

UNIX 系OS向け
Hinemos ver2.1 エージェント
作成メモ

第 1.0 版

2006 年 11 月

UNIX 系 OS 向け Hinemos エージェント
作成メモ

変更履歴

版	変更日	変更内容
1.0	2006/11/8	初期リリース

目次

1	本書について	5
1.1	本文書の位置づけと免責	5
1.2	本文書の対象とする読者	5
2	Hinemos ver2.1 エージェントの概要	6
2.1	Hinemosの概要	6
2.2	本文中での記載用語	6
2.3	Hinemosエージェントについて	7
3	Hinemosエージェントの構成	8
3.1	利用しているソフトウェア	8
3.2	エージェントパッケージの内容	8
4	UNIX版エージェントの作成	10
4.1	UNIX版エージェント作成のための作業	10
4.2	UNIXで標準的に準備されているコマンド・サービスの確認	10
4.3	Hinemosエージェントに含まれているネイティブソフトウェアの確認	11
4.4	JREの確認	12
4.5	インストールスクリプト・スタートアップスクリプトの修正	12
4.5.1	インストールスクリプトの修正	12
4.5.2	スタートアップスクリプトの修正	13
5	接続・動作確認	14
5.1	監視管理機能の動作確認	14
5.2	性能管理機能の動作確認	20
5.3	ジョブ管理機能	22
5.4	一括制御の動作確認	24
6	補足情報	26
6.1	動作しない場合、補足の情報が必要な場合	26
6.2	正式動作確認環境	26
Appendix A	Expectの動作を確認するためのスクリプト	27

UNIX系 OS 向け Hinemos エージェント 作成メモ

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

テーマ名は「分散ファシリティ統合マネージャの開発」です。

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

商標

Hinemos は、NTT データの登録商標です。

Linuxは、Linus Torvalds氏の米国およびその他の国における登録商標または商標です。

その他、本書に記載されている会社名、製品名は、各社の登録商標または商標です。

なお、本文中には TM、®マークは表記しておりません。

1 本書について

1.1 本文書の位置づけと免責

本文書は、Hinemos エージェントを正式に動作保証されている環境以外で動作させるために必要な変更項目を記載したメモです。Hinemos における記述レベルは

マニュアル > ガイド > メモ

であり、メモである本文書は作成方法を記述したもので、本文書にしたがって作成したエージェントが動作することは保証いたしません。また本文書の記載の内容および内容を実施したことによる一切の損害について NTT データは保証いたしません。

UNIX環境で正式に動作確認が必要な場合には、NTTデータ¹まで、お問い合わせください。

1.2 本文書の対象とする読者

Hinemos を利用してシステム管理を実現しており、Red Hat EL4 以外の Linux や、UNIX を監視対象に加えたいと考えている方を対象して、他の Linux ディストリビューションや、UNIX の管理対象ノードを作成する方法を説明します。

¹ osdquery@nttdata.co.jp

2 Hinemos ver2.1 エージェントの概要

2.1 Hinemos の概要

Hinemos は、複数のコンピュータを単一のコンピュータのイメージで運用することを目的とした運用管理ツールです。ユーザが運用目的ごとにコンピュータをグループ登録できる機能を備え、運用目的に応じた監視や操作を GUI で容易に行う環境を提供します。

Hinemos を使用することによって、複数の用途の異なるコンピュータによって実現している業務システムをグループ化し、より少ない操作で効率的な運用が可能となります。

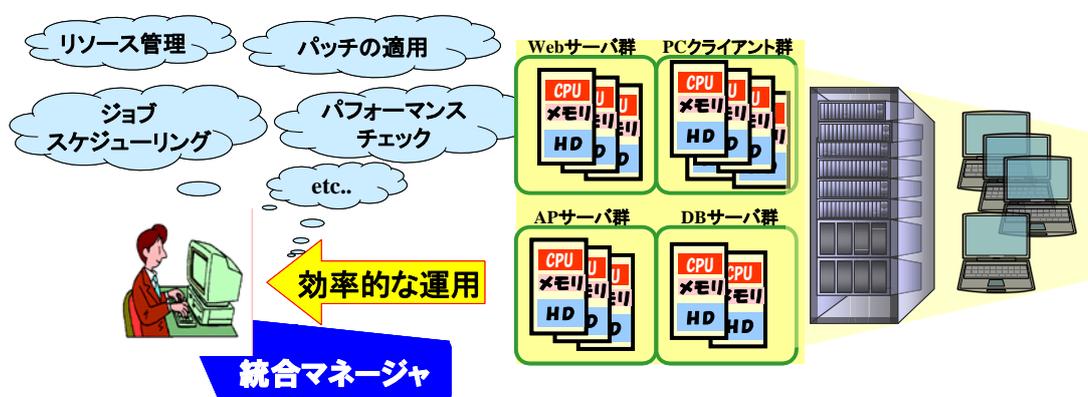


図 1 本ソフトウェアの利用イメージ

2.2 本文中での記載用語

以降の本文中では、以下の呼称を用いることがあります。

ここで、Hinemos Ver.2.1 エージェント (UNIX) は UNIX 系 OS および Linux の Red Hat EL4 以外のディストリビューションを指すものとします。

