

GroundWork Monitor Open Source

管理者ガイド

目次

- 第1部 GroundWork Monitor Open Sourceによるこそ
 - 第1章 はじめに

- 第2部 GroundWork Monitor Open Sourceのインストール
 - 第1章 必要条件
 - 第2章 インストールと構成設定への必要条件
 - 第3章 新規インストール
 - 第4章 以前のバージョンからのマイグレーション
 - 第5章 GroundWork Monitor Open Source 動作確認

- 第3部 GroundWork Monitor Open Source の管理
 - 第1章 ユーザとロールの構成設定
 - 第2章 シングル・サイン・オン管理
 - 第3章 アプリケーションの定義
 - 第4章 パッケージの管理

- 第4部 監視テクニック
 - 第1章 Nagios プラグイン

- 第5部 Monarchを使ったGroundWork Monitor Open Source の構成設定
 - 第1章 Monarch 概要
 - 第2章 構成設定プロセスの概要
 - 第3章 Configuration EZ - 始める
 - 第4章 高度な構成設定 - サービスの構成設定
 - 第5章 高度な構成設定 - プロファイル構成設定
 - 第6章 高度な構成設定 - ホストの構成設定
 - 第6b章 高度な構成設定 - ホストの管理
 - 第7章 高度な構成設定 - コンタクトの定義
 - 第8章 高度な構成設定 - エスカレーションの構成設定
 - 第9章 高度な構成設定 - コマンドの定義
 - 第10章 高度な構成設定 - 時間帯の定義
 - 第11章 高度な構成設定 - コントロールオプションを使う
 - 第12章 高度な構成設定 - ツールオプションを使う

- 第6部 GroundWork Monitor Open Source コンポーネントの構成設定
 - 第1章 概要
 - 第2章 ステータスビューアの構成設定
 - 第3章 フィーダの構成設定
 - 第4章 レポートの構成設定
 - 第5章 Bookshelfの構成設定

- 第7部 GroundWork Monitor Open Sourceの維持管理
 - 第1章 GroundWork Monitor Open Source 構成データベース
 - 第2章 ヒストリカルデータベース

- 追補: GroundWork Monitor Open Source の 高度な構成設定の詳細
 - 1. サービスの構成設定

2. プロファイル構成設定
3. ホストの構成設定
4. ホストの管理
5. コンタクトの定義
6. エスカレーションの構成設定
7. コマンドの定義
8. 時間帯の定義
9. コントロールオプションの使用
10. ツールオプションの使用

第1部 GROUNDWORK MONITOR OPEN SOURCEによるこそ

本セクションでは、GroundWork Monitor Open Source 管理者ガイドについての情報を提供します。

第1章 はじめに

[このガイドについて](#)

[関連ドキュメント](#)

[GroundWork Monitor Open Sourceのサポート](#)

[コメントを送る](#)

第1章 はじめに

GroundWork Open Source, Inc.が開発した、GroundWork Monitor Open Sourceをご使用いただきありがとうございます。

GroundWork Monitor Open Source製品は、御社のITインフラストラクチャ全体にわたる集中化した可視性とコントロールを提供するように設計されています。それは、システム、アプリケーション、データベースおよびネットワーク装置を監視します。GroundWork Monitor Open Sourceは、魅力的で、直感的なユーザインタフェース、カスタムコンフィギュレーションとカスタムレポートをサポートする柔軟なアーキテクチャを特徴とします。本システムは、選りすぐったオープンソースと、GroundWorkの専門家によって開発された IT管理のネットワーク 可用性とパフォーマンスを監視して管理するための革新的なソフトウェアを組み合わせた最新のアーキテクチャの上に構築されています。

このガイドについて

GroundWork Monitor Open Source 管理者ガイド 1.0 は、GroundWork Monitor Open Sourceの管理のための詳細が記述されています。本バージョンへの修正と更新は <http://www.groundworkopensource.com> 通して提供されるでしょう。このガイドは、特に、初期導入やシステムへのユーザアクセスを管理する一環として、システムのインストールや構成設定(コンフィギュレーション)、カスタマイズおよび保守管理することが必要な管理者のための情報を提供します。目次の表の中のセクションや左側のナビゲーション枠のリンクを選ぶことで、セクションの先頭や詳しい内容を始めることができます。(HTML版のみ)

関連ドキュメント

アプリケーション運用操作については、*GroundWork Monitor Open Source オペレータ ガイド*を参照してください。*GroundWork Monitor Open Source開発者ガイド*は、開発者のためにGroundWorkのAPI、コンポーネントの構造とGuavaについて説明しています。

GroundWork Monitor Open Sourceのサポート

- GroundWorkのサブスクリプション(年間契約サービス)契約によって、製品サポートが受けられます。
- 稼働時間帯は、米国太平洋標準時間(PST)の午前6:00 AMから午後6:00です。

GroundWorkサービスデスクにコンタクトする

サービスデスク (Webベース)	ここをクリックして、 GroundWork Support Center にログインします。	これは、当社のWebベースのサービスデスクであり、推奨するコンタクト方法です。チケットの各欄を埋めた後、サービスエンジニアがあなたのチケットをレビューし、できるだけ早くあるいは、 2営業時間 以内にあなたにコンタクトします。ログインIDとパスワードはGroundWorkの担当者から提供されます。
電話	サービスデスク 866.899.4342 に電話を掛けてください。(オプション 1)	サービスデスクの担当者は、あなたの問い合わせ番号を記録して、チケットを作成します。チケット作成後、サービスエンジニアが、あなたのチケットをできるだけ早くあるいは、 4営業時間 以内にあなたにコンタクトします。
電子メール	ここをクリックして GroundWork Support にメールを送ります。	メールを使ってチケットの各欄を埋めた後、サービスエンジニアが、あなたのチケットをできるだけ早くあるいは、 4営業時間 以内にあなたにコンタクトします。以下の情報を記述してください： <ul style="list-style-type: none">● あなたの氏名● 会社名● 問題点の説明(詳しく)● 問題の緊急性とその業務への影響度について
緊急時	サービスデスク 866.899.4342 に電話を掛けてください。(オプション 1)	時折、あなたが、あなたの業務上、GroundWorkが本当にすぐにその問題に取り掛かる必要があることがあることを我々は知っています。サービスデスクの担当者あなたのリクエストが緊急であること、それが新しいか既存の件か、を教えてください。既存の件であれば、問い合わせ番号を教えてください。新チケットや既存チケットへの情報入力の後、サービスエンジニアが緊急案件についてできるだけ早くあるいは、 30分 以内(営業時間)にあなたに回答します。

第2部 GroundWork Monitor Open Sourceのインストール

本セクションでは、必須ソフトウェアとハードウェア推奨条件をリストし、GroundWork Monitor Open Source システムのインストールと構成設定の手順を提供します。

Nagios から GroundWork Monitor Open Source へのマイグレーションの手順も、ここで説明します。一度インストールが完了すれば、提供されたガイドラインに従って、GroundWork Monitor Open Source の動作確認を行うことができます。

第1章 必要条件

- サポートするオペレーティングシステム

- 必須ソフトウェア

- 推奨するブラウザ

- 推奨ハードウェア

第2章 インストールと構成設定への必要条件

- Step 1 - SE Linuxコンフィギュレーション

- Step 2 - MySQLコンフィギュレーション

- Step 3 - ネットワークコンフィギュレーション

第3章 新規インストール

第4章 以前のバージョンからのマイグレーション

- Nagios から GroundWork Monitor Open Source 4.5 へのマイグレーション

第5章 GroundWork Monitor Open Source 動作確認

- Nagios 動作確認

- Guava フレームワークの確認

- Nagios と GroundWork のコネクション確認

- ステータスビューア同期の確認

第1章 必要条件

サポートするオペレーティングシステム

GroundWork Monitor Open Source 4.5 は RPM ベースのインストーラにより、すべてのGroundWorkコンポーネントは、商用Linux上に、一緒にインストールすることができます。これらのサポートするオペレーティングシステムは、安定性とパフォーマンスの理由から推奨されます。

SuSE Linux Enterprise Server 9.x

Red Hat Linux Enterprise WS 4

- デフォルトのインストールでよいですが、オフィスやグラフィックツールを削除してもかまいません。
- Red Hat 4 がデフォルトインストールする MySQL 4.1.10a はアンインストールする必要があります。Red Hat のデフォルトの MySQL バージョンはサポートされませんので、アンインストールして MySQL バージョン 5.0.16 以上をインストールしてください。

必須ソフトウェア

- MySQL バージョン 5.0.16 以上
- Linux カーネル バージョン 2.6
- glibc lib バージョン 2.3

推奨するブラウザ

GroundWork は、GroundWork Monitor Open Source を使用する際、下のWebブラウザを推奨します：

- Firefox 1.x
- Internet Explorer 6.x

推奨ハードウェア

シングル構成の GroundWork Monitor Open Source server の最小限のハードウェア構成を下記に示します。100台のホストに対して、デフォルトの監視インターバルで標準サービスプロファイルを使って監視する構成設定されたシステムでは、最初構成で十分でしょう。それを大きく越える場合、より強力なハードウェア(CPU速度と数、RAM)が必要になるでしょう。現時点では、64-bit Linuxカーネルはサポートしていません。

最小ハードウェア構成

- CPU: Intel Pentium 4 プロセッサ(1 GHz)
- メモリ: 1 GB
- 40 GB ハードディスク
- CD-ROM ドライブ

推奨ハードウェア構成

- CPU: Intel Pentium 4 CPU (2 GHz以上)
- メモリ: 4 GB
- 80 GB ハードディスク
- CD-ROM ドライブ

第2章 インストールと構成設定への必要条件

ステップ 1 – SE Linux 構成設定

SE Linux パッケージは、MySQLインストールを妨害するので、無効にする必要があります。MySQLインストールが終われば、再度有効にすることができます。

`/etc/selinux/config` ファイルを編集し、下記のようにします：

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
# enforcing – SELinux security policy is enforced.
# permissive – SELinux prints warnings instead of enforcing.
# disabled – SELinux is fully disabled.
```

`SELINUX=disabled`

```
# SELINUXTYPE= type of policy in use. Possible values are:
# targeted – Only targeted network daemons are protected.
# strict – Full SELinux protection.
#SELINUXTYPE=targeted
```

ステップ 2 – MySQL 構成設定

パッケージのインストール

GroundWork Monitor Open Source は、下記のパッケージ (バージョン 5.0.16 以上)がシステムにインストールされていることを必要とします。正しいバージョンがインストールされているかどうか分からない場合、

`rpm -qa |grep MySQL` コマンドを使って、各パッケージのRPMデータベースに問い合わせして下さい。

- MySQL-server
- MySQL-client
- MySQL-shared-compat

適当な MySQL の RPM ファイルをコピーして、`rpm -Uvh MySQL*` コマンドを実行することで、これらのパッケージをインストールすることができます。これ(このコマンド)で MySQL パッケージを正しい順番でインストールすることができます。MySQL 5 の前提条件として Perl-DBI モジュールがあることに注意してください。もし、あなたがこのモジュールを持っていないければ、GroundWork のダウンロードサイトからダウンロードすることができます。

発生する可能性があるアップグレードエラー

MySQL 4.1 からのアップグレード時、MySQL にアクセスしようとした際に、*Can't connect to local MySQL server through socket /var/lib/mysql/mysql.sock* というエラーが発生するかもしれません。MySQL 5.0 の動作に影響する、古い MySQL のコンフィギュレーションファイルがあるかもしれません。`/etc/my.cnf` ファイルがあるかどうか確認してください。もしあれば、そのファイルを削除するか、リネームしてください。

データベースアクセス

インストーラは、新しいデータベースを生成しパーミッション設定をするため、rootアクセスする必要があります。MySQLのrootパスワードが無い(お勧めしませんが、MySQLのインストール時のデフォルト状態です)場合、それ以上のMySQL構成は不要ですので、次のインストールステップに進めます。rootパスワードがすでに設定されているのであれば、環境変数を設定することでインストーラにそれを渡すことができます。

重要な注意: rootパスワードはインストールでのみ必要です。GroundWork Monitor Open Sourceアプリケーションは、どれも MySQLへのrootアクセスを必要としません。データベース信用証明は、

`/usr/local/groundwork/config/db.properties` に格納されています。

1. パスワードの値を持つ環境変数 `MYSQL_ROOT` (すべて大文字)を作成します
(例: `export MYSQL_ROOT=password`)。
2. 下記のインストラクションを実施します。
3. シェルをExitすると、その変数は見えなくなります。

ステップ 3 - ネットワーク構成設定

MySQL は、`/etc/hosts` ファイルの `localhost` を使用します。`localhost` エントリーが、下記の例ようになっているかを確認してください。また、`localhost` が IP アドレスの直後にあつて、その後に `localhost.localdomain` が続いていることも確認してください。`192.168.2.100` をシステムの IP アドレスに置き換え、`groundworkserver` を実際のホスト名に変更してください。

`127.0.0.1 localhost localhost.localdomain`

`192.168.2.100 groundworkserver`

例: `groundworkserver.groundworkopensource.com`

第3章 新規インストール

下記のインストール手順は、GroundWork Monitor Open Source 4.5でアップデートされました。このインストールパッケージは、すべてのGroundWorkコンポーネントを単一のRPMで導入します。

インストール

インストールを行うため、そのシステムのrootになる必要があります。インストールは、すべてのファイルを `/usr/local/groundwork` の中に置きます。GroundWorkの Apacheバージョンは、このパッケージと一緒にインストールされます。システム上に稼働中の Apache がある場合、それはインストール実行中に停止されますが、削除はされません。提供されていたコンテンツは、GroundWork版の Apacheによって提供されますが、通常のパッチ配布システムを使って適用されるパッチは、稼働中のApacheには適用されなくなりますので注意してください。このような状況を起こさないようにするため、GroundWork Monitor Open Source パッケージは、独立したシステム上にインストールするべきです。

1. パッケージのインストールには、`groundwork-monitor-os-4.5-xx.i586.rpm` ファイルをあなたのシステムにコピーし、下記のコマンドを投入します：

```
rpm -Uvh groundwork-monitor-os-4.5-xx.i586.rpm
```

- 重要な注意: `rpm -Uvh` からのインストールでは、config ディレクトリに デフォルトの `nagios.cfg` を置き (そこに何があっても上書きします)、それをスタートします。このことは、すなわち、`status.log` と `status.sav`、RRD関連、`nagios.log`と `apache` のエラーとアクセスログのファイルがなくなることを意味します。バックアップしておくべきです。それは、アップグレードをする際いつでも行うと良いことです。最低限、下記のものを含むとよいでしょう：

```
/usr/local/groundwork/nagios/etc  
/usr/local/groundwork/nagios/var  
/usr/local/groundwork/nagios/eventhandlers  
/usr/local/groundwork/nagios/libexec  
/usr/local/groundwork/apache2/logs  
/usr/local/groundwork/apache2/cgi-bin  
/usr/local/groundwork/rrd
```

- `groundwork-monitor`を削除した後、`/usr/local/groundwork/backup` というディレクトリが作られ、バックアップの `tar-gzip` ファイルが生成されます：

