036 854775807
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]	100000	0	9223372036 854775807
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	0	9223372036 854775807
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	—	—
JobMonitoring	SYS_JOB_SESSIONを有効とする場合はtrue, 無効とする場合はfalseとする。	true	—	—
ThresholdJobRunSession	実行中のジョブセッション数の増大を判定する閾値	100	0	9223372036 854775807

(注)デフォルト値から変更することは推奨されておりません。

9.2 INTERNALイベント

9.2.1 出力先の設定

Hinemos 内部で発生するイベント(INTERNALイベント)の出力先を変更できます。INTERNALイベントの設定可能な出力先は、Syslog サーバ、監視[イベント]ビュー、HinemosApl.logの3つです。/opt/hinemos/etc/hinemos.properties で、出力先を設定できます。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送信の有効無効および送信先のSyslog サーバを設定します。

```
## 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。なお、Hinemos内部スコープ(INTERNAL)に通知される情報は以下の2種類があります。

- INTERNALイベント
- セルフチェック機能からの通知情報

(セルフチェック機能からの通知情報については [セルフチェック機能](#) をご参照ください。)

```
## 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 パラメータにより、ログファイル（HinemosApl.log）への出力の有効無効を設定します。

```
## 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
```

HinemosApl.logへのログ出力はデフォルトで無効となっております。有効化するには、/opt/hinemos/etc/jboss/jboss-log4j.xml の Hinemos AplLog category のコメントアウトを外します。

```
<!-- Hinemos AplLog category for internal log -->
<!--
<category name="hinemos.apllog">
  <priority value="INFO" />
  <appender-ref ref="HINEMOS_APL_LOG"/>
</category>
-->
```

9.3 Hinemosマネージャ死活検知

Hinemosクライアントから、Hinemosマネージャの死活を検知できます。Hinemosクライアントは、Hinemosマネージャに対して定期的に接続を試み、応答がない場合はHinemosマネージャの障害として、図 9-2のようなダイアログを表示します。



図 9-2 Hinemosマネージャ死活検知

Hinemosマネージャの死活監視の間隔は、メニュー・バーの『クライアント設定』→『設定』から表示される『設定』ダイアログの設定値『マネージャ死活監視間隔(分)』より設定できます。(図 9-3 参照)