Hinemos Ver2.1 エージェント	→ 「Hinemos エージェント」
Hinemos Ver.2.1 エージェント (UNIX)	→ 「UNIX 版エージェント」
Hinemos Ver2.1 エージェント (Red Hat EL4)	→ 「Red Hat 版エージェント」
Hinemos Ver2.1 マネージャ	→ 「Hinemos マネージャ」
Hinemos Ver2.1 クライアント	→ 「Hinemos クライアント」

2.3 Hinemos エージェントについて

Hinemos では、その機能によりマネージャ・クライアント・エージェントの3つのパッケージを提供しています。

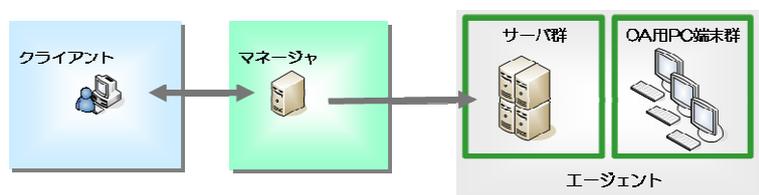


図 2 パッケージの役割

Hinemos エージェントは、管理対象のサーバを Hinemos で管理するためのソフトウェアのセットであり、複数のソフトウェアから構成されています。

本文書で説明する方法により、UNIX でセットアップすることによって、UNIX サーバを Hinemos の管理対象とすることができます。

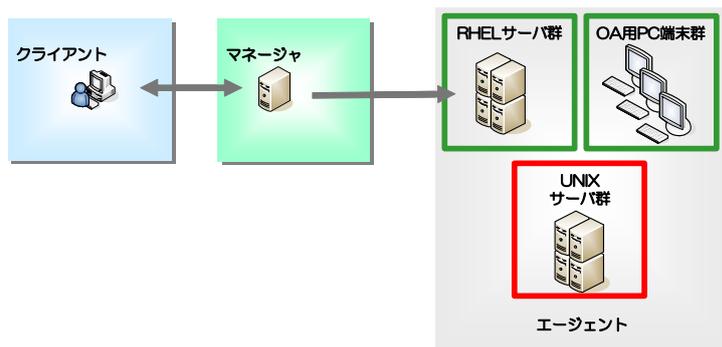


図 3 UNIX サーバの管理

3 Hinemos エージェントの構成

Red Hat 版エージェントでは、複数のソフトウェアをパッケージ内に含んでおり、また Linux に標準的にインストールされているコマンドを利用しています。そこで各 UNIX でソフトウェア、コマンドを利用可能とすることができれば、基本的にどの UNIX でも動作することになります。

3.1 利用しているソフトウェア

Hinemos では、Linux 上での下記のソフトウェアを利用していますので、対象環境でソフトウェアが準備できていることを確認します。

表 1 Red Hat 版エージェントで利用しているソフトウェア

機能		Linux版における方式			備考
機能	サブ機能	プロトコル	ソフトウェア	補助コマンド	
性能管理	リアルタイム	snmp	Net-SNMP		Disk I/Oをオンにしてビルドされている必要あり
	実績収集	snmp	Net-SNMP		
ジョブ管理機能	コマンド実行	RMI	ジョブエージェント JRE	su -c	su -cでの動作がlinuxと同等であるか？
	ファイル転送	RMI	ジョブエージェント JRE	ssh	
一括制御	コマンド実行	rsh ssh	rsh ssh	ftpクライアント expect	
監視管理	エージェント監視	RMI	hinemos エージェント		
	SQL監視	SQLコネクション	postgreSQL		
	PING監視	PING/TCP echo	iputils/xinetd		
	SNMP監視	snmp	Net-SNMP		
	syslog-ng監視	syslog-ng(TCP)	syslog-ng		linuxにも標準では含まれていない。
	リソース監視	snmp	Net-SNMP		
	アプリケーションログ監視	syslog-ng(TCP)	エージェント JRE		
	HTTP監視	HTTP	apache		
プロセス監視	snmp	Net-SNMP			

3.2 エージェントパッケージの内容

Red Hat 版エージェントのパッケージは、図 4 の構成となっています。

UNIX 系 OS 向け Hinemos エージェント
作成メモ

```
Hinemos_Agent-2.1.0_rhel4/  
├── LICENSE  
├── README  
├── README.ja  
├── agent_installer_EN.sh*  
├── agent_installer_JP.sh*  
├── hinemos_agent.cfg  
├── install.cfg  
├── jre-1.5.0_09-linux-i586.bin*  
├── hinemos_agent/  
│   ├── LICENSE  
│   ├── _version  
│   └── bin/  
│       ├── agent_start.sh*  
│       ├── agent_stop.sh*  
│       ├── hinemos_agent*  
│       ├── hinemos_agent.cfg  
│       ├── hinemos_log_agent*  
│       ├── log_agent_start.sh*  
│       └── log_agent_stop.sh*  
├── lib/  
│   ├── JobManagementEJB.jar  
│   ├── LogTransferEJB.jar  
│   ├── MonitorEJB.jar  
│   ├── RepositoryEJB.jar  
│   ├── clustercontrol.jar  
│   ├── commons-logging.jar  
│   ├── jbossall-client.jar  
│   ├── log4j.jar  
│   └── agent/  
│       ├── Agent.properties  
│       ├── agent.jar  
│       └── log4j.properties  
├── log_agent/  
│   ├── Agent.properties  
│   ├── log4j.properties  
│   └── logagent.jar  
├── var/  
│   ├── log/  
│   └── run/  
├── rpms/  
│   ├── libol-0.3.17-1.i386.rpm  
│   ├── net-snmp-5.1.2-11.EL4.7.i386.rpm  
│   ├── net-snmp-devel-5.1.2-11.EL4.7.i386.rpm  
│   ├── net-snmp-libs-5.1.2-11.EL4.7.i386.rpm  
│   ├── net-snmp-perl-5.1.2-11.EL4.7.i386.rpm  
│   ├── net-snmp-utils-5.1.2-11.EL4.7.i386.rpm  
│   └── syslog-ng-1.6.9-1.i386.rpm  
└── service/  
    ├── hinemos_agent*  
    └── hinemos_log_agent*
```

