

Hinemos

Hinemos ver5.0

管理者ガイド 第5版

目次

1	はじめに	7
2	前提条件	8
3	メンテナンス	9
3.1	Hinemosマネージャのメンテナンス	9
3.1.1	概要	9
3.1.2	メンテナンススクリプト一覧	9
3.1.3	データベース内の不要なデータの削除	10
3.1.4	データベースの再構成	11
3.1.5	データベースのバックアップ	12
3.1.6	データベースのリストア	12
3.1.7	一時キューに格納されている情報の消去	12
3.1.8	通知の抑制基準となる履歴情報の消去	13
3.1.9	環境サマリ情報の取得	13
3.1.10	OS時刻変更時のスケジューラ調整	14
3.1.11	環境構築機能のファイル入出力	14
3.2	ログファイルの削除	15
3.2.1	Hinemosマネージャのログファイル	15
3.2.2	Hinemosエージェントのログファイル	15
3.2.3	Hinemosクライアントのログファイル	16
3.2.4	Hinemos Webクライアントのログファイル	16
3.3	Hinemosマネージャのバックアップ・リカバリ	16
3.3.1	Hinemosマネージャのバックアップ	16
3.3.2	Hinemosマネージャのリカバリ	17
3.4	Hinemosエージェントのバックアップ・リカバリ	17
3.4.1	Hinemosエージェントのバックアップ	17
3.4.2	Hinemosエージェントのリカバリ	17
3.5	Hinemosリッチクライアントのバックアップ・リカバリ	18
3.6	Hinemos Webクライアントのバックアップ・リカバリ	18
3.6.1	Hinemos Webクライアントのバックアップ	18
3.6.2	Hinemos Webクライアントのリカバリ	18
4	OS/環境	20
4.1	ファイルディスクリプタの設定	20
4.2	Javaヒープメモリの設定(Hinemosマネージャ)	20
4.3	同時接続数の設定(Webクライアント)	20
4.4	OSロケールと文字コード設定	21
4.4.1	Hinemosリッチクライアント	21
4.4.2	Hinemos Webクライアント	21
4.4.3	Hinemosマネージャ	21
4.4.4	Hinemosエージェント	22
4.5	HinemosエージェントのファシリティIDの設定	22
4.6	JAVA_HOMEの変更	23
5	通知機能	23
5.1	イベント通知	23
5.1.1	最大ダウンロード件数の設定	23

5.2	メール通知	24
5.2.1	メール通知を有効にする	24
5.2.2	SMTP AUTHの設定	24
5.2.3	SSL/TLSの設定	24
5.2.4	エンベロープFromの設定	25
5.3	ログエスカレーション通知	25
5.3.1	埋め込みホスト名の設定方法	25
5.3.2	転送プロトコルにTCPを使用する場合の設定方法	25
6	監視設定/性能機能	26
6.1	SQL監視	26
6.1.1	監視対象のRDBMSの追加	26
6.2	プロセス監視	26
6.2.1	プロセス監視の動作	26
6.3	HTTPS監視	27
6.3.1	サーバ証明書を利用しない場合のHTTPS設定	27
6.3.2	サーバ証明書を利用する場合のHTTPS設定	27
6.4	システムログ監視	28
6.4.1	syslogの送信に関する設定	28
6.4.2	syslogの受信に関する設定	29
6.4.3	制御文字に対するエスケープ処理の無効化	29
6.4.4	syslogのHEADER部に対するホスト名置換処理の設定	29
6.5	SNMPTRAP監視	30
6.5.1	SNMPTRAPの受信に関する設定	30
6.5.2	デフォルトMIB	30
6.6	Windowsサービス監視	38
6.6.1	WinRMのインストール	38
6.6.2	リモートコンピュータ管理の許可	39
6.6.3	Basic認証の許可	39
6.6.4	非暗号通信の許可	39
6.6.5	サーバ証明書を利用しない場合のHTTPS設定	39
6.6.6	サーバ証明書を利用したHTTPSの設定	40
6.6.7	接続先OSユーザの準備	43
6.6.8	疎通確認	43
6.7	カスタム監視	44
6.7.1	コマンドの動作変更	44
6.7.2	標準出力からの読み込み最大量の設定	44
6.7.3	標準出力に含まれる改行コードの設定	44
6.7.4	コマンドの実行多重度の設定	44
6.8	ポーリングプロトコルの設定	45
6.8.1	Net-SNMPの設定	45
6.8.2	SNMP・WBEMの切り替え方法	45
6.9	数値監視の収集値の設定	46
6.10	リソース監視	46
6.10.1	大容量ファイルシステム使用率監視の設定	46
6.11	ログファイル監視	47

6.11.1	監視可能最大ファイル数の設定	47
6.12	JMX監視	47
6.12.1	JMX監視の設定	47
7	ジョブ機能	48
7.1	起動コマンドの動作変更	48
7.2	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	50
7.3	ファイル転送ジョブを有効にする	50
8	その他の機能	53
8.1	セルフチェック機能	53
8.1.1	セルフチェック機能の設定	54
8.1.2	セルフチェック機能の前提条件	54
8.2	INTERNALイベント	54
8.2.1	通知先の設定	54
8.2.2	通知される情報	56
8.3	Hinemosマネージャ死活検知	58
8.4	環境構築機能	60
8.4.1	HTTPS接続を利用した環境構築モジュール	60
9	セキュリティ	60
9.1	データベースアクセスのパスワードを変更する	60
9.1.1	PostgreSQL（接続先）の設定変更	60
9.1.2	Hinemosマネージャ（接続元）の設定変更	61
9.2	データベース内のパスワードの暗号化	62
9.3	JMXのアクセス権限を変更する	62
10	Hinemosコンポーネント間接続	63
10.1	HinemosマネージャへのHTTP Proxyを経由した接続	63
10.1.1	HinemosクライアントからHinemosマネージャへの接続	63
10.1.2	HinemosエージェントからHinemosマネージャへの接続	63
10.2	HinemosマネージャへのHTTPS接続	64
10.2.1	Hinemosマネージャのサーバ証明書の準備	64
10.2.2	HinemosリッチクライアントからHinemosマネージャへのHTTPS接続	66
10.2.3	Hinemos WebクライアントからHinemosマネージャへのHTTPS接続	66
10.2.4	HinemosエージェントからHinemosマネージャへのHTTPS接続	66
10.2.5	HinemosリッチクライアントからHinemosマネージャへのHTTPS接続時にホスト認証を行う	66
10.2.6	Hinemos WebクライアントからHinemosマネージャへのHTTPS接続時にホスト認証を行う	67
10.3	Hinemos WebクライアントへのHTTPSによる接続	68
11	ポート変更	69
11.1	Hinemosマネージャのポート変更	69
11.2	Hinemos Webクライアントサービスのポート変更	70
11.3	Hinemosエージェントの即時反映用ポート	70
12	動作ログ	70
12.1	Hinemosマネージャのログファイル一覧	70
12.2	javaプロセスのログ出力・ログローテーションを変更する	71
12.3	PostgreSQLのログ出力・ログローテーションを変更する	72
12.4	操作ログのログ出力・ログローテーションを変更する	72
12.5	Hinemosエージェントのログファイル一覧	72

12.6	Hinemosエージェントのログ出力・ログローテーションを変更する	74
12.7	Hinemosクライアントのログファイル一覧	74
12.8	Hinemos Webクライアントのログファイル一覧	75
13	Hinemosマネージャの設定一覧	75
14	Hinemosエージェントの設定一覧	84
15	変更履歴	94

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/about/jigyoseika/04fy-pro/open.html>

Hinemosに関する最新情報はHinemosポータルサイト (<http://www.hinemos.info>) をご確認ください。

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・履歴情報など)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
```

- スケジューラをリセットする

```
hinemos_reset_scheduler.sh
```

- 環境構築機能のファイルをインポート/エクスポートする

```
hinemos_infra_file.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の概要・起動条件

処理内容	スクリプト名	JVM	PostgreSQL
不要なデータの削除	hinemos_delete.sh	停止必須	起動必須
データベースの再構成	hinemos_cluster_db.sh	停止必須	起動必須

データベースのバックアップ	hinemos_backup.sh	任意	起動必須
データベースのリストア	hinemos_restore.sh	停止必須	起動必須
一時キューの消去	hinemos_clear_tmp.sh	停止必須	起動必須
通知抑制情報の消去	hinemos_clear_notify.sh	起動必須	起動必須
環境サマリ情報の取得	hinemos_manager_summary.sh	起動必須	起動必須
スケジューラのリセット	hinemos_reset_scheduler.sh	停止必須	起動必須
環境構築機能のファイル入出力	hinemos_infra_file.sh	停止必須	起動必須

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(ステータス)
- ・ 監視ログ情報(イベント)
- ・ ジョブの実行履歴
- ・ 数値監視の収集値
- ・ 編集ロックの取得情報

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マネージャを停止させた後、PostgreSQLを起動します。

```
(root) # service hinemos_manager stop
(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。

- ・ 全ての監視ログ情報(ステータス)を削除する場合

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

- ・ 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
```

- ・ 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)]
```

- 全ての編集ロックの取得情報を削除する場合

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

3. PostgreSQLを停止させた後、Hinemosマネージャを起動します。

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

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。

1. Hinemosマネージャを停止させた後、PostgreSQLを起動します。

```
(root) # service hinemos_manager stop  
(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_cluster_db.sh
```

3. PostgreSQLを停止させた後、Hinemosマネージャを起動します。

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

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。

1. 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
```

2. 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。また、-s オプションを指定すると設定データのみバックアップを行います。-c オプションを指定するとgz圧縮したダンプファイルを作成できます。(-c オプションは0~9の引数を指定する必要があります。数値は圧縮レベルを意味し、数値が大きいほど高い圧縮率となります。)

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_HHmss
(-s オプションを指定した場合は、hinemos_pgdump_s.YYYY-MM-DD_HHmss)
```

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まります。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マネージャを停止させた後、PostgreSQLを起動します。

```
(root) # service hinemos_manager stop
(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_HHmss
```

3. PostgreSQLを停止させた後、Hinemosマネージャを起動します。

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

hinemos_pgdump_s.YYYY-MM-DD_HHmssもhinemos_restore.shでリストア可能です。

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

- (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)

1. Hinemosマネージャを停止させた後、PostgreSQLを起動します。

```
(root) # service hinemos_manager stop
(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. PostgreSQLを停止させた後、Hinemosマネージャを起動します。

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

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
```

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
```

- Hinemosマネージャ配下にあるファイルのMD5ハッシュ値を取得する場合

```
(root) # /opt/hinemos/sbin/mng/hinemos_manager_summary.sh -md5
```

3.1.10 OS時刻変更時のスケジューラ調整

マネージャサーバのOS時刻を変更した際は、Hinemos内部のスケジューラを変更後のOS時刻にあわせるために、hinemos_reset_scheduler.shを実行します。

1. Hinemosマネージャを停止します。

```
(root) # service hinemos_manager stop
```

2. マネージャのOS時刻を変更します
3. PostgreSQLのみ起動します。

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

4. スケジューラリセットスクリプトを実行します

```
(root) # /opt/hinemos/sbin/mng/hinemos_reset_scheduler.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。

5. PostgreSQLを停止させた後、Hinemosマネージャを起動します。

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

3.1.11 環境構築機能のファイル入出力

環境構築機能のファイルはHinemosクライアントから入出力が可能ですが、hinemos_infra_file.shによる入出力も可能です。hinemos_infra_file.shを実行します。ファイルの入力方法は下記の通りです。

1. Hinemosマネージャを停止させた後、PostgreSQLを起動します。

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

2. ファイル入出力スクリプトを実行します

```
(root) # /opt/hinemos/sbin/mng/hinemos_infra_file.sh import FILE_ID FILE_PATH OWNER_ROLE_ID
```

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. PostgreSQLを停止させた後、Hinemosマネージャを起動します。

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


ファイルの出力方法は下記の通りです。

1. Hinemosマネージャを停止させた後、PostgreSQLを起動します。

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

2. ファイル入出力スクリプトを実行します

```
(root) # /opt/hinemos/sbin/mng/hinemos_infra_file.sh export FILE_ID
```

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. PostgreSQLを停止させた後、Hinemosマネージャを起動します。

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

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。どのようなログファイルがあるのかについては、[動作ログ](#)を確認してください。

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す。

- /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\Client5.0\	Hinemosクライアントのログファイル

3.2.4 Hinemos Webクライアントのログファイル

Hinemos Webクライアントのログファイルは、表3-6のディレクトリに出力されます。

表 3-6 Hinemos Webクライアントのログディレクトリ

パス	内容
/opt/hinemos_web/var/log/	TomcatとHinemos Webクライアントが出力する各種ログファイル

なお、Hinemos Webクライアント向けに 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す。

- /opt/hinemos_web/contrib/hinemos_web

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_web/contrib/hinemos_web /etc/cron.daily/
```

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
 - rsyslogの設定ファイル
 - /etc/rsyslog.d/rsyslog_hinemos_manager.conf

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
- ・ rsyslogの設定ファイル
 - ・ /etc/rsyslog.d/rsyslog_hinemos_agent.conf
- ・ snmpの設定ファイル
 - ・ /etc/snmp/snmpd.conf

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

・ 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. SNMP Serviceの再設定

SNMP Serviceの設定の詳細は、Hinemosインストールマニュアルを参照して下さい。

3.5 Hinemosリッチクライアントのバックアップ・リカバリ

Hinemosリッチクライアントでは、バックアップ対象となるファイルがありません。Hinemosリッチ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リッチ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。

3.6 Hinemos Webクライアントのバックアップ・リカバリ

Hinemos Webクライアントのバックアップ・リカバリ方法を説明します。

3.6.1 Hinemos Webクライアントのバックアップ

・ 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 Webクライアント設定ファイル
 - ・ /opt/hinemos_web/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_web
- ・ ログファイルの削除スクリプト
 - ・ /etc/cron.daily/hinemos_web

3.6.2 Hinemos Webクライアントのリカバリ

1. Hinemos Webクライアントの再インストール

Hinemos Webクライアントをインストールします。Hinemos Webクライア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でインストールします。

Hinemos Web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。

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 Webクライア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)

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/hinemos_manager.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)
or
(ファイルを開きすぎです)
```

このメッセージは、Hinemosマネージャ上のjava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 JAVA_FD_MAXNUMの設定値を変更させてください。(JAVA_FD_MAXNUMは、java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 JAVA_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 Javaヒープメモリの設定(Hinemosマネージャ)

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/hinemos.cfg

```
### JVM - Performance Tuning
# for small systems
export JVM_HEAP_OPTS="-Xms512m -Xmx512m -XX:NewSize=160m -XX:MaxNewSize=160m -XX:MaxPermSize=192m -Xss256k"
# for medium systems
#export JVM_HEAP_OPTS="-Xms1024m -Xmx1024m -XX:NewSize=320m -XX:MaxNewSize=320m -XX:MaxPermSize=256m -Xss256k"
# for large systems
#export JVM_HEAP_OPTS="-Xms2048m -Xmx2048m -XX:NewSize=640m -XX:MaxNewSize=640m -XX:MaxPermSize=360m -Xss256k"
```

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あるいは2048m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。

4.3 同時接続数の設定(Webクライアント)

Webクライアントに同時接続できるブラウザ数は上限が定められています。(デフォルトでは8ブラウザとなります。)上限を上げたい場合は、下記の設定を変更してください。

/opt/hinemos_web/conf/hinemos_web.cfg

```
### JVM - Maximum number of access users
export JVM_MAX_USER_OPTS="-Dmaximum.access.users=8"
```

16ブラウザからの接続を受け付けたい場合は、上記のmaximum.access.users=8をmaximum.access.users=16に変更してください。

/opt/hinemos_web/conf/server.xml

```
<Connector port="80" protocol="HTTP/1.1"
  connectionTimeout="20000"
  redirectPort="8443"
  maxThreads="32"
 />
```

server.xmlのmaxThreadsは、hinemos_web.cfgのmaximum.access.usersの4倍に設定してください。16ブラウザからの接続を受け付けたい場合は、maxThreads=64と設定してください。

/opt/hinemos_web/conf/hinemos_web.cfg

```
### JVM - Performance Tuning
# for small systems
JVM_HEAP_OPTS="-Xms256m -Xmx256m -XX:NewSize=40m -XX:MaxNewSize=40m -XX:MaxPermSize=128m -Xss256k"
# for medium systems
#JVM_HEAP_OPTS="-Xms512m -Xmx512m -XX:NewSize=40m -XX:MaxNewSize=40m -XX:MaxPermSize=128m -Xss256k"
# for large systems
#JVM_HEAP_OPTS="-Xms1024m -Xmx1024m -XX:NewSize=40m -XX:MaxNewSize=40m -XX:MaxPermSize=128m -Xss256k"
```

ブラウザ数が10以上の場合はmediumを利用し、20以上の場合はlargeを利用してください。16ブラウザからの接続を受け付けたい場合は、mediumの設定を利用してください。

設定変更を反映させるには、Webクライアントサービス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リッチクライアント

Hinemosリッチクライアントで日本語を表示する場合、Hinemosリッチ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リッチ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 Webクライアント

Hinemos Webクライアントで日本語を表示する場合、Hinemos Webクライアントをインストールする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。

Hinemos Webクライアントは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_web/hinemos_web.cfgに記載されています。

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

4.4.3 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利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
```

4.4.4 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・標準エラー出力の文字コード

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・標準エラー出力が含まれます。Hinemosエージェントでは、この標準出力・標準エラー出力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 job.stream.charset 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。job.stream.charset 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（uFFFD）に置換されます。

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

```
## character encoding of job's stdout and stderr
job.stream.charset=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。

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。（複数のファシリティIDを設定する場合は、カンマ区切りで設定してください）

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。

4.6 JAVA_HOMEの変更

以下にHinemosマネージャ、Hinemosエージェント、WebクライアントのJAVA_HOMEの変更手順を記載します。

- Hinemosマネージャ

HinemosマネージャのJAVA_HOMEを変更する場合、 /opt/hinemos/etc/hinemos.cfgの以下の行を変更します。

```
export JAVA_HOME=/usr/lib/jvm/jre-1.7.0-openjdk
```

Hinemosマネージャで利用するJREのパスが上記以外である場合は、上記の値を変更してください。

JAVA_HOMEを書き換えた場合、Hinemosマネージャの再起動が必要です。

- Linux版Hinemosエージェント

Linux版HinemosエージェントのJAVA_HOMEを変更する場合、 /opt/hinemos_agent/conf/hinemos_agent.cfgの以下の行を変更します。

```
JAVA_HOME=/usr
```

Linux版Hinemosエージェントが利用するJREを変更したい場合は、上記の値を変更してください。

JAVA_HOMEの書き換えた場合、Hinemosエージェントの再起動が必要です。

- Windows版Hinemosエージェント

Windows版は独自のJava Runtimeのパッケージがバンドルされているため、バンドルされているJRE以外のものは使用できません。

- Hinemosリッチクライアント

リッチクライアントは独自のJava Runtimeのパッケージがバンドルされているため、バンドルされているJRE以外のものは使用できません。

- Webクライアント

HinemosWebクライアントのJAVA_HOMEを変更する場合、 /opt/hinemos_web/conf/hinemos_web.cfgの以下の行を変更します。

```
export JAVA_HOME=/usr/lib/jvm/jre-1.7.0-openjdk
```

HinemosWebクライアントが利用するJREを変更したい場合は、上記の値を変更してください。

JAVA_HOMEの書き換えた場合、HinemosWebクライア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、Hinemosクライアントのメンテナンスパースペクティブ→メンテナンス[Hinemosプロパティ]ビューを開き、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
```

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。メール通知の設定はメンテナンスパースペクティブから設定します。

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。

```
mail.smtp.host=[SMTPサーバのIPアドレスまたはホスト名]
mail.smtp.port=[SMTPサーバの待ち受けポート番号]
mail.transport.tries=[SMTPサーバ送信の最大試行回数]
mail.transport.tries.interval=[SMTPサーバ送信の試行間隔[msec]]
mail.from.address=[送信元メールアドレス]
mail.from.personal.name=[送信先個人名]
mail.reply.to.address=[返信先メールアドレス]
mail.reply.personal.name=[返信先個人名]
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]
mail.charset.address=UTF-8
mail.charset.subject=UTF-8
mail.charset.content=UTF-8
```

mail.*.addressに関するパラメータについては、RFC822、RFC1034に準拠したフォーマットで指定して下さい。RFC822、RFC1034に準拠していない主な例としては、下記が挙げられます。