```
apache2-cgi.bin-backup.tar.gz  
apache2-log-backup.tar.gz  
etc-backup.tar.gz  
monarch-backup.tar.gz  
nagios-etc-backup.tar.gz  
nagios-eventhandlers-backup.tar.gz  
nagios-libexec-backup.tar.gz  
rrd-backup.tar.gz
```

2. パッケージをアンインストールする必要がある場合、`rpm -e groundwork-monitor-osv-4.5-XX.i586` コマンドを投入します。どのバージョンをインストールしているかを確認するためには `rpm -qa | grep groundwork-monitor` コマンドを投入します。

- 重要な注意 : `rpm -e` は、configファイルを含む、すべての GroundWork Monitor の痕跡を消し去ります。あなたがそれら環境を残しておきたい場合、最低限、すべてのRRD関連、config ファイルと log ファイルをバックアップしておくべきでしょう。それは、ソフトウェアインストールの際いつでも行うと良いことです。

3. エラーメッセージ出力があるかを確認する。

4. Nagios のファイルは、下記のディレクトリにインストールされます：

- バイナリ: `/usr/local/groundwork/nagios/bin`
- コンフィギュレーション: `/usr/local/groundwork/nagios/etc`
- イベントハンドラ: `/usr/local/groundwork/nagios/eventhandlers`
- プラグイン: `/usr/local/groundwork/nagios/libexec`
- シェア: `/usr/local/groundwork/nagios/share`
- ログ: `/usr/local/groundwork/nagios/var`
- コマンド: `/usr/local/groundwork/nagios/var/spool`
- CGI: `/usr/local/groundwork/apache2/cgi-bin`

第4章 以前のバージョンからのマイグレーション

本章では、GroundWork Monitor Open Source のさまざまなマイグレーションのオプションについて説明します。

Nagios 1.2 / 2.0 から GroundWork Monitor Open Source 4.5 へのマイグレーション

1. GroundWork Monitor Open Source を新しいサーバにインストールします。
2. 古いNagiosのディレクトリから、プラグインをコピーします：
 - たとえば：`/usr/lib/nagios/plugins` や `/usr/local/nagios/libexec` から新しいNagiosのプラグインディレクトリ `/usr/local/groundwork/nagios/libexec` へ。
 - プラグインは、プラグインディレクトリのためのリソースファイル `resource.cfg` を参照するようにしなければなりません。新システム上で、Monarchの中のリファレンスをチェックするか、そのファイルの中でこれが新しいディレクトリに設定されているかを確認します。変数 `$USER1$` は、`/usr/local/groundwork/nagios/libexec` に設定されている必要があります。
 - どれかのプラグインスクリプトに、古いNagiosディレクトリへのリファレンスが含まれている場合、新しいディレクトリ構造を反映するように変更する必要があります。
3. 古いNagiosディレクトリから、拡張サービス情報のためのカスタムCGIプログラムをコピーします：
 - たとえば：`/usr/lib/nagios/cgi` や `/usr/local/nagios/sbin` から新しい CGI ディレクトリ `/usr/local/groundwork/apache2/cgi-bin/` へ
 - どれかのCGIプログラムに、古いApacheディレクトリ構造へのリファレンスが含まれている場合、新しい構造を反映するように変更する必要があります。たとえば、スタイルシートやイメージのディレクトリはアップデートされる必要があります。
4. イベントハンドラプログラムを新しいサーバにコピーします：
 - パフォーマンスハンドラプログラムをチェックします。
 - このプログラムがカスタマイズされていれば、新しいシステム上のパフォーマンスハンドラが変更されていることを確認します。
 - 新しいパフォーマンスは `/usr/local/groundwork/nagios/eventhandlers/process_service_perf.pl` です。
5. `/etc/nagios` や `/usr/local/nagios/etc` 下にある現在の Nagios コンフィギュレーションファイルを新しいシステムのディレクトリ `/usr/local/groundwork/nagios/etc` にコピーします。
 - Monarch において、`/usr/local/groundwork/nagios/etc` ディレクトリから構成データをインポートして Monarchに搭載します。
 - `nagios.cfg` の設定をチェックし、変数やリファレンスが正しいかを確認します。
 - `cgi.cfg` の設定をチェックし、変数やディレクトリリファレンスが正しいかを確認します。
 - 構成設定(コンフィギュレーション)が正しいのを確認するため、*Pre Flight Test* を実行します。
 - 新しいNagios システム内にコンフィギュレーションを導入するため、*Commit* を実施します。

第5章 GroundWork Monitor Open Source 動作確認

Nagios 動作確認

下記のプロセスは、Nagiosに対する Guava PHPフレームワークの シングル・サインオンのオプションが無効にされている場合にのみ有効です。この機能は、デフォルトで有効になっており、GroundWork Monitor Open Source ログイン以外からのアクセスを抑制します。シングル・サインオンのオプションを無効にしたら、下記の手順でNagiosにアクセスすることができます。

1. NagiosのURL <http://<hostname>/nagios/> をブラウザで開きます。
2. Nagios のページが表示されるのを確認します。
3. Service Detailを選びます。local Hostの監視サービスが表示されるでしょう。

Guava フレームワークの確認

1. GroundWorkサーバの URL <http://<hostname>>をブラウザで開きます。
2. GroundWork Monitor Open Sourceのログインページが表示されるはずですが。
3. デフォルトのオペレータIDとパスワード、joe/joeでログインします。.
4. Status タブを選びます。
5. Overview ステータスのページを見ます。
6. NetView メニューオプションを 選びます。
7. Host and Service 情報を開きます。ステータス情報が、Nagiosの status detail ページと一致するかチェックします。もし、status detailが一致しなければ、次のステップに進みます。
8. Console タブを開きます。 Nagios のコンソールメッセージが表示されることをチェックします。

Nagios と GroundWork のコネクション確認

もし、Status Viewer のステータスと Nagios Service が一致すれば、下記を確認します。

1. 下記のコマンドを実行して `nagios2db_status.pl` が実行されているかを確認します: `ps -ef | grep nagios2db`
2. このプロセスが実行していなければ、下記のコマンドで、gwservicesを再起動します :
`/etc/init.d/gwservices restart`

ステータスビューア同期の確認

もし、最初のインストール後、Status Viewerコンフィギュレーションでホストグループの下のホストやサービスが表示されなかったら、スクリプト `/usr/local/groundwork/feeder/nagios2collage_hostgroupcheck.pl` を実行します。このスクリプトは、Monarch構成ツールデータベース内のコンフィギュレーションとステータスビューアを同期させます。同期は、Monarch の変更が Nagios にコミットされたときにも自動的に実施されます。

第3部 GroundWork Monitor Open Source の管理

本セクションでは、どのようにユーザ、ロールとパッケージを定義して構成設定するかと、GroundWork Monitor Open Sourceの中でどのように使用されるかを説明します。

第1章 ユーザとロールの構成設定

- ユーザ定義

- ロール定義

第2章 シングル・サイン・オン管理

- Nagios SSOのセットアップ

- Nagios に直接アクセスする

第3章 アプリケーションの定義

- アプリケーションの作成

- アプリケーションの追加と削除

第4章 パッケージの管理

- パッケージのインストール

- パッケージのアンインストール

- パッケージの構成設定

第1章 ユーザとロールの構成設定

GroundWork Monitor Open Sourceの管理では、システム管理者、ネットワーク管理者、アプリケーション管理者、NO Cオペレータ、CIOなどの異なるITスタッフの役割によってユーザから監視システムにアクセスできるように構成することができます。ユーザは、GroundWork Monitor Open Sourceの特定のアプリケーションに関連付けられた、ロール(役割)にアサインされます。予め定義されているOperatorのロールでは、ステータスビューア(Status Viewer)とレポート(Reports)およびブックシェルフ(Bookshelf)のアプリケーションへのアクセスが含まれています。Administratorのロールには、管理(Administration)、構成設定(Configuration)およびブックシェルフ(Bookshelf)が含まれています。

ユーザ定義

管理(Administration)のユーザ(Users)オプションで、GroundWork Monitor Open Sourceシステムのユーザを定義することができます。ユーザ名やアカウント状態とパスワードなどのユーザ属性は、ここで管理されます。一度、ユーザ(たとえば、matt)を作成されたら、それにロール(例、superuser)を追加します。デフォルトではユーザログインは、3回試行できます。パスワードの使用期限が1にセットされると、ログインには、自動的にパスワードの使用期限が設定されます。

新ユーザの追加

1. GroundWork Monitor Open Source に管理者(Administrator)としてログインします。
2. **Administration** タブを選びます。
3. **User Accounts In System** 画面で、**Add A New User** を選びます。
4. **Create A New User** 画面で、プロパティ(詳しくは、表 3.1.1.参照)を入力します。
5. 新しいユーザをセーブするため、**Finish** を選びます。そのユーザは、左枠のナビゲーションツリーの **Users** リストに追加されます。再度ユーザの属性を表示するには、ユーザ名を選びます。

図 3.1.1. 新ユーザの追加

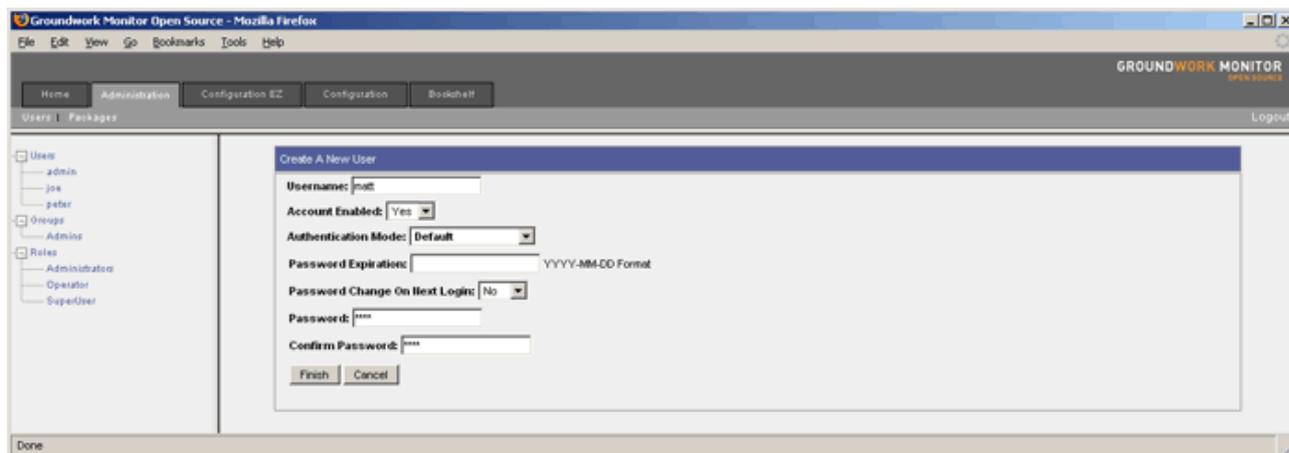


表 3.1.1. 新ユーザの追加

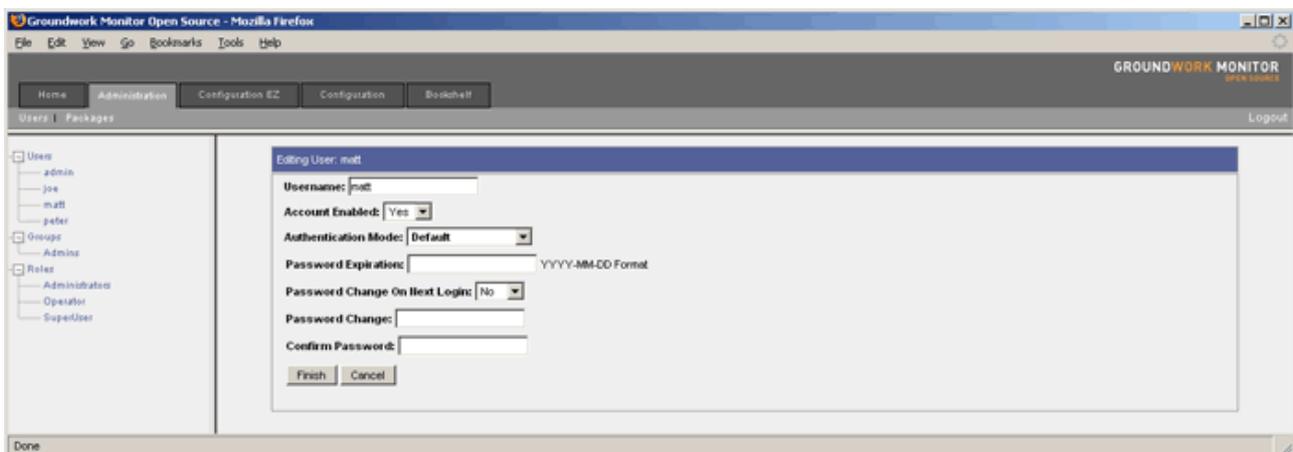
Username (ユーザ名)	GroundWork Monitor Open Sourceにアクセスするユーザの名前
Account Enabled (アクセス可能)	Yes は、アカウントをアクティブにします。 No は作成したユーザを、非アクティブ状態に保持します。
Authentication Mode (認証モード)	Default (デフォルト)の認証モードは、内部データベースによって提供されます。パスワードの変更、使用期限とアカウントの有効/無効を含むユーザ管理は、すべてそのフレームワークで行われます。 LDAPAuthModule (LDAP認証モジュール)認証モードは、Active Directory や e Directory、OpenLDAPなどの外部ディレクトリソースによって提供されます。ディレクトリサービスは、モニタ基本パッケージのコンフィギュレーションの中で指定されます。
Password Expiration (パスワード使用期限)	ユーザのパスワード期限切れの日付。入力がない場合は、期限切れなし。
Password Change On Next Login (次回ログイン時パスワード変更)	Yes は、管理上アサインされたパスワードをユーザが次回ログインしたときに変更するようにできます。 No はアサインされたパスワードが将来のログイン時にも保持されます。
Password (パスワード)	ユーザのパスワード
Confirm Password (パスワード確認)	上記で入力したパスワードの確認。

ユーザ・アトリビュートの編集と削除

ユーザ情報は、編集できます。(たとえば、新しいパスワードをセットしたり、アカウントを無効にしたり)

1. GroundWork Monitor Open Source に Administratorとしてログインします。
2. **Administration** タブを選びます。
3. 定義済みのユーザを左枠のナビゲーションツリーから選びます。ユーザのアトリビュートが表示されます。
4. ここでユーザのアトリビュートを変更するために **Edit** を選ぶか、システムからユーザ名を削除するため **Delete** を選びます。

図 3.1.2. ユーザの編集



ロールの定義

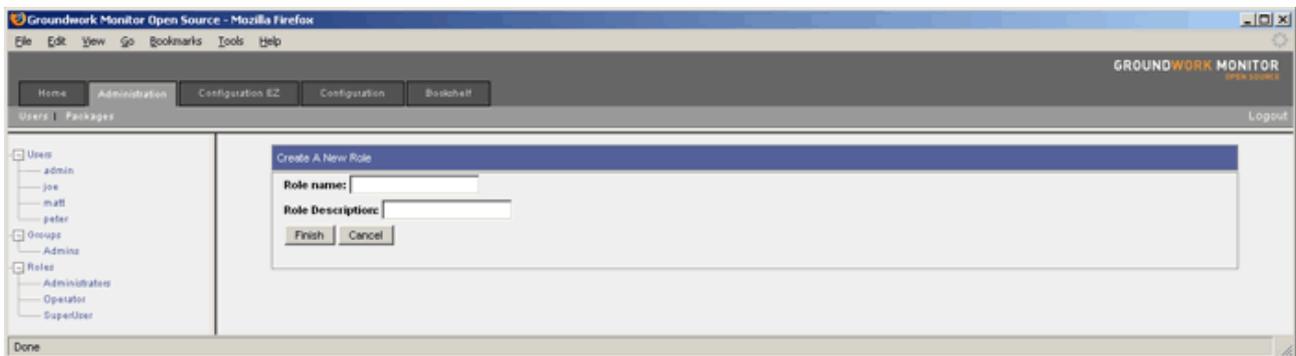
Administration の Roles オプションは、特定のアプリケーションを特定の GroundWork Monitor Open Source のユーザが見ることのできるようにします。予め定義されているロール(Role)：AdministratorとOperatorに加えて、新しいロールを作成することは、PHP フレームワークの管理カスタマイジング機能の一部です。

新ロールの追加

ここで、新しいロールに名前と説明をアサインします。下記の図 3.1.4. では、ロール名 (Role name)として *SuperUser* (説明 (Description)として Show everything)が追加し、表示しています。

1. GroundWork Monitor Open Sourceに Administrator としてログインします。
2. **Administration** タブを選びます。
3. 左側のナビゲーションツリーの中から **Roles** を選びます。
4. メインウインドウの中から、**Add A New Role** を選びます。
5. **Create A New Role** 画面で、アトリビュート **Role Name** と **Role Description** を入力します。
6. **Finish** を選びます。新しいロールが、左側のナビゲーションツリーの中の Roles のリストの中に表示されます。

図 3.1.4. 新ロールの追加

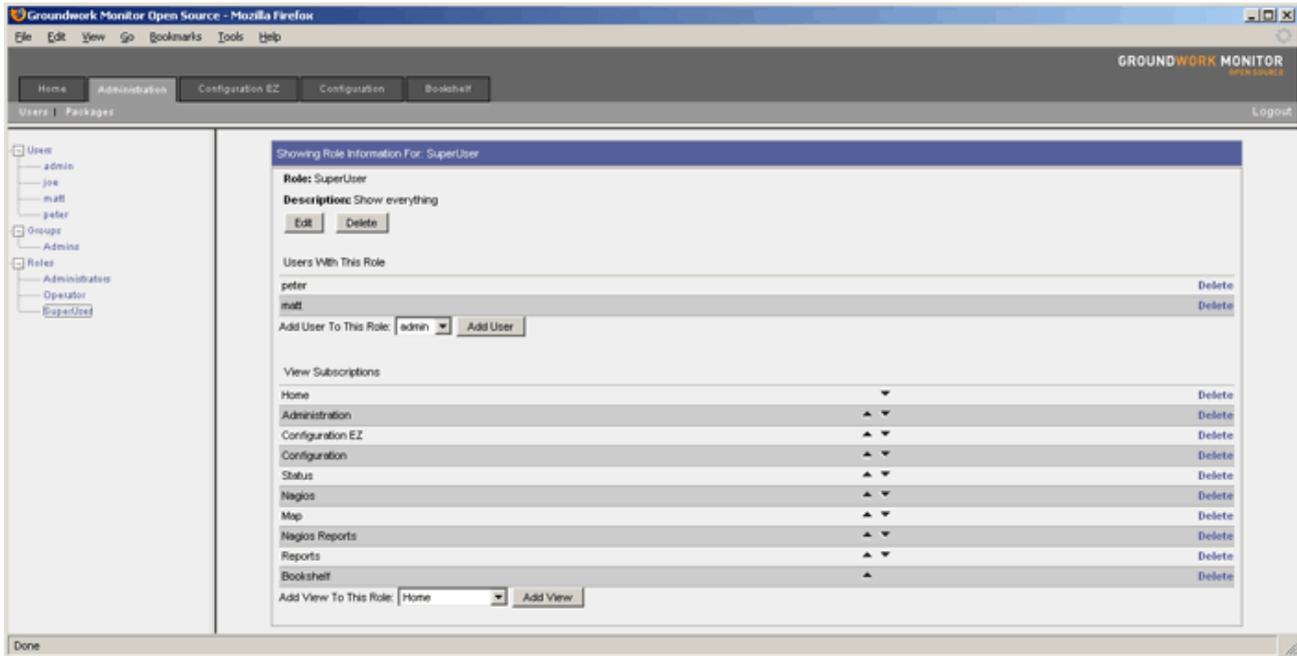


ユーザをロールに追加する

このオプションを使って、ユーザを確立されたロールに割り当てます。これらのユーザは、指定されたロールとそれに設定されたアプリケーションへのアクセスを取得します。下の図 3.1.5 では、**SuperUser** ロールにアサインされた、ユーザ **Matt** が表示されています。

1. Administratorとして GroundWork Monitor Open Source にログインします。
2. **Administration** タブを選びます。
3. 左側のナビゲーションツリーの **Roles** の下のリストからユーザを追加するロールを選択します。
4. **Add User To This Role** のドロップダウンボックスからユーザを選び、**Add User** ボタンを選びます。そのすぐ上にある、**Users with this Role** のリストにユーザが追加されます。アプリケーションをロールに追加するには、下記のステップに続きます。

図 3.1.5. ユーザをロールに追加する



ビュー(アプリケーション)をロールに追加する

Add View to this Role は、GroundWork Monitor Open Source 内の特定のビュー(アプリケーション)をロール内のユーザーにアクセスできるようにします。下の図 3.1.6. は、ユーザ *Matt* と *SuperUser* ロールの例です。そのロールにアサインされているアプリケーションには、Home、Status、Reports、Configuration、AdministrationとBookshelfがあります。

1. GroundWork Monitor Open Source に Administratorとしてログインする。
2. **Administration** タブを選びます。
3. 左側のナビゲーションツリーの**Roles** の下のリストからビューを追加するロールを選択します。
4. **Showing Role Information For:** 画面の中の**View Subscriptions** の下に、現在のこのユーザに設定されているビューがリストされています。
5. **Add View to this Role** のドロップダウンボックスからビューを選び、**Add View** ボタンを選択します。そのビューが **View Subscriptions** のリストに追加されます。このロールに追加したいアプリケーションを続けて、追加します。
6. ここで、矢印を使って、ビューの順番を変えることができます。これによりアプリケーションのタブの順番がセットできます。ビューを削除するには、削除したいビューの行の **Delete** を選択します。また、この画面において、ロールの編集や削除のために、画面の上部の **Edit** や **Delete** ボタンを選択できます。

図 3.1.6. ビューをロールに追加する

Groundwork Monitor Open Source - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

GROUNDWORK MONITOR

Home Administration Configuration EZ Configuration Bookshelf

Users Packages Logout

Users

- admin
- joe
- matt
- peter

Groups

- Admins

Roles

- Administration
- Operator
- SuperUser

Showing Role Information For: SuperUser

Role: SuperUser

Descriptions: Show everything

[Edit](#) [Delete](#)

Users With This Role

peter	Delete
matt	Delete

Add User To This Role: [Add User](#)

View Subscriptions

Home	Delete
Administration	Delete
Configuration EZ	Delete
Configuration	Delete
Status	Delete
Nagios	Delete
Map	Delete
Nagios Reports	Delete
Reports	Delete
Bookshelf	Delete

Add View To This Role: [Add View](#)

Done

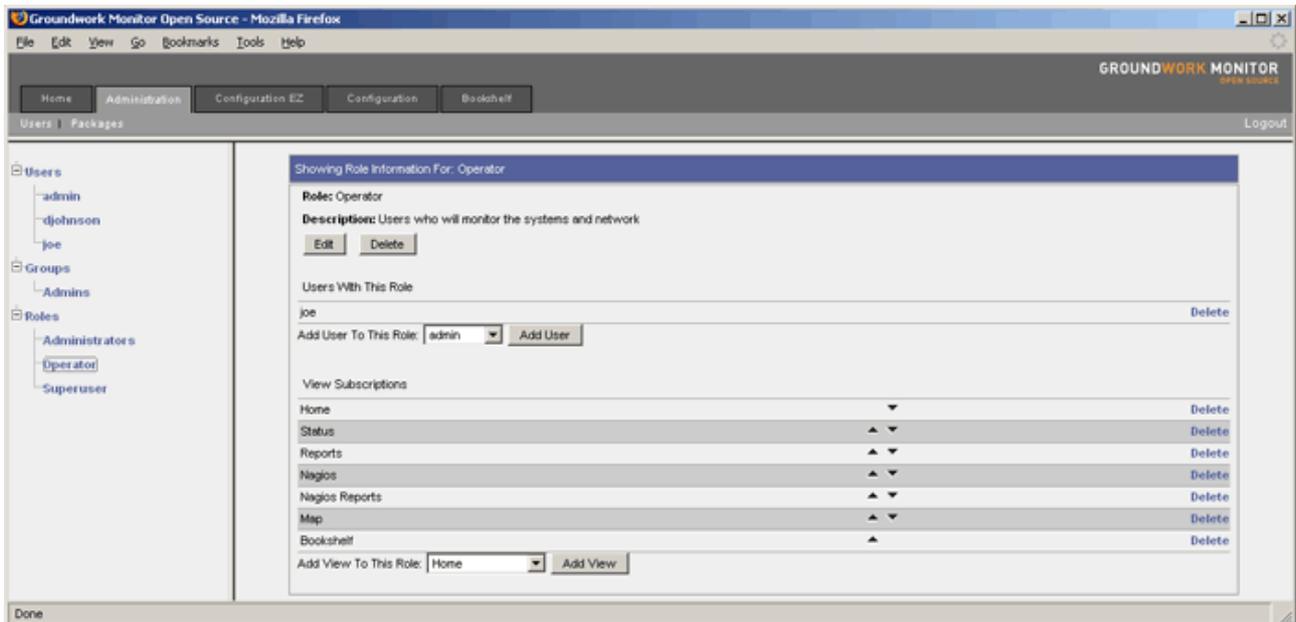
第2章 シングル・サイン・オン(Single Sign On)管理

本章では、どのようにセットアップして、GroundWork Monitor Open Source フレームワークの中から Nagios などの他のアプリケーションにアクセスするかを概説します。Operator としてログインした時にユーザが Nagios (Status)、Nagios Reports と Nagios Map を見ることができるように、Operator ロールをセットアップします。このAdministration ページで、どんなWebページへのアクセスでも、タブと制御されたアクセスを追加することができます。注意: ロールの最初のアプリケーションとして、常にHomeを最初にするのは、よいやり方です。

GroundWork Monitor Open Source内で Nagios SSOをセットアップする

1. Administratorとしてログインする。
2. **Administration** タブを選びます。
3. 左側枠のナビゲーションツリーの**Roles** の下のリストから既存の **Operator** ロールを選択します。
4. 右側枠の **View Subscriptions** の下の、**Add View To This Role** ドロップダウンボックスからNagiosを選び、**Add View** ボタンを選択します。この手順を、Nagios Reportsとd Map に繰り返します。
5. 現在のセッションをログアウトするために、**Home** タブを選びます。Operator として、再度ログインすると、Nagios のページにアクセスできる、3つのNagios のタブが追加されているでしょう。

図 3.2.1. Nagios タブを追加する



Nagios に直接アクセスする

もし、Nagios へ直接アクセスしたいならば、Apacheの `httpd.conf` ファイル:

`/usr/local/groundwork/apache2/conf/httpd.conf` 中の SSO (Single Sign On) を無効にすることができます。SSOの行をコメントにするか削除し、`ScriptAlias /nagios/cgi-bin` と `Alias /nagios` のための標準の行のコメントを外します。どのようにするか、下記のテキスト行を参照してください。

```
ScriptAlias /nagios/cgi-bin /usr/local/groundwork/apache2/cgi-bin/nagios
Directory /usr/local/groundwork/apache2/cgi-bin/nagios
# Uncomment for Guava Single Sign On
#AuthType Basic
#require valid-user
# The following line should be change to specify the default page for invalid access attempts to this directory.
#TKTAuthLoginURL http://localhost/monitor/login.php
#TKTAuthCookieName nagios_auth_tkt
# Uncomment to disable Guava Single Sign On
AllowOverride AuthConfig
Options ExecCGI
Order allow,deny
Allow from all
PassEnv LD_LIBRARY_PATH
Directory Alias /usr/local/groundwork/nagios/share
Directory /usr/local/groundwork/nagios/share
# Uncomment for Guava Single Sign On
#AuthType Basic
#require valid-user
# The following line should be change to specify the default page for invalid access attempts to this directory
#TKTAuthLoginURL http://localhost/monitor/login.php
#TKTAuthCookieName nagios_auth_tkt
# Uncomment to disable Guava Single Sign On
Options None
AllowOverride AuthConfig
Order allow,deny
Allow from all
"/Directory"
```

http を下記のコマンドでリスタートします:

```
/etc/init.d/gwhttpd stop
/etc/init.d/gwhttpd start
```

これで、`"hostname"/nagios` のURLに標準Nagiosのhtaccess IDとパスワード:nagiosadmin/nagiosadmin を使ってアクセスできるようになります。

第3章 アプリケーションの定義

アプリケーションの作成

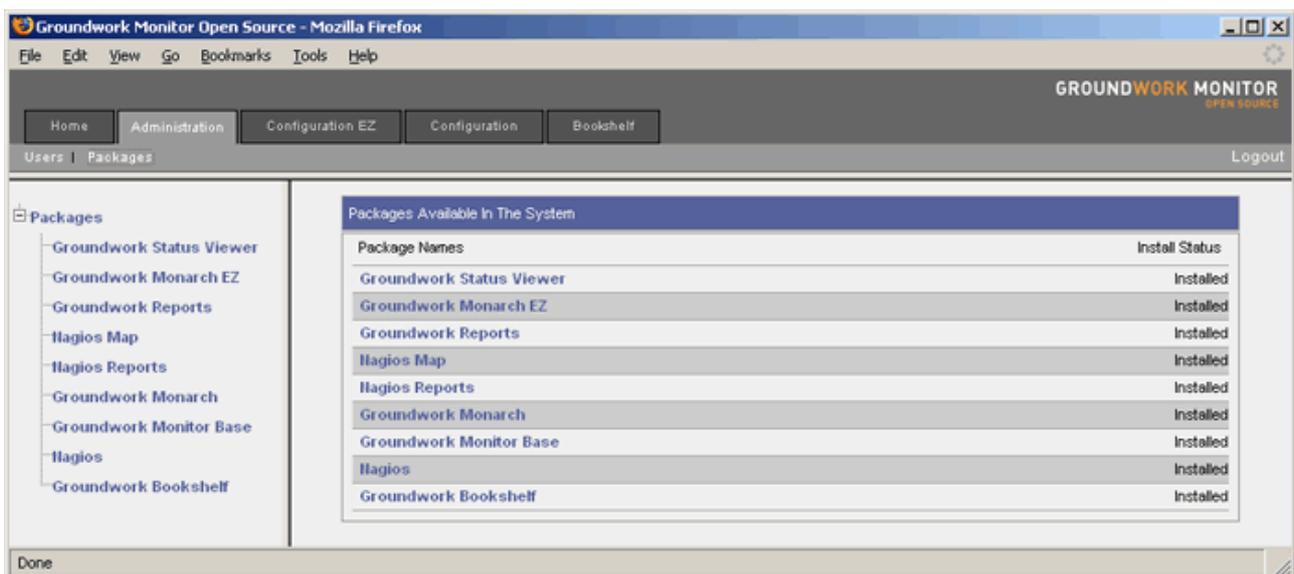
ほとんどすべてのWebアプリケーションがGuavaアプリケーションになり、GroundWork Monitor Open Sourceのユーザインタフェースに組み込みます。これらは一般的なCGIアプリケーションですが、HTTPのURLリンクで参照されるアプリケーションにすることができます。PHPアプリケーションもGuavaの要件に適合するよう修正できるかもしれません。非PHPアプリケーションからGuavaアプリケーションを作成ため、メニュー構造とそのアプリケーションへのリンクを定義するPHPラッパーを作ることができるでしょう。Guavaアプリケーションの作り方の詳細は、*GroundWork Monitor Open Source開発者ガイド*を参照してください。

アプリケーションの追加と削除

アプリケーションの追加

1. Guavaアプリケーションを `/usr/local/groundwork/guava/packages` の下の固有のサブディレクトリにコピーします。 `package.pkg` ファイルを、そのサブディレクトリの中に置きます。
2. これが済めば、Administrationの下でPackagesオプションから、アンインストールされたパッケージとして、そのパッケージが参照できるようになるでしょう。パッケージのインストールには、下記にある手順を実行します。

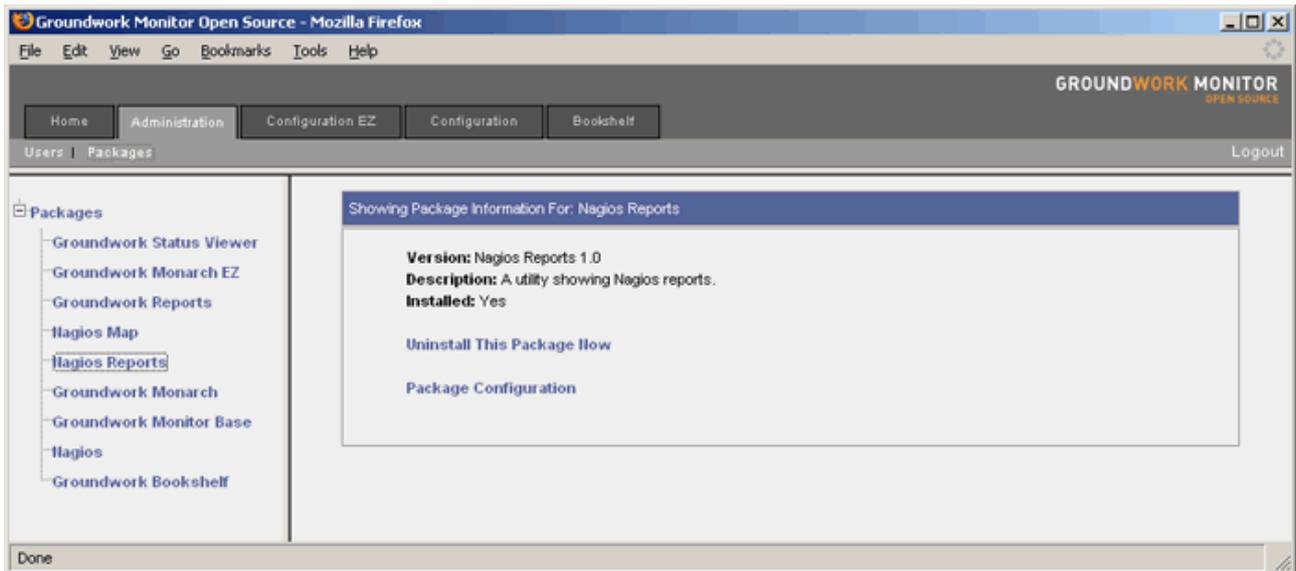
図 3.3.2. システム内で使用可能なパッケージ



アプリケーションの削除

1. GroundWork Monitor Open Source に Administratorとしてログインする。
2. **Administration** タブを選びます。
3. **Packages** オプションを選びます。
4. アンインストールするパッケージ名を選びます。
5. **Uninstall This Package Now** を選びます。
6. ここで、`/usr/local/groundwork/guava/packages` ディレクトリ下のパッケージを含むサブディレクトリを削除します。

図 3.3.3. パッケージ情報



第4章 パッケージの管理

パッケージのインストール

1. GroundWork Monitor Open Source に Administratorとしてログインする。
2. **Administration** タブを選びます。
3. **Packages** オプションを選びます。パッケージのリストがインストール状態と共に表示されます。
4. アンインストールされたパッケージ名を選びます。
5. **Install This Package Now** のリンクを選びます。パッケージのインストール状態が **Installed** に変わります。
6. パッケージに追加の構成設定が必要であれば、**Configure this package** のという文字が表示されます。その文字を選び、そのパッケージの構成設定ページを開きます。セッティングの設定や変更は、指示に従ってください。

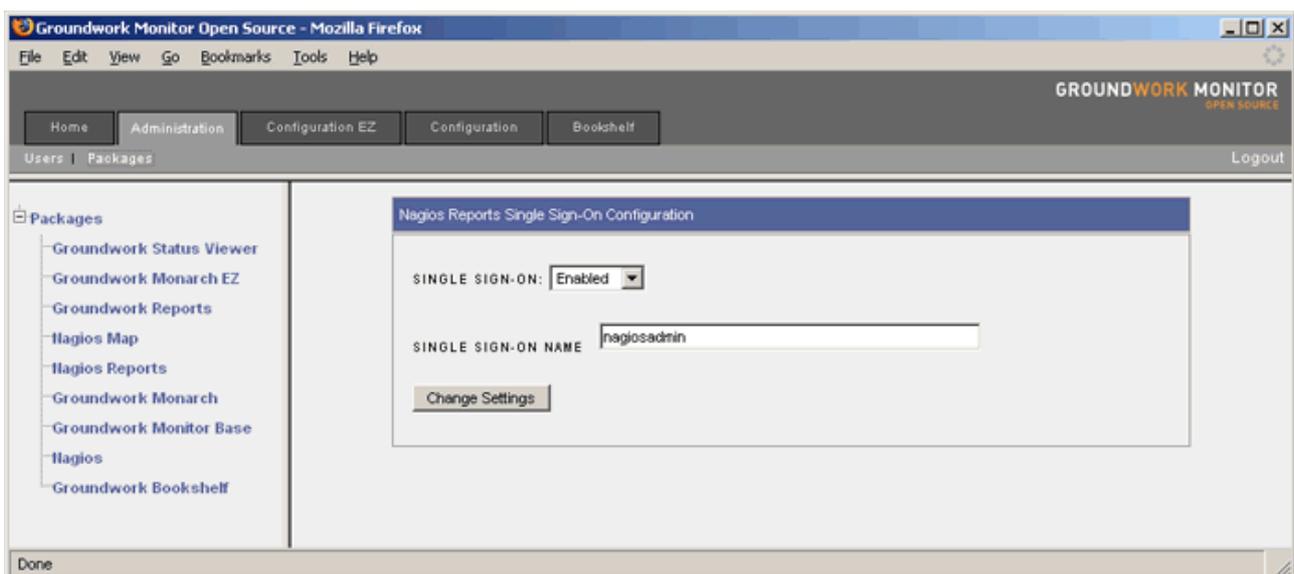
パッケージのアンインストール

1. GroundWork Monitor Open Source に Administratorとしてログインする。
2. **Administration** タブを選びます。
3. **Packages** オプションを選びます。パッケージのリストがインストール状態と共に表示されます。
4. アンインストールするため、インストールされているパッケージの名を選びます。
5. **Uninstall This Package Now** を選びます。パッケージのインストール状態が **Uninstalled** に変わります。

パッケージの構成設定

1. GroundWork Monitor Open Source に Administratorとしてログインする。
2. **Administration** タブを選びます。
3. **Packages** オプションを選びます。パッケージのリストがインストール状態と共に表示されます。
4. 構成設定するため、インストールされているパッケージの名を選びます。
5. パッケージに追加の構成設定が必要であれば、**Configure this package** リンクが表示されます。そのリンクを選び、そのパッケージの構成設定ページを開きます。セッティングの設定や変更は、指示に従ってください。

図 3.3.4. パッケージの構成設定



GroundWork Monitor Open Sourceに含まれるアプリケーション

下記のGroundWork Monitor Open Source に含まれるアプリケーションは、構成設定可能です。しかしながら、GroundWork Monitor Open Sourceのインストールプロセスで、設定をしていますので、再設定は必要ありません。

GroundWork Monitor Base

これは GroundWork Monitorのホームページです。LDAP 認証を導入したら、LDAP パラメータはここで入力します。LDAP を使わない場合、追加の構成設定は不要です。

Directory	ディレクトリサービス サーバのホストアドレスか、ホスト名
Prefix	LDAPでの認証時にユーザ名に付けるプレフィックス。例: <i>uid=</i>
Suffix	LDAPでの認証時にユーザ名に付けるサフィックス。例: <i>ou = people,</i> <i>dc = myorganization, dc = com</i>
Type	現バージョンでは、MySQLのみをサポート。
Address	データベースのホストIPアドレス、ローカルシステムでの通常は 127.0.0.1 。これをリモートのサーバへのアクセスに変更する場合、データベースはこのサーバからアクセスできるように作成する必要があることに注意。
Name	GroundWork Monitor Open Source の sv と GroundWork Monitor Professional <i>GWCollageDB</i> に設定するデータベース名
Username	データベース名にアクセスするための認証されたユーザ名
Password	データベース名にアクセスするための認証されたパスワード

Nagios

ステータスビューアで使用される Nagios パラメータがあります。

Nagios	ステータスビューアで使用される Nagios パラメータがあります。
Comment file	オペレータがNagiosに追加したコメントを格納する場所。デフォルトでは /usr/local/groundwork/nagios/var/comments.dat
Downtime file	Nagiosがホストとサービスの停止時間情報を格納する場所。デフォルトでは /usr/local/groundwork/nagios/var/downtime.dat
Command pipe	外部コマンドを追加するのに使用する、Nagiosコマンドのパイプがある場所。 デフォルトでは、これは /usr/local/groundwork/nagios/var/spool/nagios.cmd

Bookshelf

このアプリケーションは、GroundWork Monitor Open Source 製品のドキュメントとオープンソースのリファレンスへのアクセスを提供します。

Bookshelf データベース構成

Database Address	データベースのホスト IP アドレス。通常は localhost。これをリモートのサーバの指定に変更する場合、データベースはこのサーバからアクセスされるよう作成する必要があることに注意してください。
Database Name	データベース名、GroundWork Monitor Open Source 4.5 では、 <i>guava</i> にセットします。
Downtime file	データベース名にアクセスするために認証されたユーザ名。デフォルトでは <i>guava</i> とセットします。
Password	データベース名にアクセスするために認証されたパスワード。デフォルトでは <i>gwrk</i> と設定。

Mnogosearch 構成

Database Address	データベースのホスト IP アドレス。通常は localhost。これをリモートのサーバの指定に変更する場合、データベースはこのサーバからアクセスされるよう作成する必要があることに注意してください。
Database Name	データベース名、GroundWork Monitor Open Source 4.5 では、 <i>guava</i> にセットします。
Database Username	データベース名にアクセスするために認証されたユーザ名。デフォルトでは <i>mnogosearch</i> と設定。
Password	データベース名にアクセスするために認証されたパスワード。デフォルトでは <i>gwrk</i> と設定。
Indexer Binary Location	デフォルトで <i>/usr/local/groundwork/sbin/indexer</i> とセット。

ステータスビューア

Status Information Store	ステータス情報を格納するためのデータのタイプを指定します。GroundWork Monitor Open Sourceの内部(<i>Internal</i>)で設定されます。
Database	情報格納のためのデータベースパラメータを指定します。
Type	本バージョンでは MySQLのみサポートされます。
Address	データベースのホスト IP アドレス。通常はローカルシステムのための 127.0.0.1。これをリモートのサーバの指定に変更する場合、データベースはこのサーバからアクセスされるよう作成する必要があることに注意してください。
Name	データベース名。GroundWork Monitor Open Sourceのために sv を設定。
Username	データベース名にアクセスするために認証されたユーザ名。
Password	データベース名にアクセスするために認証されたパスワード。
Nagios	ステータスビューアによって使用される Nagios パラメータ
Current File	オペレータがNagiosに追加したコメントを格納する場所。デフォルトでは <code>/usr/local/groundwork/nagios/var/nagioscomment.log</code>
Downtime File	Nagiosがホストとサービスの停止時間情報を格納する場所。デフォルトでは <code>/usr/local/groundwork/nagios/var/nagiosdowntime.log</code> 。
Command Pipe	外部コマンドを追加するのに使用する、Nagiosコマンドのパイプがある場所。デフォルトでは、これは <code>/usr/local/groundwork/nagios/var/spool/nagios.cmd</code>

Nagios

このアプリケーションは、Nagios status ページを表示します。構成設定の内容は：

Single Sign-On	Enabled か Disabled。Enabledであれば、GroundWork Monitor Open Source のアプリケーション外で、NagiosのCGIページからアクセスするWebユーザを抑制します。これはデフォルトでは、Enabled です。
Single Sign-On Name	Nagiosに渡されるユーザ名。デフォルトで <code>nagiosadmin</code> に設定されます。

Nagios Reports

このアプリケーションは Nagios report ページを表示します。構成設定の内容は：

Single Sign-On	Enabled か Disabled。Enabledであれば、GroundWork Monitor Open Source のアプリケーション外で、NagiosのCGIページからアクセスするWebユーザを抑制します。これはデフォルトでは、Enabled です
Single Sign-On Name	Nagiosに渡されるユーザ名。デフォルトで <code>nagiosadmin</code> に設定されます。

Nagios Map

このアプリケーションは Nagios map ページを表示します。構成設定の内容は:

Single Sign-On	Enabled か Disabled。Enabledであれば、GroundWork Monitor Open Source のアプリケーション外で、NagiosのCGIページからアクセスするWebユーザを押し止します。これはデフォルトでは、Enabled です。
Single Sign-On Name	Nagiosに渡されるユーザ名。デフォルトで <i>nagiosadmin</i> に設定されます。

GroundWork Monarch

このアプリケーションは、GroundWork の Monarch 構成設定アプリケーションにアクセスします。

構成設定の内容は:

Single Sign-On	Enabled か Disabled。Enabledであれば、GroundWork Monitor アプリケーション外で、Monarch のCGIページからアクセスするWebユーザを押し止します。これはデフォルトでは、Enabled です。
Single Sign-On Name	Monarchアプリケーションに渡されるユーザ名。デフォルトで <i>monarch</i> に設定されます。

第4部 監視テクニック

本セクションでは、Nagios プラグインの使い方を説明します。

第一章 Nagios プラグイン

第一章 Nagios プラグイン

Nagios プラグインの概要

Nagiosは、サービスとホストの状態をチェックするのを プラグインと呼ぶ外部プログラムに依存しています。サービス定義のコマンドは、指定されたサービスやリソースをチェックし、その結果をイベントハンドラや通知などのアクションを行う Nagios に返すプラグインを含んでいます。GroundWork Monitor Open Source の監視能力を拡張するため、特定の環境のためにプラグインを書くことができます。

GroundWork Monitor Open Source は、インストール時にさまざまなプラグインを含んでいます。このパッケージに含まれているプラグインのリストと説明については、Bookshelf のオープンソースリファレンスでNagios と Nagios プラグインを選んでください。追加のプラグインのリソースについてのリンクもここで見つけることができます。

ディスク領域監視プラグインのサンプル

このプラグインは、マウントされたファイルシステムのディスクの使用パーセントをチェックし、そのパーセンテージがしきい値を超えた場合にアラートを発生させます。このチェックは、サーバ毎の異なるパーティションに適用することもできません。

チェックコマンド名

Check_Disk

コマンド

```
check_disk -w limit -c limit [-p path | -x device] [-t timeout] [-m] [-e] [--verbose]
```

パラメータ

-w, --warning=INTEGER

空きディスク領域が INTEGER キロバイト以下の場合、ワーニング状態で終了。

-w, --warning=PERCENT%

空きディスク領域の割合が PERCENT パーセント以下の場合、ワーニング状態で終了。

-c, --critical=INTEGER

空きディスク領域が INTEGER キロバイト以下の場合、状態で終了。

-c, --critical=PERCENT%

空きディスク領域の割合が PERCENT パーセント以下の場合、クリティカル状態で終了。

-p, --path=PATH

--partition=PARTITION

パスか、パーティション (指定されない場合、マウントされている全パーティション)

-m, --mountpoint

パーティションの代わりにマウントポイントを表示

-x, --exclude_device=PATH

デバイスを無視する (-p が指定されない場合のみ適用)

-e, --errors-only

エラーのあったデバイス/マウントポイントのみ表示

サンプル

Nagios がプラグインから出力される改行を切り取るので、実際にはアウトプットでは改行がなく、一行になります。