*:実行可能ファイル

図 4 Red Hat 版エージェントパッケージの構成

ここで、赤字のファイルは UNIX 版エージェントを作成する際に削除または置き換える必要があるファイル、青字のファイルは修正が必要となるファイルです。

4 UNIX 版エージェントの作成

Solaris や HP-UX など多数の UNIX 系 OS がありますが、ここでは Debian 版エージェントパッケージの作成を通して、UNIX 系 OS でエージェントを動作させるため方法を説明します。他の UNIX 系 OS も本作業と類似した作業を実施することでエージェントが動作すると考えられます。

4.1 UNIX 版エージェント作成のための作業

Hinemos エージェントの UNIX 版を作成するためには、以下の作業が必要となります。

- UNIX で標準的に準備されているコマンド・サービスの確認
- Hinemos エージェントに含まれているネイティブソフトウェアの確認
- JRE の確認
- インストールスクリプト・スタートアップスクリプトの修正

4.2 UNIX で標準的に準備されているコマンド・サービスの確認

UNIX に標準的に提供されているコマンドとその確認方法は以下の通りです。併せて、Red Hat 版エージェントの動作確認を行った Red Hat EL4 update4 でのバージョンを示します。

表 2 UNIX で確認の必要なコマンド・サービス

コマンド (引数含む)	確認方法	Red Hat EL4 update 4 バージョン
su -c	“su -c sleep 30 ユーザ名”を実行 “ユーザ”のプロセスとして sleep が実行されること。	5.2.1
ssh	マネージャでの hinemos の公開鍵をノードに置き、マネージャから hinemos でログインができること。	3.9p1
rsh	マネージャから hinemos でのアクセスを受け付ける様に設定し、マネージャからアクセスする。	0.17
ftp クライアント	ファイルのコピーなどを行うサーバにアクセスする。 /usr/kerberos/bin/ftp を利用して、アクセス可能であること。	1.3.4 (Krb5-Workstation)
TCP echo(xinetd)	サービスは起動していなくてもよい。 ファイアウォールなどで DROP しないこと。	2.3.13 (xinetd)

	確認は hinemos から行うこととする。	
expect	test_expect.sh ² を実行して正しく終了すること。	5.42

Debian Sarge では、ftp クライアントが /usr/kerberos/bin/ftp ではなかったため、
/usr/sbin/ftp のシンボリックリンクを /usr/kerberos/bin/ftp に作成することとします。

4.3 Hinemos エージェントに含まれているネイティブソフトウェアの確認

● Net-SNMP

Hinemos では、Net-SNMP の UCD-MIB を利用しているため、性能管理機能とリソース監視を利用するために、Net-SNMP をインストールする必要があります。
Net-SNMP は SourceForge の Net-SNMP サイト³からダウンロードすることができます。

Debian Sarge では、パッケージとして snmpd が用意されているので、これを利用することとします。但し、このパッケージでは DiskI/O は取得することはできません。
DiskI/O を取得できるようにするためには、DiskI/O をオンにした状態でコンパイル・再パッケージングを行う必要があります。

マネージャから接続可能であることを確認するためにマネージャから以下のコマンドを実行します。

```
#snmpwalk -v 1 -c public hostname .1.3.6.1.4.1.2021
```

このコマンドにより、以下のような出力結果が得られれば性能管理が可能となります。

```
UCD-SNMP-MIB::memIndex.0 = INTEGER: 0
UCD-SNMP-MIB::memErrorName.0 = STRING: swap
UCD-SNMP-MIB::memTotalSwap.0 = INTEGER: 755012
UCD-SNMP-MIB::memAvailSwap.0 = INTEGER: 754828
UCD-SNMP-MIB::memTotalReal.0 = INTEGER: 905404
UCD-SNMP-MIB::memAvailReal.0 = INTEGER: 242656
UCD-SNMP-MIB::memTotalFree.0 = INTEGER: 997484
UCD-SNMP-MIB::memMinimumSwap.0 = INTEGER: 16000
UCD-SNMP-MIB::memShared.0 = INTEGER: 0
UCD-SNMP-MIB::memBuffer.0 = INTEGER: 89688
UCD-SNMP-MIB::memCached.0 = INTEGER: 356088
.
.
.
```

² Appendix A を参照

³ <http://net-snmp.sourceforge.net/>

- Syslog-ng

Hinemosでは、システムログ、アプリケーションログをマネージャに転送するために利用します。管理対象ノードでログ監視を行う場合には、Syslog-ngをインストールする必要があります。ソースコードはSyslog-ngのサイト⁴からダウンロード可能です。