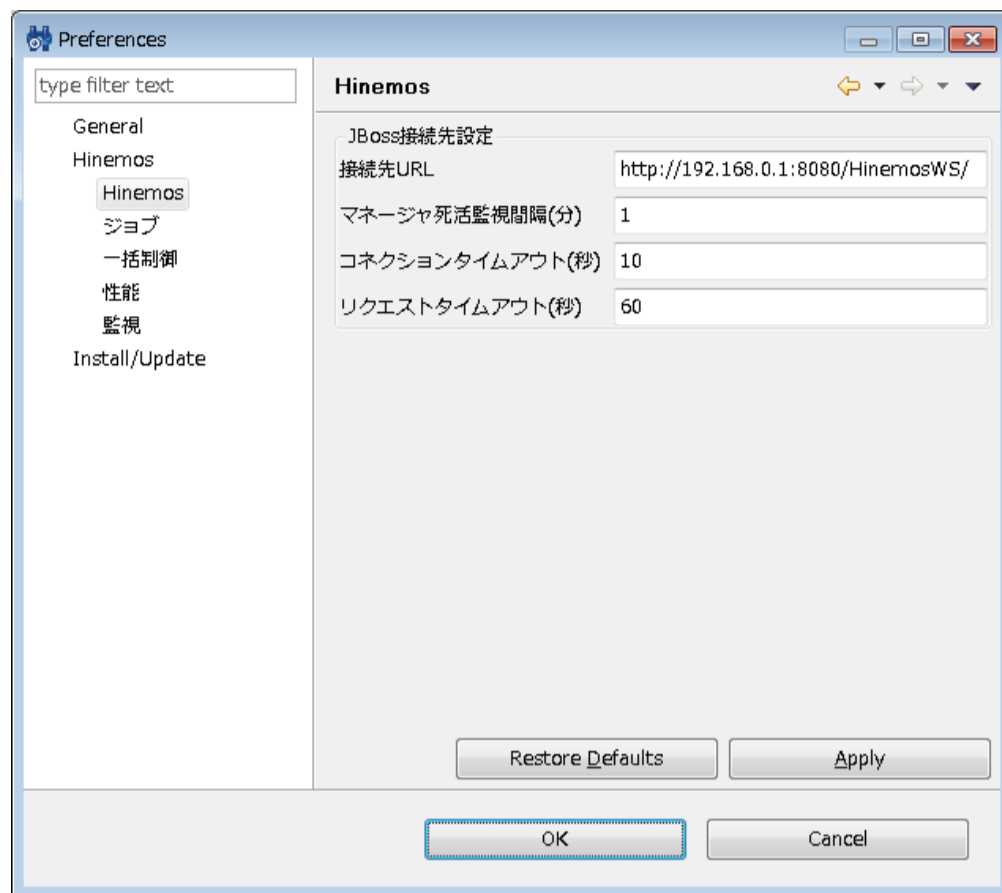


図 9-3 Hinemosマネージャ死活監視間隔

10 セキュリティ

Hinemosマネージャを停止した上で、以下の手順に従ってパスワードを変更してください。変更前に [データベースのバックアップ](#) に従い、バックアップを取得しておくことを推奨します。

パスワードおよび関連ファイルを変更した後、Hinemosマネージャを起動してください。なお、接続元・接続先でパスワードの整合性が取れていない場合、Hinemosマネージャは正常に動作せず、Hinemosクライアントから正常に接続できません。

10.1 データベースアクセスのパスワードを変更する

HinemosマネージャのPostgreSQLユーザhinemosとhinemos_quartzに対して、パスワードとアクセス権限を変更する手順を説明します。

10.1.1 PostgreSQL（接続先）の設定変更

- ・ パスワード

以下の手順でPostgreSQLのパスワードを変更します。

1. Hinemosマネージャを停止し、rootユーザで以下のコマンドを実行します。

パスワードの入力を求められますが、PostgreSQLへのログインパスワード(デフォルトは"hinemos")を入力します。

```
(root) # /opt/hinemos/bin/pg_start.sh

(root) # /opt/hinemos/postgresql/bin/psql -p 24001 -U hinemos
Password for user hinemos:
psql (9.0.4)
Type "help" for help.
```

2. psqlのプロンプトが表示されるため、以下のコマンドを実行します。

```
hinemos=# ALTER USER hinemos PASSWORD ' (新しいパスワード) ';
hinemos=# ALTER USER hinemos_quartz PASSWORD ' (新しいパスワード) ';
```

3. psqlを終了し、PostgreSQLも停止します。

```
hinemos=# \q

(root) # /opt/hinemos/bin/pg_stop.sh
```

- アクセス権限

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/etc/postgresql/pg_hba.conf

```
(略)
# TYPE DATABASE USER CIDR-ADDRESS METHOD
# "local" is for Unix domain socket connections only
local postgres hinemos md5
local hinemos hinemos md5
local hinemos hinemos_quartz md5
# IPv4 local connections:
host postgres hinemos 0.0.0.0/0 md5
host hinemos hinemos 0.0.0.0/0 md5
host hinemos hinemos_quartz 0.0.0.0/0 md5
# IPv6 local connections:
host postgres hinemos ::/0 md5
host hinemos hinemos ::/0 md5
host hinemos hinemos ::/0 md5
(略)
```

(注) 上記は設定例であり、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推奨します。

10.1.2 Hinemosマネージャ（接続元）の設定変更

以下の2つの設定ファイルを変更します。

- hinemos-ds.xml
- scheduler-ds.xml

1. hinemos-ds.xmlの編集

hinemos-ds.xmlは、PostgreSQLユーザhinemosのデータベースアクセスに関する設定ファイルです。

[PostgreSQL（接続先）の設定変更](#) で変更したパスワードを <password>（新しいパスワード）</password> に設定します。

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

```
(略)
<driver-class>org.postgresql.Driver</driver-class>
<user-name>hinemos</user-name>
<password>hinemos</password>
(略)
```

2. scheduler-ds.xmlの編集

scheduler-ds.xmlは、PostgreSQLユーザhinemos_quartzのデータベースアクセスに関する設定ファイルです。

[PostgreSQL（接続先）の設定変更](#) で変更したパスワードを <password>（新しいパスワード）</password> に設定します。

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

（略）

```
<driver-class>org.postgresql.Driver</driver-class>  
<user-name>hinemos_quartz</user-name>  
<password>hinemos_quartz</password>
```