```
check_disk -w 10% -c 5%
```

```
DISK CRITICAL 90886592 kB (84%) free on /dev/hda2,
```

```
70700 kB (73%) free on /dev/hda1,
```

```
105252112 kB (96%) free on /dev/hdc1,
```

```
479908 kB (100%) free on /dev/shm
```

第5部 Monarchを使ったGroundWork Monitor Open Source の構成設定

本セクションでは、Monarch を使った GroundWork Monitor Open Source のためのコンフィギュレーション(構成設定)プロセス全体を説明します。GroundWork は、そのGroundWork Monitor Open Source をオープンソースのネットワーク管理アプリケーションの Nagios を基盤にしました。GroundWork Monitor Open Sourceは、実際にコンフィギュレーションファイルを手で編集することなくWebベースのインターフェースを通してできるようにすることで、ITシステムのセットアップと保守をより簡単にすばやくできるようにします。

ビギナーと高度な構成設定オプションも、多数ホストのインポートとや、ホストをインポートして監視プロファイルをアサインするための検出(ディスカバリー)を通して、すばやく効果を提供します。現在 Nagios を使っているユーザのため、Monarch は Nagios の構成ファイルを読み書きし、既存システムとの統合を容易にします。経験を積んだ Nagios ユーザは、Nagios の構成ファイルを編集し、Monarch と交互に使用することもできます。インストレーション後は、GroundWorkの構成設定ユーザインタフェース Monarch を使って、ITインフラストラクチャへの追加や変更、削除をする際、監視システムのメンテナンスを容易にします。本セクションでは、Monarch を使った監視環境の構成設定を詳しく説明します。

第1章 Monarch 概要

Monarch について

ライセンスと入手

第2章 構成設定プロセスの概要

Monarch の構造概論

Monarch コンセプトの構造

Monarch のユーザインタフェース

第3章 Configuration EZ - 始める

ホストを定義する

ホストグループを定義する

プロファイルを構成する

通知を設定する

コミット(Commit)を使う

セットアップ(Setup)を使う

第4章 高度な構成設定 - サービスの構成設定

第5章 高度な構成設定 - プロファイル構成設定

第6章 高度な構成設定 - ホストの構成設定

第6B章 高度な構成設定 - ホストの管理

第7章 高度な構成設定 - コンタクトの定義

第8章 高度な構成設定 - エスカレーションの構成設定

第9章 高度な構成設定 - コマンドの定義

第10章 高度な構成設定 - 時間帯の定義

第11章 高度な構成設定 - コントロールオプションを使う

第12章 高度な構成設定 - ツールオプションを使う

第1章 Monarch 概要

Monarch について

GroundWork Monitor Architect (Monarch)は、Nagios を使うためのフル機能で使い勝手の良いWebベースの構成設定(コンフィギュレーション)システムです。Monarch は、ユーザが Nagios や GroundWork Monitor Open Source を容易に構成設定したり、保守するできるようにする一連のツールから構成されます。インストール後は、Ground Workの構成設定ユーザインタフェース Monarch を使って、ITインフラストラクチャへの追加や変更、削除をする際、監視システムのメンテナンスを容易にします。Monarch は、Nagiosの重要な機能の隙間:監視対象の各アプリケーション、サービス、デバイスなどのため監視システムを構成設定するのが、手動でコマンドラインテキストエディタを行う必要があること、を埋めます。現在 Nagios を使っているユーザのため、Monarch は Nagios の構成ファイルを読み書きし、既存システムとの統合を容易にします。経験を積んだ Nagios ユーザは、Nagios の構成ファイルを編集し、Monarch と交互に使用することもできます。

ライセンスと入手

Monarch は、オープンソースのソフトウェアライセンスの基に、下記のサイトから入手することができます:

- GroundWork Open Source <http://www.groundworkopensource.com/downloads/>
- SourceForge <http://sourceforge.net/projects/monarch/>

製品サポート

- Monarch の質問を送るには、monarch@groundworkopensource.com に電子メールを送ってください。
- Professional 製品サポートは、GroundWorkのサブスクリプション契約(年間契約)で受けることができます。
- Monarch は、GroundWork Open Source, Inc. の Scott Parris のよって開発されました。
- 本文書の範囲外のディレクティブの情報については、<http://www.nagios.org/docs/> の Nagios Documentation で見出すことができます。

第2章 構成設定プロセスの概要

Monarch の構造概論

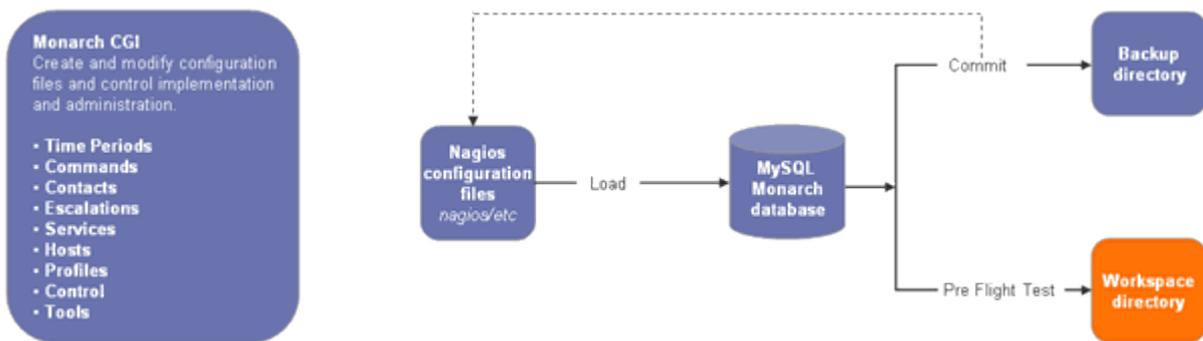
Monarch は Perl で書かれた一連の .cgi スクリプトです。

Monarch を搭載した運用を行うときには、Nagios の構成ファイルはMySQLの Monarch データベースで更新され、データベースの中の全構成がミラーリングされます。その時、Monarch はこのデータベースを操作することで機能します。Nagios の構成情報に加えて、Host と Service プロファイル、エスカレーションツリーや、Monarch アクセス権限などの、Monarchの機能を追加するための拡張データがあります。

一度、変更をし終わったら、稼働中の構成設定を制御するためのオプションを提供するため、Nagios -v コマンドと同等の、Pre Flight Test を実施することができます。これは、構成を確認し、更新された Nagios の構成ファイルを作業領域ディレクトリに書き込みます。この時点では、現在稼働中のNagios の構成ファイルには影響を及ぼさず、構成ファイルを参照したり手動で変更したりできます。

Monarch は、そのファイルを稼働中の環境構成にコミット(commit)する機能も提供します。Commit コマンドは、あなたの古い構成情報を別のディレクトリにバックアップし、更新された構成ファイルを稼働中の nagios/etc ディレクトリに入れます。そして、Nagios のリスタートを行ってその構成を使うようにさせるようコミットします。

図 5.2.1. Monarch の構造概論



Monarch コンセプトの構造

ホストとサービステンプレート

ホストとサービステンプレートは、複数のホストとサービスを定義するのに使用する共通のオブジェクト属性を格納します。テンプレートは、オブジェクトを定義するときに多くの繰り返しのエントリを減らすのに使います。たとえば、新しいホストを定義するとき、あなたは最初に共通の属性をもつホステンプレートをサクセスするでしょう。そして、ホストに固有な定義をホステンプレートに適用します。新しいサービスの定義も同様に、サービステンプレートから始めて、テンプレートにサービス定義を適用します。

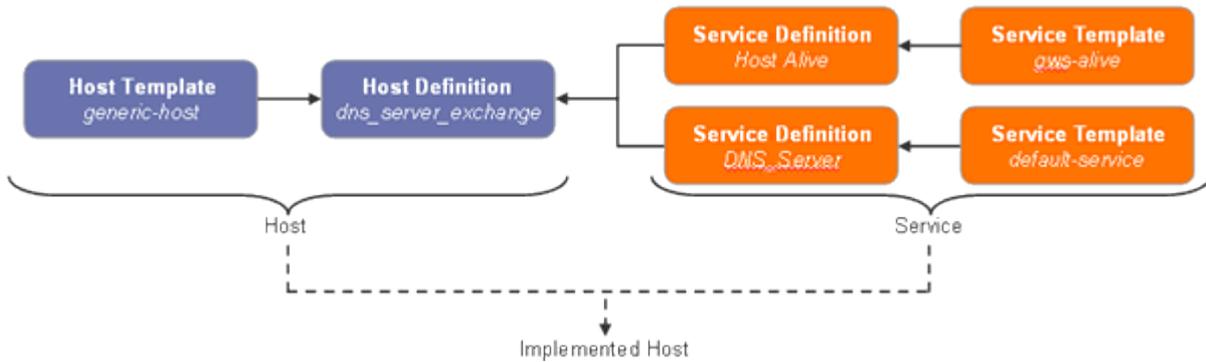
図 5.2.2. ホストとサービスのテンプレートと定義



サービス

Monarch におけるサービスは、サービステンプレートとサービス定義を組み合わせたものです。サービスは、あるホストに適用されている一般的なユニットです。ホストは、完全に導入された時点で、ホストテンプレートとユニークなホスト定義、そしてサービスを持ちます。

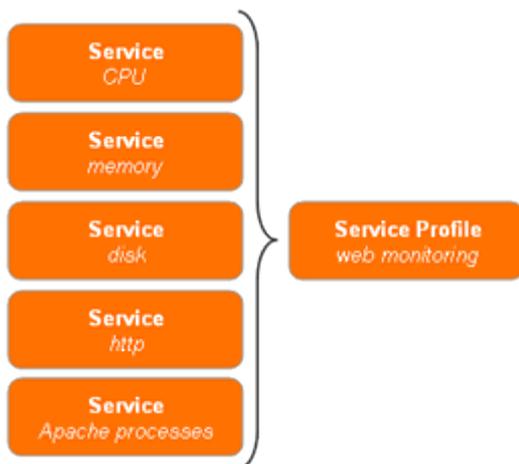
図 5.2.3. 完全に導入されたホスト



サービスプロファイル

サービスプロファイルは、複数のサービスの集合です。Monarch は、予め定義された監視パラメータ設定とユーザ設定の監視パラメータ設定を含む、デバイス固有のプロファイルを使います。プロファイルを使って、同種のデバイスを監視するためにすばやくGroundWork Monitor Open Sourceを構成設定することができ、GroundWorkの監視設計についての深い専門知識が推奨案のメリットを得ることができます。例: 50台のWebサーバを持っているとして、それらの上野CPU、メモリ、ディスク、HTTPおよび apache プロセスを監視したいとします。50の定義をする代わりに、各々のサービスCPU、メモリ、ディスクなどのサービス定義を生成するでしょう。そこで、*Web Monitoring* というサービスプロファイルを作成し、サービスプロファイルの中にすべてのサービスを含めます。そして、サービスプロファイルを50台のWebサーバ個々に関連付けることができます。Monarch は、これを、ホストプロファイルで行います。

図 5.2.4. 完全に導入されたホスト



ホストプロファイル

サービスプロファイルを定義すると、それをホストテンプレートを組み合わせて、ホストプロファイルを作ることができます。これらのプロファイルは、特定のホストに割り付けません。ホストプロファイルができたら、これを、あなたの50台の個別のWebサーバ毎の定義に組み合わせることができます。このコンセプトにより、あなたが監視し、簡単な適用する、異なるデバイスのための異なる役割を一般的に定義することができるようにします。上記の手順が完了してからどれかのパラメータひとつを変更し、それを50台すべてに適用したくなったら、全50台のサーバに適用されるホストプロファイルか、サービスプロファイルあるいは、サービスの中を変えることで変更できます。

図 5.2.5. 完全に導入されたホスト



Monarch のユーザインタフェース

Monarch をスタートすると、下記の表にリストされたメニューアイテムが表示されえます。この表で、各メニューアイテムを簡単に説明し、隣の列に、固有のカテゴリオプションをリストします。このメニューオプションは、よく使われる順に 左から右にリストされていますが、なんのアイテムで始めたいかは、その時点のあなたのデザインと管理のニーズに依存します。

表 5.2.1. メニューアイテム

Services (サービス)	Services オプションは、ホスト上で実行されるサービスやホストに関連付けられた計測基準の幾つかのタイプの定義と管理のために使用されます。サービス定義は、ホストとサービスのテンプレートおよびプロファイルの中で使用されます。	<ul style="list-style-type: none">- New- Modify- Service Templates- Service Dependency Templates- Extended Service Info Templates- Service Groups
Profiles (プロファイル)	Profiles オプションは、ホストとサービスのプロファイルを定義、変更および、インポートするのに使用されます。プロファイルは、ホストとサービスのデザインと管理の補助として使われます。	<ul style="list-style-type: none">- Host Profiles- Service Profiles- Profile Importer
Hosts (ホスト)	Hosts オプションは、ネットワーク上にある物理的なサーバ、ワークステーション、デバイスなどのホストを定義し、管理するのに使われます。ホスト定義は、ホストグループとプロファイル内で使われます。	<ul style="list-style-type: none">- New Host Wizard- Clone Host- Hosts- Host Groups- Parent Child- Host Templates- Extended Host Info Templates- Host Dependencies

<p>Contacts (コンタクト:連絡先)</p>	<p>Contacts オプションは、ネットワーク上の問題イベントを誰に連絡するかを定義するのに使います。コンタクトグループ(Contact Group)定義は、コンタクトグループ内の全コンタクトにアラート/リカバリの通知を送る目的で、ひとつ以上のコンタクト相互のグループを定義します。コンタクトの定義は、ホストとサービステンプレートとエスカレーションで使用されます。</p>	<ul style="list-style-type: none"> - Contacts - Contact Groups - Contact Templates
<p>Escalations (エスカレーション)</p>	<p>Escalations オプションは、ホストとサービスのエスカレーションツリーを定義します。エスカレーションは、個々のサービス、ホストやホストグループのためのコンタクトへの通知をエスカレート(上位に上げる)するのに使われます。エスカレーションツリーは、通知をエスカレートするため、ホスト、ホストプロフィール、ホストグループやサービスにアサインされた、複数のエスカレーションをグループリングします。</p>	<ul style="list-style-type: none"> - Escalations - Escalation Trees
<p>Commands (コマンド)</p>	<p>Commands オプションは、サービスとホストチェック、サービスとホスト通知、サービスとホストのイベントハンドラチェックに含まれるコマンドを定義するのに使用されます。コマンド定義は、ホストとサービスのテンプレートとプロフィールで使用されます。</p>	<ul style="list-style-type: none"> - New - Copy - Modify
<p>Time Periods (時間帯)</p>	<p>Time Periods オプションは、さまざまな日付にわたる通知やサービスチェックの適用時間かどうかを判断する時間のリストを停止するのに使用されます。時間帯の定義は、ホストとサービスのテンプレート、プロフィールおよび、コンタクトで使用されます。</p>	<ul style="list-style-type: none"> - New - Copy - Modify
<p>Control (コントロール:制御)</p>	<p>Control タブは、Monarch管理、セットアップと導入のコントロールで使用されるオプションを含んでいます。</p>	<ul style="list-style-type: none"> - Users - User Groups - Setup - Nagios CGI Configuration - Nagios Main Configuration - Nagios Resource Macros - Load - Files - Pre Flight Test - Commit
<p>Tools (ツール)</p>	<p>ツールタブは、構成設定ファイルをエクスポートするのに使われます。</p>	<ul style="list-style-type: none"> - Export

第3章 Configuration EZ – 始める

本章では、Monarch を使うことから始め、Monarch の構成設定ウィザード Configuration EZ の操作を最初から最後まで通します。Configuration EZ は、最もよく利用される構成設定機能へのアクセスを提供することにより、より容易な監視システムの設定を可能にします。EZ (イージー) であることの大きな条件は効果的であることです。ホストのプロファイルは予め定義されている必要があります。上級の管理者は、標準の構成設定ツールを使ってホストプロファイルに関連するすべての設定を定義することができます。監視システムにプロファイルをアサインしたり、導入することは、EZ を使って行うことが可能です。EZ が提供する機能は以下のとおりです：

表 5.3.1 Configuration EZ の機能

Hosts	監視システムのために新しいホストを定義します。 以下の3つの方法が可能です：1) EZ Monarch のWebページを使ってホスト情報を手入力する、2) カンマ区切り形式(CSV)ファイルからデータをロードする Import (インポート) オプションを使う、3) デバイスの自動検出を行うため Discover (ディスカバー) オプションを使う。
Host Groups	新しいホストグループを追加し、ホストをホストグループにアサインします。
Profiles	プロファイルをホストにアサインします。
Notifications	時間帯やコンタクトなどの通知パラメータを定義します。
Commit	Pre Flight Check であなたのコンフィギュレーションをチェックし、Commit オプションを使って、変更内容を Nagios に実施させます。
Setup	デフォルトのパラメータを設定します。

ホストを定義する

手動でホストを定義する

Configuration EZ メニューオプションから **Hosts** を選びます。

1. **New** を選びます。
2. **Host** 画面の中で、プロパティを入力します。(表 5.3.1 にリストし、説明します)。
3. **Save** を選びます。**Cancel** を選ぶと、セーブせずに Host 画面を終了します。

図 5.3.1. 新しいホストを定義する

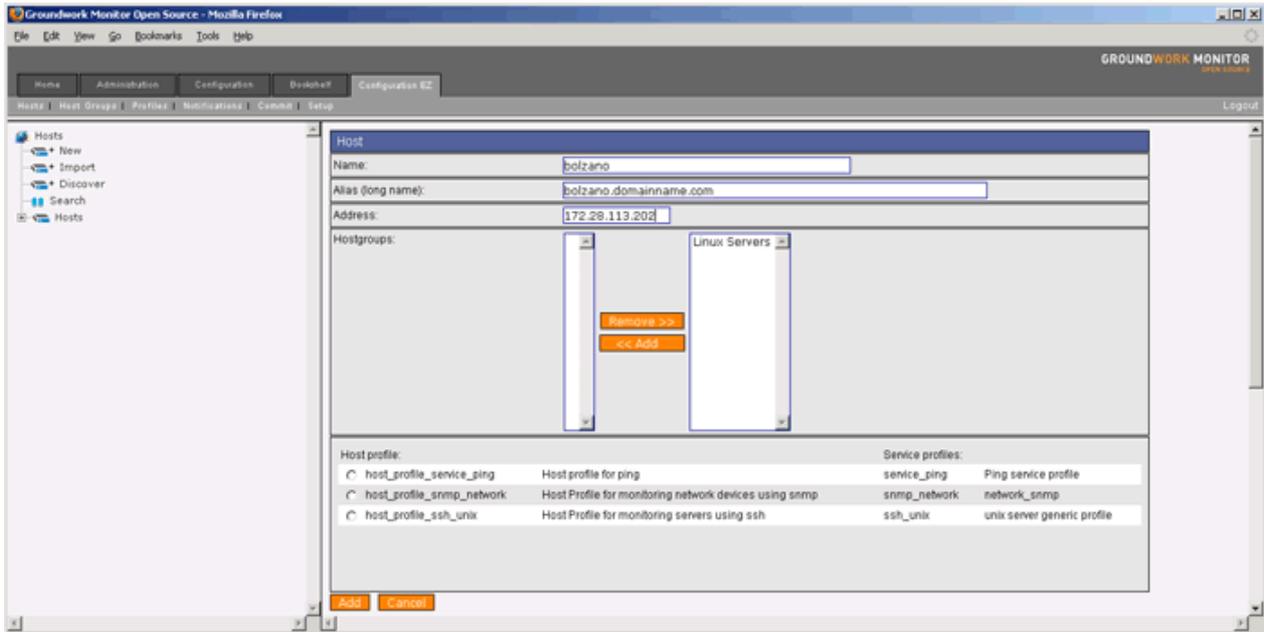


表 5.3.1. ホスト(Host)

Name	[必須] ホスト名。Host 定義の中で Address ディレクティブが指定されなかった場合、ホストの名称としてそのアドレスが使用されます。推奨方法:ホスト名には空白を入れないように。必要であれば、“_”(アンダースコア)を使いましょう。
Alias	[必須] このディレクティブは、長い名称を定義したり、ホストを識別する記述を付けるために使われます。これは、あなたが、各ホストを識別しやすくするために提供されています。
Address	[必須] このディレクティブは、ホストのアドレスを定義するために使われます。通常、これはIPアドレスですが(ホストのステータスをチェックする使うことができる限り)、実際にはどのようなものでもかまいません。IPアドレスの代わりに、ホストの識別のためにFQDNを使うことができますが、DNSサーバが使用できない場合には、問題になります。正しく使われる時には、\$HOSTADDRESS\$ マクロがこのアドレスとして入るでしょう。注意:Host 定義の中で Address ディレクティブが指定しなかった場合、ホスト名がアドレスとして使用されます。そうする場合の警告 - もしDNSに障害が発生すると、プラグインが、ホスト名を解決できなくなるので、ほとんどのサービスチェックが失敗します。
Host Groups	[必須] あなたが定義するホストは、最低限ひとつのホストグループのメンバーでなければなりません - たとえ、ホストグループの唯一のホストであったとしても。ホストは、二つ以上のホストグループに入れることができます。そのホストがダウンし、到達不能(unreachable)や回復(recovers)になった時、Nagios は、そのホストが、どのホストグループのメンバーか探し、そのホストグループ毎のコンタクトグループを調べて、それらのコンタクトグループに関連付けられたコンタクト(連絡先)に通知します。
Host Profile	[オプション] ホストプロファイルは、ホストのデザインや管理の補助情報として使用されます。

インポート(Import)を使ってホストを定義する

表計算ソフトなどのファイル中にホスト上がある場合、Import ツールでそのファイルを GroundWork の監視サーバに吸い上げ、ファイルのフィールドをホスト定義にマッピングし、新しいホスト定義を構成データベースの中に選択してイ

ンポートすることができます。Import ツールのための最低限必要なデータは、Name、Alias と Address です。

1. Configuration EZ メニューオプションから **Hosts** を選びます。
2. **Import** を選びます。
3. **Import Host Wizard** の Step 1: Upload File 画面で、ファイルのフィールドデリミタ(File delimiter)を選ぶか、入力(Other file delimiter)します。コンマ区分形式(CSV)では、","(コンマ)が一般的です。アップロードファイル(Uplod file:)を入力するか選択します。これは、あなたのシステム上にあつて、サーバにアップロードし、処理するファイルです。browse オプションを使って、ファイルをナビゲートします。以上が終了したら、**Next** を選んで、次のページに移ります。
4. 次の画面、Step 2: Set Schema 画面で、あなたのファイルのフィールドをホストのパラメータに割り当てることができます。あなたのファイルの最初の行がリファレンスとして、表示されます。あなたのファイルの各フィールド毎に、ドロップダウンリストからホストパラメータを選びます。適当なフィールドを、name、address と alias フィールドに割り当てます。 終了したら、**Next** を選んで、次のページに移ります。
5. 次の画面、Step 3: Process Hosts画面で、ホストファイルに入っているすべてのホストが表示されます。 Sort columns セクション内の選んだ名称、アイリアス、アドレスOS、プロファイル、その他のテキストを指定し、そのリストをソートすることができます。インポートしたいホストのチェックボックスを選びます。すると、Host Profile セクションの中に、選んだホストを適用するために Host Profile が現れます。それらのエントリを削除するには Discard ボタンを選びます。それらのホストを監視システムのコンフィギュレーションに追加する場合は、Add ボタンを選びます。一度ホストを追加すると、その後、左枠のナビゲーションツリー中のホストセクションの中に、それらがリストされます。

検出(Discover)を使ってホストを定義する

検出オプションは、インポートしたいホストのリストを自動的に作ろうとするために、ネットワーク検出プロトコルを使用します。指定されたIPアドレスの範囲でポートをスキャンするためにNmapツールが、使用されます。このスキャンを基に、ホスト名、IPアドレス、オペレーティングシステムを含むホストのリストが表示されます。そこで、監視システムのコンフィギュレーションにインポートしたいホストを選ぶことができます。注意:正しいホスト名を生成するために、そのホストについて、DNSの逆引き参照ができるように設定されている必要があります。そうでない場合、IPアドレスがホスト名として使用されます。

1. Configuration EZ メニューオプションから **Hosts** を選びます。
2. **Discover**を選びます。
3. Discover Host Wizard の Step 1: Scan Parameters画面で、スキャンする IPアドレスの開始(Strat)と終了(End)の幅を指定します。以上が終了したら、Next を選んで、次のページに移ります。
4. Step 2: Scanning 画面で、スキャンされた各ネットワークアドレスの結果が表示されます。Cancel ボタンを押すことで、いつでもスキャンを中止することができます。各スキャンの終了毎に、ホスト名、IPアドレス、オペレーティングシステムとステータスが表示されます。スキャンが完了したら、Next を選んで、次のページに移ります。
5. 次画面、Step 3: Process Hosts内で、スキャンの結果存在したすべてのホストが表示されます。Sort columns セクション内の名称、アイリアス、アドレスOS、プロファイル、その他のテキストを指定し、そのリストをソートすることができます。インポートしたいホストのチェックボックスを選びます。すると、Host Profile セクションの中に、選んだホストを適用するために Host Profile が現れます。それらのエントリを削除するには Discard ボタンを選びます。それらのホストを監視システムのコンフィギュレーションに追加する場合は、Add ボタンを選びます。一度ホストを追加すると、その後、左枠のナビゲーションツリー中のホストセクションの中に、それらがリストされます。

既存ホストを変更する

上記以外の方法でも、一度ホストがインポートされると、ホストのセッティングを変更することができます。ホストを見つけるには、ナビゲーションツリーのHost ノードを開いて、そこでホストを選びます。他の方法として、検索

(Search) オプションを使うこともできます。変更したいホスト名をタイプします。各文字を入力するごとに、マッチするホストがダイナミックに表示されていきます。そこで、リストの中からホストの名前を選ぶことができます。

一度ホストが選択されると、そのホストのエイリアス、IPアドレス、ホストグループやホストプロファイルを変更することができます。

ホストグループを定義する

本オプションで、新しいホストグループを作成することができ、既存のホストをホストグループにアサインすることができます。

1. Configuration EZ メニューオプションから **Hosts Groups** を選びます。
2. **New** を選びます。
3. **Host Group** 画面で、プロパティを入力します (表 5.3.2 にリストし、説明します)。
4. 新しいホストグループをセーブするため、**Add** を選びます。

図 5.3.2. 新しいホストグループを作る

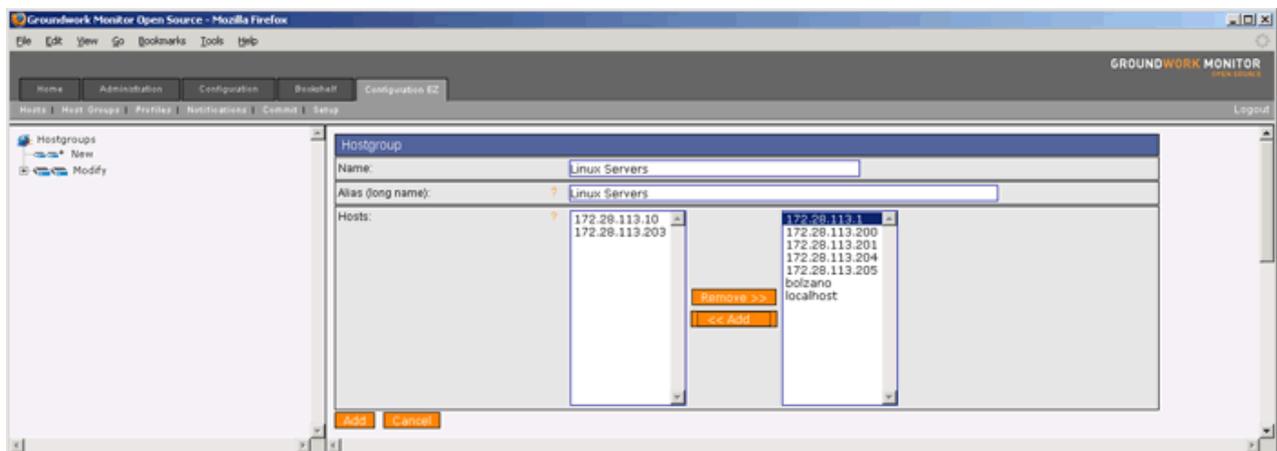


表 5.3.3. ホストグループ

Name	[必須]ホストグループ名。
Alias	[必須] このディレクティブは、長い名称を定義したり、ホストグループを識別する記述を付けるために使われます。これは、あなたが、各ホストグループを識別しやすくするために提供されています。
Hosts	[必須] これは、ホストグループに含まれるべきホストのショートネームのリストです。

プロフィールを構成する

本オプションで、既存のホストプロフィールをホストにアサインすることができます。

1. Configuration EZ メニューオプションから **Profiles** を選びます。
2. **Profiles** のドロップダウンメニューを拡大します。
3. 構成設定するプロフィールを選択します。
4. **Host Profile** 画面で、プロパティ(属性)を入力します(表 5.3.3 にリストし、説明します)。
5. **Add** を選んで、**Host Profile** 画面に移ります。

図 5.3.3. 新しいホストプロフィール

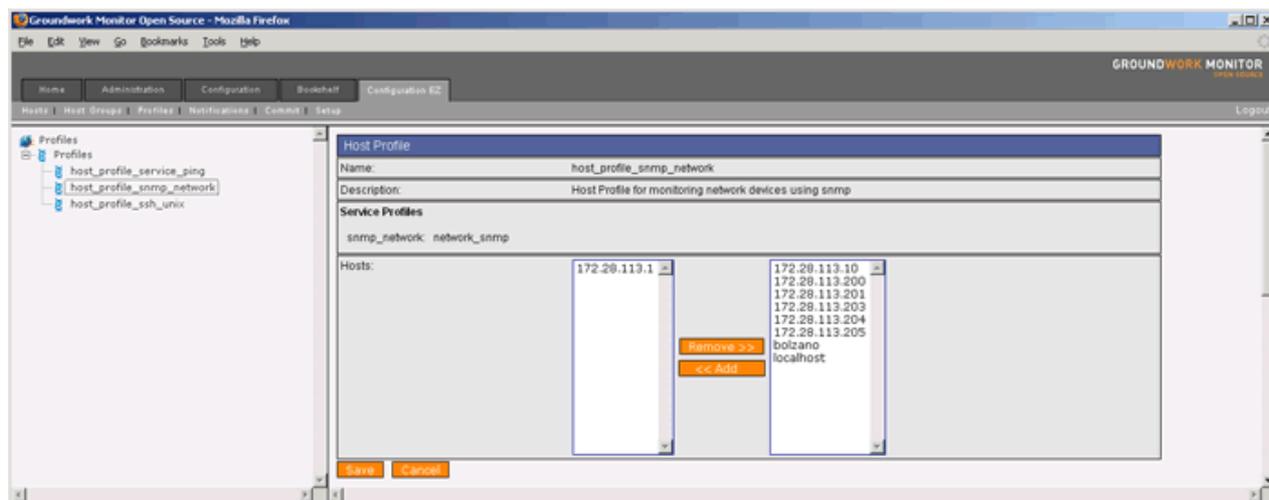


表 5.3.4. 新しいホストプロフィール

Name	[必須] ホストプロフィールの名称
Host Profile	[必須] ホストのデザインや管理の補助のために使用するホストプロフィール
Hosts	ホストプロフィールに追加するためのホストのリスト

通知を設定する

本オプションで、通知をONにしたりOFFにしたりし、通知の時間帯を作成し、通知のコンタクトを定義します。

ステップ 1 – 新しい時間帯 (Time Period) を定義する

1. Configuration EZ メニューオプションから **Notifications** を選びます。
2. **Time Periods** のドロップダウンメニューを拡大します。
3. 新しい時間帯 (Time Period) を作るため、**New** を選びます。
4. **Time Period** 画面で、プロパティ (属性) を入力します (表 5.3.4 にリストし、説明します)。
5. 新しい時間帯を追加するため、**Add** を選びます。時間帯がセーブされたことが示され、次の画面が促されます。

図 5.3.4. 時間帯のプロパティ

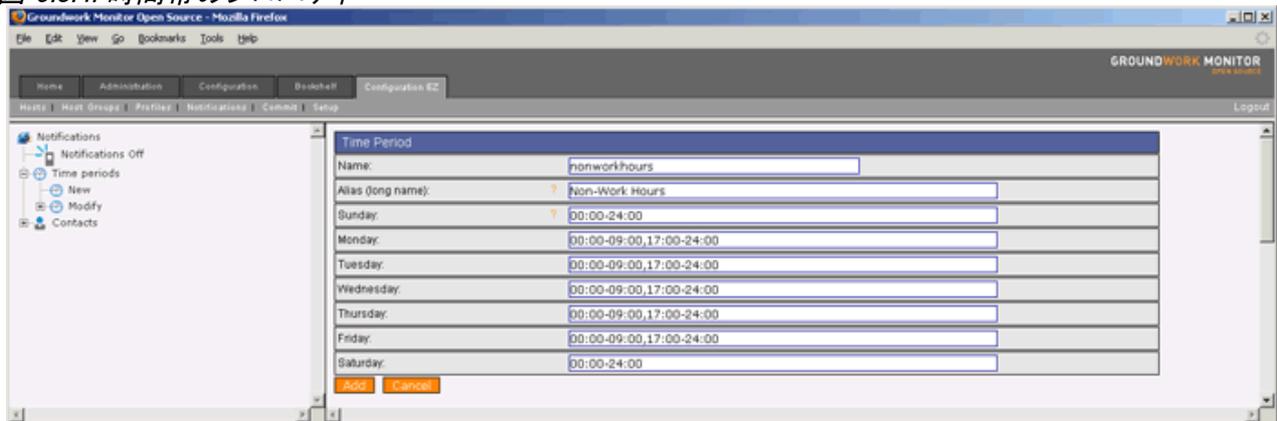


表 5.3.5. 時間帯

Name	時間帯の名称
Alias	[必須] このディレクティブは、時間帯を識別するための長い名称や説明。
Sunday-Saturday	[オプション] Sunday から Saturday のディレクティブは、コンマで区切られた時間の範囲で、どの曜日の「適用」時間を示します。ここで、時間の範囲を指定できる7つの異なる曜日 (日曜日なら土曜日) があることに注意してください。各時間の範囲は、H H: MM-HH: MM の形式で24時間時刻で指定します。たとえば、00:15-24:00 は、朝 (早朝) 0時15分から夜中の午後12:00までを示します (トータルの時間の範囲は、23時間45分)。全日を、この時間帯に指定したくない場合は、単にここに時間帯指定を入れないようにします。

ステップ 2 – 新しいコンタクトを定義する

1. Configuration EZ メニューオプションから **Notifications** を選びます。
2. **Contacts** のドロップダウンメニューを拡大します。
3. **New** を選びます。
4. **Contact Properties** 画面で、プロパティ (属性) を入力します (表 5.3.5 にリストし、説明します)。
5. **Add** を選びます。
6. コンタクトが追加されたことが表示されます。 **Continue** を選びます。

注意: コンタクトの変更や削除をするには、**Modify** を選んでください。そうすると、コンタクトを編集し、変更されたコンタクトを **Save** (セーブ) することができます。選んだコンタクトを削除するには **Delete**、名称の変更には **Rename**、変更せずに終了するには **Cancel** を選びます。

図 5.3.5. コンタクトのプロパティ

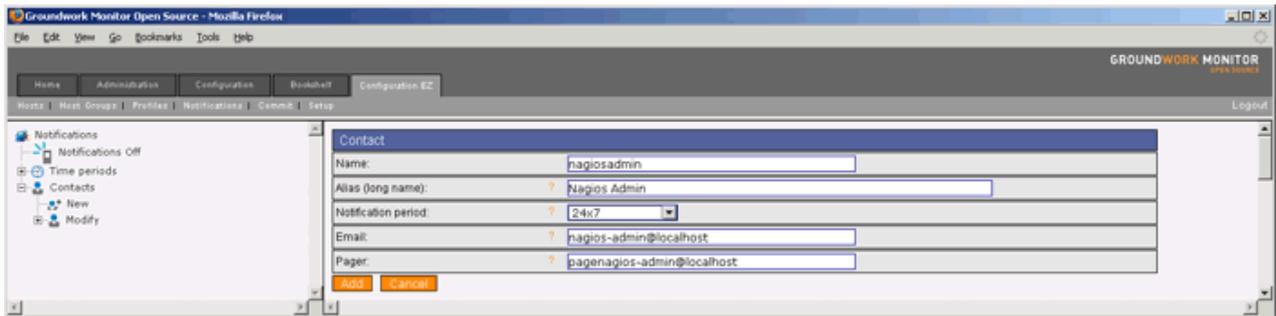


表 5.3.6. コンタクト

Name	コンタクトの名前
Notification Period	[必須] このコンタクトが通知を受けることのできる時間帯を選びます。
Alias	[必須] このディレクティブは、長い名称を定義したり、Contactの説明を定義するのに使われます。正しい環境下では、\$CONTACTALIAS\$ マクロがこの値となります。
Email	[オプション] このディレクティブは、Contactのための電子メールアドレスを定義するのに使われます。通知コマンドを設定することにより、アラートのメールをコンタクトとに送ることができます。正しい環境下では、\$CONTACTEMAIL\$ マクロがこの値となります。
Pager	[オプション] このディレクティブは、Contactのためのページャ番号を定義するのに使われます。ページャのゲートウェイの電子メールアドレスでも良いです（例： pagejoe@pagenet.com）。通知コマンドを設定することにより、アラートのページャコールをコンタクトとに送ることができます。正しい環境下では、\$CONTACTPAGER\$ マクロがこの値となります。

ステップ 3 - 通知をオンにする

1. Configuration EZ メニューオプションから **Notifications** を選びます。
2. **Notifications Off** オプションをクリックします。
3. 通知がオンに設定されます。

図 5.3.6. 通知



コミット(Commit)を使う

構成設定ツールを使ったような変更でも、その変更がコミット(実施)されるまでは、効果をおよぼしません。このオプションは、監視システムへの変更のコミットのためのプロセスを提供します。

コミットする前に、Preflight Test オプションを実施することをお勧めします。このオプションは、コンフィギュレーションにエラーがないかをチェックしますが、変更自体を導入はしません。もしエラーがあれば、PreFlight Test が選択された後で、それらエラーが Nagios Pre Flight Test ウィンドウにリスト表示されます。

変更を導入するには、ナビゲーションツリー上の Commit オプションを選びます。Abort、Backup と Commit の選択肢が表示されます。Backup オプションは、(変更が適用される前に)現在の Nagios 構成ファイルのコピーを作り、Monarch の構成データベースのデータベースダンプを行います。デフォルトでは、これらは `/usr/local/groundwork/monarch/backup` の下のタイムスタンプされたディレクトリに置かれます。必要であれば、バックアップされたどの構成でも、Nagios 構成ファイルを `/usr/local/groundwork/nagios/etc` ディレクトリにコピー、あるいは、ダンプされた Monarch の MySQL データベースをリストアすることで、元に戻すことができます。

1. Configuration EZ メニューオプションから **Commit** を選びます。
2. **Commit** を選びます。
3. **Nagios Commit** 画面で、稼働中の Nagios 構成を**バックアップ(backup)**することを促されるでしょう。