- ・ドメイン名に英数字、「-」、「.」以外の文字を含む
- ・ドメイン名に連続する「.」が含まれる
- ・ドメイン名の最初や最後が「-」、「.」である

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、Hinemosクライアントのメンテナンスパースペクティブ→メンテナンス[Hinemosプロパティ]ビューを開き、以下のパラメータを設定します。

```
mail.smtp.auth=[true:認証を行う場合, false:認証を行わない場合]
mail.transport.user=[認証において使用されるユーザ名]
mail.transport.password=[認証において使用されるパスワード]
```

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、DIGEST-MD5の順で選択されます。

5.2.3 SSL/TLSの設定

メール通知において、SSL/TLS(STARTTLS)を必要とするSMTPサーバを利用するには、Hinemosクライアントのメンテナンスパースペクティブ→メンテナンス[Hinemosプロパティ]ビューを開き、以下のパラメータを変更します。

```
mail.smtp.starttls.enable=true
```

なお、STARTTLSを使用する場合には、HinemosマネージャサーバがSSL/TLS接続を確立するため、接続先のメールサーバより発行されるセキュリティ証明書を信頼済証明書として検証する必要があります。

詳細については、[HTTPS監視](#)をご参照ください。

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、Hinemosクライアントのメンテナンスパースペクティブ→メンテナンス[Hinemosプロパティ]ビューを開き、以下のパラメータを変更します。

```
mail.smtp.from=[エンベロープFromとするメールアドレス]
```

5.3 ログエスカレーション通知

ログエスカ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

Hinemosクライアントのメンテナンスパースペクティブ→メンテナンス[Hinemosプロパティ]ビューを開き、パラメータ `notify.log.escalate.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。

```
notify.log.escalate.manager.hostname=#[NODE]
```

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

Hinemosプロパティの設定値	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...
#[FACILITY_ID]	(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...
#[NODE]	(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 hostnameコマンドの実行結果

2 オーナー別スコープ、登録ノードすべて、未登録ノード、Hinemos内部スコープ

5.3.2 転送プロトコルにTCPを使用する場合の設定方法

Hinemosクライアントのメンテナンスパースペクティブ→メンテナンス[Hinemosプロパティ]ビューを開き、パラメータ `notify.log.escalate.manager.protocol` の値をTCPへ変更することにより、転送プロトコルにTCPを指定することができます。

```
notify.log.escalate.manager.protocol=tcp
```

6 監視設定/性能機能

Hinemos 監視設定/性能機能の追加設定について説明します。

6.1 SQL監視

6.1.1 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 7.0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/plugins/ に配置します。

```
$ cp new_jdbc.jar /opt/hinemos/plugins/
```

2. Hinemosプロパティの編集

追加するRDBMSの情報を Hinemosクライアントのメンテナンスパースペクティブ→メンテナンス[Hinemosプロパティ]ビューから追加します。

まず、HinemosのSQL監視で利用できるRDBMSの種類の数 monitor.sql.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）

```
monitor.sql.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。（Hinemosプロパティ種別は文字列を選択してください。）

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

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。

6.2 プロセス監視

6.2.1 プロセス監視の動作

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 6-1 参照）

1. SNMP/WBEM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

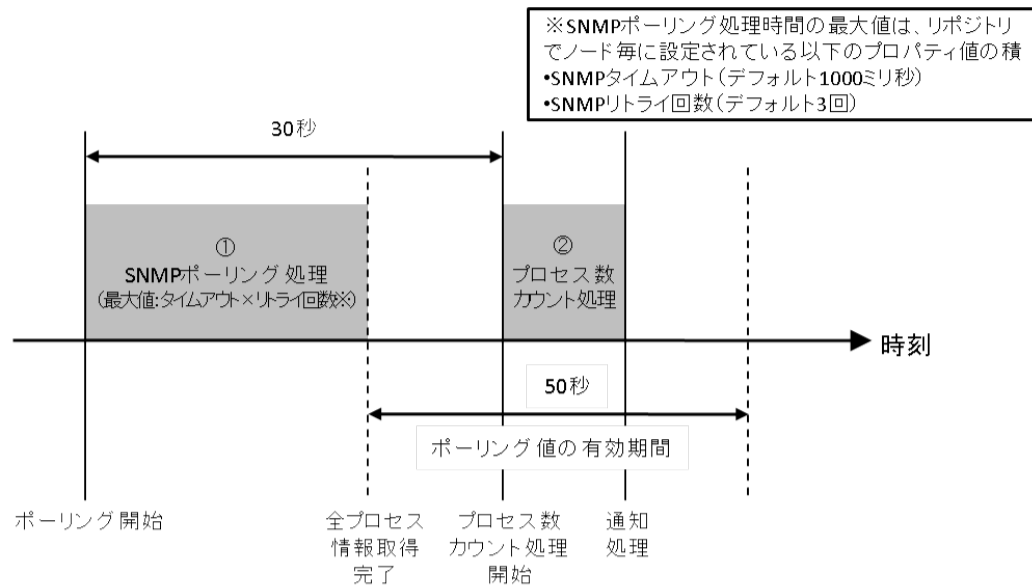


図 6-1 プロセス監視の処理

①が開始されてから30秒後に②が実行されます。①で収集したプロセス一覧は50秒間有効な情報として扱われます。

①のSNMP/WBEM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/WBEM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。

6.3 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。

6.3.1 サーバ証明書を利用しない場合のHTTPS設定

HTTPS接続をする場合は、サーバ証明書の必須有無が設定できます。サーバ証明書を必須とする場合は、次節を参照してください。本節では、サーバ証明書不要の手順を記載します。

Hinemosクライアントのメンテナンスパースペクティブを開き、メンテナンス[Hinemosプロパティ]ビューにある作成ボタンをクリックします。Hinemosプロパティ種別の「真偽値」を指定し、以下の通りに設定してください。

```
キー:monitor.http.ssl.trustall
値:true
説明:任意
```

(注) デフォルトはtrueとなっています。

6.3.2 サーバ証明書を利用する場合のHTTPS設定

サーバ証明書を利用する場合、HinemosマネージャがSSL接続を確立するため、接続先のHTTPSサーバのセキュリティ証明書を信頼済証明書として検証できる必要があります。

1. サーバ証明書をwell known public CAで署名している場合

OS同梱のOpenJDKのkeystoreは、well known CAの公開鍵を保持しているため、well known CA証明書まで順に証明書をたどり、対象のサーバ証明書を検証できます。この場合、well known public CAで署名されたサーバ証明書を信頼済証明書として検証できるため、Hinemosマネージャのkeystoreに、サーバ証明書を登録する必要はありません。

2. サーバ証明書を自己署名している場合

Hinemosマネージャのkeystoreにサーバ証明書を登録することで、対象のサーバ証明書を信頼済証明書として検証できます。

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. java起動オプションによるkeystoreファイルの指定

(注1)Hinemosマネージャの再起動は必要ありません。但し手順3の「java起動オプションによるkeystoreファイルの指定」を実施する場合は、Hinemosマネージャの再起動が必要になります。

(注2)証明書を利用する場合「6.3.1 証明書を利用しない場合のHTTPS設定」を参照し、「monitor.http.ssl.trustall」の値を「false」に設定する必要があります。

手順1～3の詳細については以下の通りです。

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。

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。

```
(hinemos) $ su - root
(root) # /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://linux.die.net/man/1/keytool-java-1.7.0-openjdk> を参照して下さい。

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(java)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/hinemos.cfgのjavax.net.ssl.trustStoreを設定するJVM_KEYSTORE_OPTSのコメントを外します。

```
### JVM - keystore (https)
#export JVM_KEYSTORE_OPTS="-Djavax.net.ssl.trustStore=${HINEMOS_HOME}/.keystore"
```

※設定変更後、Hinemosマネージャの再起動が必要です。

(注) /opt/hinemos/.keystoreからサーバ証明書を削除する場合は、以下のコマンドを実行します。

```
(root) # /usr/bin/keytool -delete -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: (証明書の登録時に入力したパスワード)
```

6.4 システムログ監視

6.4.1 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
```

6.4.2 syslogの受信に関する設定

マネージャサーバではrsyslogを使ってsyslogを受信します。 rsyslogが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.d/rsyslog_hinemos_manager.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(java)にsyslogを転送する際に使用する 待ち受けアドレス、ポート番号について、 Hinemosクライアントのメンテナンスパースペクティブ→メンテナンス[Hinemosプロパティ]ビューから変更できます。

```
monitor.systemlog.listen.address=[javaプロセスがsyslogを受信する待ち受けアドレス]
monitor.systemlog.listen.port=[javaプロセスがsyslogを受信する待ち受けポート番号]
```

この設定変更を反映させる場合は、Hinemosマネージャの再起動が必要です。

6.4.3 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(java)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 6-1 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 6-2 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.d/rsyslog_hinemos_manager.confを編集します。
/etc/rsyslog.d/rsyslog_hinemos_manager.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
```

6.4.4 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
```

6.5 SNMPTRAP監視

6.5.1 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、Hinemosクライアントのメンテナンスパースペクティブ→メンテナンス[Hinemosプロパティ]ビューを開き、以下のパラメータを変更します。