（略）

11 動作ログ

11.1 Hinemos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 11-1に示すログファイルに出力されます。

表 11-1 Hinemosマネージャのログファイル一覧

ファイル名	boot.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/jboss-log4j-boot.properties
出力レベル	priority INFO
ローテーション	ファイルサイズ 30MByte (カレントログを含めて最大5世代)
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	—
出力レベル	—
ローテーション	Hinemosマネージャ起動/停止時
内容	JBossの標準出力/標準エラー出力/スレッドダンプ
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/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/postgresql.conf
出力レベル	warning
ローテーション	Daily(無期限)
内容	PostgreSQLの動作ログ
ファイル名	startup.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/jboss-log4j.xml
出力レベル	priority INFO
ローテーション	Hinemosマネージャ起動時に上書き
内容	JBossの起動ログ
ファイル名	operation.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/jboss-log4j.xml
出力レベル	priority INFO or DEBUG
ローテーション	Daily(無期限)

内容	Hinemosの操作ログ
ファイル名	hinemos_manager_summary.*
格納ディレクトリ	/opt/hinemos/var/log/
ログ出力設定ファイル	—
出力レベル	—
ローテーション	スクリプト実行時
内容	/opt/hinemos/sbin/mng/hinemos_manager_summary.shの実行ログ
ファイル名	twiddle.log
格納ディレクトリ	/opt/hinemos/var/log/
ログ出力設定ファイル	—
出力レベル	—
ローテーション	スクリプトが実行される度に上書き
内容	/opt/hinemos/jboss-as/bin/twiddle.shの実行ログ

11.2 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/jboss-log4j.xml

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

(中略)

<!-- Hinemos AplLog category for internal log -->
<!--
<category name="hinemos.apllog">
  <priority value="INFO" />
  <appender-ref ref="HINEMOS_APL_LOG"/>
</category>
-->

(中略)

<!-- ===== -->
<!-- Setup the Root category -->
<!-- ===== -->

<root>
  <!--
    Set the root logger priority via a system property. Note this is parsed by log4j,
    so the full JBoss system property format is not supported; e.g.
    setting a default via ${jboss.server.log.threshold:WARN} will not work.
  -->
  <priority value="${jboss.server.log.threshold}"/>
  <appender-ref ref="FILE"/>
</root>
```

本ファイルでHinemosApl.log、jboss.logのログ出力レベルを変更することができます。Log4jの設定例については、http://docs.jboss.org/jbossas/docs/Installation_And_Getting_Started_Guide/5/html/The_JBoss_Server___A_Quick_Tour.html#Basic_Configuration_Issues-Logging_Service 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（自動設定読み込み機構）で、変更したログ出力のレベルが反映されます。

11.3 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。

/opt/hinemos/etc/postgresql/postgresql.conf

設定例については、<http://www.postgresql.jp/document/pg904doc/html/runtime-config-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により、変更したログ出力のレベルが反映されます。

11.4 操作ログのログ出力を変更する

- 変更ファイル

操作ログ(/opt/hinemos/var/log/operation.log)のログ出力設定を変更するには、下記の設定ファイルを編集します。

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

(中略)

```
<!-- Hinemos OperationLog category for internal log -->
<category name="HinemosOperation">
  <priority value="INFO" />
  <appender-ref ref="HINEMOS_OPE_LOG"/>
</category>
```

(中略)

なお、priority valueとログ出力対象の操作の対応関係は表11-2の通りです。

表11-2. 操作ログの設定値

priority value	ログ出力対象の操作
INFO	設定、実行
DEBUG	参照、設定、実行

- 変更適用方法

Hinemosマネージャの再起動、または10分間隔（自動設定読み込み機構）で、変更したログ出力のレベルが反映されます。

11.5 Hinemos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 11-3に示すログファイルに出力されます。

表 11-3 Linux版エージェントのログファイル一覧

ファイル名	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/conf/log4j.properties

出力レベル	priority INFO
ローテーション	ファイルサイズ 20MByte (カレントログを含めて最大5世代)
内容	Hinemosエージェントのログ
ファイル名	agent_stdouterr.log
格納ディレクトリ	/opt/hinemos_agent/var/log/
ログ出力設定ファイル	—
出力レベル	—
ローテーション	Hinemosエージェント起動時に上書き
内容	Hinemosエージェントの標準出力/標準エラー出力/スレッドダンプ
ファイル名	restart.log
格納ディレクトリ	/opt/hinemos_agent/var/log/
ログ出力設定ファイル	—
出力レベル	—
ローテーション	—
内容	Hinemosエージェント再起動時のログ

Windows版エージェントのログは表 11-4に示すログファイルに出力されます。

表 11-4 Windows版エージェントのログファイル一覧

ファイル名	agent.log*
格納ディレクトリ	[Hinemosエージェントのインストールディレクトリ]\var\log\
ログ出力設定ファイル	[Hinemosエージェントのインストールディレクトリ]\conf\log4j.properties
出力レベル	priority INFO
ローテーション	ファイルサイズ 20MByte (カレントログを含めて最大5世代)
内容	Hinemosエージェントのログ
ファイル名	agent_stdouterr.log
格納ディレクトリ	[Hinemosエージェントのインストールディレクトリ]\var\log\
ログ出力設定ファイル	—
出力レベル	—
ローテーション	Hinemosエージェント起動時に上書き
内容	Hinemosエージェントの標準出力/標準エラー出力/スレッドダンプ
ファイル名	restart.log
格納ディレクトリ	[Hinemosエージェントのインストールディレクトリ]\var\log\
ログ出力設定ファイル	—
出力レベル	—
ローテーション	—
内容	Hinemosエージェント再起動時のログ

11.6 Hinemosエージェントのログ出力を変更する

- 変更ファイル

Hinemos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/conf/log4j.properties
- [Windows版エージェントの場合]
[Hinemosエージェントのインストールディレクトリ]\conf\log4j.properties