Debian Sarge では、パッケージとして Syslog-ng が準備されているので、これを利用することとします。

4.4 JRE の確認

Debian Sarge では、Red Hat 版エージェントに含まれている JRE をそのまま利用可能なので、この作業は省略します。

Solaris の場合には、サン・マイクロシステムズの下記 Web サイト

<http://java.sun.com/j2se/1.5.0/ja/download.html> または

<http://java.sun.com/products/archive/index.html>より

Java Runtime Environment (JRE) 5.0 Update 9 をダウンロードしてください。

その他アーキテクチャ・OS 用の JVM は各ベンダーのサイトより取得してください。

4.5 インストールスクリプト・スタートアップスクリプトの修正

4.5.1 インストールスクリプトの修正

インストールスクリプト(Hinemos_Agent-2.1.0_rhel4/agent_install_EN.sh⁵)は、Red Hat EL 4 向けに作成されているため、以下の部分を修正する必要があります。

- rpm によるインストール部分

Debian Sage では apt-get に書き換え

- chkconfig によるサービスの設定

Debian Sage では update-rc.d に書き換え

apt-get install stalog-ng でインストールした syslog-ng は既にオンとなっているので、

⁴ <http://www.balabit.com/products/syslog-ng/>

⁵ Red Hat版エージェントのインストーラは Red Hat EL4 のロケールである ja_JP.UTF8 を想定しているため、Debian Sageなどのja_JP.EUC-JPの環境においてはENのインストーラを使用すると良いでしょう。

```
update-rc.d -f syslog remove
update-rc.d -f klog remove
```

のみを行うこととします。

- Syslog-ng の設定部分の修正
s_local→s_all, f_mesg → f_messages
gawk → awk

必要に応じて、アンインストール部分も修正します。

4.5.2 スタートアップスクリプトの修正

Hinemos_Agent-2.1.0_rhel4/bin/hinemos_agent.cfg

- HINEMOS_AGENT_LOCK_FILE=/var/lock/subsystem/hinemos_agent
→HINEMOS_AGENT_LOCK_FILE=/var/lock/hinemos_agent
へ変更します。
- HINEMOS_LOG_AGENT_LOCK_FILE=/var/lock/subsystem/hinemos_log_agent
→HINEMOS_LOG_AGENT_LOCK_FILE=/var/lock/hinemos_log_agent
へ変更します。

Hinemos_Agent-2.1.0_rhel4/service/hinemos_agent

- ./etc/rc.d/init.d/functions をコメントアウトします。
- daemon \${HINEMOS_AGENT} → \${HINEMOS_AGENT} へ変更します。
- killproc \$prog
→ PID=`cat /var/run/\$prog.pid`
kill \$PID

へ変更します。

Hinemos_Agent-2.1.0_rhel4/service/hinemos_log_agent

- ./etc/rc.d/init.d/functions のコメントアウト
- daemon \${HINEMOS_LOG_AGENT} → \${HINEMOS_LOG_AGENT} へ変更
- killproc \$prog
→ PID=`cat /var/run/\$prog.pid`
kill \$PID

へ変更します。

5 接続・動作確認

前節までの修正を行ったパッケージを元にインストールを行います。インストール内容の詳細は、Hinemos インストールマニュアルを参照してください。

インストール後に

```
/opt/hinemos_agent/sbin/agent_start.sh
```

```
/opt/hinemos_agent/sbin/log_agent_start.sh
```

を実行してエージェントを起動してください。また、Net-SNMP、Syslog-ng、sshd も利用可能な状態としてください。

その後マネージャから接続・動作確認を行います。マネージャでは事前準備として以下の2点を行っておく必要があります。

- ステータス監視を行う通知設定を定義します。(既存のもの流用も可)
- 上記インストールノードの登録とテスト用スコープの定義

設定方法の詳細については Hinemos ユーザマニュアルを参照してください。併せて、以降の操作全般についても Hinemos ユーザマニュアルと併せて操作させることをお勧めします。