```
monitor.snmptrap.listen.address=[javaプロセスがSNMPTRAPを受信する待ち受けアドレス]
monitor.snmptrap.listen.port=[javaプロセスがSNMPTRAPを受信する待ち受けポート番号]
```

6.5.2 デフォルトMIB

Hinemosマネージャにデフォルトで登録されているMIBを表6-3に示します。

表6-3 Hinemosマネージャにデフォルトで登録されたMIBの一覧

GENERIC TRAP	A3COM-SWITCHING-SYSTEMS-FDDI-MIB
Centrum-MIB	A3Com-DLSW-r1-MIB
LANPLEX-SYSTEMS-MIB	LBHUB-ECS-MIB
SYNC-RESEARCH-MIB	A3Com-SdIc-r1-MIB
SECURITY-MIB	A3Com-System-r8-MIB
A3COM-SWITCHING-SYSTEMS-POLL-MIB	A3COM-SWITCHING-SYSTEMS-QOS-MIB
A3COM-SWITCHING-SYSTEMS-BRIDGE-MIB	A3COM-SWITCHING-SYSTEMS-MIB
CHIPCOMMIB	CHIPCOM-MIB
PRODUCTMIB	A3COM0007-SYSLOADER
LANPLEX-MIB	LBHUB-BLC-MIB
LBHUB-BRIDGE-MIB	LBHUB-MSH-MIB
LINKB-OPT-FDDI-MIB	LB3GH-1-0-7
LANPLEX-OPT-FDDI-MIB	NCDCHASS-MIB
SWITCHING-SYSTEMS-MIB	USR-TRAP-MIB
VRRP-MIB	ADICLIBMIB

IBM-AIX-MIB	SPAGENT-MIB
BESTPOWER-MIB	XUPS-MIB
PowerNet-MIB	ATM-FORUM-ILMI40-MIB
DPT-SCSI-MIB	ADTRAN-ATLAS-550-MIB
ADTRAN-ATLAS-HSSI-V35-MIB	ADTRAN-ATLAS-MODULE-MIB
ADTRAN-ATLAS-T1-MIB	ADTRAN-ATLAS-UNIT-MIB
ADTRAN-ATLAS-V35NX-MIB	ADTRAN-GENCHASSISTRAP-MIB
Aedilis-MIB	AirDefense-Product-MIB
ALCATEL-IND1-CHASSIS-MIB	ALCATEL-IND1-GROUP-MOBILITY-MIB
ALCATEL-IND1-GVRP-MIB	ALCATEL-IND1-HEALTH-MIB
ALCATEL-IND1-INLINE-POWER-MIB	ALCATEL-IND1-INTERSWITCH-PROTOCOL-MIB
ALCATEL-IND1-IP-MIB	ALCATEL-ISIS-MIB
ALCATEL-IND1-LAG-MIB	ALCATEL-IND1-LPS-MIB
ALCATEL-IND1-MAC-ADDRESS-MIB	ALCATEL-IND1-NETSEC-MIB
ALCATEL-IND1-PIM-BSR-MIB	ALCATEL-IND1-PIM-STD-MIB
ALCATEL-IND1-POLICY-MIB	ALCATEL-IND1-PORT-MIB
ALCATEL-IND1-PORT-MIRRORING-MONITORING-MIB	ALCATEL-IND1-SESSION-MGR-MIB
ALCATEL-IND1-SLB-MIB	ALCATEL-IND1-STACK-MANAGER-MIB
ALCATEL-IND1-TRAP-MGR-MIB	ALCATEL-IND1-UDLD-MIB
ALCATEL-IND1-VLAN-STP-MIB	ALCATEL-IND1-VRRP3-MIB
ALCATEL-IND1-WCCP-MIB	ALCATEL-IND1-WEBMGT-MIB
LUCENT-SECURE-VPN-SOLUTIONS-LSMS-NOTIFICATION-MIB	ALLOT-NX-MIB
ALLOT-SMP-SNMP-MIB	ALLOT-MIB
ALTEON-TRAP-MIB	CHEETAH-TRAP-MIB
ALTIGA-trap-event	WLSR-AP-MIB
WLSX-SWITCH-MIB	WLSX-TRAP-MIB
ASCEND-TRAP	AcBoard
ALARM-MIB	DS1-MIB
ENTITY-MIB	IF-MIB
RMON-MIB	RTCPXR-MIB
SNMPv2-MIB	ACS-TRAP-MIB
CYCLADES-ACS5000-TRAP-MIB	AMX5000-TRAP-MIB
AMX5010-TRAP-MIB	AMX5020-TRAP-MIB
AMX5030-TRAP-MIB	AVCT-CCM-TRAP-MIB
DSR-TRAP-MIB	DSR1021-TRAP-MIB
DSR2010-TRAP-MIB	AVOCENT-MERGEPOINT-TRAP-MIB
PM-TRAP-MIB	BEA-Weblogic-Server-Startup-Event
BEA-Weblogic-Server-Shutdown-Event	BEA-Weblogic-Attribute-Change-Received-Event
BEA-Weblogic-Monitor-Trap-Received-Event	BEA-Weblogic-Log-Filter-Trap-Received-Event
BGP4-MIB	RFC1269-MIB
Backup-Exec-MIB	BLACKBERRYSERVERMIB
BLACKBERRYSERVER-MIB	BNT-GbESM-10Ub-RS-MIB
BLUECOAT-DIRECTOR-TRAP-MIB	BLUECOAT-SG-ATTACK-MIB

BLUECOAT-SG-DISK-MIB	BLUECOAT-SG-HEALTHCHECK-MIB
BLUECOAT-SG-HEALTHMONITOR-MIB	BLUECOAT-SG-POLICY-MIB
BLUECOAT-SG-SENSOR-MIB	BLUECOAT-SG-USAGE-MIB
BLUECOAT-AV-MIB	ATTACK-MIB
POLICY-MIB	SENSOR-MIB
USAGE-MIB	ADONIS-DNS-MIB
FCMGMT-MIB	SW-TRAP
ARCserve-Alarm-MIB	CPQN54NN-MIB
CPQCLUSTER-MIB	CPQCMC-MIB
CPQCR-MIB	CPQDMII-MIB
CPQDSCCS-MIB	CPQFCA-MIB
CPQGEN-MIB	CPQHLTH-MIB
CPQHOST-MIB	CPQHSV110V3-MIB
CPQICA-MIB	CPQIDA-MIB
CPQIDE-MIB	CPQSINFO-MIB
CPQN5226A-MIB	CPQNIC-MIB
CPQPOWER-MIB	CPQRACK-MIB
CPQRECOV-MIB	CPQRPM-MIB
CPQSANAPP-MIB	CPQSANEVENT-MIB
CPQSCSI-MIB	CPQSERVICE-MIB
CPQSM2-MIB	CPQSRVMN-MIB
CPQSTDEQ-MIB	CPQSTSYS-MIB
CPQSWCC-MIB	CPQTHRSH-MIB
CPQ-TRAPS-MIB	CPQUPS-MIB
CPQWCRM-MIB	CPQOS-MIB
CRITAPP-MIB	CISCO-CIDS-MIB
PCUBE-SE-MIB	CISCO-SERVICE-CONTROL-RDR-MIB
CISCO-SERVICE-CONTROL-LINK-MIB	CISCO-SERVICE-CONTROL-SUBSCRIBERS-MIB
AIRESPACE-WIRELESS-MIB	CISCO-LWAPP-DOT11-CLIENT-MIB
ACCOUNTING-CONTROL-MIB	ADSL-LINE-MIB
APPN-MIB	APPN-TRAP-MIB
ATM-SOFT-PVC-MIB	AWCVX-MIB
CISCO-5800-HEALTH-MON-MIB	CISCO-6400-CHASSIS-MIB
Cisco90Series-MIB	CISCO-AAA-SERVER-MIB
CISCO-ACCESS-ENVMON-MIB	CISCO-ALPS-MIB
CISCO-APS-MIB	CISCO-ATM-DUAL-PHY-MIB
CISCO-ATM-NETWORK-CLOCK-MIB	CISCO-BSTUN-MIB
CISCO-C2900-MIB	CISCO-C3800-MIB
CISCO-C8500-REDUNDANCY-MIB	CISCO-CALL-TRACKER-MIB
CISCO-CASA-FA-MIB	CISCO-CASA-MIB
CISCO-CCM-MIB	CISCO-CHANNEL-MIB
CISCO-CIPCMPC-MIB	CISCO-CIPCSNA-MIB
CISCO-CLUSTER-MIB	CISCO-CONFIG-COPY-MIB

CISCO-CONFIG-MAN-MIB	CISCO-CONTENT-ENGINE-MIB
CISCO-CONTENT-NETWORK-MIB	CISCO-DLSW-EXT-MIB
CISCO-DLSW-MIB	CISCO-DOCS-EXT-MIB
CISCO-DOCS-REMOTE-QUERY-MIB	cdspMIB
CISCO-DSPU-MIB	CISCO-ENTITY-ALARM-MIB
CISCO-ENTITY-FRU-CONTROL-MIB	CISCO-ENTITY-SENSOR-MIB
CISCO-ENVMON-MIB	CISCO-EVENT-DISTR-MIB
CISCO-FASTHUB-MIB	CISCO-FIREWALL-MIB
CISCO-FLASH-MIB	CISCO-GATEKEEPER-MIB
CISCOTRAP-MIB	CISCO-GPRS-GTP-MIB
CISCO-GPRS-L2RLY-MIB	CISCO-HSRP-MIB
CISCO-ICSUDSU-MIB	CISCO-IETF-ATM2-PVCTRAP-MIB
CISCO-IF-THRESHOLD-MIB	CISCO-IP-ENCRYPTION-MIB
GWPOA-MIB	CISCO-IPMROUTE-MIB
CISCO-IPSEC-FLOW-MONITOR-MIB	CISCO-IPSEC-MIB
CISCO-ISDN-MIB	CISCO-ISDNU-IF-MIB
CISCO-ITP-SCCP-MIB	CISCO-ITP-SP-MIB
CISCO-LOCAL-DIRECTOR-MIB	CISCO-NETWORK-REGISTRAR-MIB
CISCO-OAM-MIB	CISCO-OPTICAL-PATCH-MIB
CISCO-OSCP-MIB	CISCO-PIM-MIB
CISCO-PING-MIB	CISCO-POP-MGMT-MIB
CISCO-PPPOE-MIB	CISCO-REPEATER-MIB
CISCO-RF-MIB	CISCO-RHINO-MIB
CISCO-RSRB-MIB	CISCO-RTTMON-MIB
CISCO-SDLLC-MIB	CISCO-SIBU-MANAGERS-MIB
CISCO-SIBU-STACKABLE-DUAL-SPEED-HUB-MIB	CISCO-SLB-MIB
CISCO-SNA-LLC-MIB	CISCO-SP-MIB
CISCO-SRP-MIB	CISCO-STACK-MIB
CISCO-STP-EXTENSIONS-MIB	CISCO-STUN-MIB
CISCO-SYSLOG-MIB	CISCO-SYSTEM-MIB
CISCO-TS-STACK-MIB	CISCO-VLAN-MEMBERSHIP-MIB
CISCO-VOICE-APPS-MIB	CISCO-VOICE-DIAL-CONTROL-MIB
CISCO-VTP-MIB	CISCO-WIRELESS-DOCS-EXT-MIB
CISCO-WIRELESS-IF-MIB	CISCO-WIRELESS-P2MP-LINK-METRICS-MIB
CISCO-WIRELESS-P2MP-PHY-MIB	CISCO-WIRELESS-P2MP-RF-METRICS-MIB
STAND-ALONE-ETHERNET-SWITCH-MIB	MADGERSW-MIB
METRO1500-MIB	CISCO-ES-STACK-MIB
CISCO-MVPN-MIB	CISCO-ATM-IF-MIB
CISCO-ATM-PVCTRAP-EXTN-MIB	CISCO-BBSM-MIB
CISCO-BGP4-MIB	CISCO-BULK-FILE-MIB
CISCO-CABLE-AVAILABILITY-MIB	CISCO-CABLE-METERING-MIB
CISCO-CABLE-QOS-MONITOR-MIB	CISCO-CABLE-SPECTRUM-MIB
CISCO-CALLHOME-MIB	CISCO-CAT6K-CROSSBAR-MIB

CISCO-CDL-MIB	CISCO-CDMA-AHDLC-MIB
CISCO-CDMA-PDSN-MIB	CISCO-CSG-MIB
CISCO-DDP-IAPP-MIB	CISCO-DEVICE-EXCEPTION-REPORTING-MIB
CISCO-DIST-DIRECTOR-MIB	CISCO-DM-MIB
CISCO-DOT11-CONTEXT-SERVICES-MIB	CISCO-ENTITY-PFE-MIB
CISCO-EPM-NOTIFICATION-MIB	CISCO-EXT-SCSI-MIB
ciscoFabricC12kMIB	CISCO-FABRIC-HFR-MIB
CISCO-FC-FE-MIB	CISCO-FCC-MIB
CISCO-FCPING-MIB	CISCO-FCS-MIB
CISCO-FCTRACEROUTE-MIB	CISCO-FDMI-MIB
CISCO-FEATURE-CONTROL-MIB	CISCO-FSPF-MIB
CISCO-GGSN-MIB	CISCO-GPRS-ACC-PT-MIB
CISCO-GPRS-CHARGING-MIB	CISCO-GTP-DIRECTOR-MIB
CISCO-GTP-MIB	CISCO-HC-ALARM-MIB
CISCO-HEALTH-MONITOR-MIB	CISCO-IETF-DOT11-QOS-EXT-MIB
CISCO-IETF-PW-MIB	CISCO-IETF-SCTP-EXT-MIB
CISCO-IETF-VDSL-LINE-MIB	CISCO-IMAGE-UPGRADE-MIB
CISCO-IP-LOCAL-POOL-MIB	CISCO-ISCSI-MIB
CISCO-ITP-GRT-MIB	CISCO-ITP-GSCCP-MIB
CISCO-ITP-GSP-MIB	CISCO-ITP-MLR-MIB
CISCO-ITP-MONITOR-MIB	CISCO-ITP-RT-MIB
CISCO-ITP-XUA-MIB	CISCO-IVR-MIB
CISCO-L2-CONTROL-MIB	CISCO-L2-DEV-MONITORING-MIB
CISCO-LICENSE-MGR-MIB	CISCO-MAC-NOTIFICATION-MIB
CISCO-MOBILE-IP-MIB	CISCO-MODULE-AUTO-SHUTDOWN-MIB
CISCO-NBAR-PROTOCOL-DISCOVERY-MIB	CISCO-NMS-APPL-HEALTH-MIB
CISCO-NS-MIB	CISCO-OPTICAL-MONITOR-MIB
CISCO-OSPF-TRAP-MIB	CISCO-OUTAGE-MONITOR-MIB
CISCO-PORT-SECURITY-MIB	CISCO-PORT-STORM-CONTROL-MIB
CISCO-PSA-MICROCODE-MIB	CISCO-PSM-MIB
CISCO-RPMS-MIB	CISCO-RSCN-MIB
CISCO-SCSI-MIB	CISCO-SLB-EXT-MIB
CISCO-SONET-MIB	CISCO-SSG-MIB
CISCO-SSL-PROXY-MIB	CISCO-SYS-INFO-LOG-MIB
CISCO-SYSTEM-EXT-MIB	cTapMIB
CISCO-TAP-MIB	CISCO-VIRTUAL-NW-IF-MIB
CISCO-VISM-TRAPS-MIB	CISCO-VOICE-DNIS-MIB
CISCO-VPDN-MGMT-MIB	CISCO-VSAN-MIB
CISCO-WAN-TOPOLOGY-MIB	CISCO-WLAN-VLAN-MIB
CISCO-WWNMGR-MIB	CISCO-ZS-MIB
CISCOWORKS-MIB	CISCO-SME-MIB
CISCO-SLB-HEALTH-MON-MIB	CISCO-STACKWISE-MIB
CISCO-ENHANCED-SLB-MIB	CISCO-MODULE-VIRTUALIZATION-MIB

CLARENT-MIB	CSI-P2-MIB
COLUBRIS-802DOT11-MIB	COLUBRIS-MAINTENANCE-MIB
COLUBRIS-PUBLIC-ACCESS-MIB	COLUBRIS-PUBLIC-ACCESS-RETENTION-MIB
COLUBRIS-SATELLITE-MANAGEMENT-MIB	COLUBRIS-SYSLOG-MIB
COLUBRIS-SYSTEM-MIB	COLUBRIS-TOOLS-MIB
COLUBRIS-VPN-MIB	SERVERVANTAGE-TRAP-MIB
CDM-625	DiagnosticsMonitor
CXC-MIB	Cricket-Threshold-exceeded
Cricket-Threshold-cleared	Crossbeam-Hardware-Event
Crossbeam-Module-Event	Crossbeam-VAP-Group-Event
Crossbeam-VRRP-Event	DISMAN-PING-MIB
DISMAN-EVENT-MIB	DMTF-DMI-MIB
DMTF-MOBILE-MIB	DMTF-SYSTEMS-MIB
DPS-MIB	VM-MIB
MIB	ArrayManager-MIB
DELL_ASF-MIB	StorageManagement-MIB
DELL-RAC-MIB	INTEL-LAN-ADAPTERS-MIB
EMC-CELERRA	CLARIION-MIB
EMC-MIB	A3COM51-SS9000SX
EXTREME-CABLE-MIB	EXTREME-CLEARFLOW-MIB
EXTREME-DOS-MIB	EXTREME-ENH-DOS-MIB
EXTREME-ESRP-MIB	EXTREME-IP-SECURITY-MIB
EXTREME-PORT-MIB	EXTREME-SOFTWARE-MONITOR-MIB
EXTREME-STACKING-MIB	EXTREME-SYSTEM-MIB
EXTREME-TRAP-MIB	EXTREME-UPM-MIB
EXTREME-V2TRAP-MIB	EXTREME-WIRELESS-MIB
LOAD-BAL-SYSTEM-MIB	WAN-TRAP-MIB
F5-3DNS-MIB	F5-BIGIP-COMMON-MIB
Fore-Switch-MIB	FORTIOS-300-MIB
FOUNDRY-SN-TRAP-MIB	SNI-HD-MIB
SNI-MYLEX-MIB	SNI-SERVER-CONTROL-MIB
FSC-SERVERCONTROL2-MIB	SERVERVIEW-STATUS-MIB
SERVERVIEW-DUPLEXDATAMANAGER-MIB	SIEMENS-DUPLEXWRITE-MIB
DESKTRAP-MIB	SIEMENS-MULTIPATH-MIB
SNI-NT-CLUSTER-MIB	PCI-HOTPLUG-MIB
SNI-SERVERVIEW-MIB	SNI-TRAP-MIB
FSC-HACL-MIB	WSA-TRAP-MIB
GGSN-MIB	GWAPIMIB
GWADA-MIB	NGWASYNC
GWIAMIB	GWMTA-MIB
GWOVMMIB	GWPAGERMIB
GWSMTMIB	GWSNADSMIB
NGWX400MIB	HPNSAECC-MIB

HP-ENTITY-MIB	HP-httpManageable-MIB
HP-ICF-8023-RPTR	HP-ICF-BASIC
HP-ICF-CHAIN	HP-ICF-CHASSIS
HP-ICF-FAULT-FINDER-MIB	HP-ICF-GENERIC-RPTR
HP-ICF-VG-RPTR	ICF-VG-RPTR
HP-MCSG	HP-SN-TRAP-MIB
JETDIRECT3-TRAP	HPNSATRAP-MIB
TapeAlert-MIB	UMSEVENT-MIB
IBM-Director-Alert-MIB	IBM-SERVERAID-MIB
Converged-Power-System-Trap	RSASPPALT-MIB
IEEE802dot11-MIB	IPUNITY-SES-MIB
IPV6-MIB	ISS-MIB
INTEL-GEN-MIB	INTEL-S500-MIB
RMM2-MIB	PET-MIB
AOLAN-MIB	PET_EVENTS
I3IC-MIB	ASYN COS-MAIL-MIB
Juniper-System-MIB	Juniper-CLI-MIB
Juniper-RADIUS-CLIENT-MIB	Juniper-System-Clock-MIB
Juniper-ADDRESS-POOL-MIB	Juniper-REDUNDANCY-MIB
Juniper-MROUTER-MIB	BGP4-V2-MIB
JUNIPER-CFGMGMT-MIB	JUNIPER-MIB
JUNIPER-LDP-MIB	MPLS-MIB
JUNIPER-MPLS-LDP-MIB	JUNIPER-PING-MIB
JUNIPER-PMon-MIB	JUNIPER-RMON-MIB
JUNIPER-SONET-MIB	APS-MIB
JUNIPER-VPN-MIB	JUNIPER-USER-AAA-MIB
JUNIPER-COLLECTOR-MIB	JUNIPER-SP-MIB
OSPFV3-MIB	JUNIPER-SYSLOG-MIB
JUNIPER-CHASSIS-CLUSTER-MIB	JUNIPER-JS-AUTH-MIB
JUNIPER-V1-TRAPS-BGP	JUNIPER-V1-TRAPS-CHAS
JUNIPER-V1-TRAPS-MPLS	JUNIPER-V1-TRAPS-OSPF
LLDP-MIB	LIEBERT-SERIES-600-UPS-MODULE-MIB
LIEBERT-GP-AGENT-MIB	LIEBERT-GP-NOTIFICATIONS-MIB
Linksys-Connection-Trap	AGG-TRAP-MIB
CDR-TRAP-MIB	EXCEL-SWITCH-MIB
FC-TRAP-MIB	H323-TRAP-MIB
MANTRA-TRAP-MIB	PSAX-TRAP-MIB
RM-TRAP-MIB	SIP-TRAP-MIB
SPINS-TRAP-MIB	MG-SNMP-UPS-MIB
MPLS-VPN-MIB	MPLS-L3VPN-STD-MIB
MPLS-LSR-STD-MIB	MPLS-TE-STD-MIB
DEV-CFG-MIB	MSCR-MIB
VLAN-MIB	NSTACK-MIB

OADWDM-MIB	OA-VDSL-MIB
OA-VOICE-MIB	OAATERESCOPE-MIB
DRAFT-MSDP-MIB	MADGECAL-MIB
TVD-MIB	mcafee_EVENT_NEW_MIB
LanMgr-Alerts-II-MIB	PIM-MIB
MSDP-MIB	MYLEXRAID-MIB
CONTIVITY-TRAPS-V1-MIB	NETWORK-APPLIANCE-MIB
NET-SNMP-AGENT-MIB	NETBOTZ-MIB
NETGEAR-SWITCHING-MIB	NS-ROOT-MIB
NETSCREEN-TRAP-MIB	NOKIA-ENHANCED-SNMP-SOLUTION-SUITE-ALARM-IRP
NOKIA-ENHANCED-SNMP-SOLUTION-SUITE-PM-IRP	NOKIA-IPSO-LBCLUSTER-MIB
NOKIA-IPSO-SYSTEM-MIB	DHCP-MIB
IPX	Novell-Directory-Services-Trap-MIB
Windows-NT-Server-Trend-MIB	NetWare-Server-Alarm-MIB
NetWare-Server-Trend-MIB	NWTRAPCONFIGURATION
OSPF-TRAP-MIB	OPENNMS-MIB
RDBMS-MIB	ORALISTENER-MIB
ORAINTERCHANGE-MIB	ORACLE-AGENT-MIB
ORACLE-ENTERPRISE-MANAGER-4-MIB	OVERTURE-FAULTS-MIB
BLUECOAT-PACKETSHAPER-MIB	PATROL-MIB
SIPXECs-ALARM-NOTIFICATION-MIB	PIXMET-ATM-MIB
PIXMET-COFDM-MIB	PIXMET-DVBT-MIB
PIXMET-DVSTATION-MIB	PIXMET-IQ-MIB
PIXMET-QAM-MIB	PIXMET-QMM-MIB
PIXMET-QPSK-MIB	PIXMET-SLF-MIB
PIXMET-TSP-MIB	PIXMET-VSB-MIB
POLYCOM-VIDEO-MIB	CV-MIB
RADLAN-MIB	RAPID-CITY
RFC1382-MIB	UPS-MIB
RANCID-CUSTOM-MIB	REDLINE-TRAPv2-MIB
AVTC-COMMON-MIB	STEELHEAD-MIB
SNA-NAU-MIB	SNMP-REPEATER-MIB
SENSAPHONE-MIB	Sentry3-MIB
SNORT-INTRUSION-DETECTION-ALERT-MIB	SONICWALL-FIREWALL-TRAP-MIB
SONUS-COMMON-MIB	SONUS-TRUNK-GROUP-RESOURCES-MIB
SONUS-SYSTEM-TIMING-MIB	SONUS-SOFTSWITCH-CLIENT-SERVICES-MIB
SONUS-SS7-SERVICES-MIB	SONUS-SS7-MTP3-MIB
SONUS-SS7-MTP2-MIB	SONUS-SONET-MIB
SONUS-SOFTWARE-UPGRADE-SERVICES-MIB	SONUS-SIP-SIGNALING-MIB
SONUS-RTCP-MIB	SONUS-REDUNDANCY-SERVICES-MIB
SONUS-OSPF-MIB	SONUS-NTP-SERVICES-MIB
SONUS-IP-INTERFACE-MIB	SONUS-NODE-RESOURCES-MIB
SONUS-NODE-MIB	SONUS-MASTER-TRUNK-RESOURCE-MANAGER-MIB

SONUS-MGCP-SERVICES-MIB	SONUS-LOG-STREAMING-SERVICES-MIB
SONUS-JAPANST-MIB	SONUS-ISUP-SERVICE-GROUP-MIB
SONUS-ISDN-SERVICE-GROUP-MIB	SONUS-IP-FILTER-MIB
SONUS-H323-SIGNALING-MIB	SONUS-GATEWAY-SIGNALING-MIB
SONUS-EVENT-LOG-MIB	SONUS-DS3-MIB
SONUS-DS3THRESHOLD-MIB	SONUS-DS1-MIB
SONUS-DS1THRESHOLD-MIB	SONUS-DSP-RESOURCES-MIB
SONUS-CAS-MIB	SONUS-ACCOUNTING-SERVICES-MIB
SONUS-COMMON-CALL-PROCESS-MIB	SONUS-BT-SERVICE-GROUP-MIB
SONUS-ATM-EXTENSIONS-MIB	SONUS-ANNOUNCEMENT-RESOURCES-MIB
SONUS-APS-MIB	SONUS-SONET-APS-MIB
SONUS-ALARM-CONTACT-MIB	IPOA-MIB
ATM2-MIB	SONUS-DATASTREAM-INTEGRATOR-MIB
SONUS-DSI-TRANSPORTER-MIB	SONUS-AGT-SGX-EVENT-MIB
SONUS-HA-MIB	SONUS-HSX-MIB
SONUS-SOFTSWITCH-DBREP-MIB	SONUS-SOFTSWITCH-PIPE-MIB
SONUS-SOFTSWITCH-POLICY-EXECUTION-SERVER-MIB	SONUS-SOFTSWITCH-PROXY-GATEKEEPER-MIB
SONUS-SOFTSWITCH-SCPA-MIB	SONUS-SOFTSWITCH-SIP-ENGINE-MIB
SONUS-SOFTSWITCH-SSREQ-MIB	BRIDGE-MIB
DIAL-CONTROL-MIB	DLSW-MIB
DOCS-CABLE-DEVICE-TRAP-MIB	DS3-MIB
IMA-MIB	ISDN-MIB
PTOPO-MIB	RFC1315-MIB
SNA-SDLC-MIB	TN3270E-RT-MIB
SUN-PLATFORM-MIB	SUN-HW-TRAP-MIB
SWISSQUAL-NQAGENT-MIB	SYMBOL-CC-WS2000-MIB
SYMBOL-DSSS-ENTERPRISE-PRIVATE-MIB	SYMBOL-WS5000-MIB
EMPIRE	TUT-T2-MIB
TRIPPUPS-MIB	UPTIME-ROOT-MIB
CCU3000PMAC-TRAPS-MIB	WBSN-APPLIANCE-MIB
XEROX-HOST-RESOURCES-EXT-MIB	XEROX-JOB-MONITORING-EXT-MIB
XEROX-JOB-MONITORING-MIB	XEROX-RESOURCES-MIB
XEROX-SERVICE-MONITORING-MIB	XEROX-SIMPLE-JOB-MGMT-MIB
IPVREMS-MIB	

6.6 Windowsサービス監視

6.6.1 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、WinRM 2.0、WinRM 3.0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、Windows 7の場合はWinRM 2.0が、Windows Server 2012, 2012 R2、Windows 8, 8.1, 10の場合はWinRM 3.0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。

インストールされているWinRMのバージョンを確認するには、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ます。実行結果最終行の"Stack"の値がバージョン番号です。

```
> winrm id
IdentifyResponse
  ProtocolVersion = http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd
  ProductVendor = Microsoft Corporation
  ProductVersion = OS: 6.1.7600 SP: 0.0 Stack: 2.0
```

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。

6.6.2 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, WinRM 3.0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
```

6.6.3 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
```

6.6.4 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
```

6.6.5 サーバ証明書を利用しない場合のHTTPS設定

HTTPS接続をする場合は、サーバ証明書の必須有無が設定できます。サーバ証明書を必須とする場合は、次節を参照してください。本節では、サーバ証明書不要の手順を記載します。

Hinemosクライアントのメンテナンスパースペクティブを開き、メンテナンス[Hinemosプロパティ]ビューにある作成ボタンをクリックします。Hinemosプロパティ種別の「真偽値」を指定し、以下の通りに設定してください。

キー:monitor.winservice.ssl.trustall

値:true

説明:任意

(注) デフォルトはtrueとなっています。

登録後は、「6.6.6 証明書を利用したHTTPSの設定 4-1 リポジトリの設定」を参照してください。

(注)本設定はHinemos ver5.0.3以降で設定可能です。

Hinemos5.0.2以前でHTTPS接続を利用する場合は、「6.6.6 証明書を利用したHTTPSの設定」を参照してください。

6.6.6 サーバ証明書を利用したHTTPSの設定

Windowsサービス監視でHTTPSを利用する場合、下記の順番で設定します。

1. 証明書の作成
2. 監視対象サーバの設定
3. Hinemosマネージャの設定
4. Hinemosクライアントの設定

設定例

今回一例としてOpenSSLの独自認証局の証明書（自己署名の証明書）を使用し設定を行います。（既にサーバ証明書を持っている場合は「2. 監視対象サーバの設定」から参考に設定してください。）

1. 証明書の作成

以下のコマンドを実行します。

```
(root)# export OPENSSL_CONF=/etc/pki/tls/openssl.cnf
```

サーバ認証用の証明書にするために、/etc/pki/tls/openssl.cnf内の[v3_ca] セクションに以下の行を追記してください。

```
[ v3_ca ]
extendedKeyUsage = serverAuth ← 追記
```

以下のコマンドを実行し、任意の値を入力し証明書と秘密鍵を作成します。

```
(root)# /etc/pki/tls/misc/CA -newca

CA certificate filename (or enter to create)
(何も入力せずEnter)
Making CA certificate ...
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to '/etc/pki/CA/private/./cakey.pem'
Enter PEM pass phrase: (任意のパスワード) ※1
Verifying - Enter PEM pass phrase: (※1と同じパスワードを使用)
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:JP ←任意の値を入れてください。
State or Province Name (full name) []:Tokyo ←任意の値を入れてください。
Locality Name (eg, city) [Default City]:Tokyo ←任意の値を入れてください。
Organization Name (eg, company) [Default Company Ltd]:Company ←任意の値を入れてください。
Organizational Unit Name (eg, section) []:section ←任意の値を入れてください。
```



```
Common Name (eg, your name or your server's hostname) []:xx.xx.xx.xx ← 監視対象のIPアドレスを入れてください。  
Email Address []:sample@example.jp ← 任意のアドレスを入れてください。
```

```
Please enter the following 'extra' attributes  
to be sent with your certificate request  
A challenge password []:xxx ← 任意の値を入れてください。  
An optional company name []:company ← 任意の値を入れてください。  
Using configuration from /etc/pki/tls/openssl.cnf  
Enter pass phrase for /etc/pki/CA/private/./cakey.pem: (※1と同じパスワードを使用)  
Check that the request matches the signature  
Signature ok
```