```
### direct messages to file agent.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 syslog ###
log4j.appender.syslog=org.apache.log4j.net.SyslogAppender
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=info, file

log4j.appender.file.File=/opt/hinemos_agent/var/log/agent.log
log4j.appender.syslog.SyslogHost={管理対象ノードのホスト名}
```

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。

注) org.apache.log4j.net.SyslogAppenderで定義される出力は、Hinemosエージェントの機能自身であるため、org.apache.log4j.net.SyslogAppenderに関連する設定を変更した場合に、Hinemos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

Hinemosエージェントの再起動、または10分間隔（自動設定読み込み機構）で、変更したログ出力のレベルが反映されます。

11.7 Hinemosクライアントのログファイル一覧

Hinemosクライアントのログは表 11-5に示すログファイルに出力されます。

表 11-5 Hinemosクライアントのログファイル一覧

ファイル名	client.log
格納ディレクトリ	C:\Users\【ユーザ名】\AppData\Roaming\hinemos\
ログ出力設定ファイル	[Hinemosクライアントインストールディレクトリ]\log4j.properties
出力レベル	—
ローテーション	ファイルサイズ 20MByte (カレントログを含めて最大5世代)
内容	Hinemosクライアントのログ

12 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 に定義されています。(propertiesファイルにマルチバイト文字を記述する場合、propertiesファイル編集用のエディタが必要です)

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。

パラメータ[common.notify.date.format]

プロパティ	common.notify.date.format
プロパティ名	(通知)通知に利用可能な変数(#[GENERATION_DATE])の日付書式
説明	通知に利用可能な変数(#[GENERATION_DATE])の日付書式
データ型	文字列
デフォルト値	yyyy/MM/dd HH:mm:ss
参考	http://java.sun.com/javase/6/docs/ja/api/java/text/SimpleDateFormat.html

パラメータ[common.mail.from.address]

プロパティ	common.mail.from.address
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	admin@hinemos.com

パラメータ[common.mail.from.personal.name]

プロパティ	common.mail.from.personal.name
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

パラメータ[common.mail.reply.to.address]

プロパティ	common.mail.reply.to.address
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	admin@hinemos.com

パラメータ[common.mail.reply.personal.name]

プロパティ	common.mail.reply.personal.name
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

パラメータ[common.mail.errors.to.address]

プロパティ	common.mail.errors.to.address
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	admin@hinemos.com

パラメータ[common.mail.retry.count]

プロパティ	common.mail.retry.count
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

パラメータ[common.mail.retry.sleep]

プロパティ	common.mail.retry.sleep
プロパティ名	(通知)メール通知 再送待ち時間
説明	メール通知によるSMTPサーバへの送信失敗時の再送待ち時間(ミリ秒)
データ型	整数
デフォルト値	10000

パラメータ[common.manager.hostname]

プロパティ	common.manager.hostname
プロパティ名	(通知)ログエスケレーション通知 送信元ホスト名
説明	ログエスケ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(\${NODE}、\${FACILITY_ID}、固定のホスト名、空:マネージャサーバのホスト名)
デフォルト値	\${NODE}

パラメータ[common.repository.snmp.timeout]

プロパティ	common.repository.snmp.timeout
プロパティ名	リポジトリ管理機能のfind by snmpに関するタイムアウト値
説明	リポジトリ管理機能のfind by snmpにおけるSNMPポーリングのタイムアウト時間
データ型	整数(msec)
デフォルト値	5000

パラメータ[common.repository.snmp.retry]