5.1 監視管理機能の動作確認

監視管理の確認を行うことでエージェントの動作の大半を把握することができます。これ以降では、実際に監視設定を行い、結果を確認する作業を行います。

UNIX系OS向けHinemos エージェント 作成メモ

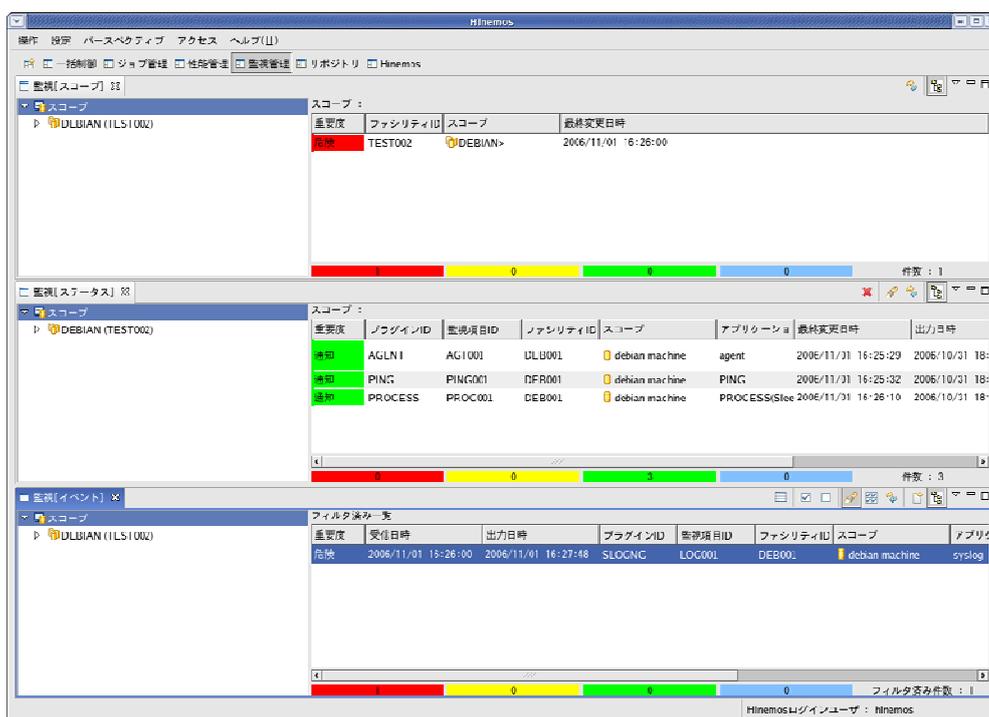


図 5 監視管理パースペクティブ

● Ping 監視

まず、マネージャと管理対象ノードとの疎通を確認するためにping監視の設定を行います。スコープにはあらかじめ設定されているテスト用スコープを用いてping監視が行えるよう設定します。(設定例は図 6を参照してください。)

Ping監視が監視管理パースペクティブ(図 5)で”危険”と表示される場合には、以下の2点をチェックしてください。

- マネージャと管理対象ノードがネットワーク的に到達可能であるか？
- ファイアウォールなどでtcp 7のパケットがドロップしていないか？



図 6 ping 監視設定ダイアログ

UNIX系OS向けHinemosエージェント 作成メモ

● エージェント監視

Hinemosエージェントがマネージャから認識されているかを確認します。Hinemosエージェントの確認を行うためには、エージェント監視を利用します。(設定例は図7を参照。) エージェント監視で以下の設定を行い、監視管理パースペクティブ(図5)で”危険”と表示される場合には、以下の2点をチェックしてください。

- エージェントが正しく起動しているか?
- ファイアウォールなどで通信が遮断されていないか?

The screenshot shows the 'Hinemos Agent Configuration' dialog box. The '監視項目ID' field is set to 'AGT001' and the '説明' field is 'agent'. The 'スコープ' is 'DEBIAN'. Under '条件', the interval is '1分'. The 'チェック設定' section has 'ジョブエージェント' and 'ログ転送エージェント' checked. The '単位' section has 'ノード' selected. The 'ノード単位の重要度' section has '値取得の成功時' set to 'OK: 通知', 'NG: 危険', and '値取得の失敗時' set to '不明'. The '通知' section has '通知ID' set to 'INFO001' and 'アプリケーション' set to 'agent'. At the bottom, there is a table for '実行' with columns for '実行', 'ジョブID', '通知抑制と運動', and '呼出失敗'. The 'この設定を有効にする' checkbox is checked. 'OK' and 'キャンセル' buttons are at the bottom right.

図7 エージェント監視設定ダイアログ

● SNMP・プロセス監視

管理対象ノードのSNMPがマネージャから利用可能かを確認するためにここではプロセス監視を設定します。図 8の設定例では、sleepプロセスの数をカウントしますので、監視設定後に”sleep 120”といったsleepコマンドを複数実行し、監視の状態に変化があるか確認します。



図 8 プロセス監視設定ダイアログ

UNIX系OS向けHinemos エージェント 作成メモ

● ログ監視機能

管理対象ノードのsyslog-ngが正しく動作しているかを確認するためにsyslog-ng監視の設定を行います。図 9の設定では、管理対象ノード上でtestというメッセージがログに出力された場合に、危険というイベントを生成するので、管理対象ノードで”logger test”等のコマンドを実行し、監視管理パースペクティブ(図 5)で”危険”と表示されることを確認してください。必要に応じて、ログ転送の設定も行い管理対象ノードのHinemosエージェントのログ転送機能の確認も行ってください。

The screenshot shows a dialog box for configuring a syslog-ng filter. The title is "syslog-ng [フィルタの作成・変更]". The "監視項目ID" (Monitoring Item ID) is "LOG001", the "説明" (Description) is "log", and the "条件" (Condition) is "log". Under "チェック設定" (Check Settings), the "スコープ" (Scope) is "DEBIAN>" and the "パターンマッチ表現" (Pattern Match Expression) is "*.test.*". The "条件に一致したら処理する" (Process if condition matches) radio button is selected. The "通知ID" (Notification ID) is "INFO001", the "アプリケーション" (Application) is "syslog", the "重要度" (Priority) is "危険" (Danger), the "メッセージID" (Message ID) is "log001", and the "メッセージ" (Message) is "log". The "実行" (Execute) checkbox is unchecked. At the bottom, the "ジョブID" (Job ID) field has a "参照" (Reference) button, the "通知抑制と連動" (Notification suppression and linkage) checkbox is unchecked, and the "呼出失敗時" (On call failure) dropdown is set to "不明" (Unknown). The "この設定を有効にする" (Enable this setting) checkbox is checked. "OK" and "キャンセル" (Cancel) buttons are at the bottom right.