正常に実行されると以下の様に表示されます。

```
Certificate Details:  
Serial Number: 18161293535955209561 (0xfc09e03f90354d59)  
Validity  
Not Before: May 16 07:29:47 2016 GMT  
Not After : May 16 07:29:47 2019 GMT  
Subject:  
countryName = JP  
stateOrProvinceName = Tokyo  
organizationName = Company  
organizationalUnitName = section  
commonName = xx.xx.xx.xx  
emailAddress = sample@example.jp  
X509v3 extensions:  
X509v3 Extended Key Usage:  
TLS Web Server Authentication  
X509v3 Subject Key Identifier:  
C6:34:58:EF:37:50:DB:15:0E:D4:54:3B:94:2B:19:C6:92:1D:DD:00  
X509v3 Authority Key Identifier:  
keyid:C6:34:58:EF:37:50:DB:15:0E:D4:54:3B:94:2B:19:C6:92:1D:DD:00  
  
X509v3 Basic Constraints:  
CA:TRUE  
Certificate is to be certified until May 16 07:29:47 2019 GMT (1095 days)  
  
Write out database with 1 new entries  
Data Base Updated
```

以下のコマンドを実行し、秘密鍵をpkcs12形式に変更します。

```
(root)# openssl pkcs12 -export -inkey /etc/pki/CA/private/cakey.pem -in /etc/pki/CA/cacert.pem -out /etc/pki/CA/test.p12  
  
Enter pass phrase for /etc/pki/CA/private/cakey.pem: (※1と同じパスワードを使用)  
Enter Export Password: (任意のパスワード) ※2  
Verifying - Enter Export Password: (※2と同じパスワードを使用)
```

以下のコマンドを実行しtest.p12が作成されていることを確認してください。

```
(root)# ls /etc/pki/CA/  
test.p12
```

2. 監視対象サーバの設定

(注)既にサーバ証明書を持っている場合は、このセクションから参考になさってください。

ここではWindowsServer 2012R2の設定例を紹介します。

2-1 WinRMの設定

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てください。

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

2-2 証明書のインポート

- 1.mmc.exeを実行し、[ファイル] - [スナップインの追加と削除] - [証明書]を選択 - 追加ボタンをクリックします。
 - 2.証明書スナップインダイアログで「コンピュータアカウント」を選択して次へ - [ローカルコンピュータ]にチェックが入っていることを確認して完了をクリックします。選択されたスナップインに[証明書 (ローカルコンピュータ)]が追加されたら、OKをクリックします。
 - 3.コンソール1のダイアログで、コンソールルートにある[個人]を右クリックし、[すべてのタスク]を選択し、[インポート]を選択します。
 - 4.証明書のインポートウィザードで以下の様に指定し、「1. 証明書の作成」で作成した「test.p12」をインポートします。
- (注)既にサーバ証明書を持っている場合は、「test.p12」を、お持ちのサーバ認証用の証明書に置き換えて、インポートしてください。

```
インポートする証明書ファイル: 「test.p12」を指定
秘密キーの保護: 「1. 証明書の作成」で指定した ※2のパスワードを指定
証明書ストア: 証明書をすべて次のストアに配置するにチェックし、個人を指定
```

2-3 winrmに証明書の登録

コマンドプロンプトで以下のコマンドを実行してください。(注)1行で入力してください。

```
> winrm create winrm/config/Listener?Address=*+Transport=HTTPS @{Hostname="【監視対象のIPアドレス】";
CertificateThumbprint="【証明書の拇印 (16進数) ※3】" }
```

※3 証明書の拇印 (16進数) は、2-2でインポートした証明書をコンソール1で指定し、右クリックをし、[開く]をクリックします。ダイアログが表示されたら、[詳細]タブにある拇印の値を入れてください。

以下の結果が表示されることを確認してください。

```
ResourceCreated
Address = http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
ReferenceParameters
ResourceURI = http://schemas.microsoft.com/wbem/wsman/1/config/listener
SelectorSet
Selector: Address = *, Transport = HTTPS
```

コマンドプロンプトで以下のコマンドを実行し、5986 ポートがLISTENINGされていることを確認してください。

```
> netstat -na | findstr 5986

TCP 0.0.0.0:5986 0.0.0.0:0 LISTENING
TCP [::]:5986 [::]:0 LISTENING
```

3 Hinemosマネージャの設定

3-1 キーストアの作成

「1. 証明書の作成」で作成した証明書を使用し、キーストアを作成します。

証明書をマネージャサーバの/root配下に配置した場合を例として、以下のコマンドを実行します。

```
(root)# keytool -import -keystore /opt/hinemos/.keystore -storepass changeit -file /root/ (証明書)

所有者: EMAILADDRESS=sample@example.jp, CN=172.16.44.240, OU=section, O=Company, ST=Tokyo, C=JP
発行者: EMAILADDRESS=sample@example.jp, CN=172.16.44.240, OU=section, O=Company, ST=Tokyo, C=JP
シリアル番号: fc09e03f90354d59
有効期間の開始日: Mon May 16 16:29:47 JST 2016終了日: Thu May 16 16:29:47 JST 2019
証明書のフィンガプリント:

(省略)
```

```
この証明書を信頼しますか。 [いいえ]: はい ←はいを入力
```

3-2 キーストアの設定

「6.3.2 証明書を利用する場合のHTTPS設定 3. 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。

3-3 Hinemosマネージャの再起動

```
(root)# systemctl restart hinemos_manager
```

4 Hinemosクライアントの設定

4-1 リポジトリの設定

リポジトリパースpekティブのリポジトリ[ノード]ビューの管理対象ノードを選択し、リポジトリ[ノードの作成・変更]ダイアログを開きます。

サービスのWinRMを指定し、以下の内容に変更してください。

ポート番号:5986

プロトコル:https

4-2 Hinemosプロパティの設定

Hinemosクライアントのメンテナンスパースpekティブを開き、メンテナンス[Hinemosプロパティ]ビューにある作成ボタンをクリックします。Hinemosプロパティ種別の「真偽値」を指定し、以下の通りに設定してください。

キー:monitor.winservice.ssl.trustall

値:false

説明:任意

6.6.7 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ユーザを用意する必要があります。

※このユーザを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, WinRM 3.0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。

※注意 ActiveDirectoryドメインに参加しているサーバを監視対象とする場合は、ドメインユーザではなく、ローカルユーザを用意する必要があります。

6.6.8 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。なお、下記コマンドを実行するためには、操作を行う Red Hat Enterprise Linux環境に wsmancliパッケージをインストールしてください。

```
(root) # wsmancli -v -c -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/wsmancli/1/wmi/root/cimv2/Win32_Service  
(root) # wsmancli -v -c -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/wsmancli/1/wmi/root/cimv2/Win32_Service?Name=wudfsvc
```

6.7 カスタム監視

6.7.1 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`monitor.custom.command.mode`パラメータで変更できます。なお、`monitor.custom.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)

```
monitor.custom.command.mode=auto
```

カスタム監視の`monitor.custom.command.mode`パラメータは、Hinemosエージェントの`job.command.mode`パラメータと同様に動作します。`monitor.custom.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。エージェント起動ユーザと起動コマンド実効ユーザが異なる場合、コマンド実行時の環境変数を取得する`monitor.custom.command.login`パラメータについてもHinemosエージェントの`job.command.login`パラメータと同様に動作します。

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。

6.7.2 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)

```
monitor.custom.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。

6.7.3 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)

```
monitor.custom.lineseparator=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。

6.7.4 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)

```
monitor.custom.thread=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。

6.8 ポーリングプロトコルの設定

性能機能や監視機能（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で通信できる必要があります。

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。

6.8.1 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
```

6.8.2 SNMP・WBEMの切り替え方法

性能機能、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 -U hinemos -c "UPDATE setting.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、XFS/EXT4/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 -U hinemos -c "UPDATE setting.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）。

6.9 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、Hinemosクライアントのメンテナンスパースペクティブ→メンテナンス[Hinemosプロパティ]ビューを開き、以下のパラメータ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。

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

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。

6.10 リソース監視

6.10.1 大容量ファイルシステム使用率監視の設定

リソース監視で大容量ファイルシステム使用率の項目を利用する場合、以下の手順が必要です。この項目で監視できるファイルシステムはデバイス名に設定されているもののみです。※dfコマンドで確認できます。

1. 監視対象側で、/etc/snmp/snmpd.confを追記する。

```
disk / 10000    ← 追記
disk /dev/shm 10000 ← 追記
disk /boot 10000 ← 追記
```

※上記の例は、[/][/dev/shm][/boot]の3つがデバイス名に設定されている場合です。

2. 監視対象のsnmpdを再起動する。

```
# service snmpd restart
```

6.11 ログファイル監視

6.11.1 監視可能最大ファイル数の設定

以下の設定により、ログファイル監視で監視できるファイルの最大数（デフォルトは1エージェントにつき500）を設定することが可能です。

変更するにはAgent.propertiesに下記を追記します。

※500より大きな値に変更しないでください。

- /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)

```
monitor.logfile.filter.maxfiles=500
```

上限を超えたファイルは監視されません。上限を超えると、下記のログがagent.logに出力されます。

```
refresh() too many files for logfile. not-monitoring file=<監視されないファイル名>
```

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。

6.12 JMX監視

6.12.1 JMX監視の設定

JMX監視をするには、管理対象ノードに対し以下の手順を実施してください。

1. 監視対象ノードのホストの確認

以下のコマンドを実行し、ホスト名がホストのIPアドレスに正しく解決されることを確認してください。

```
(root)# hostname -i  
【ホストのIPアドレス】
```

(注)上記のコマンドを実行したとき、「127.0.0.1」が出力される場合正しくJMX監視ができないため、ご注意ください。

2. JMXリモート接続の設定

以下のシステムプロパティを監視対象としたいJavaプロセスの起動引数として設定してください。

(注)以下を設定する場合は改行は入れず、1行で追加してください。

```
-Dcom.sun.management.jmxremote  
-Dcom.sun.management.jmxremote.port=[任意のポート]  
-Dcom.sun.management.jmxremote.authenticate=false ※1  
-Dcom.sun.management.jmxremote.ssl=false ※1
```

※1 上記の設定ではパスワード認証、SSLの設定は無効になっているため、必要な場合は適宜設定をしてください。

3. JMX監視の設定

設定方法については「Hinemosユーザマニュアル 7.17 JMX監視」を参照してください。

その際、JMX[作成・変更]ダイアログの「条件」-「チェック設定」で指定するポートは「2. JMXリモート接続の設定」で設定したものにしてください。

7 ジョブ機能

Hinemosジョブ機能の追加設定について説明します。

7.1 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job.command.mode/パラメータで変更できます。なお、job.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)

```
job.command.mode=auto
```

job.command.mode/パラメータに指定できる値の一覧を表7-1に示します。

表 7-1 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
unix.su	Linuxプラットフォーム用にコマンド生成 ※1

※1 unix.suを設定すると、以下の通り動作します。

- エージェント起動ユーザと実効ユーザが同じ場合、unixと同様の動作します。
- エージェント起動ユーザと実効ユーザが異なる場合、compatibleと同様の動作します。

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 : echo
 第1引数 : XXX

Hinemosは、与えられた起動コマンドについて、半角スペースを区切り文字として分割します。
 分割された先頭の文字列をコマンドとし、以降の文字列を第1引数、第2引数・・・としてOSに渡し、実行します。
 引数内の半角スペースで区切られないようにするためには、該当する箇所を""でくくってください。

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：

```
sh -c [起動コマンド]  
コマンド : sh  
第1引数 : -c  
第2引数 : echo XXX
```

Hinemosでは特に起動コマンドの文字列処理は行いません。
起動コマンドとして指定した文字列をそのまま sh コマンドの第3引数としてOSに渡し、実行します。

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：

```
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 : -i ※1  
第4引数 : sh  
第5引数 : -c  
第6引数 : echo XXX
```

Hinemosでは特に起動コマンドの文字列処理は行いません。
起動コマンドとして指定した文字列をそのまま sudo コマンドの第6引数としてOSに渡し、実行します。

※1 job.command.loginがtrueの時に第3引数を追加してコマンドを実行します。

• 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
```

Hinemosは、与えられた起動コマンドについて、半角スペースを区切り文字として分割します。
分割された先頭の文字列をコマンドとし、以降の文字列を第1引数、第2引数・・・としてOSに渡し、実行します。

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 : - ※1  
第2引数 : [実効ユーザ]  
第3引数 : -c  
第4引数 : echo XXX
```

Hinemosでは特に起動コマンドの文字列処理は行いません。
起動コマンドとして指定した文字列をそのまま su コマンドの第3引数としてOSに渡し、実行します。

※1 job.command.loginがtrueの時に第1引数を追加してコマンドを実行します。

- 起動コマンドをunix.su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：

```
sh -c [起動コマンド]
コマンド : sh
第1引数  : -c
第2引数  : echo XXX
```

Hinemosでは特に起動コマンドの文字列処理は行いません。
起動コマンドとして指定した文字列をそのまま sh コマンドの第3引数としてOSに渡し、実行します。

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  : - ※1
第2引数  : [実効ユーザ]
第3引数  : -c
第4引数  : echo XXX
```

Hinemosでは特に起動コマンドの文字列処理は行いません。
起動コマンドとして指定した文字列をそのまま su コマンドの第3引数としてOSに渡し、実行します。

※1 job.command.loginがtrueの時に第1引数を追加してコマンドを実行します。

設定を変更した後、Hinemosエージェントの再起動が必要です。

7.2 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 13.1 javaプロセス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の動作について」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。

- Hinemosクライアントのメンテナンスパースペクティブ→メンテナンス[Hinemosプロパティ]ビュー

```
quartz.dbms.jobStore.misfireThreshold = 3600000
```

なお、Hinemosは再起動等で一時的に停止している時間帯のジョブスケジュールに関しては、再起動後に動作する仕様となっておりますが、上記のパラメータを小さくしすぎると、再起動後に動作しない仕様となってしまいます。そのため、デフォルト値の3600000を大幅に小さくすることは推奨されておりません。

7.3 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。

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 13.4.1 ジョブ機能の制限事項」を参照してください）

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。

※必要に応じて、hinemosを他のユーザ名(rootなど)に当てはめて設定してください。

- 転送元ノード：ファイルの転送元となるノード(scpコマンド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(scpコマンドが実行されるサーバ)
- 転送するユーザ：ジョブ[ファイル転送ジョブの作成・変更]ダイアログの実効ユーザグループで指定したユーザ (scpコマンドを実行するユーザ)。

なお、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) $ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hinemos/.ssh/id_rsa): ←Enterを入力
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hinemos/.ssh/id_rsa.
Your public key has been saved in /home/hinemos/.ssh/id_rsa.pub.
(略)

(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/conf/Agent.properties

##
## Common Function
##

## Common : For JAX-WS XML Invalid Char(true : replace specified char, false : replace Hexa expression)
common.invalid.char.replace=false
(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sshでログインし、known_hostsに鍵を登録します。なお、ipv6を利用している場合に、sshの引数として2重コロン (::) を用いた省略表記のIPアドレスを用いて、鍵を登録しようとすると、IPアドレスが省略表記のまま、known_hostsに登録されてしまうため、ファイル転送ジョブが正常に動作しません。ipv6でsshログインする際は、省略表記は使用しないようにしてください。

```
(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/conf/Agent.properties

##
## Common Function
##

## Common : For JAX-WS XML Invalid Char(true : replace specified char, false : replace Hexa expression)
common.invalid.char.replace=false
(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。

8 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。

8.1 セルフチェック機能

Hinemos マネージャの内部状態を定期的を確認して、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 (selfcheck.monitoring.db)
 - ・ 内部スケジューラの異常に伴う機能不全 (selfcheck.monitoring.scheduler.delay)
- ・ 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 (selfcheck.monitoring.jvm)
 - ・ 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 (selfcheck.monitoring.filesystem)
- ・ パフォーマンス低下
 - ・ 大量のsyslogやsnmptrap受信による一時情報格納領域の肥大化 (selfcheck.monitoring.systemlog, selfcheck.monitoring.snmptrap)
 - ・ 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 (selfcheck.monitoring.asyncntask)
 - ・ 履歴情報(イベント、性能情報、ジョブ履歴など)の肥大化 (selfcheck.monitoring.table.size)
 - ・ 実行中のジョブセッション数の増大 (selfcheck.monitoring.job.runningssession)
 - ・ 動作中のスレッド数の増大 (selfcheck.monitoring.thread.activity)
 - ・ マネージャサーバのスワップアウト発生 (selfcheck.monitoring.swapout)

異常が検知された場合、デフォルトで監視履歴[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。それらの出力の変更方法については、[INTERNAL イベント](#) を参照してください。

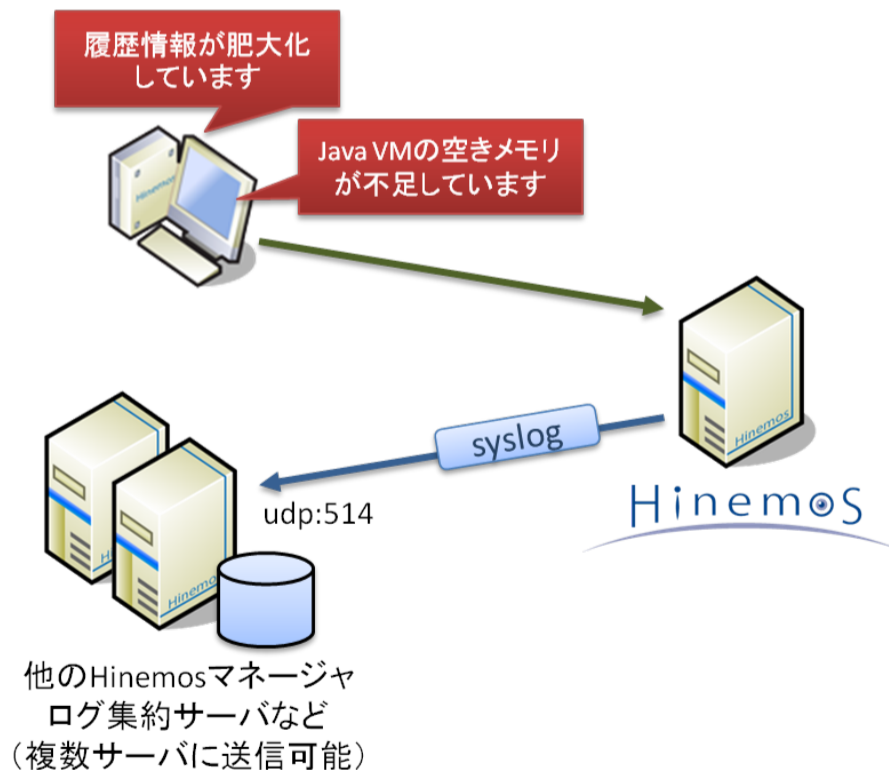


図8-1. セルフチェック機能の概要

8.1.1 セルフチェック機能の設定

セルフチェック機能の設定はHinemosクライアントのメンテナンスパースペクティブ→メンテナンス[Hinemosプロパティ]ビューから行います。設定値の詳細については [Hinemosマネージャの設定一覧](#) の「表13-8. セルフチェック機能の設定値」を参照してください。

8.1.2 セルフチェック機能の前提条件

SNMPのインストール

セルフチェック機能を使用するためにはマネージャサーバにnet-snmp、net-snmp-utilsがインストールされている必要があります。

セルフチェック機能に関する通信要件（マネージャサーバ内の接続）

マネージャサーバ内の接続を行う場合、snmpdの設定ファイルの/etc/snmp/snmpd.confに127.0.0.1に対するアクセス許可がされていることが必要です。（デフォルトであれば、127.0.0.1へのアクセスは許可されています。）

なお、Hinemosマネージャサーバ上で下記のコマンドを実行することで、アクセス許可を確認可能です。

```
(root)# snmpwalk -c public -v 2c <マネージャのIPアドレス> .1.3.6.1.2.1.1
```