プロパティ	common.repository.snmp.retry
プロパティ名	リポジトリ管理機能のfind by snmpに関するポーリング最大試行回数
説明	リポジトリ管理機能のfind by snmpにおけるSNMPポーリングの最大試行回数
データ型	整数
デフォルト値	1

パラメータ[common.ws.address]

プロパティ	common.ws.address
プロパティ名	Hinemosマネージャが待ち受けるIPアドレス
説明	Hinemosマネージャがクライアントやエージェントからの接続を待ち受けるIPアドレス

データ型	文字列
デフォルト値	http://0.0.0.0:8080

パラメータ[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

パラメータ[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

パラメータ[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	2000

パラメータ[monitor.common.report.event.bom]

プロパティ	monitor.common.report.event.bom
プロパティ名	Hinemos イベントダウンロードで生成するファイルのBOM付与定義
説明	監視[イベント]ビューのダウンロード機能の生成するファイルにBOMを付与するかどうか
データ型	論理 (true, false)
デフォルト値	true

パラメータ[monitor.common.report.event.separator]

プロパティ	monitor.common.report.event.separator
プロパティ名	Hinemos イベントダウンロードで生成するファイルのカラム区切り文字
説明	監視[イベント]ビューのダウンロード機能の生成するファイルのカラム区切り文字
データ型	文字列
デフォルト値	,

パラメータ[monitor.common.report.event.format]

プロパティ	monitor.common.report.event.format
プロパティ名	Hinemos イベントダウンロードで生成するファイルの日時カラムの日時書式
説明	監視[イベント]ビューのダウンロード機能の生成するファイルの日時カラムの日時書式
データ型	文字列(SimpleDateFormatの日付/時刻パターンに従う)

デフォルト値	yyyy/MM/dd HH:mm:ss
参考	http://java.sun.com/javase/ja/6/docs/ja/api/java/text/SimpleDateFormat.html

パラメータ[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

パラメータ[monitor.port.protocol.dns]

プロパティ	monitor.port.protocol.dns
プロパティ名	サービス・ポート監視においてDNSの動作確認に利用するホスト名
説明	サービス・ポート監視においてDNSの動作確認として問い合わせ対象とするホスト名
データ型	文字列
デフォルト値	localhost

パラメータ[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(秒)
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(秒)
データ型	整数
デフォルト値	30

パラメータ[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(秒)
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(秒)
データ型	整数
デフォルト値	50

パラメータ[monitor.resource.start.second]

プロパティ	monitor.resource.start.second
プロパティ名	(リソース)リソース監視の閾値判定処理の開始間隔(秒)
説明	リソー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(秒)
データ型	整数
デフォルト値	30

パラメータ[monitor.resource.valid.second]

プロパティ	monitor.resource.valid.second
プロパティ名	(リソース)リソース情報有効間隔(秒)
説明	リソー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(秒)
データ型	整数

デフォルト値	50
--------	----

パラメータ[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(秒)
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(秒)
データ型	整数
デフォルト値	10

パラメータ[monitor.systemlog.udp.socket.buffer.size]

プロパティ	monitor.systemlog.udp.socket.buffer.size
プロパティ名	(システムログ)システムログ監視で利用するSocketの受信バッファサイズ(Byte)
説明	システムログ監視で利用するSocketの受信バッファサイズ(Byte)
データ型	整数
デフォルト値	8388608

パラメータ[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

パラメータ[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(秒)
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(秒)
データ型	整数
デフォルト値	60

パラメータ[repository.find.by.snmp.verbose]

プロパティ	repository.find.by.snmp.verbose
プロパティ名	find by snmpのデバイスの冗長追加の有無