4. **Commit** を選び、稼働中の Nagios 構成を上書きして、Nagios を再起動します。

図 5.3.7. Nagios のコミット

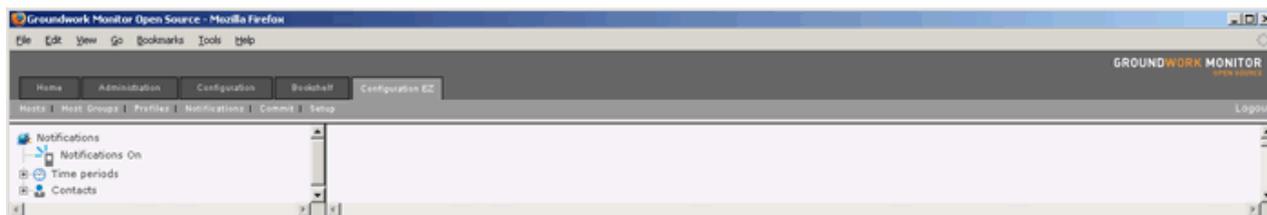


表 5.3.7. Nagios のコミット

Commit	稼働中の Nagios 構成を上書きして Nagios を再起動するには Commit を選びます。
Backup	データベース内の Nagios 構成レコードをバックアップするため Backup を選びます。
Abort	Nagios 構成レコードをロードするのを中止やキャンセルします。

セットアップ(Setup)を使う

このオプションは、新しい定義が作られたときに適用されであろうデフォルトの値をセットアップすることができるようにします。

1. Configuration EZ メニューオプションから **Setup** を選びます。
2. **Configuration Defaults** を選びます。
3. **Configuration Defaults** 画面で、プロパティ(属性)を入力します(表 5.3.7 にリストし、説明します)。
4. **Save** を選びます。

図 5.3.8. 構成設定のデフォルト

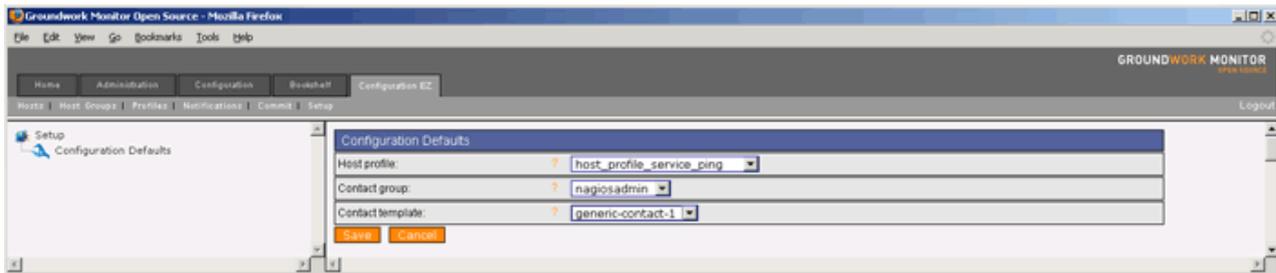


表 5.3.8. 構成設定のデフォルト

Host Profile	[必須] このツールで追加されるすべてのホストのために適切なホストプロファイルを選びます。これは、ほとんどの環境下では単純PING (simple ping) プロファイルになります。
Contact Group	[必須] このツールで追加されるすべてのコンタクトのために適切なコンタクトグループを選びます。
Contact Template	[必須] このツールで追加されるすべてのコンタクトのために適切なコンタクトテンプレートを選びます。

第4章 高度な構成設定 - サービスの構成設定

本章では、Monarch を使ったサービスの構成設定に焦点をあてます。このリファレンスは、サービス構成設定のすべての観点でより深い情報を提供するのを意図しています。各「どうやって(How To)」では、構成設定プロセスをステップ・バイ・ステップで説明します。始める前に、各ハウツーの下にあるリストをチェックして、要求される前提条件がないかを確認してください。

サービス(Services)

サービス定義は、ホスト上で実行されるサービス(Service)を定義するのに使用されます。サービスは、ホスト上で実行されている実際のサービス(POPやSMTP、HTTPなど)または、ホストに関係付けられた他のタイプの計測基準(PINGへの応答、ログインしているユーザ数、空きディスクスペース、など)です。サービスは、複数の定義(サービステンプレート、チェックコマンド、コマンドライン、サービス依存テンプレート、拡張サービス情報テンプレートとサービスエスケーションツリー)を組み込んで、ひとつにしたものです。これらのハウツーは、Monarch を使って、サービスを定義するプロセスを通して説明します。リストされた前提条件と共に始める必要があります。

- どうやって、新しいサービスを定義するか？ (追補1参照)
 - 時間帯 (追補8参照)
 - サービステンプレート (追補1参照)
 - コマンド(テンプレート内で定義されていなければ必須。) (追補7参照)

サービステンプレート(Service Template)

サービステンプレートは、複数のサービスに共通の属性を持つ場合に便利です。サービステンプレートは、複数のサービスの詳細なディレクティブとサービスチェック定義を盛り込んでいます。このハウツーは、新しいサービステンプレートを定義するプロセスを紹介します。

例: サービステンプレート (`generic-win-disk-c`)において、チェックコマンド `check_nt_disk` は、`$USER1$/check_nt -H $HOSTADDRESS$ -p 1248 -v USEDDISKSPACE -I $ARG1$ -w $ARG2$ -c $ARG3$` というコマンドラインを持っており、値(Value)として `Service check_disk_c` のためのデフォルトのコマンドラインの値である `check_nt_disk!C!85!95` がセットされています。ここで、`check_disk_d` というサービスを作りたいなら、サービステンプレートの `generic-win-disk-c` を選び、`Inherit all values from template` をチェックしないで、Cパラメータを Dに置き換えま

す。

- どうやって、サービステンプレートをつくるか？ (追補1参照)
 - 時間帯 (追補8参照)
 - コマンド (追補7参照)
 - コンタクトグループ(通知が指定されていれば、必須) (追補5参照)

サービス依存性テンプレート(Service Dependency Template)

サービス依存性は、ひとつあるいはそれ以上の他のサービスのステータスに起因する、通知とアクティブチェックを抑制することを可能にします。

- どうやって、サービス依存性テンプレートを構成設定するか？ (追補1参照)
 - サービス (追補1参照)

拡張サービス情報テンプレート(Extended Service Info Templates)

拡張サービス情報テンプレートは、基本的に status と extinfo の CGI の確認の出力を作るのに使用されます。それらは監視には影響を及ぼさず、完全にオプション的なものです。このハウツーは、拡張サービス情報テンプレートの作成ステップを説明します。

→ どうやって、拡張サービス情報テンプレートをつくるか？ (追補1参照)

サービスグループ(Service Group)

サービスグループ定義は、通知を単純化する目的で、ひとつ以上のサービスをグループにまとめるために使用します。サービスが停止(Down)し、到達不能(unreachable)や停止になったとき、Nagios はそのサービスが、どのサービスグループのメンバーを見出し、それらの各サービスグループのためのコンタクトグループを確認して、そのコンタクトグループに関連付けられているすべてのコンタクトに通知します。サービスグループは、誰がどのような問題のときにページャ通知を受けられるかを決定するのに柔軟性を与えます。サービスグループは表示のため、CGIでサービスお互いにグループ化できるようにし、構成設定を少し容易にするためにサービス依存性とサービスエスカレーション定義の中で参照できるようにします。このハウツーは、新しいサービスグループの作成ステップを説明します。サービスグループのための前提条件は、ホストとサービスを定義し、サービスをホストにアサインしておくことです。

→ どうやって、サービスグループをつくるか？ (追補1参照)

第5章 高度な構成設定 – プロファイル構成設定

本章では、Monarch を使った、プロファイルの構成設定に焦点をあてます。このリファレンスは、プロファイルの構成設定のすべての観点でより深い情報を提供することを意図しています。各「どうやって(How To)」では、構成設定プロセスをステップ・バイ・ステップで説明します。始める前に、各ハウツーマニュアルの下にあるリストをチェックして、要求される前提条件がないかを確認してください。

ホストプロファイル(Host Profiles)

ホストプロファイルは、ホストテンプレートとサービスプロファイルを格納します。一度ホストプロファイルができたなら、それを複数のホスト、たとえば50台の特定のWebサーバ、をまとめた定義として適応することができます。ホストプロファイルは、特定のホストに関係付けられるものではありません。このコンセプトは、監視する異なるデバイスの異なる役割を一般的に定義し、それを適応することができるようにします。ホストプロファイルを定義されたら、指定されたすべてのホストに適用する、ひとつのパラメータの変更をしたくなるかもしれません。あなたは既にすべてのホストに適用されホストプロファイルやサービスプロファイルあるいは、サービスの中を変更することができます。

- どうやって、ホストプロファイルを定義するか？（追補2参照）
 - サービスプロファイル(使われている場合)（追補2参照）
 - ホスト（追補3参照）
 - ホストテンプレート（このホストプロファイルを適用）（追補3参照）
 - ホストグループ（このホストプロファイルを適用）（追補3参照）

サービスプロファイル(Service Profile)

サービスプロファイルは、複数のサービスの集まりです。サービスプロファイルができると、これとホストテンプレートを組み合わせ、ホストプロファイルを作成します。GroundWork Monitor Professionalでは、あなたが使うことのできる独自の、複数種類のサーバやデバイスなどのために予めロードされたサービスチェックを入手できます。サービスプロファイルは、一連のサービスとプラグインおよび特定のデバイスやプロトコルを監視するために適用するベストプラクティスをカプセル化します。サービスプロファイルを使う利点は、それが予め統合化されていることです。プロファイルを使って、デバイスのグループを同じ方法で監視するように、GroundWork Monitor Open Source をすばやく構成設定することができます。このハウツーマニュアルは、新しいサービスプロファイルを定義するプロセス：ステップ 1 – 新しいサービスプロファイル; 名前のアサインと構成ファイルのアサイン、および ステップ 2 – サービスプロファイルのプロパティの定義、を説明します。

- どうやって、サービスプロファイルを定義するか？（追補2参照）
 - サービス（追補1参照）
 - サービステンプレート（追補1参照）

プロファイルのインポート

予め定義された監視パラメータとユーザ定義の監視パラメータを含むプロファイルは、ホストとサービスをデザインや管理するのに使うため、Monarch の中にインポートすることができます。プロファイルは Profile Detail 画面の Export ボタンによって、エクスポートすることができ、プロファイルを移動と、ファイル共有を可能にします。

- どうやって、プロファイルをインポートするか？（追補2参照）

第6章 高度な構成設定 – ホストの構成設定

本章では、Monarch を使った、ホストの構成設定に焦点をあてます。このリファレンスは、ホスト構成設定のすべての観点でより深い情報を提供するのを意図しています。各「どうやって(How To)」では、各手順をステップ・バイ・ステップで説明します。始める前に、各ハウツーの下にあるリストをチェックして、要求される前提条件がないかを確認してください。

ホスト (Hosts)

ホストは、あなたのネットワーク上にある物理的なサーバ、ワークステーション、デバイスなどを定義するのに使われます。このハウツーは、Monarch を使って新しく監視するホストを定義するプロセスを説明します。まず、前提条件からはじめる必要があります。ホストを生成するため、ホストの要点 (Host Vitals)、ホストのアトリビュート (Host Attributes) の1と2、および このホストに特定のサービスを追加するための、ホストサービス選択の数ステップを続ける必要があります。新しいホストを作成する時、ホストプロファイルが選ばれたら、Parents、Host Groups、Extended Host Info、Host Escalation Tree と Service Escalation Tree のフィールドが自動的に埋められ、それを変更できます。

- どうやって、新しいホストを定義するか？ (追補3参照)
 - ホストテンプレート (追補3参照)
 - ホストとサービス構成ファイル (追補9参照)

ホストテンプレート (Host Templates)

ホストテンプレートは、複数のホストを定義するのに使われる共通のオブジェクト属性を格納します。テンプレートは、オブジェクトを定義するときに繰り返しエントリを削減するのに使用します。たとえば、新しいホストを定義するときに、最初に共通の属性のホストテンプレートを作成します。そして、個別のホストを定義するため、ホストテンプレートを使って、その属性を適用します。

- どうやって、ホストテンプレートを構成設定するか？ (追補3参照)
 - 時間帯 (追補8参照)

ホストグループ (Host Group)

ホストグループの定義は、通知を簡略化のために、ひとつ以上のホストをグループ化するのに使います。あなたが定義した各ホストは、最低限ひとつのホストグループのメンバである必要があります – それがホストグループ内の唯一のホストであったとしても。ホストは、ひとつ以上のホストグループに入れることができます。ホストがダウンして、到達不能 (unreachable) や回復 (recovers) になったとき、Nagios は、そのホストがどのホストグループのメンバーを調べて、各ホストグループのためのコンタクトグループを入手し、それらのコンタクトグループに関連付けられたすべてのコンタクトに通知します。ホストグループは、どのような種類の問題のときに誰がページャ通知を受け取るかを柔軟に定義できるようにします。

- どうやって、新しいホストグループを作成するか？ (追補3参照)
 - Hosts (追補3参照)
 - Contact Groups (追補5参照)

親/子 (Parent/Child)

親ホストは、通常、監視ホストとリモートのホスト間にあるルータ、スイッチ、ファイウォールなどです。リモートのホストの最も近くにあるルータ、スイッチなどは、ホストの親(Parent)となるとみなされます。もしこのホストが(仲介するルータなしに)監視を実施しているホストと同じネットワークセグメント上にある場合、そのホストはローカルネットワーク上にあり、親ホストを持たないと認識されます。

- どうやって、親/子関係(Parent/Child Relationship)を作るか？ (追補3参照)
 - ホスト (追補3参照)

拡張ホスト情報テンプレート(Extended Host Info Templates)

拡張ホスト情報のエントリーは、基本的に status、statusmap、statuswri および extinfo の CGI の確認のアウトプットを作るのに使われます。それらは監視には影響を及ぼさず、完全にオプション的なものです。

- どうやって、拡張ホスト情報テンプレートを作るか？ (追補3参照)

ホスト依存性(Host Dependencies)

ホスト依存性は、他のひとつ以上のホストのステータスに基づく、ホストのための通報を抑制するのに使います。サービス依存性は、ひとつあるいはそれ以上の他のサービスのステータスに起因する、通知とアクティブチェックを抑制することを可能にします。

- どうやって、ホスト依存性を作るか？ (追補3参照)
 - Hosts (追補3参照)

第6B章 高度な構成設定 – ホストの管理

本章では、Monarch を使った、ホスト構成の管理に焦点をあてます。New Host ウィザードやホストのクローニング (コピー) で新しいホストを作成した後、前に定義したホストを見つけ出すために検索機能を使ったり、ナビゲーション ツリー内 Hosts オプションを開いて、既存のホストを選ぶことができます。定義されたホストを選択するとホスト詳細 (Host Detail)、ホストとサービスのプロファイル (Host and Service Profiles)、親 (Parents)、ホストグループ (Host Groups)、エスカレーション (Escalations) とサービス (Services) を、一画面で管理するためにアクセスできます。それぞれのホスト管理のタブを下記のリストします。ホスト属性を変更するためのステップ・バイ・ステップの指示を見るに、そのリンクをクリックしてください。

ホスト詳細 (Host Detail) の管理

Manage Host 画面の Host Detail タブでは、すべて変種可能な、定義されたホスト名、エイリアス、ホストのアドレスと、先に選択されたホストテンプレートがリストされます。選んだテンプレートからすべてのディレクティブの値を引き継ぐには、**Set Inheritance** をクリックします。テンプレートの値を上書きするには、画面の下半分にある、そのディレクティブの左側のチェックボックスのチェックを外します。

→ ホスト詳細 (追補4参照)

ホストプロファイル (Host Profiles) の管理

Manage Host 画面の Profile タブでは、ホストプロファイルとサービスプロファイルをアサインしたり適用したりできます。変更を行った後、このページ上のサービスのリストを更新するため refresh ボタンを使います。また、Apply を選ぶ前に、適切なサービスオプションを選択しているかを確認してください。現在のホストから全サービスを削除してあなたが選んだもの置き換える **Replace Existing Services** を選ぶこともできます。

Merge with Existing Services は、既存のサービスをそのままにします。Manage Host 画面の Profile タブからホストプロファイルをアサインすることで関連を作り、プロファイルからホストを管理でき、ホストの構成に他の影響を及ぼしません。

→ ホストプロファイル (追補4参照)

ホスト – 親 (Host – Parents) の管理

Manage Host 画面の Parents タブは、ホスト定義に、親ホストをアサインします。

また、親子関係は Hosts>Parent Child と行くことで直接管理することもできるでしょう。親ホストは、通常、監視ホストとリモートのホスト間にあるルータ、スイッチ、ファイウォールなどです。

→ ホスト – 親 (追補4参照)

ホスト – ホストグループ (Host – Host Groups) の管理

Manage Host 画面の Host Groups タブは、ホスト定義にホストグループをアサインするのに使います。また、ホストグループは Hosts>Host Groups と行くことで直接管理することもできるでしょう。ホストがダウンして、到達不能 (unreachable) や回復 (recovers) になったとき、Nagios は、そのホストがどのホストグループのメンバーを調べて、各ホストグループのためのコンタクトグループを入手し、それらのコンタクトグループに関連付けられたすべてのコンタクトに通知します。

→ ホスト – ホストグループ (追補4参照)

ホスト - エスカレーション(Host - Escalations)の管理

Manage Host 画面の Escalations タブは、ホストと、ホストのため適当なサービスエスカレーションツリーをオプションとして、選ぶことができるようにします。サービスエスカレーションが選択されると、選択されたホスト上のすべてのサービスが同じエスカレーションを使います。異なるサービスに異なるエスカレーションを使うには、各サービスが、それ自体のエスカレーションを持つ必要がありますし、この場合は、サービスエスカレーションをアサインするべきではありません。エスカレーションオプションは、ホストとサービスエスカレーションツリーを定義するのに使われます。エスカレーションは、特定のサービス、ホスト、ホストグループのためのコンタクト通知をエスカレートさせるのに使用します。エスカレーションツリーは、通知をエスカレートするため、複数のエスカレーションをグルーピングして、ホストやホストプロファイル、ホストグループとサービスにアサインしたものです。

→ ホスト - エスカレーション (追補4参照)

ホスト - サービス(Host - Services)の管理

Manage Host 画面の Service タブで、ホストにアサインされたサービスを管理できます。あなたは、ホストに、サービスの追加、変更、削除ができます。このページからサービスを管理すると、そのホストは、十中八九、そのサービスプロファイルとの同期から外れるでしょう。変更を行った後、このホストにプロファイルを適用するとき用心してください。

→ ホスト - サービス (追補4参照)

第7章 高度な構成設定 – コンタクトの定義

本章では、Monarch を使った、コンタクト、コンタクトテンプレートおよびコンタクトグループ構成設定に焦点をあてます。各「どうやって(How To)」では、手順をステップ・バイ・ステップで説明します。始める前に、各ハウツーの下にあるリストをチェックして、要求される前提条件がないかを確認してください。

コンタクト(Contacts)

コンタクト定義はあなたのネットワーク上の問題イベントの場合に連絡を受ける誰かを調べるために使います。

- どうやって、新しいコンタクトを定義するか？ (追補5参照)
 - コンタクトテンプレート (追補5参照)
 - 時間帯 (追補8参照)

コンタクトグループ(Contact Groups)

コンタクトグループは、アラート/回復の通知を送出するためのひとつ以上のコンタクトを定義します。ホストやサービスに問題が発生したり、回復した時、Nagios は通知を送るべき適切なコンタクトグループを見つけ、それらのコンタクトグループ内にすべてのコンタクトに通知を送ります。Nagios 2.0 では、コミットを行う前に、最低ひとつのコンタクトグループをホストに設定する必要があります。コンタクトグループは、ホストやサービスの通知のセットアップに使用するため、エスカレーションテンプレートとエスカレーションツリーに適用することができます。

- どうやって、コンタクトグループを作成するか？ (追補5参照)
 - コンタクト (追補5参照)

コンタクトテンプレート(Contact Templates)

コンタクトテンプレートは、共通の属性を持つ複数のコンタクトを作るときに便利です。コンタクト定義は、あなたのネットワーク上の問題イベントの場合に連絡を受ける誰かを調べるために使います。

- どうやって、コンタクトテンプレートを定義するか？ (追補5参照)
 - 時間帯 (追補8参照)

第8章 高度な構成設定 - エスカレーションの構成設定

本章では、Monarch を使った、エスカレーション設定に焦点をあてます。各「どうやって(How To)」では、手順をステップ・バイ・ステップで説明します。始める前に、各ハウツーの下にあるリストをチェックして、要求される前提条件がないかを確認してください。

エスカレーション(Escalations)

通知とエスカレーションは、監視サービスがステータス(OK、ワーニング、クリティカルと未知)の変化したとき、GroundWork Monitor の Nagios エンジンがどのようにユーザにアラートを送るかを定義します。エスカレーションは、通知がエスカレートされた時に通知を受ける指定されたコンタクトグループを組み合わせます。エスカレーションツリーは、ホスト、ホストプロファイル、ホストグループやサービスにアサインされた複数のエスカレーションがグループ化されたものです。エスカレーションは、任意選択です。

- どうやって、ホストやサービスのエスカレーションを構成設定するか？ (追補6参照)
 - 時間帯 (追補8参照)

エスカレーションツリー(Escalation Trees)

エスカレーションツリーは、ホスト、ホストプロファイル、ホストグループやサービスにアサインされた複数のエスカレーションがグループ化されたものです。

- どうやって、ホストやサービスのエスカレーションツリーを定義するか？ (追補6参照)
 - 定義されたホスト (追補3参照)
 - 定義されたホストグループ (追補3参照)
 - 定義されたサービス (追補1参照)
 - 定義されたサービスグループ (追補1参照)

第9章 高度な構成設定 - コマンドの定義

本章では、Monarch を使った、コマンド設定に焦点をあてます。コマンドを定義するコマンドラインは、特にコンタクト、ホストとサービス定義内で参照されます。コマンドは、サービスチェック、サービス通知、サービスイベントハンドラ、ホストチェック、ホスト通知、およびホストイベントハンドラに含まれるよう定義されます。コマンド定義は、マクロを含むことができますが、コマンドが使用される時の環境において、それらのマクロが使用可能な場合のみ含めることを必ず確認してください。コマンドの実行直前に、Nagios はコマンドの中のすべてのマクロを、それらに対応する値で置き換えます。このことは、あなたの必要なことすべてを取り扱うため、少数の一般的なコマンドを定義できるようにします。コマンドが使用される時の環境において、それらのマクロが使用可能な場合のみ含めることを必ず確認してください。マクロの完全なリストと使い方は、[Nagios のドキュメント](#)を参照してください。。各「どうやって(How To)」では、手順をステップ・バイ・ステップで説明します。始める前に、各ハウツーの下にあるリストをチェックして、要求される前提条件がないかを確認してください。

- どうやって、新しいコマンドを定義するか？ (追補7参照)

第10章 高度な構成設定 – 時間帯の定義

本章では、Monarch を使った時間帯設定に焦点をあてます。時間帯は、通知やサービスチェックの適用時間とみなす、さまざまな日に渡る時間のリストです。各「どうやって(How To)」では、手順をステップ・バイ・ステップで説明します。始める前に、各ハウツーの下にあるリストをチェックして、要求される前提条件がないかを確認してください。

- どうやって、時間帯を定義するか？ (追補8参照)

第11章 高度な構成設定 – コントロールオプションを使う

本章では、Nagios ディレクトリ、CGI とメインの構成設定、コマンドの中で定義されるリソースマクロなど、Monarch を使った Nagios 内のコントロールオプションに焦点をあてます。各「どうやって(How To)」では、手順をステップ・バイ・ステップで説明します。

- どうやって、セットアップの中の設定を参照/編集するか？ (追補9参照)
- どうやって、Nagios CGI 構成のための設定を参照/編集するか？ (追補9参照)
- どうやって、Nagios メイン構成のための設定を参照/編集するか？ (追補9参照)
- どうやって、Nagios リソースマクロを参照するか？ (追補9参照)
- どうやって、Nagios 構成設定をロードするか？ (追補9参照)
- どうやって、ファイルを追加し、ファイルの関連を参照するか？ (追補9参照)
- どうやって、飛行前チェック(Pre Flight Test)を実行するか？ (追補9参照)
- どうやって、Nagios 構成設定をコミットするか？ (追補9参照)

第12章 高度な構成設定 – ツールオプションを使う

本章で、あなたはすべての構成設定ファイルを見たり、デバッグするためエクスポートすることができます。これらは、ディレクトリ `/usr/local/groundwork/apache2/htdocs/monarch/download` にダウンロードされる正しい構成設定ファイルです。

- どうやって、構成設定ファイルをエクスポートするか？ (追補10参照)

第6部 GroundWork Monitor Open Source コンポーネントの構成設定

本セクションでは、GroundWork Monitor Open Sourceのコンポーネントとそれらのパッケージ構成についての情報を提供します。

第1章 概要

第2章 ステータスビューアを構成設定する

第3章 フィーダを構成設定する

第4章 レポートを構成設定する

- ディレクトリ構造

- コンフィギュレーション(構成設定)

- データロードスクリプト

- レポート更新をスケジュールする

第5章 ブックシュelfを構成設定する

- パッケージインストール

- ディレクトリ構造

- コンフィギュレーション(構成設定)

- カスタマイズ作業

第1章 概要

GroundWork Monitor Open Source は、ステータスビューア、レポートとドキュメンテーションシステムである ブックシェルフ(Bookshelf)のコンポーネントから構成されます。これらのコンポーネントは、それぞれ独立して操作することができます。フィーダーコンポーネントは、Nagios の構成とステータスファイルからデータを読み、sv データベースをアップデートします。sv データベースは、ステータスビューアのアプリケーションによって Nagios オブジェクトのリアルタイム状態を問い合わせるのに使用されます。レポートシステムは、cron スクリプトによって毎日更新される独自のデータベースを持ちます。またブックシェルフも、ドキュメンテーションシステム内の項目検索を可能にする、独自のデータベースを持っています。

第2章 ステータスビューアを構成設定する

ステータスビューアのアプリケーションによって使用される設定をカスタマイズするには、Administration タブを使います。(第3部 GroundWork Monitor Open Source の管理、の第4章 パッケージの管理を参照ください)。下記のパラメータが設定されるでしょう。

表 6.2.1. パッケージの構成設定

Status Information Store	これはステータスデータをデータ格納のタイプを指定します。これはGroundWork Monitor Open Source では Internal 、GroundWork Monitor Professional では Foundation となります。
Database	これは、情報格納のためのデータベースパラメータを指定します。
Type	このバージョンでは、MySQL のみがサポートされます。
Address	データベースのホスト IP アドレス、通常はローカルシステムの 127.0.0.1。これをリモートのサーバの指定に変更する場合、データベースはこのサーバからアクセスされるよう作成する必要があることに注意してください。
Name	データベース名、GroundWork Monitor Open Source では "sv" を、GroundWork Monitor Professional では "GWCollageDB" をセットします。
Username	データベース名にアクセスするための認証されたユーザ名
Password	データベース名にアクセスするための認証されたパスワード
Nagios	ステータスビューアで使用される Nagios パラメータがあります。
Comment file	オペレータが Nagios に追加したコメントを格納する場所。 デフォルトでは <code>/usr/local/groundwork/nagios/var/comments.dat</code>
Downtime file	Nagios がホストとサービスの停止時間情報を格納する場所。 デフォルトでは <code>/usr/local/groundwork/nagios/var/downtime.dat</code>
Command pipe	外部コマンドを追加するのに使用する、Nagios コマンドのパイプがある場所。デフォルトでは、これは、 <code>/usr/local/groundwork/nagios/var/spool/nagios.cmd</code>

第3章 フィーダを構成設定する

ステータスビューアが使用するデータベースは、Nagios ログとステータスファイルを読みアップデートを行う一連のプロセスによって行われます。

これらのプロセスは、フィーダ(feeder)と呼ばれ、ディレクトリ `/usr/local/groundwork/feeder` 下に置かれ、下記の名前です：

- `nagios2db_status.pl` - Nagios のステータスログを読みステータスビューアデータベースのホストとサービスのステータス情報をアップデートします。
- `nagios2db_synch.pl` - Monarch の構成設定データベースを読み、ホストグループ、ホストおよびサービスの名前をステータスビューアと同期させます。このプロセスは、Monarch から Nagios に新しい構成がコミットされたら時に自動的に実行されますが、手動でこのスクリプトを実行することも可能です。
- `nagios2db_eventlog.pl` - Nagios のイベントログを読み、データベースの Log Message テーブルを更新します。Nagios のイベントログメッセージをデータベースに格納する必要がなければ、これを無効にすることができます。イベントメッセージは、GroundWork Monitor Open Source版では使われませんが、GroundWork Monitor Professional のコンソールアプリケーションでは、使用します。これは、disabled (無効)がデフォルトです。

フィーダは、常に監視されます。それらが停止すると、サービスが自動的にそれらを再起動します。フィーダを停止したい場合、`/etc/init.d/gwservices stop` コマンドを実行します。フィーダを起動したい場合、`/etc/init.d/gwservices start` コマンドを実行します。

sv データベースは、ステータスビューア以外のアプリケーションでも使用できます。一般的な使用法は、被監視環境にあるホストグループ、ホスト、サービスやイベントの複雑な問い合わせを行う必要がある Nagios プラグインでしょう。

たとえば、ホストグループ内のすべてのサービスステータスを問い合わせ、50%のサービスがOKでない場合にアラームを発生させるために Perl API を使う Perl スクリプトを書くことができます。これらのプラグインを作成するのに使える API と サンプルについては GroundWork Monitor Open Source Developers Guide を参照してください。

ひとつのコンフィギュレーションファイルが、フィーダプログラムの操作をコントロールします。このファイルは、`/usr/local/groundwork/nagios/etc/nagiosfeeders.cfg` にあります。デフォルトの内容と説明を下記に示します。