8.2 INTERNALイベント

8.2.1 通知先の設定

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、メール、コマンド、監視履歴[イベント]ビュー、hinemos_internal.logの5つです。Hinemosクライアントのメンテナンスパースペクティブ→メンテナンス[Hinemosプロパティ]ビューで、出力先を設定できます。

(注)出力先がイベント通知の場合、イベント通知のオリジナルメッセージ欄にINTERNALイベントの詳細な内容が出力されます。出力先がSyslogサーバ、およびHinemosApl.logの場合、INTERNALイベントの詳細な内容は転送・出力されません。

- Syslog送信

internal.syslog パラメータにより、Syslog送信の有効無効およびSyslogの送信先を設定します。

```
internal.syslog=false
internal.syslog.priority=info
internal.syslog.host=192.168.1.1,192.168.1.2
internal.syslog.port=514
internal.syslog.facility=daemon
internal.syslog.severity=alert
```

以下にinternal.syslog パラメータによりSyslogに送信された際のINTERNALイベントの出力例を記載します。

```
Jun  2 14:26:13 sample_machine hinemos: MNG,Hinemos Manager Monitor,SYS,001,情報,
Hinemosマネージャが起動しました。 [sample_machine]
```

- 監視履歴[イベント]ビュー

internal.event パラメータにより、INTERNALスコープへの出力有無および出力レベルを設定します。

```
internal.event=true
internal.event.priority=info
```

以下は監視履歴[イベント]ビューに出力された、INTERNALイベントの監視[イベントの詳細]の出力例です。(図 8-2参照)

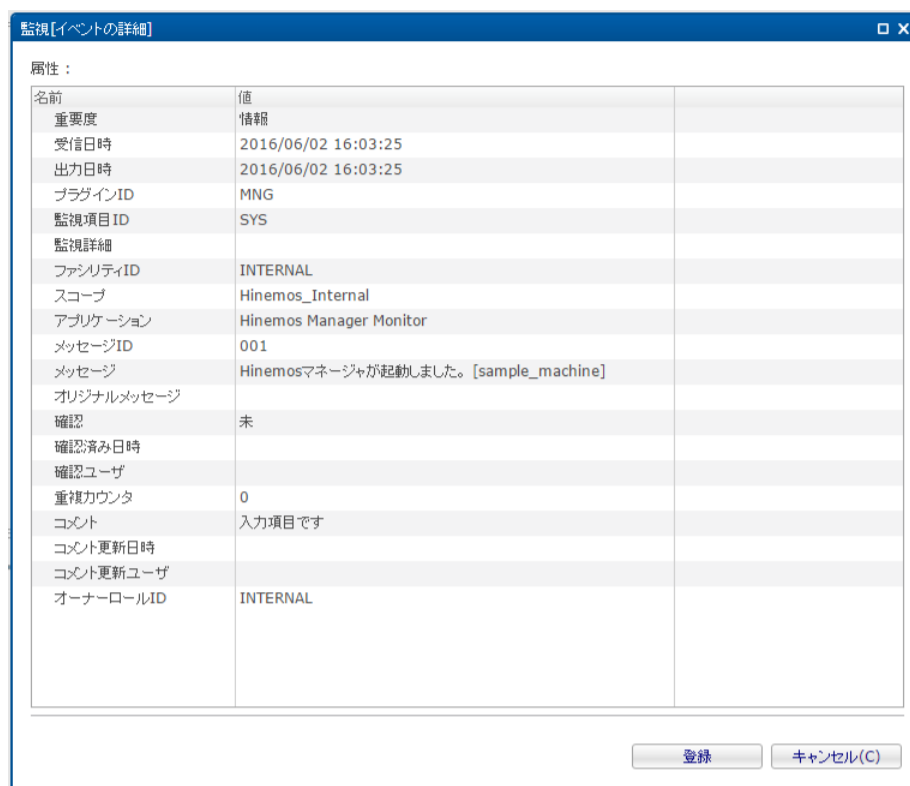


図 8-2. 監視[イベントの詳細]:INTERNALイベント出力例

- hinemos_internal.log

internal.file パラメータにより、ログファイル（hinemos_internal.log）への出力有無および出力レベルを設定します。

```
internal.file=true
internal.file.priority=info
```

以下にinternal.file パラメータによりhinemos_internal.logへINTERNALイベントが出力された際の出力例を記載します。

```
2016-06-02 14:26:13,656 INFO [HinemosInternal] (main) 2016/06/02 14:26:13
MNG,Hinemos Manager Monitor,SYS,001,3,Hinemosマネージャが起動しました。 [sample_machine]
```

- メール送信

internal.mail パラメータにより、メールの送信有無、出力レベル、および送信先を設定します。

```
internal.mail=false
internal.mail.priority=info
internal.mail.address=user1@host.domain,user2@host.domain
```

以下にinternal.mail パラメータによりINTERNALイベントをメール送信し、受信した際の出力例を記載します。

```
From admin@hinemos.com Thu Jun 2 15:05:04 2016
Return-Path: <admin@hinemos.com>
Received: from sample_machine ([xx.xx.xx.xx])

(省略)

出力日時 : 2016-06-02 15:04:53
アプリケーション : Hinemos Manager Monitor
重要度 : 情報
メッセージ : Hinemosマネージャが起動しました。 [sample_machine]
スコープ : Hinemos_Internal
```

- コマンド実行

internal.command パラメータにより、コマンドの実行有無、出力レベル、実効ユーザ、コマンド、およびタイムアウトを設定します。

```
internal.command=false
internal.command.priority=info
internal.command.user=root
internal.command.commandline=echo #[GENERATION_DATE] #[MESSAGE] >> /tmp/test.txt
internal.command.timeout=15000
```

以下にinternal.command パラメータにより、コマンド実行された際に作成された/tmp/test.txtの内容を出力例として記載します。

```
(root)# cat /tmp/test.txt
2016/06/02 15:09:28 Hinemosマネージャが起動しました。 [sample_machine]
```

8.2.2 通知される情報

INTERNALイベントの一覧を以下の表に示します。

表 9-2 INTERNALイベント

重要度	プラグインID	監視項目ID	アプリケーション	メッセージID	メッセージ
正常	MNG	SYS	Hinemos Manager Monitor	001	Hinemosマネージャが起動しました。
正常	MNG	SYS	Hinemos Manager Monitor	002	Hinemosマネージャが停止しました。
警告	COMMON	SYS	内部共通機構	001	長時間（{0}分）動作が続くポーリングスレッドを検出しました。 PollerGroup={1}, PollerName={2} (現在の統計値：長時間動作スレッド={3}, 全ポーリングスレッド={4})
警告	SYS_SFC	SYS	セルフチェック	001	データベースが利用できません。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。
警告	SYS_SFC	SYS	セルフチェック	002	ファイルシステム（{0}）の使用量（[{1} [%]）が閾値（{2} [%]）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。
警告	SYS_SFC	SYS	セルフチェック	003	利用可能なメモリ容量（{0}[MByte]）が閾値（{1} [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。
警告	SYS_SFC	SYS	セルフチェック	004	スケジューラ（{0}:{1}:{2} - 次回実行予定 {3}）に遅延（{4} [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。
警告	SYS_SFC	SYS	セルフチェック	005	メモリのスワップアウト（{0} [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。
警告	SYS_SFC	SYS	セルフチェック	006	内部DB内のテーブル（{0}）が非常に多く蓄積（{1} [mbyte], {2} [rows] > 閾値 {3} {4}）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。

警告	SYS_SF C	SYS	セルフチェッ ク	007	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され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。
警告	SYS_SF C	SYS	セルフチェッ ク	008	Hinemosマネージャ(tcp : 8080)に対するリクエストが多く (処理待ちのリクエスト数 {0} > 閾値 {1})、処理遅延が生じています。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。
警告	SYS_SF C	SYS	セルフチェッ ク	009	Hinemosマネージャに送信されたsyslogが多く(処理待ちのsyslog数 {0} > 閾値 {1})、処理遅延が生じています。監視対象から大量のsyslogが送信されていないかを確認してください。
警告	SYS_SF C	SYS	セルフチェッ ク	010	Hinemosマネージャに送信されたsnmptrapが多く(処理待ちのsnmptrap数 {0} > 閾値 {1})、処理遅延が生じています。監視対象から大量のsnmptrapが送信されていないかを確認してください。
警告	SYS_SF C	SYS	セルフチェッ ク	011	非同期処理の蓄積量が多く(処理待ちの非同期処理数 {0} > 閾値 {1})、処理遅延が生じています。通知やジョブが非常に多く実行されていないかをご確認ください。
警告	SYS_SF C	SYS	セルフチェッ ク	012	内部ロジック(スレッドID {0}, スレッド名 {1}, クラス名 {2}, 開始日時 {3})で {4} [sec]以上の時間を要しています。
警告	NOTIFY	SYS	通知	004	通知の取得に失敗しました。(NotifyId={0})
警告	NOTIFY	SYS	通知	006	通知一覧の取得に失敗しました。
危険	NOTIFY	SYS	通知	007	Failed to notify Notification ID "{0}"
危険	NOTIFY	SYS	通知	008	通知に失敗しました。ジョブ通知の通知先のジョブ定義が存在しません。(NotifyId={0},MonitorId={1},JobunitId={2},JobunitId={3})
警告	MAILTE MP	SYS	メールテンプレ ート	004	メールテンプレートの取得に失敗しました。(MailTemplateID={0})
警告	MAILTE MP	SYS	メールテンプレ ート	005	メールテンプレートID一覧の取得に失敗しました。
警告	MAILTE MP	SYS	メールテンプレ ート	006	メールテンプレート一覧の取得に失敗しました。
警告	MON	SYS	監視	001	スコープ情報の取得に失敗しました。(FacilityId={0})
警告	MON	SYS	監視	004	イベントの確認/未確認に失敗しました。 (FacilityId={0}, MonitorId={1}, PluginId={2}, ConfirmType={3})
警告	MON	SYS	監視	010	監視情報の取得に失敗しました。(MonitorTypeId={0}, MonitorId={1})
警告	MON	SYS	監視	011	監視情報一覧の取得に失敗しました。(MonitorTypeId={0}, MonitorId={1})
警告	MON	SYS	監視	012	監視の実行に失敗しました。(MonitorTypeId={0}, MonitorId={1})
警告	PROC	SYS	プロセス監視	001	ポーラーの登録に失敗しました。(FacilityId={0})
警告	PROC	SYS	プロセス監視	002	ポーラーの削除に失敗しました。(FacilityId={0})
危険	TRAP	SYS	SNMPTRAP 監視	009	SNMP TRAP の通知に失敗しました。(trapOid={0}, genericId={1}, specificId={2})
危険	MON_P NG	SYS	PING監視	001	Fpingが応答を返しませんでした。(監視項目ID={0})

警告	PERF	SYS	性能管理	010	監視情報の取得に失敗しました。(MonitorTypeId={0}, MonitorId={1})
警告	PERF	SYS	性能管理	014	ポーラーの登録に失敗しました。(FacilityId={0})
警告	PERF	SYS	性能管理	015	ポーラーの削除に失敗しました。(FacilityId={0})
危険	JOB	SYS	ジョブ管理	003	ジョブの履歴情報の作成に失敗しました。(JobID={0})
警告	JOB	SYS	ジョブ管理	008	ジョブの開始[中断解除]に失敗しました。(SessionID={0}, JobID={1})
警告	JOB	SYS	ジョブ管理	009	ジョブの開始[保留解除]に失敗しました。(SessionID={0}, JobID={1})
警告	JOB	SYS	ジョブ管理	010	ジョブの開始[スキップ解除]に失敗しました。(SessionID={0}, JobID={1})
警告	JOB	SYS	ジョブ管理	011	ジョブの停止[即時]に失敗しました。(SessionID={0}, JobID={1}, FacilityID={2})
警告	JOB	SYS	ジョブ管理	012	ジョブの停止[中断]に失敗しました。(SessionID={0}, JobID={1})
警告	JOB	SYS	ジョブ管理	013	ジョブの停止[保留]に失敗しました。(SessionID={0}, JobID={1})
警告	JOB	SYS	ジョブ管理	014	ジョブの停止[スキップ]に失敗しました。(SessionID={0}, JobID={1})
警告	JOB	SYS	ジョブ管理	016	ジョブのスケジュール実行に失敗しました。(JobID={0}, ScheduleInfo={1})
警告	JOB	SYS	ジョブ管理	017	ジョブのファイルチェック実行に失敗しました。(JobID={0}, ScheduleInfo={1})

8.3 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 8-3のようなダイアログを表示します。

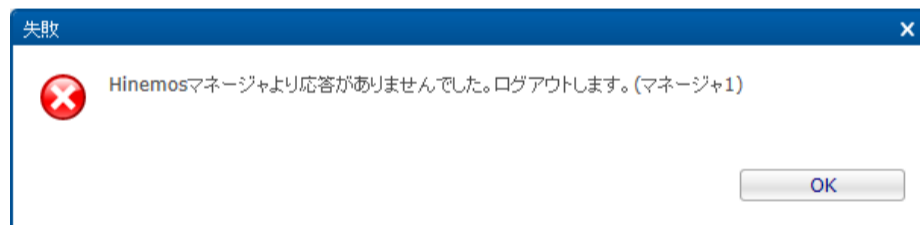


図 8-3 Hinemosマネージャ死活検知

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 8-4 参照)

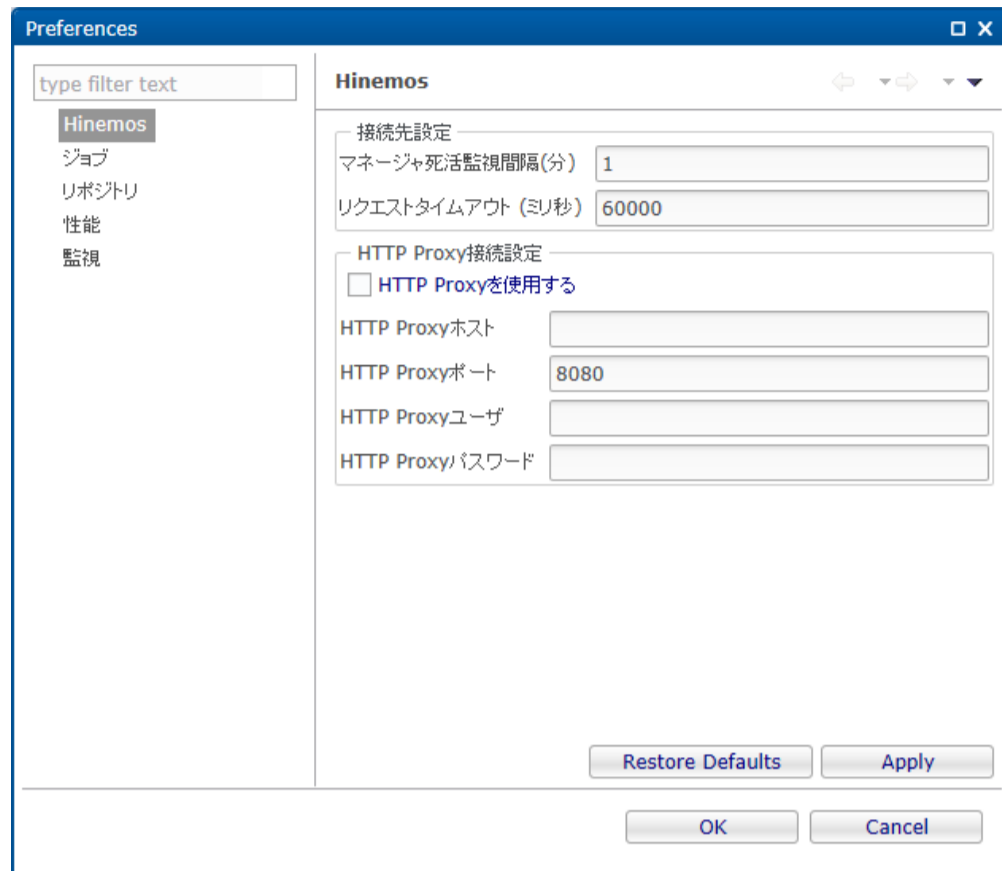


図 8-4 Hinemosマネージャ死活監視間隔

8.4 環境構築機能

8.4.1 HTTPS接続を利用した環境構築モジュール

ここでは、HTTPS接続で環境構築モジュールを実行する際のサーバ証明書の必須有無が設定できます。

サーバ証明書を利用しない場合

Hinemosクライアントのメンテナンスパースペクティブを開き、メンテナンス[Hinemosプロパティ]ビューにある作成ボタンをクリックします。Hinemosプロパティ種別の「真偽値」を指定し、以下の通りに設定してください。

```
キー:infra.winrm.ssl.trustall  
値:true  
説明:任意
```

(注) デフォルトではtrueとなっています。

登録後は、「6.6.6 証明書を利用したHTTPSの設定 4-1 リポジトリの設定」を参照してください。

(注)本設定はHinemos ver5.0.3以降で設定可能です。

サーバ証明書を利用する場合

「6.6.6 証明書を利用したHTTPSの設定」を参照してください。

その際、「4-2 Hinemosプロパティの設定」のキーには「infra.winrm.ssl.trustall」を指定してください。

9 セキュリティ

セキュリティの観点から、Hinemosマネージャの内部DBのパスワードを変更する場合には、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。

9.1 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。

9.1.1 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 (9.3.5)
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
```

4. Hinemosマネージャを起動します。

- 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)

# "local" is for Unix domain socket connections only
local postgres hinemos md5
local hinemos hinemos md5
local hinemos hinemos_quartz md5
local replication hinemos_repl 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
host replication hinemos_repl 0.0.0.0/0 md5
# IPv6 local connections:
host postgres hinemos ::/0 md5
host hinemos hinemos ::/0 md5
host hinemos hinemos_quartz ::/0 md5
host replication hinemos_repl ::/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。

9.1.2 Hinemosマネージャ（接続元）の設定変更

以下の2つの箇所を変更します。

- /opt/hinemos/etc/META-INF/persistence.xml
- /opt/hinemos/etc/db_account.properties

1. persistence.xmlの編集

persistence.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javax.persistence.jdbc.passwordに設定します。

/opt/hinemos/etc/META-INF/persistence.xml

(略)

```
<property name="javax.persistence.jdbc.user" value="hinemos"/>
<property name="javax.persistence.jdbc.password" value="hinemos"/>
```

(略)

2. db_account.propertiesの編集

PostgreSQLユーザhinemos, hinemos_quartzのデータベースアクセスに関する設定を編集します。

[PostgreSQL（接続先）の設定変更](#) で変更したパスワードを hinemos_pass, hinemos_quartz_pass に設定します。

```
hinemos_user=hinemos
hinemos_pass=hinemos
hinemos_quartz_user=hinemos_quartz
hinemos_quartz_pass=hinemos_quartz
```

9.2 データベース内のパスワードの暗号化

Hinemosマネージャのデータベース(PostgreSQL)にはノードプロパティのパスワード(WBEMパスワードや仮想化ソフト接続パスワード)や SQL監視のパスワードなどが暗号化されて格納されています。暗号化のキーは下記の通りです。

- db_crypt.keyの編集

/opt/hinemos/etc/db_crypt.key

```
hinemos
```

なお、この文字列を変更する場合は、インストール直後に実施してください。設定登録後に実施してしまうと、登録済のパスワードが復号化されなくなってしまいます。

9.3 JMXのアクセス権限を変更する

HinemosマネージャはJMX(Java Management Extensions)にてHinemosマネージャ内部の状態をチェックすることが可能です。アクセス可能なユーザやパスワードは下記の通り変更可能です。

- jmxremote.access

/opt/hinemos/etc/jmxremote.access

ロールごとに許可するアクセス権を定義するアクセス制御ファイルです。ロールとアクセス権を追加します。

```
hinemos readwrite
```

- jmxremote.password

/opt/hinemos/etc/jmxremote.password

アクセス制御ファイルで定義したロールのパスワードを追加します。