図 9 ログ監視設定ダイアログ

5.2 性能管理機能の動作確認

Net-SNMP の UCD-MIB の情報が取得可能であるか確認します。UCD-MIB の情報を利用する Hinemos の機能としては、性能管理機能のリアルタイム表示、実績収集と監視管理のリソース監視があります。ここでは、性能管理のリアルタイムグラフ表示を実行することで動作を確認します。

今回の Debian Sarge では、Net-SNMP において DiskIO を有効にしていなかったため、DiskIO を設定してもグラフ表示できないことにご注意ください。設定例を図 10 に示します。

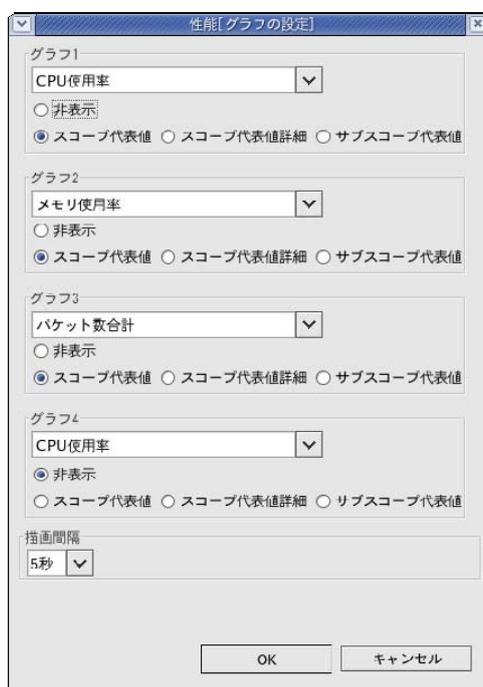


図 10 リアルタイムグラフ設定ダイアログ

リアルタイムグラフビューでグラフが表示されると成功です。

UNIX 系 OS 向け Hinemos エージェント 作成メモ

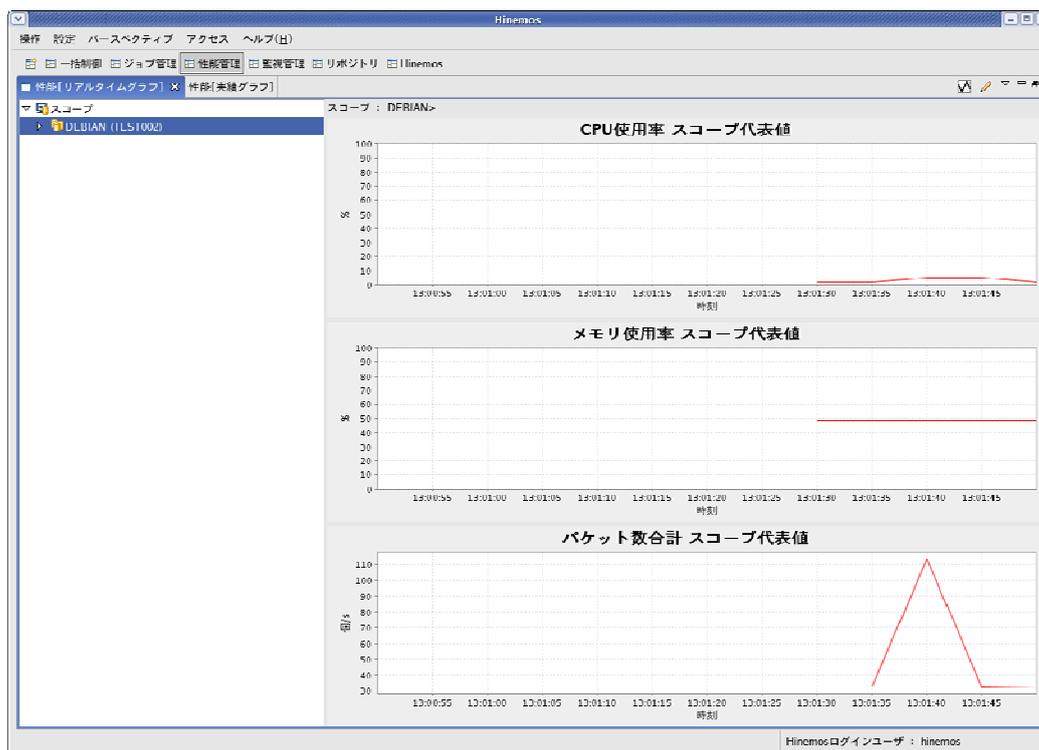


図 11 リアルタイム表示ビュー

5.3 ジョブ管理機能

ジョブエージェントの動作を確認するためには実際にジョブを実行させて、正常に終了するかを確認します。前提として、監視管理機能のエージェント監視において、マネージャ・エージェント間の接続がなされていることが必要となります。

実際に指定するジョブとしては、この例(図 12)では”sleep 30”を指定しています。このジョブを実行したときに、管理対象ノードでsleepが実行されていることを確認し、30秒後に図 13 ジョブ履歴ビューでジョブが終了したことを確認します。また、同時に管理対象ノードにおいてもsleepジョブが終了していることを確認します。