説明	find by snmpにより登録されるデバイス (disk,nic) はOSの起動後に、一度以上データのIN/OUTがあった物のみ取得しますが、本パラメータをtrueにすると、IN/OUTのなかったデバイスも追加されます。
データ型	論理
デフォルト値	false

パラメータ[performance.graph.max.plot]

プロパティ	performance.graph.max.plot
プロパティ名	性能管理機能のグラフとして出力される最大プロット数
説明	性能管理機能のグラフとして出力される最大プロット数。このプロット数を超える範囲が指定された場合は、先頭から最大プロット数だけグラフに表示されます。

データ型	整数
デフォルト値	10000

パラメータ[performance.export.dir]

プロパティ	performance.export.dir
プロパティ名	性能管理機能のエクスポートで利用されるワークディレクトリ
説明	性能管理機能のエクスポートで利用されるワークディレクトリで、このディレクトリにエクスポートデータが一時的に出力された後、Hinemosクライアントに送信されます。
データ型	文字列
デフォルト値	/opt/hinemos/var/export/

パラメータ[performance.export.encode]

プロパティ	performance.export.encode
プロパティ名	性能管理機能のエクスポートで生成されるファイルの文字コード
説明	性能管理機能のエクスポートで生成されるファイルの文字コード
データ型	文字列(MS932, UTF-8, EUC_JP)
デフォルト値	MS932

パラメータ[performance.export.line.separator]

プロパティ	performance.export.line.separator
プロパティ名	性能管理機能のエクスポートで生成されるファイルの改行コード
説明	性能管理機能のエクスポートで生成されるファイルの改行コード
データ型	文字列(\r\n, \r, \n)
デフォルト値	\r\n

パラメータ[performance.export.fetchsize]

プロパティ	performance.export.fetchsize
プロパティ名	性能管理機能のエクスポートをデータベースからの読み出し単位
説明	性能管理機能のエクスポートをデータベースからの読み出し単位(SELECTのFetch Size)
データ型	整数
デフォルト値	1000

パラメータ[common.notify.command.thread.pool.count]

プロパティ	common.notify.command.thread.pool.count
プロパティ名	コマンド通知に利用するコマンド実行用のスレッドプール
説明	コマンド通知に利用するコマンド実行用のスレッドプール(プールするスレッド数)
データ型	整数
デフォルト値	10

パラメータ[common.notify.command.create.mode]

プロパティ	common.notify.command.create.mode
プロパティ名	コマンド通知のコマンド動作に関するOSプラットフォーム定義
説明	コマンド通知のコマンド動作に関するOSプラットフォーム定義
データ型	文字列 (auto, windows, unix, compatible)

デフォルト値	auto
--------	------

パラメータ[common.notify.command.success.exit]

プロパティ	common.notify.command.success.exit
プロパティ名	コマンド通知のコマンド戻り値の正常値
説明	コマンド通知のコマンド戻り値の正常値。これ以外の戻り値の場合は、INTERNALイベントとして通知される。
データ型	整数
デフォルト値	0

パラメータ[log.line.max.length]

プロパティ	log.line.max.length
プロパティ名	システムログ監視の#[LOG_LINE]に埋め込む最大文字列長(byte)
説明	システムログ監視の#[LOG_LINE]に埋め込む最大文字列長(byte)。この長さを超えるSyslogのメッセージ部は切り捨てられて、通知のメッセージに格納される。
データ型	整数
デフォルト値	256

パラメータ[common.repository.restart.sleep]

プロパティ	common.repository.restart.sleep
プロパティ名	Hinemosエージェントアップデートの試行間隔(ミリ秒)
説明	Hinemosエージェントアップデートはエージェントごとに間隔をあけて実施される。その試行間隔を定義する。
データ型	整数
デフォルト値	500

パラメータ[monitor.snmptrap.filter.queue.size]

プロパティ	monitor.snmptrap.filter.queue.size
プロパティ名	SNMPTRAP監視のフィルタリングタスクの最大キューサイズ
説明	SNMPTRAP監視のフィルタリングタスクの最大キューサイズ。SNMPTRAP監視のフィルタリング性能より、受信量が多くなった場合、このキューに蓄積されていき、最大キューサイズを超えると、SNMPTRAPはdropされる。
データ型	整数
デフォルト値	3000

パラメータ[monitor.snmptrap.filter.thread.size]

プロパティ	monitor.snmptrap.filter.thread.size
プロパティ名	SNMPTRAP監視のフィルタリングタスクのスレッド数
説明	SNMPTRAP監視のフィルタリングタスクを行う最大スレッド数。
データ型	整数
デフォルト値	8

パラメータ[snmptrap.stats.interval]

プロパティ	snmptrap.stats.interval
プロパティ名	SNMPTRAP監視における統計情報の出力間隔

説明	SNMPTRAPの受信数がこの倍数となった際に統計情報がログファイルに出力される。
データ型	整数
デフォルト値	100

パラメータ[common.notify.replace.before]

プロパティ	common.notify.replace.before