```
hinemos hinemos
```

10 Hinemosコンポーネント間接続

10.1 HinemosマネージャへのHTTP Proxyを経由した接続

Hinemos5.0は、Hinemosリッチクライアント及びHinemos Webクライアント、Hinemosエージェントからの、HTTP Proxyを経由したHinemosマネージャへの接続に対応します。

10.1.1 HinemosクライアントからHinemosマネージャへの接続

Hinemosクライアントから、HTTP Proxyの情報を設定します。

1. Hinemosクライアントメニューのクライアント設定 -> 設定に表示される、HTTP Proxy接続設定から設定します（図 10-1 参照）。

HTTP Proxyホスト

HTTP ProxyサーバのIPアドレスまたはホスト名を入力します

HTTP Proxyポート

HTTP Proxyサーバの待ち受けポートを入力します

HTTP Proxyユーザ

HTTP Proxyサーバの認証ユーザを入力します

HTTP Proxyパスワード

HTTP Proxyサーバの認証ユーザパスワードを入力します

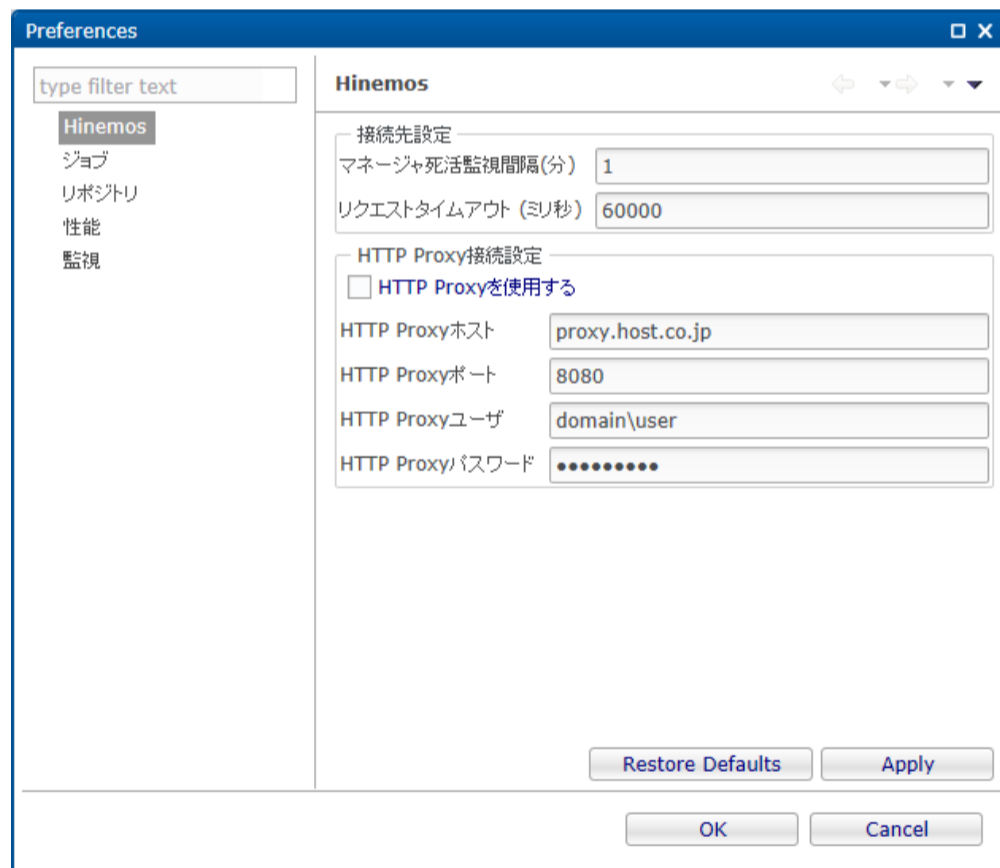


図 10-1 Hinemosコンポーネント間の接続設定

10.1.2 HinemosエージェントからHinemosマネージャへの接続

Hinemosエージェントの、以下のファイルを編集します。

/opt/hinemos_agent/conf/Agent.properties

```
http.proxy.host=192.168.100.100
http.proxy.port=8080
http.proxy.user=proxyuser1
http.proxy.password=password
```

以下のパラメータを設定します。

- http.proxy.host=HTTP ProxyサーバのIPアドレスまたはホスト名
- http.proxy.port=HTTP Proxyサーバの待ち受けポート
- http.proxy.user=HTTP Proxyサーバの認証ユーザ
- http.proxy.password=HTTP Proxyサーバの認証ユーザパスワード

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。

10.2 HinemosマネージャへのHTTPS接続

Hinemos5.0は、HinemosクライアントとHinemosマネージャ間、HinemosエージェントとHinemosマネージャ間共に、HTTPS通信に対応します。

10.2.1 Hinemosマネージャのサーバ証明書の準備

HinemosマネージャへHTTPSにより接続する場合、Hinemosマネージャ側でサーバ証明書を作成する必要があります。

まず、Hinemosマネージャサーバのサーバ証明書（PKCS#12）を準備します。本節では、OpenSSL(OpenSSL 1.0.0-fips 29 Mar 2010)を使用した自己署名証明書の作成例について記載します。

1. デフォルトのopenssl.cnfをコピーします。

```
# mkdir /opt/hinemos/etc/ssl
# cd /opt/hinemos/etc/ssl
# cp /etc/pki/tls/openssl.cnf .
```

2. 以下のように、openssl.cnfを修正します。

```
# vi openssl.cnf

[ req ]
# x509_extensions = v3_ca # The extensions to add to the self signed cert ←コメントアウト
x509_extensions = v3_req ←追加

req_extensions = v3_req # The extensions to add to a certificate request ←コメントを外す

[ v3_req ]
subjectAltName=IP:【HinemosマネージャサーバのIPアドレス】 ←追加
```


3. 以下のコマンドにより、証明書を作成します。

```
# openssl genrsa -des3 -out server.key 1024
Generating RSA private key, 1024 bit long modulus
.....++++++
.++++++
e is 65537 (0x10001)
Enter pass phrase for server.key: (hinemosを入力)
Verifying - Enter pass phrase for server.key: (hinemosを入力)
```

```
# openssl req -new -x509 -key server.key -out server.crt -config openssl.cnf -days 3650
Enter pass phrase for server.key: (hinemosを入力)
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]: (何も入力せずEnter)
State or Province Name (full name) []: (何も入力せずEnter)
Locality Name (eg, city) [Default City]: (何も入力せずEnter)
Organization Name (eg, company) [Default Company Ltd]: (何も入力せずEnter)
Organizational Unit Name (eg, section) []: (何も入力せずEnter)
Common Name (eg, your name or your server's hostname) []: (何も入力せずEnter)
Email Address []: (何も入力せずEnter)
```

```
# openssl pkcs12 -export -in server.crt -inkey server.key -out /root/keystore
Enter pass phrase for server.key: (hinemosを入力)
Enter Export Password: (hinemosを入力)
Verifying - Enter Export Password: (hinemosを入力)
```

次に、上記手順により作成したサーバ証明書を、Hinemosマネージャから読み込みます。Hinemosマネージャからサーバ証明書を読み込むには、Hinemosクライアントのメンテナンスパースペクティブ →メンテナンス[Hinemosプロパティ]ビューを開き、以下のパラメータを変更します。

```
ws.client.address=https://0.0.0.0:8443 ←Hinemosクライアントからの接続をHTTPS化したい場合に変更
ws.agent.address=https://0.0.0.0:8444 ←Hinemosエージェントからの接続をHTTPS化したい場合に変更
```

HTTPS接続に関連するパラメータは以下の通りです。

ws.client.address:

クライアントからマネージャへの接続アドレス
 プロトコルを必要に応じてhttpsとすることが可能です。httpsとした場合、httpの場合とは異なるポート番号に変更する必要があります。

ws.agent.address:

エージェントからマネージャへの接続アドレス
 プロトコルを必要に応じてhttpsとすることが可能です。httpsとした場合、httpの場合とは異なるポート番号に変更が必要です。また、httpsプロトコルにした場合、全ての管理対象エージェントからの接続設定をhttpsに変更する必要があります。

ws.https.keystore.path:

キーストアのディレクトリパス

ws.https.keystore.password:

キーストアのパスワード

ws.https.keystore.type:

キーストアのタイプ

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。

10.2.2 HinemosリッチクライアントからHinemosマネージャへのHTTPS接続

Hinemosリッチクライアントを起動し、接続[ログイン]ダイアログの接続先URLに以下を設定します。

```
https://【HinemosマネージャのIPアドレス】:8443/HinemosWS/
```

その他の項目は、HTTPプロトコルで接続する際と同様に設定します。これにより、HinemosリッチクライアントとHinemosマネージャ間が、HTTPSで暗号化された通信を行うこととなります。（HTTPSのホスト認証は行われません）

10.2.3 Hinemos WebクライアントからHinemosマネージャへのHTTPS接続

ブラウザからHinemos Webクライアントにアクセスし、ログイン画面にてHinemosマネージャ接続先URLとして以下を指定します。

```
https://【HinemosマネージャのIPアドレス】:8443/HinemosWS/
```

その他の項目は、HTTPプロトコルで接続する際と同様に設定します。これにより、Hinemos WebクライアントとHinemosマネージャ間が、HTTPSで暗号化された通信を行うこととなります。（HTTPSのホスト認証は行われません）

10.2.4 HinemosエージェントからHinemosマネージャへのHTTPS接続

エージェントの設定ファイル

- ・ /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)

を以下のように修正します。

```
managerAddress=https://【HinemosマネージャのIPアドレス】:8444/HinemosWS/
```

設定変更後、エージェントを再起動します。これにより、HinemosエージェントとHinemosマネージャ間が、HTTPSで暗号化された通信を行うこととなります。

10.2.5 HinemosリッチクライアントからHinemosマネージャへのHTTPS接続時にホスト認証を行う

HinemosリッチクライアントからHinemosマネージャへ、HTTPSプロトコルによるアクセスをする際、接続先のHinemosマネージャのホスト認証を行う必要がある場合には、以下の設定を行います。

[Hinemosマネージャのサーバ証明書の準備](#)にて作成したserver.crtを、Hinemosリッチクライアントがインストールされた端末に配置します。（本書では、server.crtをC:\tmp\に配置するとして説明します）

1. コマンドプロンプトを「管理者として実行」で起動します。
2. 配置したサーバ証明書を、トラストストアにインポートします。コマンドプロンプトから、以下のコマンドを1行で実行します。（以下は32bit環境での例です。64bit環境の場合は、「Program Files」を「Program Files (x86)」に読み換えてください）

```
>C:\Users>"C:\Program Files\Hinemos\Client5.0.0\eclipse-rcp\jre\bin\keytool.exe" -import -alias hinemos -file "C:\tmp\server.crt" -keystore "C:\tmp\cacerts"
```

```
キーストアのパスワードを入力してください: (changeitを入力)
所有者: O=Default Company Ltd, L=Default City, C=XX
発行者: O=Default Company Ltd, L=Default City, C=XX
```

(省略)

```
この証明書を信頼しますか? [いいえ]: はい
証明書がキーストアに追加されました。
```

3. Hinemosリッチクライアントの起動スクリプトを修正します。

Hinemosリッチクライアントをインストールしたディレクトリに存在する、起動スクリプト（client_start.vbs と client_clean_start.vbs）を修正します。

(修正前)

```
...
strCmdLine = strCmdLine & strVmArgs & eclipseLocale & "-Dorg.apache.commons.logging.Log= ...(省略)..."
' strCmdLine = strCmdLine & " -Dhttps.hostVerify=true"
' strCmdLine = strCmdLine & " -Djavax.net.ssl.trustStore=C:\tmp\cacerts"
' strCmdLine = strCmdLine & " -Djavax.net.ssl.trustStorePassword=changeit"
objWshShell.CurrentDirectory = strExecFolder
...
```

(修正後)

```
...
strCmdLine = strCmdLine & strVmArgs & eclipseLocale & "-Dorg.apache.commons.logging.Log= ...(省略)..."
strCmdLine = strCmdLine & " -Dhttps.hostVerify=true" ← コメントを外す
strCmdLine = strCmdLine & " -Djavax.net.ssl.trustStore=C:\tmp\cacerts" ← コメントを外す
strCmdLine = strCmdLine & " -Djavax.net.ssl.trustStorePassword=changeit" ← コメントを外す
objWshShell.CurrentDirectory = strExecFolder
...
```

指定する環境変数は以下の通りです

https.hostVerify:

trueの場合ホスト認証を行います。設定しない場合、false（ホスト認証を行わない）として動作します。

javax.net.ssl.trustStore:

トラストストアファイルのファイルパスを指定します。

※注意 トラストストアファイルのパスを指定する際、パスに空白が含まれる場合は、パスを二つのダブルクォーテーション("")で囲む必要があります。

設定例

```
strCmdLine = strCmdLine & " -Djavax.net.ssl.trustStore=""C:\Program Files(x86)\Hinemos\Client5.0.x\
\eclipse-rcp\jre\lib\security\cacerts"""
```

javax.net.ssl.trustStorePassword:

トラストストアファイルのパスワードを指定します。

4. 修正済みのclient_start.vbs を使ってクライアントを起動し、HTTPSで接続します。

修正済みのclient_start.vbs を使って起動したクライアントを使用する場合、HTTPSで接続した場合にはホスト認証が行われることになります。

10.2.6 Hinemos WebクライアントからHinemosマネージャへのHTTPS接続時にホスト認証を行う

Hinemos WebクライアントからHinemosマネージャへ、HTTPSプロトコルによるアクセスをする際、接続先のHinemosマネージャのホスト認証を行う必要がある場合には、以下の設定を行います。

Hinemosマネージャのサーバ証明書の準備にて作成したserver.crtを、Hinemos Webクライアントがインストールされた端末に配置します。（本書では、server.crtを /opt/hinemos/etc/ssl/ に配置するとして説明します）

1. Hinemos Webクライアントがインストールされたサーバで、以下のコマンドを実行します。

```
# mkdir /opt/hinemos_web/conf/ssl
# keytool -import -import -alias hinemos -file /opt/hinemos/etc/ssl/server.crt -keystore /opt/hinemos_web/conf/ssl/cacerts

キーストアのパスワードを入力してください: (changeitを入力)
新規パスワードを再入力してください: (聞かれた場合にはchangeitを入力)

所有者: O=Default Company Ltd, L=Default City, C=XX
発行者: O=Default Company Ltd, L=Default City, C=XX

(省略)

この証明書を信頼しますか? [いいえ]: はい
証明書がキーストアに追加されました。
```

2. Hinemos Webクライアントの設定ファイルを修正します。
/opt/hinemos_web/conf/hinemos_web.cfg を修正します。

```
(変更前)
...
### JVM - HTTPS HostVerify
#export JVM_SSL_OPTS="-Dhttps.hostVerify=true -Djavax.net.ssl.trustStore=
/opt/hinemos_web/conf/ssl/cacerts -Djavax.net.ssl.trustStorePassword=changeit"

(変更後)
...
### JVM - HTTPS HostVerify
export JVM_SSL_OPTS="-Dhttps.hostVerify=true -Djavax.net.ssl.trustStore=
/opt/hinemos_web/conf/ssl/cacerts -Djavax.net.ssl.trustStorePassword=changeit"

↑コメントを外す
```

指定する環境変数は、以下の通りです

https.hostVerify:

trueの場合ホスト認証を行います。設定しない場合、false（ホスト認証を行わない）として動作します。

javax.net.ssl.trustStore:

トラストストアファイルのファイルパスを指定します。

javax.net.ssl.trustStorePassword:

トラストストアファイルのパスワードを指定します。

3. Hinemos Webクライアントを再起動します。

以降、Hinemos Webクライアントの接続[ログイン]ダイアログにて、接続先URLにHTTPSを指定すると、接続時にマネージャのホスト認証が行われます。

10.3 Hinemos WebクライアントへのHTTPSによる接続

Hinemos Webクライアントサービスは、各ブラウザからのHTTPS通信に対応します。

1. /opt/hinemos_web/conf/server.xmlを編集します。

```
<!-- ←コメントアウト
<Connector port="80" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="8443"
    maxThreads="32"
    />
--> ←コメントアウト
<!-- Define a SSL HTTP/1.1 Connector on port 8443
    This connector uses the NIO implementation that requires the JSSE
    style configuration. When using the APR/native implementation, the
    OpenSSL style configuration is required as described in the APR/native
    documentation -->
    ←コメントを外す
<Connector port="443" protocol="org.apache.coyote.http11.Http11NioProtocol"
    maxThreads="32" SSLEnabled="true" scheme="https" secure="true"
    clientAuth="false" sslProtocol="TLS"
    keystoreFile="/opt/hinemos_web/.keystore" keystorePass="changeit"
    />
    ←コメントを外す
```

2. 以下のコマンドにより、証明書を作成します。

```
keytool -genkey -alias tomcat -keyalg RSA -keystore /opt/hinemos_web/.keystore
```

キーストアのパスワードを入力してください:(1でコメントを外したパラメータ「keystorePass」に設定した値を入力)
新規パスワードを再入力してください:

姓名を入力してください。

[Unknown]: (何も入力せずENTER)

織単位名を入力してください。

[Unknown]: (何も入力せずENTER)

組織名を入力してください。

[Unknown]: (何も入力せずENTER)

都市名または地域名を入力してください。

[Unknown]: (何も入力せずENTER)

州名または地方名を入力してください。

[Unknown]: (何も入力せずENTER)

この単位に該当する 2 文字の国番号を入力してください。

[Unknown]: (何も入力せずENTER)

CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown でよろしいですか?

[いいえ]: (はいを入力)

<tomcat> の鍵パスワードを入力してください。

(キーストアのパスワードと同じ場合は RETURN を押してください): (何も入力せずENTER)

3. Hinemos Webクライアントサービスを再起動すると設定が反映されます。

11 ポート変更

11.1 Hinemosマネージャのポート変更

Hinemosマネージャのポート番号の変更には、Hinemosクライアントのメンテナンスパースペクティブ→メンテナンス[Hinemosプロパティ]ビューを開き、以下のパラメータを変更します。

```
ws.client.address=http://0.0.0.0:8080
ws.agent.address=http://0.0.0.0:8081
```

以下のパラメータを設定します。

- ws.client.address=Hinemosクライアントからマネージャへの接続アドレス（ポートを変更します）
- ws.agent.address=Hinemosエージェントからマネージャへの接続するアドレス（ポートを変更します）

なお、これらのパラメータの変更反映には、Hinemosマネージャの再起動が必要です。

11.2 Hinemos Webクライアントサービスのポート変更

Hinemos Webクライアントのポート番号の変更には、/opt/hinemos_web/conf/server.xmlの以下のパラメータを変更します。

```
<Connector port="80" protocol="HTTP/1.1"
  connectionTimeout="20000"
  redirectPort="8443"
  maxThreads="32"
 />
```

以下のパラメータを設定します。

- port=ブラウザからHinemos Webクライアントサービスへの接続ポート

設定の反映には、Webクライアントサービスの再起動が必要です。

11.3 Hinemosエージェントの即時反映用ポート

Hinemosエージェントに設定を即時に反映させるために、HinemosエージェントはデフォルトでUDP24005をLISTENしています。このポート番号を変更する場合は、Hinemosクライアントより、ノードプロパティの即時反映用ポートを変更してください。

12 動作ログ

12.1 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 12-1に示すログファイルに出力されます。

表 12-1 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/log4j.properties
出力レベル	priority INFO
ローテーション	起動時に常に上書き
内容	javaプロセス内の起動シーケンスに関するログ
ファイル名	jvm_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
内容	javaプロセスの標準出力/標準エラー出力/スレッドダンプ
ファイル名	hinemos_manager.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/log4j.properties

出力レベル	priority INFO
ローテーション	Daily(無期限)
内容	javaプロセスの動作ログ
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/hinemos.cfg
出力レベル	—
ローテーション	—
内容	PostgreSQLの標準出力/標準エラー出力
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	hinemos_internal.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/log4j.properties
出力レベル	priority INFO
ローテーション	Daily(無期限)
内容	INTERNALイベントのログ
ファイル名	hinemos_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/log4j.properties
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	gc.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/hinemos.cfg
出力レベル	—
ローテーション	20MByteごとに最大5世代
内容	javaプロセスのGarbage Collectionログ

12.2 javaプロセスのログ出力・ログローテーションを変更する

- 変更ファイル

Hinemosで利用するjavaプロセスのログ出力のレベル、ログローテーションの方式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/log4j.properties

```
log4j.rootCategory=info, manager
```

本ファイルでhinemos_manager.logのログ出力レベルおよびログローテーションの設定を変更することができます。

ログローテーションの設定方法等Log4jの設定例については、<http://logging.apache.org/log4j/1.2/manual.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60秒間隔（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。

12.3 PostgreSQLのログ出力・ログローテーションを変更する

- 変更ファイル

Hinemosで利用するPostgreSQLのログ出力のレベルおよびログローテーションの設定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/9.3/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。

12.4 操作ログのログ出力・ログローテーションを変更する

- 変更ファイル

操作ログ(/opt/hinemos/var/log/hinemos_operation.log)のログ出力設定 およびログローテーションの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/log4j.properties