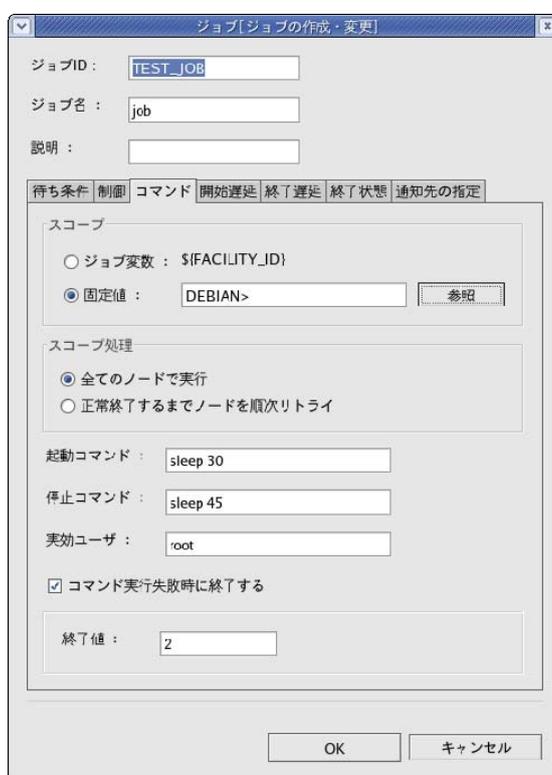


図 12 ジョブ作成ダイアログ

UNIX系OS向けHinemos エージェント 作成メモ

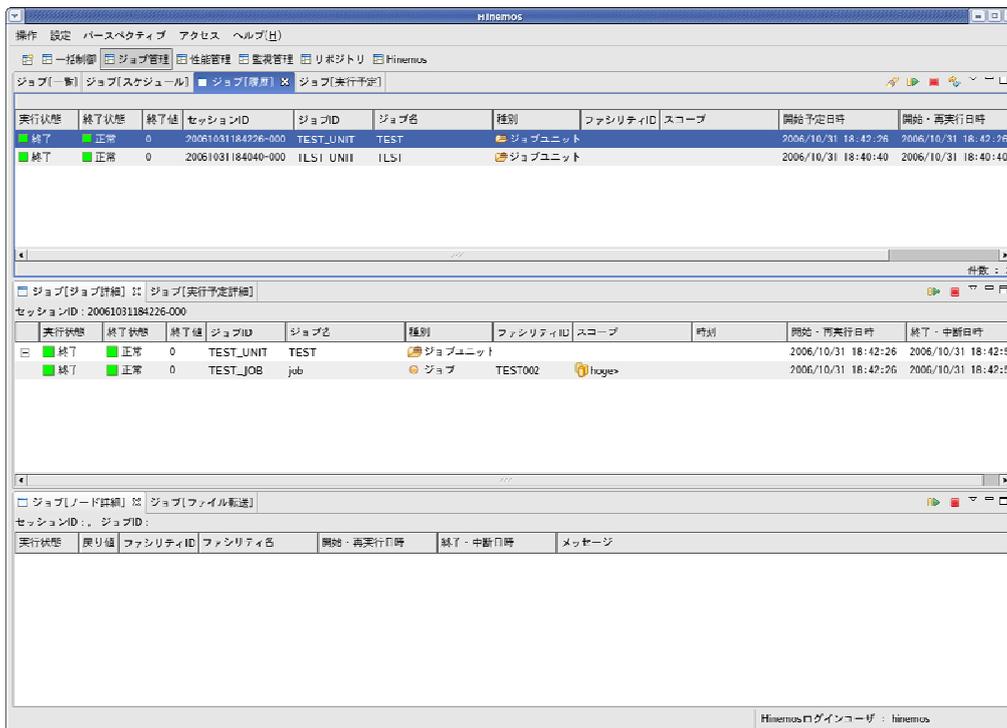


図 13 ジョブ履歴ビュー

5.4 一括制御の動作確認

一括制御は、UNIXのコマンドを多用するため、Hinemos マネージャに準備されているRed Hat用の一括制御スクリプトでは、他のUNIXやLinuxディストリビューションではコマンドの相違により動作しないことが多く発生します。そこで、ここでは比較的コマンド差異の少ないファイルの削除を実行して接続を確認してみます。