```
#####  
#  
# Global properties  
#  
#####  
#  
#  
# Version of Nagios. This us used to determine the format of the status and event logs.  
# Should be 1 or 2  
nagios_version = 2  
#  
# IP address of host name of the Nagios server.  
# This field is used to set the monitoring server entry in the SV database.  
nagios_hostname = localhost  
  
#####  
#  
# Status feeder directives  
#  
#####  
#
```

```
# Debug flag for status feeder. Set to 1 to write debug messages to the feeder_status_log file
status_debug = 0
#
# Nagios log file. This is read by the status feeder and contains Host and status data.
# There are different formats for Nagios 1 vs Nagios 2
nagios_status_log = /usr/local/groundwork/nagios/var/status.log
#
# This is the amount of time in seconds that the status プログラム will wait before
# reading the nagios status log again.
loop_wait_time = 10
#
# This is the log file for the nagios2db_status.pl プログラム.
# Debug and informational messages will be contained here
feeder_status_log = /usr/local/groundwork/feeder/log/nagios2db_status.log
#
# This field determines when database updates will be performed. If smart_update is set
# to 0, every time the nagios status log is read, the プログラム will update each host and
# service entry in the database. If this is set to 1, the プログラム will only update the
# database when it recognizes a host or service status has changed. Change is defined as
# when the last check time changes, or when the acknowledged flag changes.
# Setting this to 1 will improve efficiency by reducing the number of database updates.
smart_update=1

#####
#
# Event log feeder directives
#
#####
#
# Enable event log feed into the database. Default is 0 since no applications use the log database. feeder_event_log
file
event_enable = 0
#
# Debug flag for event log feeder. Set to 1 to write debug messages to the feeder_event_log file
event_debug = 0
#
# Nagios event log file. This is read by the event feeder and contains an entry
# for nagios actions. The HOST ALERT, HOST NOTIFICATION, SERVICE ALERT and
# SERVICE NOTIFICATION event logs are read and inserted into the status viewer database.
nagios_event_log = /usr/local/groundwork/nagios/var/nagios.log
#
# This is a temporary file used by the プログラム to store the location of the last
# record read from the nagios event log. Every time the プログラム loops, it will
# start to read from this location, read to the end of the file, then write the
# last read file location to this file. This file is automatically created by
# the event log feeder. If you want to start reading from the beginning of the
# event log, erase this file.
nagios_event_seekfile = /usr/local/groundwork/nagios/var/nagios_seek.tmp
#
# This is the log file for the nagios2db_eventlog.pl プログラム.
# Debug and informational messages will be contained here
feeder_event_log = /usr/local/groundwork/feeder/log/nagios2db_event.log
#
# This is the amount of time in seconds that the プログラム will wait before
# checking to see if the nagios event log has more events to process
event_sleep_time = 15
```

```
#####  
#  
# Synchronization Check feeder directives  
#  
#####  
#  
# Debug flag for status feeder. Set to 1 to write debug messages to the feeder_status_log file  
hostgroup_debug = 0  
#  
# This is the log file for the nagios2db_synch.pl プログラム.  
# Debug and informational messages will be contained here  
feeder_hostgroup_log = /usr/local/groundwork/feeder/log/nagios2db_hostgroup.log  
#  
# This is the host group name to assign to all hosts that do not belong to a host group.  
# Since status viewer navigation starts at the host group, hosts that do not belong  
# to a host group will not appear. These hosts will be assigned this host group name  
# so that they can be shown in status viewer.  
orphan_hostgroupname = _hosts not in any hostgroup
```

第4章 レポートを構成設定する

パッケージの構成

Insight Reports は Guava のパッケージで、`/usr/local/groundwork/guava/reports` ディレクトリにインストールされます。

ディレクトリ構造

Insight Reports は下記の二つの機能から構成されます：

1. Nagios ログデータをレポートデータベースの中にロードする。これらのスクリプトはディレクトリ：
`/usr/local/groundwork/reports/utils` にあります。
これらのスクリプトは、ユーザ Nagios の下の Cron を使ってスケジュールされます。これらのスクリプトによって作られるログメッセージは下記のディレクトリにあります：
`/usr/local/groundwork/reports/utils/log`
`/usr/local/groundwork/nagios/var`
2. CGI プログラムs that create and present the report to a browser. These プログラムs are run under the Guava portal framework and are in the directory:
`/usr/local/groundwork/apache2/cgi-bin/reports`

コンフィギュレーション(構成設定)

構成設定ファイルは、`/usr/local/groundwork/nagios/etc/gwir.cfg` にあり、Insight Report の CGI とデータロードプログラムで使用される設定が含まれています。デフォルトの内容と説明を下記に示します。

```
# Configuration file for Nagios reports.
# This file is used by the following プログラムs:
# dashboard_data_load.pl - Updates the reports database with Nagios log data.
# dashboard_lwp_load.pl - Updates the reports database with Nagios availability report data.
# nagios_alarms1.pl - CGI プログラム that generates the alerts report page.
# nagios_notifications1.pl - CGI プログラム that generates the notifications report page.
# nagios_outages1.pl - CGI プログラム that generates the outage report page.
#
#####
#
# Global properties
#
#####
#
# Database parameters. Used to connect to the Status Viewer mysql database
dbusername = root
dbpassword =
dbname = dashboard
dbhost = localhost
#
#
# Version of Nagios. This is used to determine the format of the status and event logs.
# Should be 1 or 2
nagios_version = 2;
#####
#
# CGI parameters
```

```
#
#####
#
# Directory to place the generated graph image files
graphdirectory = /usr/local/groundwork/apache2/htdocs/reports
#
# Relative URL to access the generated graph image files
graphhtmlref = /reports

#####
#
# Nagios log data load parameters
#
#####
#
# The main nagios.cfg configuration file. The data load プログラム will read this file to get
# all the other nagios configuration files that are reference. All the reference files will
# then be read to identify host group directives. The host group and host members are identified
# and used in the reports.
nagios_cfg_file = /usr/local/groundwork/nagios/etc/nagios.cfg
#
# Nagios event log file. This is read by the event feeder and contains an entry
# for nagios actions. The HOST ALERT, HOST NOTIFICATION, SERVICE ALERT and
# SERVICE NOTIFICATION event logs are read and inserted into the status viewer database.
nagios_event_log = /usr/local/groundwork/nagios/var/nagios.log
#
# This is the log file for the dashboard_data_load.pl プログラム.
# Debug and informational messages will be contained here
dashboard_data_log = /usr/local/groundwork/reports/utils/log/dashboard_data.log
#
# Debug setting for the dashboard data load プログラム
# Set to 1 to write debug messages to the dashboard_data_log file
dashboard_data_debug = 1

#####
#
# Nagios availability data load parameters
# There are two methods for getting this data.
# - The lwp method uses Perl LWP to open a browser to the Nagios availability
# reports.
# - The avail method execute the report cgi local and read the output. This is the
# default method since it will work if Guava Single Sign authorization is on.
#
#####
#
# Debug flag for status feeder. Set to 1 to write debug messages to the feeder_status_log file
dashboard_lwp_debug = 1
#
# LWP method - IP Address of the nagios server. Used to call the Nagios availability report cgi
nagios_server_address = localhost
nagios_server_port = 80
#
# LWP method - Nagios security realm. This is on the userid/password dialog box when accessing a secure Nagios
page.
nagios_realm = Nagios Access
# LWP method - Authorized user ID to access Nagios reports page.
```