```
log4j.category.HinemosOperation=info, operation
```

なお、priority valueとログ出力対象の操作の対応関係は表 12-2の通りです。

表 12-2. 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60秒間隔（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。

12.5 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 12-3に示すログファイルに出力されます。

表 12-3 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_stdout.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/スレッドダンプ
ファイル名	agent_stderr.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エラー出力
ファイル名	gc.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/hinemos_agent.cfg
出力レベル	—
ローテーション	10MByteごとに最大5世代
内容	javaプロセスのGarbage Collectionログ

Windows版エージェントのログは表 12-4に示すログファイルに出力されます。

表 12-4 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	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
ファイル名	gc.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]\bin\RegistAgentService.bat
出力レベル	—
ローテーション	java-1.7.0を利用している場合：10MByteごとに最大5世代 java-1.6.0を利用している場合：なし
内容	javaプロセスのGarbage Collectionログ

12.6 Hinemosエージェントのログ出力・ログローテーションを変更する

・ 変更ファイル

Hinemosエージェントの出力レベルおよびログローテーションの設定を変更するには、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 ← Appenderの指定（デフォルトではファイルサイズでローテーションする）
log4j.appender.file.MaxFileSize = 20MB ← 最大ファイルサイズ
log4j.appender.file.MaxBackupIndex = 4 ← 最大バックアップログファイル世代数
log4j.appender.file.Append=true ← 起動時に追記するか否か
log4j.appender.file.layout=org.apache.log4j.PatternLayout ← レイアウトを行うクラスの指定
log4j.appender.file.layout.ConversionPattern=%d %-5p [%t] [%c] %m%n ← 出力フォーマットパターン

### 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 ← ログレベル、システムログ転送出力先の指定
log4j.logger.com.clustercontrol=info, file ← ログレベル、エージェントログ出力先の指定

log4j.appender.file.File=${hinemos.agent.log.dir}/agent.log ← 出力ファイル名の設定
# 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://logging.apache.org/log4j/1.2/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。

12.7 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 12-5に示すログファイルに出力されます。

表 12-5 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\Client5.0
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
----	------------------

12.8 Hinemos Webクライアントのログファイル一覧

Hinemos Webクライアントのログは表 12-6に示すログファイルに出力されます。

表 12-6 Hinemosクライアントのログファイル一覧

ファイル名	webclient.log
格納ディレクトリ	/opt/hinemos_web/var/log
ログ出力設定ファイル	[Hinemos Web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	Daily(無期限)
内容	Hinemos Webクライアントのログ
ファイル名	catalina.out
格納ディレクトリ	/opt/hinemos_web/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 Webクライアントサービス起動
内容	Hinemos Webクライアントサービスのログ
ファイル名	catalina.boot.log
格納ディレクトリ	/opt/hinemos_web/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 Webクライアントサービスのログ
ファイル名	gc_webclient.log.*
格納ディレクトリ	/opt/hinemos_web/var/log
ログ出力設定ファイル	—
出力レベル	—
ローテーション	java-1.7.0を利用している場合：10MByteごとに最大5世代 java-1.6.0を利用している場合：なし
内容	javaプロセスのGarbage Collectionログ

13 Hinemosマネージャの設定一覧

Hinemosマネージャの設定は、データベースに定義されています。Hinemosクライアントのメンテナンスパースペクティブ→メンテナンス[Hinemosプロパティ]ビューから設定を追加・変更できます。

表13-1. INTERNALイベントの設定値

プロパティ名	説明	データ型	デフォルト値
internal.command	INTERNALイベントによりコマンド実行する場合はtrue、しない場合はfalseとする。	真偽値	false

internal.command.commandline	INTERNALイベントにより実行するコマンド	文字列	echo #[GENERATION_DATE] #[MESSAGE] >> /tmp/test.txt
internal.command.priority	コマンド実行を行う場合の、INTERNALイベントの重要度「危険(critical)/不明(unknown)/警告(warning)/ 情報(info)」を設定する。設定した重要度以上のINTERNAL イベントが通知される。 重要度の優先順位については「Hinemosユーザマニュアル 5.4 監視[スコープ]ビューでの監視結果確認」を参照してください。	文字列	info
internal.command.timeout	INTERNALイベントによるコマンド実行におけるタイムアウト値[msec]	数値	15000
internal.command.user	INTERNALイベントによるコマンド実行における実効ユーザ	文字列	root
internal.event	INTERNALイベントを監視履歴[イベント]ビューへ出力する場合はtrue、出力しない場合はfalseとする。	真偽値	true
internal.event.priority	INTERNALイベントの監視履歴[イベント]ビューへの出力における重要度「危険(critical)/不明(unknown)/ 警告(warning)/情報(info)」を設定する。設定した重要度 以上のINTERNAL イベントが通知される。 重要度の優先順位については「Hinemosユーザマニュアル 5.4 監視[スコープ]ビューでの監視結果確認」を参照してください。	文字列	info
internal.file	INTERNALイベントをログファイルへ出力する場合はtrue、出力しない場合はfalseとする。	真偽値	true
internal.file.priority	INTERNALイベントのログファイルへの出力における重要度「危険(critical)/不明(unknown)/警告(warning)/ 情報(info)」を設定する。設定した重要度以上のINTERNAL イベントが通知される。 重要度の優先順位については「Hinemosユーザマニュアル 5.4 監視[スコープ]ビューでの監視結果確認」を参照してください。	文字列	info
internal.mail	INTERNALイベントをメール送信する場合はtrue、送信しない場合はfalseとする。	真偽値	false
internal.mail.address	INTERNALイベントのメール送信先メールアドレス	文字列	user1@host.domain, user2@host.domain
internal.mail.priority	メール送信を行う場合の、INTERNALイベントの重要度「危険(critical)/不明(unknown)/警告(warning)/ 情報(info)」を設定する。設定した重要度以上のINTERNAL イベントが通知される。 重要度の優先順位については「Hinemosユーザマニュアル 5.4 監視[スコープ]ビューでの監視結果確認」を参照してください。	文字列	info
internal.syslog	INTERNALイベントをsyslog送信する場合はtrue、送信しない場合はfalseとする。	真偽値	false
internal.syslog.facility	INTERNALイベントのsyslog送信におけるfacility	文字列	daemon
internal.syslog.host	INTERNALイベントをsyslog送信するホスト	文字列	192.168.1.1, 192.168.1.2
internal.syslog.port	INTERNALイベントをsyslog送信するポート	数値	514
internal.syslog.priority	Syslog送信を行う場合の、INTERNALイベントの重要度「危険(critical)/不明(unknown)/警告(warning)/ 情報(info)」を設定する。設定した重要度以上のINTERNAL イベントが通知される。 重要度の優先順位については「Hinemosユーザマニュアル 5.4 監視[スコープ]ビューでの監視結果確認」を参照してください。	文字列	info

internal.syslog.severity	INTERNALイベントのsyslog送信におけるseverity	文字列	alert
--------------------------	-----------------------------------	-----	-------

表13-2. 監視機能の設定値

プロパティ名	説明	データ型	デフォルト値
monitor.sql.jdbc.driver	SQL監視で利用するJDBCドライバの数	数値	3
monitor.sql.jdbc.driver.classname.1	SQL監視で利用する1つ目のJDBCドライバのクラス名	文字列	org.postgresql.Driver
monitor.sql.jdbc.driver.classname.2	SQL監視で利用する2つ目のJDBCドライバのクラス名	文字列	com.mysql.jdbc.Driver
monitor.sql.jdbc.driver.classname.3	SQL監視で利用する3つ目のJDBCドライバのクラス名	文字列	oracle.jdbc.driver.OracleDriver
monitor.sql.jdbc.driver.logintimeout.1	SQL監視で利用する1つ目のJDBCドライバのタイムアウト [sec]	数値	30
monitor.sql.jdbc.driver.logintimeout.2	SQL監視で利用する2つ目のJDBCドライバのタイムアウト [sec]	数値	30
monitor.sql.jdbc.driver.logintimeout.3	SQL監視で利用する3つ目のJDBCドライバのタイムアウト [sec]	数値	30
monitor.sql.jdbc.driver.name1	SQL監視で利用する1つ目のJDBCドライバにおける、設定ダイアログ上の表示名	文字列	PostgreSQL
monitor.sql.jdbc.driver.name2	SQL監視で利用する2つ目のJDBCドライバにおける、設定ダイアログ上の表示名	文字列	MySQL
monitor.sql.jdbc.driver.name3	SQL監視で利用する3つ目のJDBCドライバにおける、設定ダイアログ上の表示名	文字列	Oracle
monitor.sql.jdbc.driver.properties.1	SQL監視で利用する1つ目のJDBCドライバの追加プロパティ	文字列	socketTimeout=3600&tcpKeepAlive=true
monitor.sql.jdbc.driver.properties.2	SQL監視で利用する2つ目のJDBCドライバの追加プロパティ	文字列	未定義
monitor.sql.jdbc.driver.properties.3	SQL監視で利用する3つ目のJDBCドライバの追加プロパティ	文字列	未定義
monitor.log.line.max.length	システムログ監視の#[LOG_LINE]に埋め込む最大文字列長 [文字数]。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。	数値	256
monitor.common.delete.cascade.perfdata	監視設定削除時に、性能データも同時に削除する場合は "on"、しない場合は "off"。	文字列	on
monitor.common.report.event.bom	監視履歴[イベント]ビューのダウンロード機能で生成するファイルに BOMを付与する場合はtrue、付与しない場合はfalse。	真偽値	true
monitor.common.report.eventt.count	監視履歴[イベント]ビューのダウンロード機能で生成するファイルに含めるイベント情報の最大件数	数値	2000
monitor.common.report.event.format	監視履歴[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	文字列	yyyy/MM/dd HH:mm:ss
monitor.common.report.event.separator	監視履歴[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	文字列	,
monitor.snmptrap.listen.address	[要Hinemosマネージャ再起動] javaプロセスが監視対象とするSNMPTRAPを受信するための待ち受けアドレス	文字列	0.0.0.0
monitor.snmptrap.listen.port	[要Hinemosマネージャ再起動] javaプロセスが監視対象とするSNMPTRAPを受信するための待ち受けポート番号	数値	162
monitor.snmptrap.org.message.community	SNMPTRAP監視において、オリジナルメッセージにコミュニティ名を出力する場合はtrue、出力しない場合はfalse。	真偽値	true

monitor.snmptrap.org.message.varbind	SNMPTRAP監視において、オリジナルメッセージにvarbindを出力する場合はtrue、出力しない場合はfalse。	真偽値	true
monitor.snmptrap.v3.user	[要Hinemosマネージャ再起動] 受信するバージョン3のSNMPTRAPのユーザ	文字列	hinemos
monitor.snmptrap.v3.security.level	[要Hinemosマネージャ再起動] 受信するバージョン3のSNMPTRAPのセキュリティレベル (noauth_nopriv/auth_nopriv/auth_priv)	文字列	noauth_nopriv
monitor.snmptrap.v3.auth.password	[要Hinemosマネージャ再起動] 受信するバージョン3のSNMPTRAPの認証パスワード	文字列	authPassword
monitor.snmptrap.v3.auth.protocol	[要Hinemosマネージャ再起動] 受信するバージョン3のSNMPTRAPの認証プロトコル(MD5/SHA)	文字列	MD5
monitor.snmptrap.v3.priv.password	[要Hinemosマネージャ再起動] 受信するバージョン3のSNMPTRAPの暗号化パスワード	文字列	privPassword
monitor.snmptrap.v3.priv.protocol	[要Hinemosマネージャ再起動] 受信するバージョン3のSNMPTRAPの暗号化プロトコル(DES/AES)	文字列	DES
monitor.snmptrap.stats.interval	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。	数値	100
monitor.systemlog.listen.address	[要Hinemosマネージャ再起動] javaプロセスが監視対象とするsyslogを受信するための待ち受けアドレス	文字列	0.0.0.0
monitor.systemlog.listen.port	[要Hinemosマネージャ再起動] javaプロセスが監視対象とするsyslogを受信するための待ち受けポート番号	数値	24514
monitor.systemlog.stats.interval	syslog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。	数値	未定義
monitor.http.ssl.trustall	HTTPS接続でHTTP監視を行う際にサーバ証明書を利用しない場合はtrue、利用する場合はfalseをとする。	真偽値	true
monitor.winservice.ssl.trustall	HTTPS接続でWindowsサービス監視を行う際にサーバ証明書を利用しない場合はtrue、利用する場合はfalseをとする。	真偽値	true

表13-3. 通知機能の設定値

プロパティ名	説明	データ型	デフォルト値
notify.log.escalate.manager.hostname	ログ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	文字列	未定義
notify.log.escalate.manager.protocol	ログエスカレーション通知の転送プロトコル(udp, tcp)	文字列	udp
notify.log.escalate.manager.retry.count	ログエスカレーション通知の試行回数	数値	1
notify.log.escalate.manager.retry.interval	ログエスカレーション通知のリトライ時の間隔[msec]	数値	10000
notify.command.create.mode	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	文字列	auto
notify.command.success.exit	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。	数値	0
notify.date.format	通知に利用可能な変数(#[GENERATION_DATE])の日付書式	文字列	yyyy/MM/dd HH:mm:ss
notify.remove.subkey.MON_HTTP_S	監視詳細の削除有無。trueを指定すると、HTTP監視(文字列)の結果から通知を行う際に、監視詳細が削除され空となる。	真偽値	false
notify.remove.subkey.MON_HTTP_SCE	監視詳細の削除有無。trueを指定すると、HTTP監視(シナリオ)の結果から通知を行う際に、監視詳細が削除され空となる。	真偽値	false

notify.remove.subkey. MON_SNMP_S	監視詳細の削除有無。 trueを指定すると、SNMP監視(文字列)の結果から通知を行う際に、監視詳細が削除され空となる。	真偽値	false
notify.remove.subkey. MON_SNMP_TRP	監視詳細の削除有無。 trueを指定すると、SNMPTRAP監視(トラップ)の結果から通知を行う際に、監視詳細が削除され空となる。	真偽値	false
notify.remove.subkey. MON_SQL_S	監視詳細の削除有無。 trueを指定すると、SQL監視(文字列)の結果から通知を行う際に、監視詳細が削除され空となる。	真偽値	false
notify.remove.subkey. MON_WINEVENT_S	監視詳細の削除有無。 trueを指定すると、Windowsイベント監視(文字列)の結果から通知を行う際に、監視詳細が削除され空となる。	真偽値	false
notify.remove.subkey. MON_SYSLOG_S	監視詳細の削除有無。 trueを指定すると、システムログ監視(文字列)の結果から通知を行う際に、監視詳細が削除され空となる。	真偽値	false
notify.remove.subkey. MON_LOGFILE_S	監視詳細の削除有無。 trueを指定すると、ログファイル監視(文字列)の結果から通知を行う際に、監視詳細が削除され空となる。	真偽値	false

表13-4. メール通知の設定値

プロパティ名	説明	データ型	デフォルト値
mail.charset.address	メール通知のアドレスの文字コード	文字列	UTF-8
mail.charset.content	メール通知の本文の文字コード	文字列	UTF-8
mail.charset.subject	メール通知の題名の文字コード	文字列	UTF-8
mail.errors.to.address	エラーメールの送信先アドレス (Error-To)	文字列	admin@hinemos.com
mail.from.address	送信元メールアドレス (From)	文字列	admin@hinemos.com
mail.from.personal.name	送信元名 (From)	文字列	Hinemos Admin
mail.reply.personal.name	返信先名 (Reply-To)	文字列	Hinemos Admin
mail.reply.to.address	返信先メールアドレス (Reply-To)	文字列	admin@hinemos.com
mail.transport.tries	メール送信試行回数	数値	1
mail.transport.tries.interval	メール送信の再試行時のインターバル[msec]	数値	10000
mail.transport.password	メール送信時の認証パスワード	文字列	password
mail.transport.user	メール送信時の認証ユーザ	文字列	nobody
mail.smtp.auth	SMTP AUTHの利用有無	真偽値	false
mail.smtp.connectiontimeout	SMTPサーバとの接続タイムアウト[msec]	数値	15000
mail.smtp.host	SMTPサーバのIPアドレス	文字列	127.0.0.1
mail.smtp.port	SMTPサーバのポート番号	数値	25
mail.smtp.timeout	SMTPサーバのリクエストタイムアウト[msec]	数値	30000
mail.smtp.starttls.enable	SSL/TLS(STARTTLS)を必要とするSMTPサーバを利用する	真偽値	false
mail.transport.protocol	SMTPサーバとの接続プロトコル	文字列	smtp

表13-5. 性能機能の設定値

プロパティ名	説明	データ型	デフォルト値
--------	----	------	--------

performance.export.encode	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	文字列	MS932
performance.export.line.separator	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	文字列	CRLF
performance.use.old	監視機能の前回値保持期間[sec] ※監視機能に「収集」があるもの(数値監視)のみ対象	数値	660

表13-6. ジョブ機能の設定値

プロパティ名	説明	データ型	デフォルト値
quartz.dbms.jobStore.misfireThreshold	[要Hinemosマネージャ再起動] Hinemosマネージャ再起動時に遡って実行されるジョブの有効期間[msec]	数値	3600000
job.message.max.length	ジョブ履歴のログの最大文字数 Agent.propertiesのjob.message.length も合わせて変更する必要があります。 /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)	数値	2048

表13-7. リポジトリ機能の設定値

プロパティ名	説明	データ型	デフォルト値
repository.device.search.interval	自動デバイスサーチの実行間隔[min]。0の場合は自動デバイスサーチは無効となる。	数値	5
repository.device.search.verbose	デバイスサーチ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するが、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る。	真偽値	true
repository.device.search.prop.device.cpu	ノードプロパティのデバイス->CPU情報を自動デバイスサーチの対象とする場合はtrue、対象としない場合はfalseとする。	真偽値	true
repository.device.search.prop.device.memory	ノードプロパティのデバイス->メモリ情報を自動デバイスサーチの対象とする場合はtrue、対象としない場合はfalseとする。	真偽値	true
repository.device.search.prop.device.nic	ノードプロパティのデバイス->NIC情報を自動デバイスサーチの対象とする場合はtrue、対象としない場合はfalseとする。	真偽値	true
repository.device.search.prop.device.disk	ノードプロパティのデバイス->ディスク情報を自動デバイスサーチの対象とする場合はtrue、対象としない場合はfalseとする。	真偽値	true
repository.device.search.prop.device.filesystem	ノードプロパティのデバイス->ファイルシステム情報を自動デバイスサーチの対象とする場合はtrue、対象としない場合はfalseとする。	真偽値	true
repository.device.search.prop.device.general	ノードプロパティのデバイス->汎用デバイス情報を自動デバイスサーチの対象とする場合はtrue、対象としない場合はfalseとする。	真偽値	false
repository.device.search.prop.basic.hardware	ノードプロパティのサーバ基本情報->ハードウェアを自動デバイスサーチの対象とする場合はtrue、対象としない場合はfalseとする。	真偽値	false
repository.device.search.prop.basic.network	ノードプロパティのサーバ基本情報->ネットワークを自動デバイスサーチの対象とする場合はtrue、対象としない場合はfalseとする。	真偽値	true

repository.device.search.prop.basic.os	ノードプロパティのサーバ基本情報->OSを自動デバイスサーチの対象とする場合はtrue、対象としない場合はfalseとする。	真偽値	false
repository.device.search.prop.basic.agent	ノードプロパティのサーバ基本情報->Hinemosエージェントを自動デバイスサーチの対象とする場合はtrue、対象としない場合はfalseとする。	真偽値	false

表13-8. セルフチェック機能の設定値

プロパティ名	説明	データ型	デフォルト値
selfcheck.alert.threshold	通知するまでの異常の検知回数	数値	3
selfcheck.interval	[要Hinemosマネージャ再起動] 内部コンポーネントの状態確認間隔[sec]	数値	150
selfcheck.monitoring.asyncntask.queue	非同期処理用キューのチェックを有効とする場合はtrue、無効とする場合はfalseとする。	真偽値	true
selfcheck.monitoring.asyncntask.queue.list	非同期処理用キューのチェックするキューの名前と最大メッセージ数	文字列	(注2)
selfcheck.monitoring.db	データベースへのアクセスチェックを有効とする場合はtrue、無効とする場合はfalseとする。	真偽値	true
selfcheck.monitoring.db.validationquery	データベースへのアクセスチェックするSQL	文字列	SELECT 1 FOR UPDATE
selfcheck.monitoring.filesystem.usage	ファイルシステムの空き領域チェックを有効にする場合はtrue、無効とする場合はfalseとする	真偽値	false
selfcheck.monitoring.filesystem.usage.list	Hinemosマネージャサーバのファイルシステム監視閾値。半角カンマ区切りで複数指定可能。デフォルトでは「/」配下を上限50%で監視。	文字列	/:50