プロパティ名	SNMPTRAP監視のvarbindに含まれる不正文字の置換設定
説明	SNMPTRAP監視のvarbindに含まれる不正文字の置換設定。置換対象とする文字のバイトコードを定義する。
データ型	文字列(半角カンマ区切りで複数のバイトコードを定義できる)
デフォルト値	0

パラメータ[common.notify.replace.after]

プロパティ	common.notify.replace.after
プロパティ名	SNMPTRAP監視のvarbindに含まれる不正文字の置換設定
説明	SNMPTRAP監視のvarbindに含まれる不正文字の置換設定。置換対象とする文字が見つかった場合、この設定文字に置換する。
データ型	文字列(1バイト)
デフォルト値	?

13 Hinemosエージェントの設定一覧

Hinemosエージェントの設定は、下記の設定ファイルに定義されています。

- /opt/hinemos_agent/conf/Agent.properties (Linux版エージェント)
- [Hinemosエージェントのインストールディレクトリ]\conf\Agent.properties (Windows版エージェント)

(propertiesファイルにマルチバイト文字を記述する場合、propertiesファイル編集用のエディタが必要です)

設定変更を反映させるには、Hinemosエージェントの再起動が必要です。

パラメータ[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

パラメータ[runhistory.clear.delay]

プロパティ	runhistory.clear.delay
プロパティ名	Hinemosマネージャから受信したジョブ情報の最大生存期間(秒)
説明	Hinemosマネージャから受信したジョブ情報の最大生存期間(秒)
データ型	整数
デフォルト値	604800

パラメータ[limit.size.jobmsg]

プロパティ	limit.size.jobmsg
プロパティ名	ジョブの実行結果(標準出力、標準エラー出力)として扱う最大バイト数
説明	ジョブの実行結果(標準出力、標準エラー出力)として扱う最大バイト数
データ型	整数
デフォルト値	1024

パラメータ[limit.size.logmsg]

プロパティ	limit.size.logmsg
プロパティ名	ログファイル監視で読み込む1行の最大バイト数(byte)
説明	ログファイル監視で読み込む1行の最大バイト数(byte)。このバイト数を超える部分は切り捨てられて、ログファイル監視の対象となる。
データ型	整数
デフォルト値	1024

パラメータ[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)

パラメータ[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, unix, windows, compatible)
デフォルト値	auto

パラメータ[run.interval]

プロパティ	notify.retry.count
プロパティ名	ログファイル監視のファイルチェック間隔(ミリ秒)
説明	ログファイル監視のファイルチェック間隔(ミリ秒)
データ型	整数
デフォルト値	10000

パラメータ[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

パラメータ[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	ログファイル監視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

パラメータ[syslog.message.priority]

プロパティ	syslog.message.priority
プロパティ名	ログ転送機能に利用するsyslog priority(ログファイル転送機能互換)
説明	ログファイルに追記されたメッセージをsyslogで転送する設定です。syslogのpriorityを定義することで、従来のログファイル転送機能の互換性のある動作が可能となります。未定義の場合は、syslogによる転送を行いません。
データ型	文字列
デフォルト値	未定義

パラメータ[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(MS932, UTF-8, EUC_JP)
デフォルト値	Linux版エージェント : UTF-8, Windows版エージェント : MS932

パラメータ[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

パラメータ[collector.command.workerthreads]

プロパティ	collector.command.workerthreads
プロパティ名	カスタム監視におけるコマンド実行の最大スレッド数
説明	カスタム監視におけるコマンド実行の最大スレッド数
データ型	整数
デフォルト値	8

パラメータ[collector.command.mode]

プロパティ	collector.command.mode
プロパティ名	カスタム監視のコマンド動作に関するOSプラットフォーム定義
説明	カスタム監視のコマンド動作に関するOSプラットフォーム定義
データ型	文字列(auto, unix, windows, compatible)
デフォルト値	auto

パラメータ[collector.command.buffer]

プロパティ	collector.command.buffer
プロパティ名	カスタム監視における標準出力読み込みの最大量(byte)
説明	カスタム監視における標準出力読み込みの最大量(byte)
データ型	整数
デフォルト値	512

パラメータ[collector.command.charset]

プロパティ	collector.command.charset
プロパティ名	カスタム監視における標準出力の文字コード定義
説明	カスタム監視における標準出力の文字コード定義
データ型	文字列(MS932, UTF-8, EUC_JP)
デフォルト値	Linux版エージェント : UTF-8, Windows版エージェント : MS932