```
nagios_user = nagiosadmin
# Authorized password to access Nagios reports page.
nagios_password = nagiosadmin
# This is the log file for the dashboard_lwp_load.pl プログラム.
# Debug and informational messages will be contained here
dashboard_lwp_log = /usr/local/groundwork/reports/utils/log/dashboard_lwp.log
#
# Debug flag for status feeder. Set to 1 to write debug messages to the feeder_status_log file
dashboard_avail_debug = 1
# This is the log file for the dashboard_lwp_load.pl プログラム.
# Debug and informational messages will be contained here
dashboard_avail_log = /usr/local/groundwork/reports/utils/log/dashboard_avail.log
# The directory where the Nagios report CGI is located
#
nagios_cgi_dir = /usr/local/groundwork/apache2/cgi-bin/nagios
```

データロードスクリプト

データ収集プロセスは、Nagios データを分析してMySQLデータベースのダッシュボードにロードします。このプロセスをコントロールする二つのスクリプトがあります。

1. dashboard_data_load.pl

このスクリプトは、`nagios.log` ファイルを読み、アラートと通知エントリをダッシュボードデータベースにロードします。下記のオプションが利用可能です。

- **-c "configuration file"**
レポートロードプロセスのための設定が含まれる構成設定ファイル。
デフォルトは `/usr/local/groundwork/nagios/etc/gwir.cfg`
- **-a "DIRECTORY"**
このディレクトリ下の全ログファイルを読みます。通常、アーカイブディレクトリの中のすべてのログファイルを読みます。たとえば、`/usr/local/groundwork/nagios/var/archives` では、Nagios アーカイブディレクトリ内の全ファイルを処理します。
- **-C "NAGIOS CONTACT GROUP CONFIG FILE"**
コンタクトグループ定義が含まれている Nagios 構成設定ファイル。このプログラムは、これがセットされていないと実行されませんが、通知コンタクトレポート (Notification Contact reports) は正しく動作しません。デフォルトは `/usr/local/groundwork/nagios/etc/contactgroups.cfg`
- **-f "NAGIOS LOG FILE"**
この Nagios ログファイルを読みます。
デフォルトは `/usr/local/groundwork/nagios/var/nagios.log`
- **-L "OUTPUT LOG"**
このプログラムからのステータスメッセージを含むログファイル。
デフォルトは `/usr/local/groundwork/nagios/var /dashboard.log`
- **-H "NAGIOS HOST GROUP CONFIG FILE"**
ホストグループ定義が含まれている Nagios 構成設定ファイル。このプログラムは、これがセットされていないと実行されませんが、ホストグループレポート (Host Group reports) は正しく動作しません。デフォルトは `/usr/local/groundwork/nagios/etc /hostgroups.cfg`
- **-d**
デバッグモード。ログファイルに追加メッセージがログされます。

- `-h` または `-help`
ヘルプメッセージが表示されます。

2. `dashboard_avail_load.pl`

このスクリプトは、Nagios CGI ユーザインタフェースから、レポートページを開くためにHTTP GETを実行します。このページの詳細なレポート情報は、このスクリプトで処理され、ダッシュボードデータベースにロードされます。下記のオプションが利用可能です。

- `-c "configuration file"`
レポートロードプロセスのために設定を含む構成設定ファイル。
デフォルトは `/usr/local/groundwork/nagios/etc/gwir.cfg`
- `-L "OUTPUT LOG"`
このプログラムからのステータスメッセージを含むログファイル。
デフォルトは `/usr/local/groundwork/feeder/log/dashboard_lwp.log`
- `-s "YYYYMMDD"`
開始日 (デフォルト: 前日の00時)
- `-e "YYYYMMDD"`
終了日 (デフォルト: 前日の24時)
- `-d`
デバッグモード。ログが追加されます。

レポート更新をスケジュールする

デフォルトでは、毎晩、深夜12時に `dashboard_data_load.pl` スクリプトがスケジュールされます。
`dashboard_lwp_load.pl` スクリプトは毎晩、午前1時にスケジュールされます。スケジュールを変更するには:

1. `root` の ID を使って、GroundWork サーバへのセッションを開きます。
2. `crontab` のエントリを編集するため下記のコマンドを投入します: `crontab -u nagios -e`
3. `vi` エディタのセッションが開き、Nagios のための `crontab` エントリが表示されます:

```
59 23 * * * /usr/local/groundwork/reports/utils/dashboard_data_load.pl  
/usr/local/groundwork/reports/utils/log/dashboard_data_load.log 2;1  
0 1 * * * /usr/local/groundwork/reports/utils/dashboard_lwp_load.pl >  
/usr/local/groundwork/reports/utils/log/dashboard_avail_load.log 2;1
```
4. `crontab` エントリを適当なスケジュールに変更します。たとえば、実行時間を 11:59pm にする場合、下記のようにエントリを変更します:

```
59 23 * * * /usr/local/groundwork/reports/utils/dashboard_data_load.pl  
/usr/local/groundwork/reports/utils/log/dashboard_data_load.log 2";1
```

第5章 ブックシェルフを構成設定する

パッケージインストール

Bookshelf ライブラリは、Groundwork Monitor と共に予めインストールされています。Bookshelf アプリケーションがアンインストールされた場合、アプリケーション管理のパッケージ管理で再インストールすることができます。

ディレクトリ構造

Bookshelf パッケージは、パッケージリポジトリの 'bookshelf' ディレクトリにあります。このディレクトリの中に 'docs' と呼ばれる '/usr/local/groundwork/docs' へのシンボリックリンクがあります。docs ディレクトリは、Bookshelf アプリケーションが使用する、各 bookshelf ライブラリを含んでいます。各 bookshelf ライブラリは、別々のディレクトリです。これらのディレクトリの中身は、ライブラリ定義と関連するドキュメントです。

コンフィギュレーション(構成設定)

Bookshelf の構成設定は、管理 (Administration) アプリケーション内のパッケージ管理内で行われます。Bookshelf 構成設定の中にも、3つのサブセクション:ライブラリ(Libraries)、インデックス(Index)と構成設定(Configuration)があります。

ライブラリ(Libraries)

Libraries では、'docs' ディレクトリ内のすべての使用可能なライブラリのリストが表示されます。また、そのインストール状態も見ることができます。ライブラリは、Bookshelf アプリケーションの中で見るようにするには、インストールされている必要があります。ライブラリがインストールされたいなければ、その上をクリックして、詳細情報を見ることができます。これは、ライブラリ内の全ドキュメントの概要を表示します。Install をクリックすると、ライブラリはインストールされます。

インデックス(Index)

Bookshelf は、Bookshelf のドキュメントすべてにわたって索引を付ける複雑な検索エンジンを使います。もし、ライブラリがインストールされているか、アンインストールされている場合、インデクサー(Indexer)を手動で起動し、新しいドキュメントを索引付けし、古いドキュメントの削除を確実に行うべきです。Index Now をクリックすると、インデックスプロセスがバックグラウンドで動きます。

構成設定(Configuration)

Bookshelf は、Groundwork Monitor のインストールで予め構成設定されています。構成設定の中で、あなたがシステム構成設定をカスタマイズしたのであれば、その値を変更することができます。mnogosearch データベースは、検索問い合わせの結果を見つけるためにサーチエンジンが使用するデータベースです。インデクサーのパスは、新しいドキュメントの索引付けを行うときに使う、検索インデクサーのバイナリの場所です。

カスタマイズ作業

Bookshelf は、新しいドキュメントライブラリをソートする機能を持ちます。あなたの環境に Bookshelf を通じてアクセスしたい追加ドキュメントがある場合、ライブラリ定義ファイルを作成して、新しいライブラリをインストールすることができます。

ブックシェルフのコンテンツを追加する

あなたのカスタムブックシェルフのライブラリ(文献集)は、下記のようなディレクトリ構造になるでしょう:

```
"shortname"  
  library.pkg  
  "chapterdirectory"  
  "chapterdirectory2"
```

shortname は、あなたのライブラリを表すショートネームです。このディレクトリ中に、あなたの library.pkg 定義ファイルと、あなたのライブラリのカテゴリを表現するであろうサブディレクトリがあります。library.pkg ファイルは、ライブラリ名、カテゴリ、ドキュメントとドキュメントのリンクを含む、あなたのライブラリ構造を定義します。この library.pkg 構造は、下記に示します:

```
define library {  
    name = My Library  
    basepath = mylibrary/  
}  
  
define category {  
    name = My Top Level Category  
}  
  
define category {  
    name = My Sub Category  
    parent = My Top Level Category  
}  
  
define document {  
    name = My First Document  
    category = My Sub Category  
    filename = mysubcat/myfirstdoc.html  
}  
  