selfcheck.monitoring.job.runningssession	実行中のジョブチェックを有効とする場合はtrue、無効とする場合はfalseとする。	真偽値	true
selfcheck.monitoring.job.runningssession.threshold	実行中のジョブセッション数の増大を判定する閾値	数値	1000
selfcheck.monitoring.jvm.freeheap	Java VMの空きメモリ領域チェックを有効とする場合はtrue、無効とする場合はfalseとする。	真偽値	true
selfcheck.monitoring.jvm.freeheap.threshold	Java VMの空きメモリ領域の閾値[MByte]	数値	32
selfcheck.monitoring.scheduler.delay	スケジューラのチェックを有効とする場合はtrue、無効とする場合はfalseとする。	真偽値	true
selfcheck.monitoring.scheduler.delay.threshold	スケジューラの動作遅延を判定する最大遅延時間 [sec]	数値	300
selfcheck.monitoring.snmptrap.queue	SNMPTRAP監視の内部キューのチェックを有効とする場合はtrue、無効とする場合はfalseとする。	真偽値	true
selfcheck.monitoring.snmptrap.queue.threshold	SNMPTRAP監視の内部キューの最大メッセージ数	数値	10000
selfcheck.monitoring.swapout	スワップアウトの発生チェックを有効とする場合はtrue、無効とする場合はfalseとする。	真偽値	false
selfcheck.monitoring.systemlog.queue	システムログ監視の内部キューのチェックを有効とする場合はtrue、無効とする場合はfalseとする。	真偽値	true
selfcheck.monitoring.systemlog.queue.threshold	システムログ監視の内部キューの最大メッセージ数	数値	10000
selfcheck.monitoring.table.size	内部テーブルのサイズチェックを有効とする場合はtrue、無効とする場合はfalseとする。	真偽値	true

selfcheck.monitoring.table.size.list	サイズチェックする、内部テーブル名と最大サイズ	文字列	(注3)
selfcheck.monitoring.thread.activity	停滞中スレッド数のチェックを有効とする場合はtrue、無効とする場合はfalseとする。	真偽値	true
selfcheck.monitoring.thread.activity.threshold	停滞中スレッドの停滞時間の最大値[sec]。	数値	300
selfcheck.monitoring.ws.queue	Webサービスのアクセスキューのチェックを有効とする場合はtrue、無効とする場合はfalseとする。	真偽値	true
selfcheck.monitoring.ws.queue.threshold	Webサービスのアクセスキューの最大メッセージ数	数値	10000
selfcheck.snmp.community	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	文字列	public
selfcheck.snmp.port	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	数値	161
selfcheck.snmp.retries	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	数値	3
selfcheck.snmp.timeout	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]	数値	3000
selfcheck.snmp.version	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)	文字列	2c
selfcheck.starup.delay	[要Hinemosマネージャ再起動] セルフチェック開始時間[sec]	数値	90
selfcheck.threadpool.size	セルフチェック機能が利用するスレッド数	数値	4

(注1) selfcheck.snmp.* を除き、デフォルト値からの変更は非推奨です。

(注2) NotifyStatusTaskFactory:10000, NotifyEventTaskFactory:10000, NotifyMailTaskFactory:10000, NotifyCommandTaskFactory:10000, NotifyLogEscalationTaskFactory:10000, NotifyJobTaskFactory:10000

(注3) log.cc_event_log:5120:MBYTE, log.cc_calculated_data:20480:MBYTE, log.cc_job_session:100000:COUNT

表13-9. Hinemosマネージャの接続に関する設定値

プロパティ名	説明	データ型	デフォルト値
ws.client.address	[要Hinemosマネージャ再起動] Hinemosマネージャがクライアントからの接続を待ち受けるIPアドレス	文字列	http://0.0.0.0:8080
ws.agent.address	[要Hinemosマネージャ再起動] Hinemosマネージャがエージェントからの接続を待ち受けるIPアドレス	文字列	http://0.0.0.0:8081
ws.https.keystore.password	[要Hinemosマネージャ再起動] キーストアのパスワード	文字列	hinemos
ws.https.keystore.path	[要Hinemosマネージャ再起動] キーストアのディレクトリパス	文字列	/root/keystore
ws.https.keystore.type	[要Hinemosマネージャ再起動] キーストアのタイプ	文字列	PKCS12
ws.https.protocol	[要Hinemosマネージャ再起動] HTTPS通信のセキュアプロトコル(TLS/TLSv1.2)	文字列	TLS※1

※1 TLSを選択した場合、HinemosマネージャはSSL3.0、TLS1.0での通信が可能となります。TLSv1.2を選択した場合、HinemosマネージャはSSL3.0、TLS1.0、TLS1.1、TLS1.2での通信が可能となります。実際にどのバージョンで通信が行われるかは、エージェント・クライアント側で通信可能なプロトコルバージョン内から通信開始時に決定されます。

表13-10. 環境構築に関する設定値

プロパティ名	説明	データ型	デフォルト値
infra.transfer.winrm.url	HinemosマネージャのURL(環境構築機能のファイル転送モジュール(WinRM)で利用される)	文字列	http://[hostname コマンドの実行結果]:8080 ※1

infra.winrm.ssl.trustall	HTTPS接続で環境構築機能の環境構築モジュール(WinRM)を実行する際にサーバ証明書を利用しない場合はtrue、利用する場合はfalseをとする。	真偽値	true
infra.max.file.size	アップロードするファイルの制限サイズ(byte単位)	数値	67108864

※1 上記のデフォルト値にはホスト名が設定されるため、WindowsノードからアクセスできるようにDNS設定を行うか、IPアドレスを設定する必要があります。

14 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)

パラメータを変更した場合、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[common.invalid.char.replace]

プロパティ	common.invalid.char.replace
プロパティ名	オリジナルメッセージに含まれる制御文字の置換方法
説明	本パラメータをtrueとした場合、オリジナルメッセージに含まれる制御文字は、invalid.char.replace.toで指定した値で置換されます。falseとした場合、オリジナルメッセージに含まれる制御文字は、16進数で置換されます。
データ型	文字列
デフォルト値	false

パラメータ[common.invalid.char.replace.to]

プロパティ	common.invalid.char.replace.to
プロパティ名	オリジナルメッセージに含まれる制御文字の置換後文字列
説明	invalid.char.replaceをtrueとした場合、オリジナルメッセージに含まれる制御文字は、本パラメータで指定された文字列に置換されます。
データ型	文字列
デフォルト値	?

パラメータ[monitor.message.length]

プロパティ	monitor.message.length
プロパティ名	ログファイル監視/Windowsイベント監視のメッセージの最大バイト数
説明	ログファイル監視/Windowsイベント監視の監視結果のメッセージの最大バイト数 ログファイル監視の場合、読み込む1行の最大バイト数(byte)と一致する。メッセージのうち、このバイト数を超える部分は切り捨てられる。
データ型	整数
デフォルト値	1024

パラメータ[job.history.period]

プロパティ	job.history.period
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
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(ミリ秒)
データ型	整数
デフォルト値	604800000

パラメータ[job.message.length]

プロパティ	job.message.length
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 Hinemosクライアントのメンテナンスパースペクティブ→メンテナンス [Hinemosプロパティ] ビューを開き、 job.message.max.lengthも合わせて設定する必要があります。
データ型	整数
デフォルト値	1024

パラメータ[job.stream.charset]

プロパティ	job.stream.charset
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	Linux版エージェント：UTF-8, Windows版エージェント：MS932

パラメータ[job.command.mode]

プロパティ	job.command.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, unix.su)
デフォルト値	auto

パラメータ[job.command.login]

プロパティ	job.command.login
プロパティ名	ジョブ実行時のコマンド動作に関する環境変数設定
説明	ジョブ実行時のコマンド動作に関する環境変数設定 true … 実効ユーザの環境変数に切り替えてコマンドを実行する false … 実効ユーザとしてコマンドを実行する。環境変数はHinemosエージェント起動ユーザの設定で行う
データ型	真偽値
デフォルト値	false

パラメータ[job.filecheck.interval]

プロパティ	job.filecheck.interval
プロパティ名	ジョブファイルチェックの実行間隔
説明	ジョブファイルチェックの実行間隔 (ミリ秒)
データ型	整数
デフォルト値	10000

パラメータ[job.filecheck.sender.tries]

プロパティ	job.filecheck.sender.tries
プロパティ名	ジョブファイルチェックの結果送信処理の最大試行回数
説明	ジョブファイルチェック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[job.filecheck.sender.interval]

プロパティ	job.filecheck.sender.tries
プロパティ名	ジョブファイルチェックの結果送信処理の試行間隔
説明	ジョブファイルチェック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[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における転送先ノードの公開鍵
説明	ファイル転送ジョブにおける転送先ノードの公開鍵
データ型	文字列
デフォルト値	未定義

パラメータ[root.authorized.keys.path]

プロパティ	root.authorized.keys.path
プロパティ名	ファイル転送ジョブにおける転送元ノードのauthorized_keysのパス
説明	ファイル転送ジョブにおける転送元ノードのauthorized_keysのパス
データ型	文字列
デフォルト値	/root/.ssh/authorized_keys

パラメータ[monitor.logfile.filter.interval]

プロパティ	monitor.logfile.filter.interval
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[monitor.logfile.filter.filesizecheck.period]

プロパティ	monitor.logfile.filter.filesizecheck.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[monitor.logfile.filter.fileheadercheck.period]

プロパティ	monitor.logfile.filter.fileheadercheck.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[monitor.logfile.filter.fileheadercheck.size]

プロパティ	monitor.logfile.filter.fileheadercheck.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[monitor.logfile.filter.maxsize]

プロパティ	monitor.logfile.filter.maxsize
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	info

パラメータ[monitor.logfile.syslog.program]

プロパティ	monitor.logfile.syslog.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_agent

パラメータ[monitor.logfile.forwarding.queue.maxsize]

プロパティ	monitor.logfile.forwarding.queue.maxsize
プロパティ名	ログファイル監視の内部キューの最大メッセージ数
説明	ログファイル監視の内部キューの最大メッセージ数
データ型	整数
デフォルト値	5000

パラメータ[monitor.logfile.forwarding.transport.maxsize]

プロパティ	monitor.logfile.forwarding.transport.maxsize
プロパティ名	ログファイル監視の結果送信処理における送信分割単位
説明	ログファイル監視の結果送信処理における一度に送信可能なメッセージ数の上限。内部キューのメッセージ数が設定値よりも大きい場合、送信処理を分割して行う。
データ型	整数
デフォルト値	100

パラメータ[monitor.logfile.forwarding.transport.maxtries]

プロパティ	monitor.logfile.forwarding.transport.maxtries
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	900

パラメータ[monitor.logfile.forwarding.transport.interval.size]

プロパティ	monitor.logfile.forwarding.transport.interval.size
プロパティ名	ログファイル監視の結果送信処理の試行契機(メッセージ数)
説明	ログファイル監視の結果送信処理の試行契機となるメッセージ数。 前回の結果送信処理時から内部キューにメッセージが格納された数が設定値に達すると結果送信処理を行う。
データ型	整数
デフォルト値	15

パラメータ[monitor.logfile.forwarding.transport.interval.msec]

プロパティ	monitor.logfile.forwarding.transport.interval.msec
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	1000

パラメータ[monitor.logfile.random.access.file]

プロパティ	monitor.logfile.random.access.file
プロパティ名	Windows環境上におけるログファイル監視のログローテーション方式
説明	Windows環境上にてログファイル監視をする場合に本パラメータを"windows"とすることで、ログローテーションの際に対象ログのリネームを行うmv方式に対応する。
データ型	文字列
デフォルト値	Linux版エージェント : linux, Windows版エージェント : windows

パラメータ[monitor.custom.thread]

プロパティ	monitor.custom.thread
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[monitor.custom.command.mode]

プロパティ	monitor.custom.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, unix.su)
デフォルト値	auto

パラメータ[monitor.custom.command.login]

プロパティ	monitor.custom.command.login
プロパティ名	カスタム監視のコマンド動作に関する環境変数設定
説明	カスタム監視のコマンド動作に関する環境変数設定 true … 実効ユーザの環境変数に切り替えてコマンドを実行する false … 実効ユーザとしてコマンドを実行する。環境変数はHinemosエージェント起動ユーザの設定で行う
データ型	真偽値
デフォルト値	false

パラメータ[monitor.custom.buffer]

プロパティ	monitor.custom.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[monitor.custom.charset]

プロパティ	monitor.custom.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[monitor.custom.lineseparator]

プロパティ	monitor.custom.lineseparator
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[monitor.custom.forwarding.queue.maxsize]

プロパティ	monitor.custom.forwarding.queue.maxsize
プロパティ名	カスタム監視の内部キューの最大メッセージ数
説明	カスタム監視の内部キューの最大メッセージ数
データ型	整数
デフォルト値	5000

パラメータ[monitor.custom.forwarding.transport.maxsize]

プロパティ	monitor.custom.forwarding.transport.maxsize
プロパティ名	カスタム監視の結果送信処理における送信分割単位
説明	カスタム監視の結果送信処理における一度に送信可能なメッセージ数の上限。内部キューのメッセージ数が設定値よりも大きい場合、送信処理を分割して行う。
データ型	整数
デフォルト値	100

パラメータ[monitor.custom.forwarding.transport.maxtries]

プロパティ	monitor.custom.forwarding.transport.maxtries
プロパティ名	カスタム監視の結果送信処理の最大試行回数
説明	カスタム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	900

パラメータ[monitor.custom.forwarding.transport.interval.size]

プロパティ	monitor.custom.forwarding.transport.interval.size
プロパティ名	カスタム監視の結果送信処理の試行契機(メッセージ数)
説明	カスタム監視の結果送信処理の試行契機となるメッセージ数。 前回の結果送信処理時から内部キューにメッセージが格納された数が設定値に達すると結果送信処理を行う。
データ型	整数
デフォルト値	15

パラメータ[monitor.custom.forwarding.transport.interval.msec]

プロパティ	monitor.custom.forwarding.transport.interval.msec
プロパティ名	カスタム監視の結果送信処理の試行間隔(ミリ秒)
説明	カスタム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	1000

パラメータ[monitor.winevent.filter.interval]

プロパティ	monitor.winevent.filter.interval
プロパティ名	Windowsイベント監視の監視間隔
説明	Windowsイベント監視のイベントログ取得の実行間隔(ミリ秒)
データ型	整数
デフォルト値	10000

パラメータ[monitor.winevent.filter.timeout]

プロパティ	monitor.winevent.filter.timeout
プロパティ名	Windowsイベント監視のコマンドタイムアウト
説明	Windowsイベント監視のイベントログ取得用コマンドのタイムアウト値(ミリ秒)。-1を指定するとタイムアウトが無効となる。
データ型	整数
デフォルト値	-1

パラメータ[monitor.winevent.buffer]

プロパティ	monitor.winevent.buffer
プロパティ名	Windowsイベント監視のバッファサイズ
説明	Windowsイベント監視のイベントログ取得用コマンドの実行結果の標準出力を格納するバッファサイズ(byte)。
データ型	整数
デフォルト値	100000

パラメータ[monitor.winevent.maxevents]

プロパティ	monitor.winevent.maxevents
プロパティ名	Windowsイベント監視のコマンド実行あたりの最大イベントログ件数
説明	Windowsイベント監視のコマンド実行あたりに取得するイベントログ件数の上限値。-1を指定すると無制限となる。
データ型	整数
デフォルト値	-1

パラメータ[monitor.winevent.mode]

プロパティ	monitor.winevent.mode
プロパティ名	Windowsイベント監視のイベントログ取得モード
説明	Windowsイベント監視のイベントログ取得モード。
データ型	文字列(auto, get-winevent, get-eventlog, wevtutil)
デフォルト値	auto
その他	auto : OSをもとに自動判定 (Windows Server 2008/2008R2/2012/2012R2、Windows 7, 8, 8.1の場合: wevtutilとなる) get-winevent : Get-WinEventコマンドレットを使用 http://technet.microsoft.com/en-us/library/hh849682.aspx get-eventlog : Get-EventLogコマンドレットを使用 http://technet.microsoft.com/en-us/library/hh849834.aspx wevtutil : wevtutil.exeコマンドを使用

パラメータ[monitor.winevent.forwarding.queue.maxsize]

プロパティ	monitor.winevent.forwarding.queue.maxsize
プロパティ名	Windowsイベント監視の内部キューの最大メッセージ数
説明	Windowsイベント監視の内部キューの最大メッセージ数
データ型	整数
デフォルト値	5000

パラメータ[monitor.winevent.forwarding.transport.maxsize]

プロパティ	monitor.winevent.forwarding.transport.maxsize
プロパティ名	Windowsイベント監視の結果送信処理における送信分割単位
説明	Windowsイベント監視の結果送信処理における一度に送信可能なメッセージ数の上限。内部キューのメッセージ数が設定値よりも大きい場合、送信処理を分割して行う。
データ型	整数
デフォルト値	100

パラメータ[monitor.winevent.forwarding.transport.maxtries]

プロパティ	monitor.winevent.forwarding.transport.maxtries
プロパティ名	Windowsイベント監視の結果送信処理の最大試行回数
説明	Windowsイベント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	900

パラメータ[monitor.winevent.forwarding.transport.interval.size]

プロパティ	monitor.winevent.forwarding.transport.interval.size
-------	---

プロパティ名	Windowsイベント監視の結果送信処理の試行契機(メッセージ数)
説明	Windowsイベント監視の結果送信処理の試行契機となるメッセージ数。 前回の結果送信処理時から内部キューにメッセージが格納された数が設定値に達すると結果送信処理を行う。
データ型	整数
デフォルト値	15

パラメータ[monitor.winevent.forwarding.transport.interval.msec]

プロパティ	monitor.winevent.forwarding.transport.interval.msec
プロパティ名	Windowsイベント監視の結果送信処理の試行間隔(ミリ秒)
説明	Windowsイベント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	1000

パラメータ[monitor.winevent.return.char.replace]

プロパティ	monitor.winevent.return.char.replace
プロパティ名	Windowsイベント監視の改行文字を置換するための文字
説明	Windowsイベント監視の改行文字を置換するための文字
データ型	文字列
デフォルト値	#n;

パラメータ[monitor.winevent.gt.char.replace]

プロパティ	monitor.winevent.gt.char.replace
プロパティ名	Windowsイベント監視の"<"を置換するための文字
説明	Windowsイベント監視の"<"を置換するための文字
データ型	文字列
デフォルト値	#gt;

パラメータ[monitor.winevent.lt.char.replace]

プロパティ	monitor.winevent.lt.char.replace
プロパティ名	Windowsイベント監視の">"を置換するための文字
説明	Windowsイベント監視の">"を置換するための文字
データ型	文字列
デフォルト値	#lt;

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]:8081/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(ミリ秒)
説明	Hinemosマネージャへの問い合わせ間隔(ミリ秒)
データ型	整数
デフォルト値	30000

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(ミリ秒)
説明	Hinemosマネージャと接続する際のタイムアウト時間(ミリ秒)
データ型	整数
デフォルト値	10000

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(ミリ秒)
説明	Hinemosマネージャからの受信タイムアウト時間(ミリ秒)
データ型	整数
デフォルト値	60000

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

15 変更履歴

変更履歴

版	変更日	変更内容
第1版	2015/5/27	初版発行
第2版	2015/9/18	Windowsサービス監視のユーザについて補足 ログエスケレーション通知の Protokolについて補足 ファイル転送ジョブのセットアップにおける鍵の準備方法を記載 その他日本語の不備等を修正
第3版	2016/2/15	ジョブ起動コマンドの区切り文字の仕様について詳細化 その他日本語の不備等を修正
第4版	2016/8/5	JAVA_HOMEの変更について記載 HTTPS接続時にサーバ証明書を利用しない設定を記載 サーバ証明書を利用したHTTPSの設定について詳細化 JMX監視の設定を記載 INTERNALイベントの通知先設定の出力例を記載 job.message.max.lengthについて記載 その他、細かい記載漏れ、記載内容誤り等を修正
第5版	2017/2/28	infra.max.file.sizeについて記載 job.command.mode、monitor.custom.command.modeのunix.suについて記載 job.command.login、monitor.custom.command.loginについて記載

Hinemos ver5.0 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。