パラメータ[collector.command.returncode]

プロパティ	collector.command.returncode
プロパティ名	カスタム監視における標準出力の改行コード定義
説明	カスタム監視における標準出力の改行コード定義
データ型	文字列(LF, CRLF, CR)
デフォルト値	Linux版エージェント : LF, Windows版エージェント : CRLF

パラメータ[user]

プロパティ	user
-------	------

プロパティ名	Hinemosマネージャにログインするためのユーザ
説明	Hinemosマネージャにログインするためのユーザ
データ型	文字列
デフォルト値	HINEMOS_AGENT

パラメータ[password]

プロパティ	password
プロパティ名	Hinemosマネージャにログインするためのパスワード
説明	Hinemosマネージャにログインするためのパスワード
データ型	文字列
デフォルト値	HINEMOS_AGENT

パラメータ[managerAddress]

プロパティ	managerAddress
プロパティ名	Hinemosマネージャに接続するためのURL
説明	Hinemosマネージャに接続するためのURL
データ型	文字列
デフォルト値	http://[HinemosマネージャのIPアドレス]:8080/HinemosWS/

パラメータ[topic.interval]

プロパティ	topic.interval
プロパティ名	Hinemosマネージャへの問い合わせ間隔(msec)
説明	Hinemosマネージャへの問い合わせ間隔(msec)
データ型	整数
デフォルト値	30000

パラメータ[topic.fast.interval]

プロパティ	topic.fast.interval
プロパティ名	ジョブ実行時におけるHinemosマネージャへの問い合わせ間隔(msec)
説明	ジョブ実行時におけるHinemosマネージャへの問い合わせ間隔(msec)
データ型	整数
デフォルト値	1000

パラメータ[topic.fast.count]

プロパティ	topic.fast.count
プロパティ名	topic.fast.intervalで動作する問い合わせの最大回数
説明	topic.fast.intervalで動作する問い合わせの最大回数
データ型	整数
デフォルト値	10

パラメータ[connect.timeout]

プロパティ	connect.timeout
プロパティ名	Hinemosマネージャへの接続タイムアウト(秒)

説明	Hinemosマネージャと接続する際のタイムアウト時間
データ型	整数
デフォルト値	10

パラメータ[request.timeout]

プロパティ	request.timeout
プロパティ名	Hinemosマネージャからの受信タイムアウト(秒)
説明	Hinemosマネージャからの受信タイムアウト時間
データ型	整数
デフォルト値	60

パラメータ[facilityId]

プロパティ	facilityId
プロパティ名	Hinemosエージェントと対応するノードのファシリティID
説明	Hinemosエージェントと対応するノードのファシリティID。未定義の場合は、ホスト名とIPアドレスから該当するノードを識別する。定義した場合は、ファシリティIDが固定される。
データ型	文字列
デフォルト値	未定義

パラメータ[first.part.data.check.period]

プロパティ	first.part.data.check.period
プロパティ名	ファイルの冒頭チェック期間設定(秒)
説明	ファイルサイズが指定秒間変化がない場合でも、ファイルの冒頭をチェックする機構を発動させる間隔
データ型	整数
デフォルト値	300

パラメータ[first.part.data.check.size]

プロパティ	first.part.data.check.size
プロパティ名	ファイルの冒頭チェックにおける比較バイト数
説明	ファイルの冒頭チェックにおける比較バイト数
データ型	整数
デフォルト値	256

パラメータ[notify.retry.count]

プロパティ	notify.retry.count
プロパティ名	ログファイル監視の結果送信処理の最大試行回数
説明	ログファイル監視の結果送信処理の最大試行回数
データ型	整数
デフォルト値	15

パラメータ[notify.retry.sleep]

プロパティ	notify.retry.count
プロパティ名	ログファイル監視の結果送信処理の試行間隔(ミリ秒)

説明	ログファイル監視の結果送信処理の試行間隔(ミリ秒)
データ型	整数
デフォルト値	60000

パラメータ[file.transfer.skip.keyfile.update]

プロパティ	file.transfer.skip.keyfile.update
プロパティ名	ファイル転送ジョブ実行時の鍵ファイル編集スキップの有無
説明	本パラメータをtrueとすると、Hinemosエージェントによる authorized_keyの編集が行われずに、ファイル転送ジョブが動作します。ユーザは事前にファイル送信元のauthorized_keyにファイル送信先の公開鍵を登録しておく必要があります。(Hinemos4.0.1以降で利用可)
データ型	論理
デフォルト値	false

14 変更履歴

変更履歴

版	変更日	変更内容
第1.0版	2012/4/25	初版発行
第1.1版	2012/10/1	3.1.2 hinemos_clear_notify.shの実行する上での前提条件を修正
		4.3 run.confの設定変更を反映
		5.2.5 メール通知におけるSMTPサーバのポート番号の変更方法を記載
		7.6.6 WinRMへアクセスするユーザの必要権限を記載
		8.1 OSプラットフォームの識別の説明を修正
		9.1 Hinemos内部スコープに通知される情報を記載
		13 ジョブエージェントのファイル転送ジョブ用の設定パラメータを追記

Hinemos ver4.0 管理者ガイド

非売品

- 禁無断複製
- 禁無断転載
- 禁無断再配布

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。