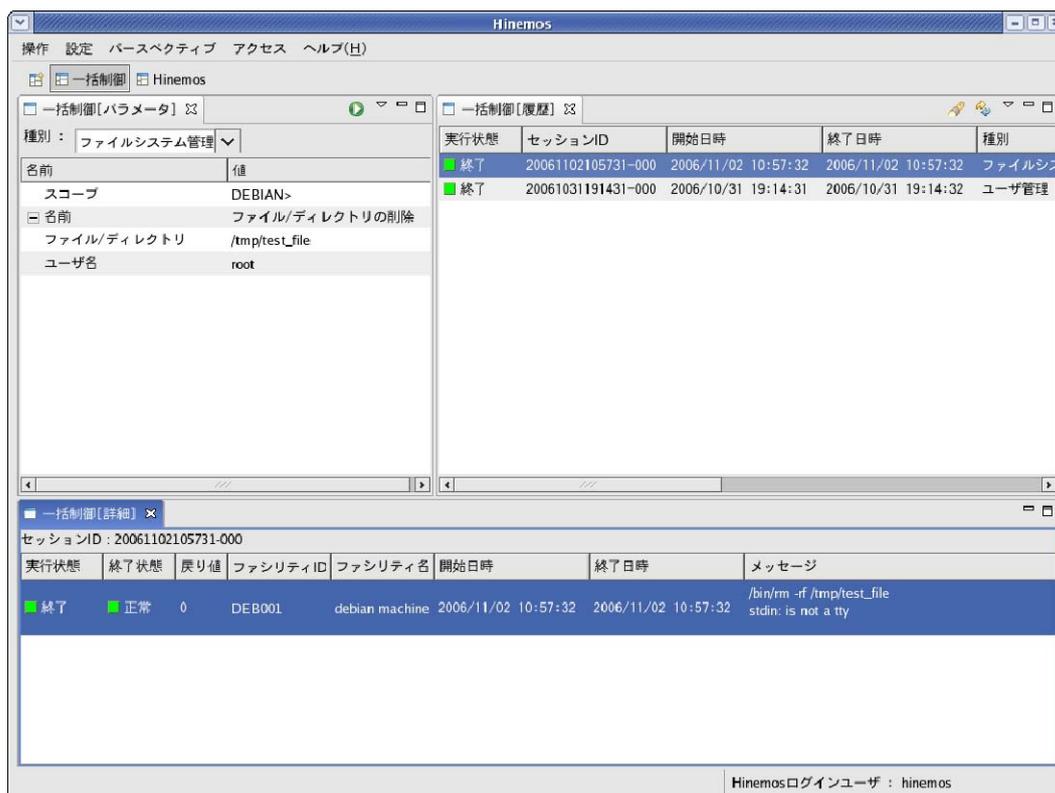


図 14 一括制御パースペクティブ

図 14の例では、まず、あらかじめ管理対象ノード上で手動でファイル/tmp/test_fileを作成します。そして、図 14の設定を行い、実行します。結果の履歴情報が正常終了となっていることを確認し、管理対象ノードの/tmp/test_fileが削除されていることを確認します。UNIXノードに対して積極的に一括制御を利用するためには、該当UNIXで動作するシェルスクリプトを準備し、一括制御の操作を編集する機能（図 15）がありますので登録して使用することとなります。

UNIX系OS向けHinemosエージェント
作成メモ

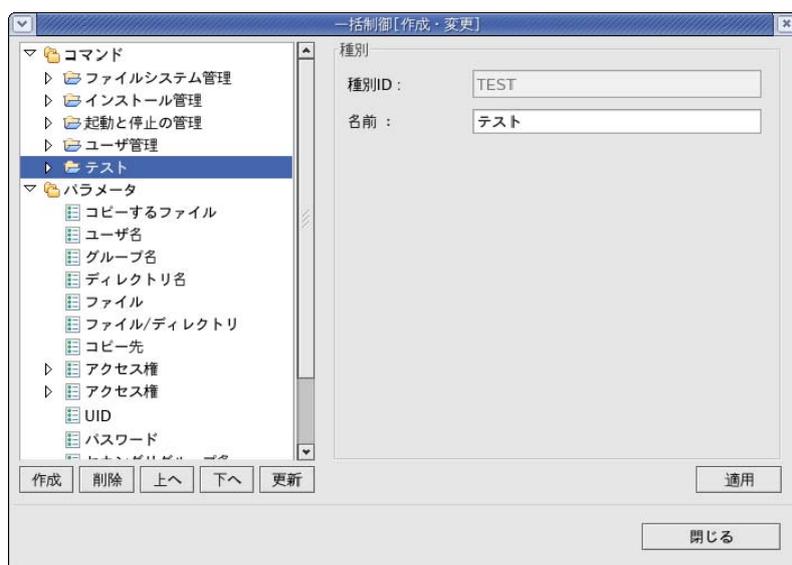


図 15 一括制御操作作成ダイアログ

6 補足情報

6.1 動作しない場合、補足の情報が必要な場合

上記の行った際に問題が発生した場合には、以下から情報を収集してみてください。

SourceForge の Hinemos サイト

<http://sourceforge.jp/projects/hinemos/>

SourceForge の Hinemos メーリングリスト

hinemos-mailing-list@lists.sourceforge.jp

Doblog Hinemos 開発日記 (公式情報ではありません)

<http://www.doblog.com/weblog/myblog/59074>

NTT データの問い合わせ窓口 (できれば、技術の問い合わせは ML へ)

osdquery@nttdata.co.jp

6.2 正式動作確認環境

上記の確認作業で Hinemos エージェントとしておおむね動作させることが可能であることを確認できました。

しかし Red Hat 版エージェントは上記の動作確認に加えてリリースのための試験を実施しています。UNIX 版エージェントを作成し、同等の試験を実施する必要がある場合には、NTT データの問い合わせ窓口までお問い合わせください。

Appendix A Expect の動作を確認するためのスクリプト

test_expect.sh

```
#!/bin/sh

FTP_PROC="/usr/kerberos/bin/ftp "
FTP_HOST="192.168.1.1" ←Hinemos で利用する FTP サーバを指定する。
FTP_USER="hinemos"
FTP_PASSWD="hinemos"
EXPECT_PROC="expect"

$EXPECT_PROC -c "
    set timeout 300
    spawn $FTP_PROC $FTP_HOST
    expect Name*:
    send $FTP_USER\r
    expect Password:
    send $FTP_PASSWD\r
    expect {
        successful.    {}
        failed.        {exit 1}
        timeout        {exit 1}
    }
    expect >
    send quit\r
    expect eof
```