define link {  
    category = My Sub Category  
    document = My First Document  
    title = My Link  
    anchor = link  
}
```

ライブラリの定義 (define library) は必須です。その中に入れる二つの識別子があります。それらは、名称(name)とベースパス(basepath)です。名称はライブラリの名前。ベースパスは、ライブラリがあるディレクトリです。末尾のスラッシュ(/)は、ドキュメントのファイル名に、この値をプレフィックスとして付けるので、重要です。

カテゴリは、category definition で定義されます。そこには、ひとつの必須の識別子と、ひとつのオプション識別子があります。必須識別子は、name で、そのカテゴリの名前を宣言します。オプション識別子は、parent で、このカテゴリが、どのカテゴリの子供になるか指定します。カテゴリの名称は、大文字と小文字を区別をします。

ドキュメントは、document definition で定義されます。そこには、三つの必須識別子があります。name 識別子はドキュメント名。category 識別子はこのドキュメントが所属するカテゴリ。カテゴリ名は大文字と小文字を区別することに注意してください。filename は、ライブラリの basepath に対する、ドキュメントへのパスです。

ドキュメントリンクは、オプションです。ドキュメントリンクは、ドキュメント内のアンカータグと並行して使用されます。これは、あなたのドキュメントのサブセクションへのすばやいナビゲーションを提供します。category と document 識別子は、あなたがリンクしたいドキュメントと必ず一致しなければなりません。title は、リンクにあたなが付与する名称です。anchor 識別子は、あなたのドキュメント内のアンカー名と必ず一致しなければなりません。

一度、あなたのライブラリのディレクトリ構造を bookshelf パッケージ内の 'docs' ディレクトリ内にコピーすると、Books helf Configuration の Libraries セクションからインストールすることができます。

ブックシェルフのコンテンツを変更する

あなたのブックシェルフのコンテンツを変更する必要がある場合、ライブラリ定義ファイルを変更する前に、ライブラリをアンインストールする必要があります。これにより、ライブラリが完全に更新されることが保証されます。一度、ライブラリ定義ファイルを変更したら、ライブラリを再インストールし、ドキュメントの索引付けを再度行います。

ブックシェルフのコンテンツを削除する

ブックシェルフからライブラリを削除するには、Bookshelf Configuration のアプリケーションの中の Libraries セクションでそれを削除 (remove) します。それをアンインストールした後で、安全に 'docs' ディレクトリからライブラリのディレクトリを削除することができます。

第7部 GroundWork Monitor Open Sourceの維持管理

本セクションでは、構成データベースおよびヒストリカルデータのバックアップとリストア手順を含む、GroundWork Monitor Open Sourceの維持に関する情報を提供します。

第1章 GroundWork Monitor Open Source 構成データベース

- 構成データベースのバックアップ

- 構成データベースのバックアップのリストア

第2章 ヒストリカルデータベース

- 古いデータのアーカイブ化

- 使用しないデータの削除

- バックアップデータのリストア

第1章 GroundWork Monitor Open Source 構成データベース

構成データベースのバックアップ

インタフェースから、Commit を実行するたびに Monarch データベースをバックアップすることができます。Monarch のナビゲーション画面から Control>Commit のオプションを投入した時、Abort、Backup と Commit が表示されます。Backup オプションを選んでください。これで、`/usr/local/groundwork/monarch/backup` ディレクトリの下に、Monarch データベースと Nagios 構成ファイルのバックアップが作られます。バックアップ終了後、メッセージが表示されますので、commit で最後まで行いたくない場合、abort を選ぶことができます。下記の手順のコマンドラインを使って、Monarch データベースのバックアップを手動で行うことができます。」

1. Monarch システムへのターミナルセッションを開き、root としてログインします。
2. 下記のコマンドを発行し、バックアップを作ります：
`mysqldump monarch > monarch_backup_file.sql`
この `monarch_backup_file.sql` は、データベースのバックアップファイルの名前です。

構成データベースのバックアップのリストア

1. Monarch システムへのターミナルセッションを開き、root としてログインします。
2. ディレクトリを移動します：
`cd /usr/local/groundwork/monarch/backup`
3. バックアップが行われた日時のタイムスタンプのディレクトリへ移ります。たとえば：
`2005-08-26_11-53-06`
4. `monarch-"timestamp".sql` という名前のファイルがあるはずですが。("timestamp"は、上記のディレクトリと同じ様な表現)
5. 現在の Monarch データベースを下記のコマンドでドロップします (mysqlの root パスワードがない前提)。root パスワードがあれば、`-u root -p password` を `mysql` コマンドに付加します。
`mysql`
`drop database monarch;`
`create database monarch;`
`exit`
6. 古いデータベースをリストアします。OSのプロンプトで、下記のコマンドを投入します：
`cat monarch-"timestamp".sql | mysql monarch`

第2章 ヒストリカルデータベース

GroundWork Monitor Professional が使用する運用データベースは下記のとおりです：

- sv - 状態とイベントを収容するデータベース
- dashboard - インサイトレポートのデータベースで使用されるヒストリカル的な情報を収容

古いデータのアーカイブ化

古いデータをアーカイブにするには、mysqldump コマンドを使ってデータベースのダンプファイルを作ります。バックアップファイルを作るため、下記のコマンドを投入します：

- `mysqldump sv > sv_backup_file.sql`
- `mysqldump dashboard > dashboard_backup_file.sql`

上記の `sv_backup_file.sql` と `dashboard_backup_file.sql` はデータベースバックアップファイルの名前です。

使用しないデータの削除

データベース全体を削除し、初期設定にリセットするためには、下記の手順を使用します。

1. `cd /usr/local/groundwork/databases/load` を投入します。このディレクトリに、データベース作成のSQLステートメントがあります。
2. `mysql` を投入します。mysql プロンプトが表示されるでしょう。
3. `drop database dashboard;` を投入します。これにより、レポートデータベースがクリアされます。
4. `create database dashboard;` を投入します。
5. `drop database sv;` を投入します。これにより、ステータスデータベースがクリアされます。
6. `create database sv;` を投入します。
7. `exit` を投入します。 You should be back at the system command prompt.
8. コマンド `cat dashboard_nagios_create.sql | mysql dashboard` を投入します。
9. コマンド `cat sv.sql | mysql sv` を投入します。

ある日付よりも古いイベントメッセージを `sv` データベースから選択的に削除するには、下記の手順に従ってください。
注意:これはあなたがイベントロギングをONにしている場合にのみ有効です。デフォルトでは、これはOFFになっています。

1. `mysql` を投入します。mysql プロンプトが表示されるでしょう。
2. `use sv` を投入します。
3. `delete from LogMessage where ReportDate is less than '2006-01-01 00:00:00';` を投入します。
この例は、2006年1月1日以前のイベントメッセージを削除します。
4. `exit` を投入します。

ある日付よりも古いレポートエントリを `dashboard` データベースから選択的に削除するには、下記の手順に従ってください。

1. `mysql` を投入します。mysql プロンプトが表示されるでしょう。
2. `use dashboard` を投入します。
3. `delete from measurements where TIMESTAMP is less than '2006-01-01';` を投入します。
この例は、2006年1月1日以前の毎日のアラートエントリを削除します。
4. `delete from host_availability where TIMESTAMP is less than '2006-01-01';` を投入します。
この例は、2006年1月1日以前の毎日のホスト稼働率エントリを削除します。
5. `delete from Service_availability where TIMESTAMP is less than '2006-01-01';` を投入します。
この例は、2006年1月1日以前の毎日のサービス稼働率エントリを削除します。
6. `delete from hostgroup_host_availability where TIMESTAMP is less than '2006-01-01';` を投入します。
この例は、2006年1月1日以前の毎日のホストグループ稼働率エントリを削除します。
7. `delete from hostgroup_service_availability where TIMESTAMP is less than '2006-01-01';` を投入します。
この例は、2006年1月1日以前の毎日のサービス稼働率エントリを削除します。

8. `exit` を投入します。

バックアップデータのリストア

以前にバックアップしたデータベースをリストアするには、下記の手順を使います。現在のデータベースをクリアしてから、バックアップしたデータベースをリストアしたい場合は、上で述べたように、データベースのドロップと作成を行います。

1. `cat sv_backup_file.sql | mysql sv` を投入します。
これは GWCollageDB データベースをバックアップファイル `sv_backup_file.sql` からリストアします。
2. `cat dashboard_backup_file.sql | mysql dashboard` を投入します。
これは、dashboard データベースをバックアップファイル `dashboard_backup_file.sql` からリストアします。

追補1. Monarchを使ってサービスを構成設定する

目次

- どうやって、新しいサービスを定義するか？
- どうやって、サービステンプレートを作成するか？
- どうやって、サービス依存性テンプレートを構成設定するか？
- どうやって、拡張サービス情報テンプレートを作成するか？
- どうやって、新しいサービスグループを作成するか？

Monarchを使ったサービス構成設定： どうやって、新しいサービスを定義するか？

ステップ1 – New Service 画面

1. Configuration メニューオプションから Services を選びます。
2. New を選びます。
3. New Service画面で、プロパティを入力します。(項目の一覧と説明は、表 5.4.1を参照してください)
4. Add を選び、Manage Service 画面に移ります。

図 5.4.1. New Service 画面

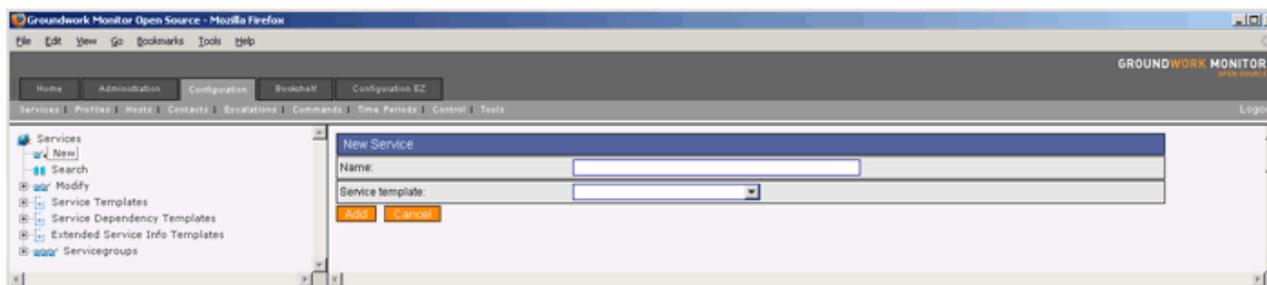


表 5.4.1 New Service

Name	[必須] 新しいサービスの名前。例: check-disk-sda1。 注: Monarch と GroundWork Monitor はホストとサービス名にバックスラッシュをサポートしていません。
Service Template	[必須] このオプションは、サービス名定義のデフォルトテンプレートをセットします。 例: generic-service.

ステップ2 – Service Detail タブ

Manage Service 画面は、Service Detail、Service Check、Service Dependencies、Services Profiles、および Apply Hosts の5つのタブが組み込まれています。

これらのサービス管理オプションについて、この章で説明します。

1. Service Detail タブの中で、プロパティを入力します。(項目の一覧と説明は、表 5.4.2 を参照してください)
2. 新しいサービスを追加するため、Save を選びます。次の画面でサービスが保存されたとのメッセージを受けます。Close を選びます。Delete は、現在のサービスを削除し、Rename は現在のサービスの名前を変更し、Close はセーブしないで現在のサービスを終了します。

図 5.4.2. Service Detail タブ画面

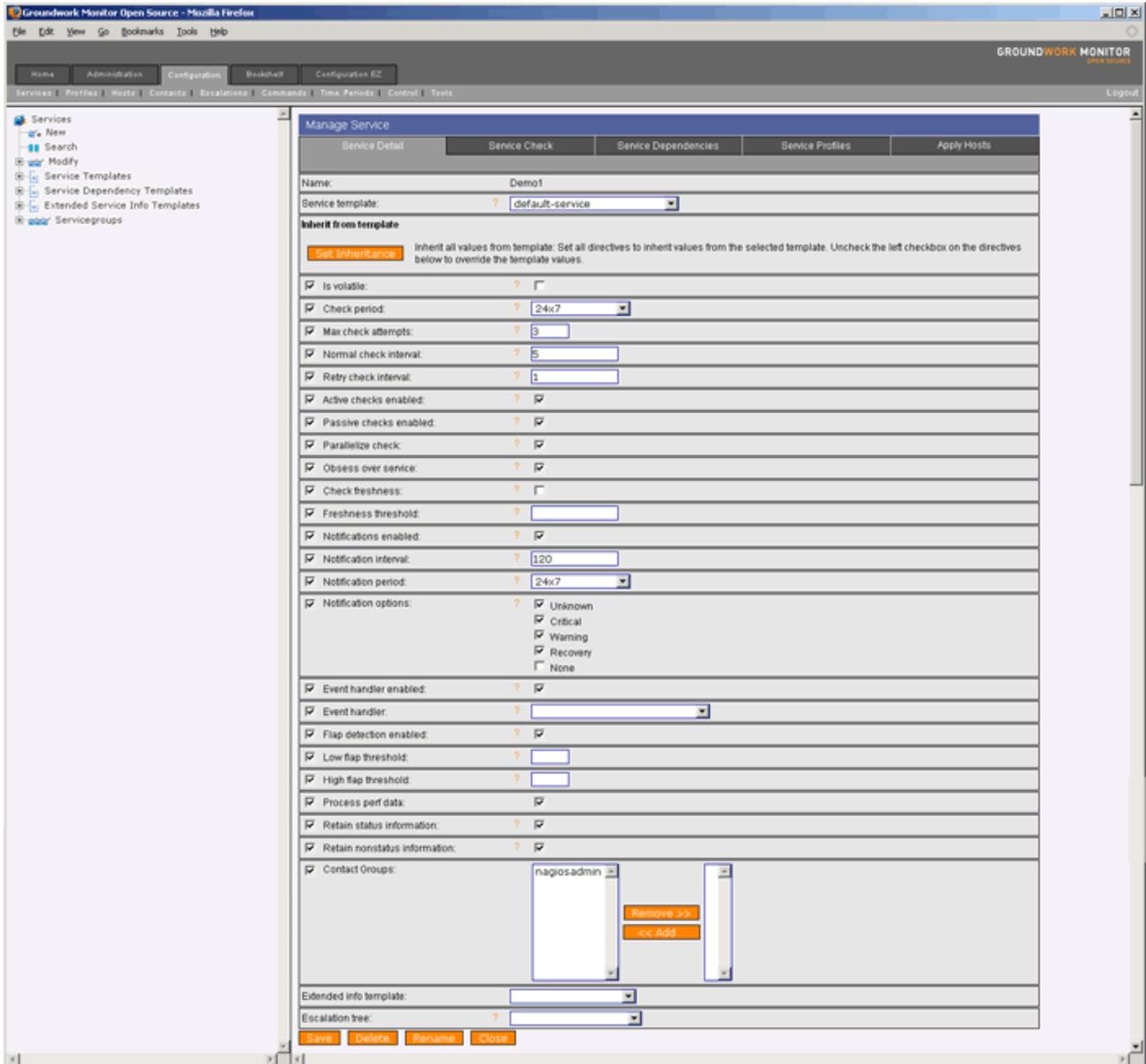


表 5.4.2 Service Detail

Name	前の画面で入力したサービスの名前
Service Template	[必須] 前画面で選択されたこのサービスに最も適したサービステンプレート。ここで選び直してよいです。
Inherit All Values from Template	[オプション] チェックすると選んだテンプレートのすべてのディレクティブが継承されます。チェックしないと、これ以下の値がテンプレートの値に上書きされます。
Is Volatile	[必須] このディレクティブは、このサービスがVolatile(変化が激しい)かどうかを示すのに使われます。サービスは通常、Volatileではありません。チェックしないと、サービスは、Volatileでない、チェックするとサービスはVolatile であると示します。
Check Period	[必須] このディレクティブは、このサービスのアクティブチェックが実施される時間帯を区別する名称を指定するのに使います。
Max Check Attempts	[必須] このディレクティブは、OK状態ではない状態値が返ってきたときに Nagios が何回サービスチェックコマンドをリトライするかを指定するのに使います。この値を 1

	にセットしたら、Nagiosはサービスチェックのリトライをせずに、アラートを発行します。
Normal Check Interval	[必須] このディレクティブは、サービスの次の定期チェックまで待つタイムユニットの数を定義するのに使います。定期チェックは、サービスがOK状態やサービスがOKでない状態だがすでに max_attempts number 回数だけ再チェックした場合にスケジュールされます。あなたが interval_length ディレクティブのデフォルトの値 60 を変更していない限り、この数値は分を意味します。
Retry Check Interval	[必須] このディレクティブは、次のサービス再チェックまで待つタイムユニットの数を定義するのに使います。サービスは、OKでない状態に変わったときに、リトライ間隔を置いて再スケジュールされます。ステータス変化なしに、サービスの再チェックが max_attempts 回行われた場合、スケジュールは check_interval で定義された通常のレートに戻ります。あなたが interval_length ディレクティブのデフォルトの値 60 を変更していない限り、この数値は分を意味します。
Active Checks Enabled	[オプション] このディレクティブは、このサービスのアクティブチェックが有効かどうか決定するために使用します。チェックなしはサービスチェック無効、チェックはサービスチェックを有効にします。
Passive Checks Enabled	[オプション] このディレクティブは、このサービスのパッシブチェックが有効かどうか決定するために使用します。チェックなしはパッシブサービスチェック無効、チェックはパッシブサービスチェックを有効にします。
Parallelize Check	[オプション] このディレクティブは、サービスチェックが並行動作させられるかどうかを判断するために使われます。デフォルトでは、サービスチェックは並行動作可能されます。サービスの並行チェックを無効にすると、重大なパフォーマンスの問題が起る可能性があります。チェックしないと、サービスチェックは並行動作しません(警告が表示されます)。チェックするとサービスチェックは並行動作します。
Obsess Over Service	このディレクティブは Nagios が、サービスチェック結果に obsess over (取り付かれる)し、あなたが定義した強迫観念的で強制的なサービスプロセスコマンドを実行するかどうかを判断します。このオプションは、分散監視を実施する場合に便利です。分散監視を行わない場合は、これを有効にしないでください。チェックすると有効になります。
Check Freshness	[オプション] このディレクティブは、新鮮度チェックがこのサービスで有効かどうかを定義するのに使います。チェックすると新鮮度チェックが有効になります。
Freshness Threshold	[オプション] このディレクティブは、このサービスの新鮮度チェックのしきい値(秒)を指定するのに使います。このディレクティブの値を 0 にセットすると、Nagios は使用する新鮮度のしきい値を自動的に決めます。
Notifications Enabled	[オプション] このディレクティブは、このサービスのための通知が有効かどうかを判断するために使用します。チェックなしは、サービス通知を無効。チェックは、サービス通知有効にする。
Notification Interval	[必須] このディレクティブは、このサービスが、まだOKでない状態であることを、コンタク(通知先)トへの再通知を行うまでに待つ「タイムユニット」の数を定義するのに使います。あなたが interval_length ディレクティブのデフォルト値 60 を変更していない限り、この数値は分を意味します。この値を 0 にセットすると、Nagios は、このテンプレートを使ったホストについて問題は、再通知しません - 状態変化がない限り、問題の通知は一度だけ送出されます。
Notification Period	[必須] このディレクティブは、このサービスについての通知がコンタク(通知先)に送られる時間帯のショートネームを指定するのに使います。時間帯の範囲にない時間のは通知は送出されません。
Notification Options	[必須] このディレクティブは、サービスに対する通知をいつ送るかを判断するのに使います。適用されるオプションは下記のひとつか、組み合わせです: Unknown をチェックしたら UNKNOWN (未知)状態では通知を送り、Critical チェックしたら CRITICAL (クリティカル)状態では通知を送り、Warning をチェックしたら WARNING

	(ワーニング)状態で通知を送り、Recovery をチェックしたら OK(正常)状態に復旧したら通知を送ります。もし、n (none) をオプションとして指定したら、サービス通知は送出されません。
Event Handler Enabled	[オプション] このディレクティブは、このサービスのイベントハンドラが有効かどうかを判断するために使用します。チェックなしはイベントハンドラを無効、チェックはイベントハンドラを有効にします。
Event Handler	[オプション] このディレクティブは、ホストの状態の変化が検出された時(つまり、停止もしくは復旧した時)かならず実行されるコマンドのショートネームを指定するのに使います。イベントハンドラコマンドを最大何回実行できかについては event_handler_timeout オプションで制御します。
Flap Detection Enabled	[オプション] このディレクティブは、このサービスのフラップ検出を有効にするかどうかを決めるのに使用します。チェックなしはサービスフラップ検出を無効にし、チェックはサービスフラップ検出を有効にします。
Low Flap Threshold	[オプション] このディレクティブは、このサービスのフラップ検出の低い値を指定するのに使用します。このディレクティブの値を 0 に設定したら、low_service_flap_threshold ディレクティブで指定されている、プログラム全体に指定されている値を使用します。
High Flap Threshold	[オプション] このディレクティブは、このサービスのフラップ検出の高い値を指定するのに使用します。このディレクティブの値を 0 に設定したら、high_service_flap_threshold ディレクティブで指定されている、プログラム全体に指定されている値を使用します。
Process Perf Data	[オプション] このディレクティブは、このテンプレートを使ったサービスのパフォーマンスデータ処理を行うかどうかを決めるのに使用します。チェックなしはパフォーマンスデータ処理を無効にし、チェックはパフォーマンスデータ処理を有効にします。
Retain status information	[オプション] このディレクティブは、サービスに関するステータス関連情報がプログラムの再起動以降も保持されるかどうかを決めるのに使用します。これは、retain_state_information ディレクティブを使って状態保持を有効にしたときにだけ便利です。チェックなしはステータス情報保持を無効にし、チェックはステータス情報保持を有効にします。
Retain Non-status Information	[オプション] このディレクティブは、サービスに関する非ステータス情報がプログラムの再起動以降も保持されるかどうかを決めるのに使用します。これは、retain_state_information ディレクティブを使って状態保持を有効にしたときにだけ便利です。チェックなしは非ステータス情報保持を無効にし、チェックは非ステータス情報保持を有効にします。
Contact Groups	このサービスのために通知を受けるコンタクトグループを選択します。
Extended Info Template	[オプション] このサービス名のための拡張サービス情報を定義します。そのテンプレートは、このサービス定義をアサインしたホストのための Nagios 内で見られるアイコンとURLリンクをコントロールします。
Escalation Tree	[オプション] このサービスに最適なエスカレーションツリーを選びます。通知の拡大(つまり、同じイベントへの複数通知)を回避するため、サービスにアサインされたサービスエスカレーションは、ホストやホストグループにアサインするべきではありません。

Monarchを使ったサービス構成設定： どうやって、サービステンプレートを作成するか？

ステップ1 – New Service Template 画面

1. Configuration メニューオプションから Services を選びます。
2. Service Templates を選びます。
3. New を選びます。
4. New Service Template 画面で、プロパティを入力します。
(項目の一覧と説明は、表 5.4.3を参照してください)
5. テンプレート名を追加するため Add を選び、テンプレートを定義するため次画面に移ります。

図 5.4.3. New Service Template画面

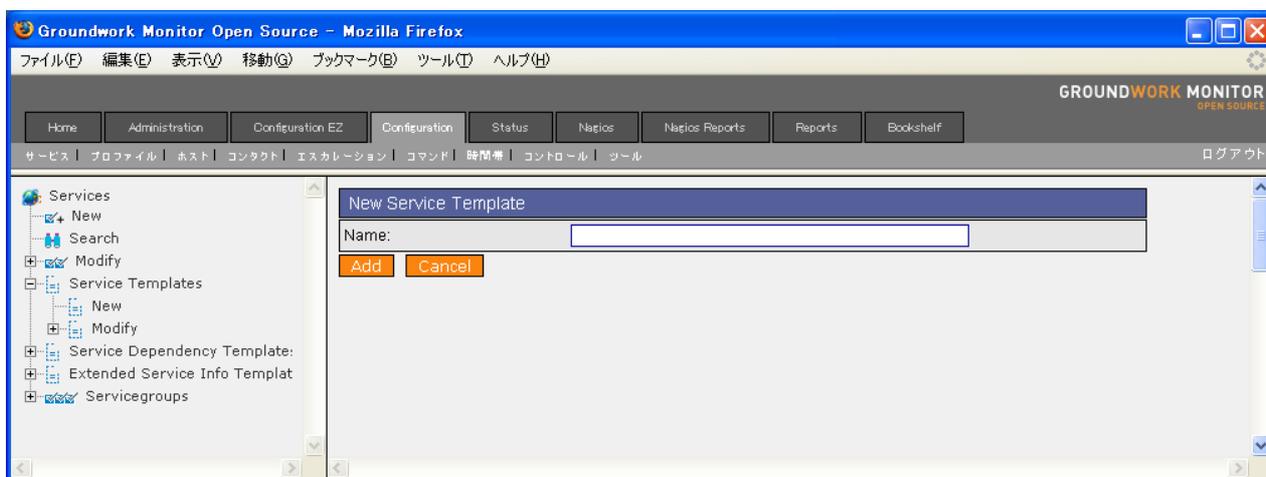


表 5.4.3 New Service Template

Name	新しいサービスの名前
------	------------

ステップ2 - Manage Service Template 画面 - Service Detail タブ

1. Service Detail タブの中で、プロパティを入力します。(項目の一覧と説明は、表 5.4.4 を参照してください)
2. 新しいサービステンプレートを追加するため、Save を選びます。Delete は現在のサービステンプレートを削除し、Rename は現在のサービステンプレートの名前を変更し、Close はセーブしないで現在のサービステンプレートを終了します。重要な注意事項: 各タブにはそれぞれ保存(save)コマンドがあります。各タブの間で Save を選んでおかないと、入力したものを失ってしまうでしょう。

図 5.4.4. Service Detail 画面

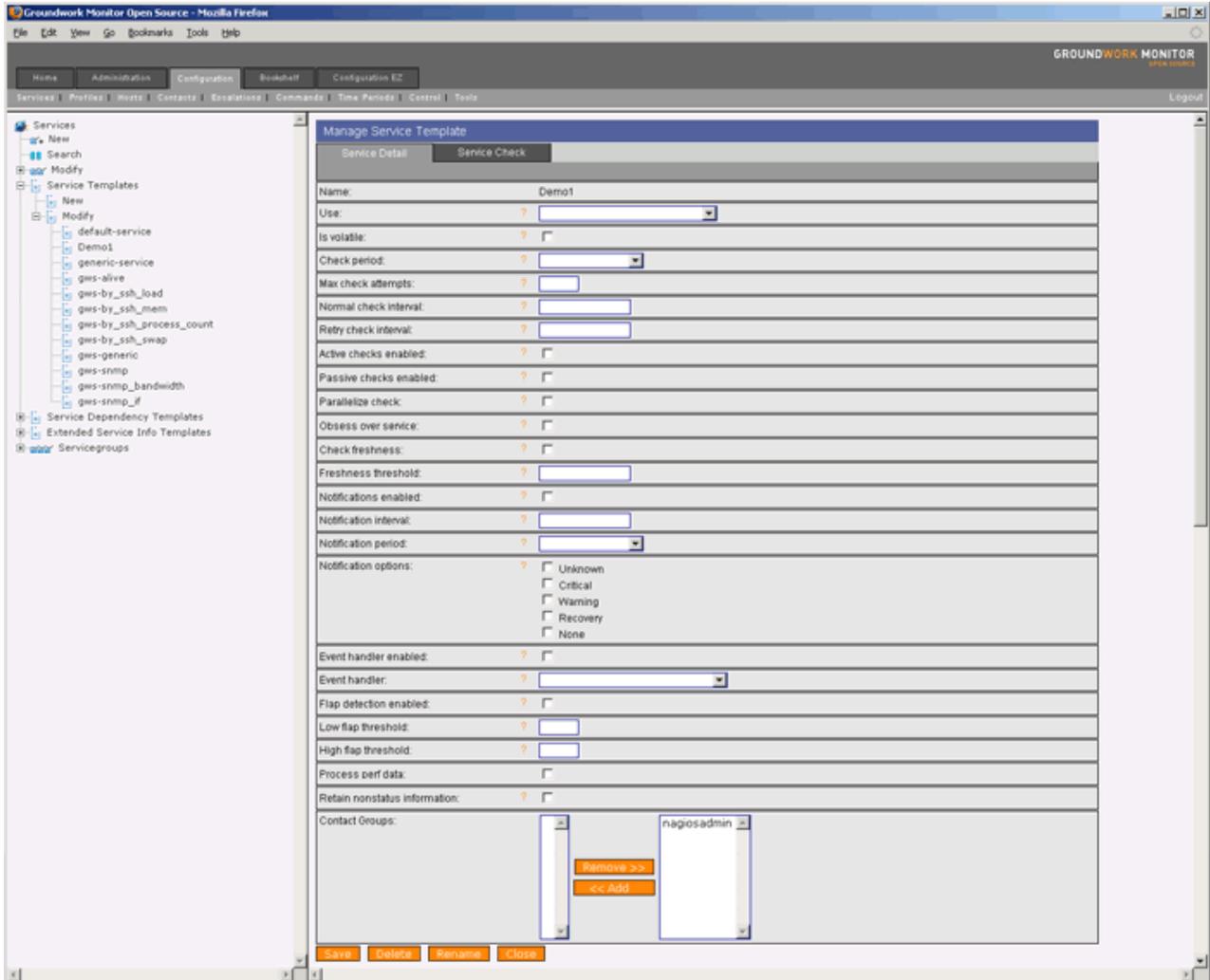


表 5.4.4 - Service Detail

Use	[必須] このサービスに最も適したサービステンプレートを選びます。この画面でのディレクティブ変更はテンプレートの値を上書きします。
Is Volatile	このディレクティブは、このサービスがVolatile(変化が激しい)かどうかを示すのに使われます。サービスは通常、Volatileではありません。チェックしないと、サービスは、Volatileでない、チェックするとサービスはVolatile であると示します。
Check Period	[必須] このディレクティブは、このサービスのアクティブチェックが実施される時間帯を区別する名称を指定するのに使います。
Max Check Attempts	[必須] このディレクティブは、OK状態ではない状態値が返ってきたときに Nagios が何回サービスチェックコマンドをリトライするかを指定するのに使います。この値を 1

	にセットしたら、Nagiosはサービスチェックのリトライをせずに、アラートを発行します。
Normal Check Interval	[必須] このディレクティブは、サービスの次の定期チェックまで待つタイムユニットの数を定義するのに使います。定期チェックは、サービスがOK状態やサービスがOKでない状態だがすでに max_attempts number 回数だけ再チェックした場合にスケジュールされます。あなたが interval_length ディレクティブのデフォルトの値 60 を変更していない限り、この数値は分を意味します。
Retry Check Interval	[必須] このディレクティブは、次のサービス再チェックまで待つタイムユニットの数を定義するのに使います。サービスは、OKでない状態に変わったときに、リトライ間隔を置いて再スケジュールされます。ステータス変化なしに、サービスの再チェックが max_attempts 回行われた場合、スケジュールは check_interval で定義された通常のレートに戻ります。あなたが interval_length ディレクティブのデフォルトの値 60 を変更していない限り、この数値は分を意味します。
Active Checks Enabled	[オプション] このディレクティブは、このサービスのアクティブチェックが有効かどうか決定するために使用します。チェックなしはサービスチェック無効、チェックはサービスチェックを有効にします。
Passive Checks Enabled	[オプション] このディレクティブは、このサービスのパッシブチェックが有効かどうか決定するために使用します。チェックなしはパッシブサービスチェック無効、チェックはパッシブサービスチェックを有効にします。
Parallelize Check	[オプション] このディレクティブは、サービスチェックが並行動作させられるかどうかを判断するために使われます。デフォルトでは、サービスチェックは並行動作可能されます。サービスの並行チェックを無効にすると、重大なパフォーマンスの問題が起る可能性があります。チェックしないと、サービスチェックは並行動作しません（警告が表示されます）。チェックするとサービスチェックは並行動作します。
Obsess Over Service	このディレクティブは Nagios が、サービスチェック結果に obsess over (取り付かれる) し、あなたが定義した強迫観念的で強制的なサービスプロセスコマンドを実行するかどうかを判断します。このオプションは、分散監視を実施する場合に便利です。分散監視を行わない場合は、これを有効にしないでください。
Check Freshness	[オプション] このディレクティブは、新鮮度チェックがこのサービスで有効かどうかを定義するのに使います。チェックなしは新鮮度チェック無効、チェックすると新鮮度チェックが有効になります。
Freshness Threshold	[オプション] このディレクティブは、このサービスの新鮮度チェックのしきい値(秒)を指定するのに使います。このディレクティブの値を 0 にセットすると、Nagios は使用する新鮮度のしきい値を自動的に決めます。
Notifications Enabled	[オプション] このディレクティブは、このサービスのための通知が有効かどうかを判断するために使用します。チェックなしはサービス通知を無効にし、チェックはサービス通知を有効にします。
Notification Interval	[必須] このディレクティブは、このサービスが、まだOKでない状態であることを、コンタク(通知先)トへの再通知を行うまでに待つ「タイムユニット」の数を定義するのに使います。あなたが interval_length ディレクティブのデフォルト値 60 を変更していない限り、この数値は分を意味します。この値を 0 にセットすると、Nagios は、このテンプレートを使ったホストについて問題は、再通知しません - 状態変化がない限り、問題の通知は一度だけ送出されます。
Notification Period	[必須] このディレクティブは、このサービスについての通知がコンタク(通知先)に送られる時間帯のショートネームを指定するのに使います。時間帯の範囲にない時間のはサービス通知は送出されません。
Notification Options	[必須] このディレクティブは、サービスに対する通知をいつ送るかを判断するのに使います。適用されるオプションは下記のひとつか、組み合わせです: Unknown をチェックしたら UNKNOWN (未知)状態 で通知を送り、Critical チェックしたら CRITICAL (クリティカル)状態 で通知を送り、Warning をチェックしたら WARNING

	(ワーニング)状態で通知を送り、Recovery をチェックしたら OK(正常)状態に復旧したら通知を送ります。もし、n (none) をオプションとして指定したら、サービス通知は送出されません。
Event Handler Enabled	[オプション] このディレクティブは、このサービスのイベントハンドラが有効かどうかを判断するために使用します。チェックなしはイベントハンドラを無効、チェックはイベントハンドラを有効にします。
Event Handler	[オプション] このディレクティブは、ホストの状態の変化が検出された時(つまり、停止もしくは復旧した時)、かならず実行されるコマンドのショートネームを指定するのに使います。イベントハンドラコマンドを最大何回実行できかについては event_handler_timeout オプションで制御します。
Flap detection enabled	[オプション] このディレクティブは、このサービスのフラップ検出を有効にするかどうかを決めるのに使用します。チェックなしはサービスフラップ検出を無効にし、チェックはサービスフラップ検出を有効にします。
Low Flap Threshold	[オプション] このディレクティブは、このサービスのフラップ検出の低い値を指定するのに使用します。このディレクティブの値を 0 に設定したら、low_service_flap_threshold ディレクティブで指定されている、プログラム全体に指定されている値を使用します。
High Flap Threshold	[オプション] このディレクティブは、このサービスのフラップ検出の高い値を指定するのに使用します。このディレクティブの値を 0 に設定したら、high_service_flap_threshold ディレクティブで指定されている、プログラム全体に指定されている値を使用します。
Process Perf Data	[オプション] このディレクティブは、このテンプレートを使ったサービスのパフォーマンスデータ処理を行うかどうかを決めるのに使用します。チェックなしはパフォーマンスデータ処理を無効にし、チェックはパフォーマンスデータ処理を有効にします。
Retain status information	[オプション] このディレクティブは、サービスに関するステータス関連情報がプログラムの再起動以降も保持されるかどうかを決めるのに使用します。これは、retain_state_information ディレクティブを使って状態保持を有効にしたときにだけ便利です。チェックなしはステータス情報保持を無効にし、チェックはステータス情報保持を有効にします。
Retain Non-status Information	[オプション] このディレクティブは、サービスに関する非ステータス情報がプログラムの再起動以降も保持されるかどうかを決めるのに使用します。これは、retain_state_information ディレクティブを使って状態保持を有効にしたときにだけ便利です。チェックなしは非ステータス情報保持を無効にし、チェックは非ステータス情報保持を有効にします。
Contact Groups	このサービスのために通知を受けるコンタクトグループを選択します。

ステップ3 – Service Check タブ

1. 次に、Service Check タブを選び、このテンプレートのためのサービスチェックを定義します。
2. Service Check タブ画面で、プロパティを入力します。(項目の一覧と説明は、表 5.4.5を参照してください)
3. チェックコマンドを保存するため Save を選びます。保存したことの確認メッセージが表示されます。Close を選びます。

図 5.4.5. Service Check

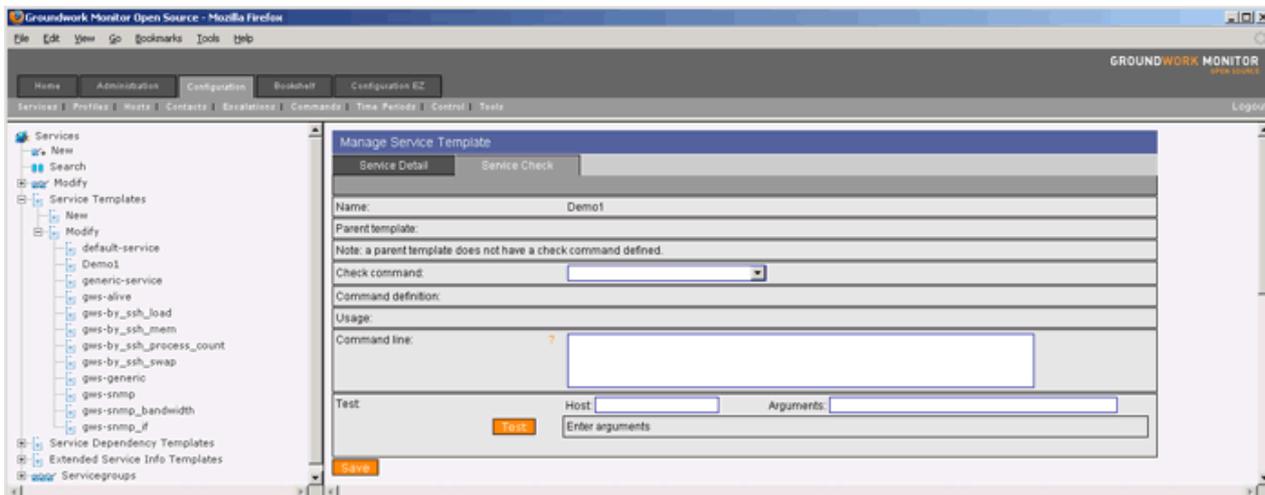


表 5.4.5. Service Check

Check Command	これは、Nagios がサービスの状態をチェックするために実行するコマンドです。使用できる三つのコマンドフォーマットがあります：コマンド、引数付きコマンド、およびコマンドラインです。
Command Definition	チェックコマンド選択後、コマンド定義はここに表示されます。
Usage	コマンドの使用パラメータ
Command Line	チェックコマンドが引数を要求する場合、チェックコマンドと引数を「！」文字で区分して、ここに入力します。例：check_disk!/dev/sda1
Test	コマンドラインのホストと引数を入力します。“Test” ボタンをクリックすると、コマンド定義をパースして、マクロの置換、Hostボックスのホスト名の挿入を行い、結果として作成されたコマンドをシェルコマンドラインに入れて実行し、実効の返された結果を右側のボックスに表示します。

Monarchを使ったサービス構成設定:

どうやって、サービス依存性テンプレートを構成設定するか？

ステップ1 - 新しいサービステンプレート

1. Configuration メニューオプションから Services を選びます。
2. Service Dependency Templates を選びます。
3. New を選びます。
4. Service Dependency Template 画面で、プロパティを入力します。
(項目の一覧と説明は、表 5.4.6 を参照してください)
5. サービス依存性テンプレートをセーブするため Save を選びます。Delete は現在のテンプレートを削除し、Rename は現在のテンプレートの名前を変更します。次の画面で、サービス依存性テンプレートがセーブされたことを知らせるでしょう。Continue を選びます。

図 5.4.6. Service Dependency Template 画面

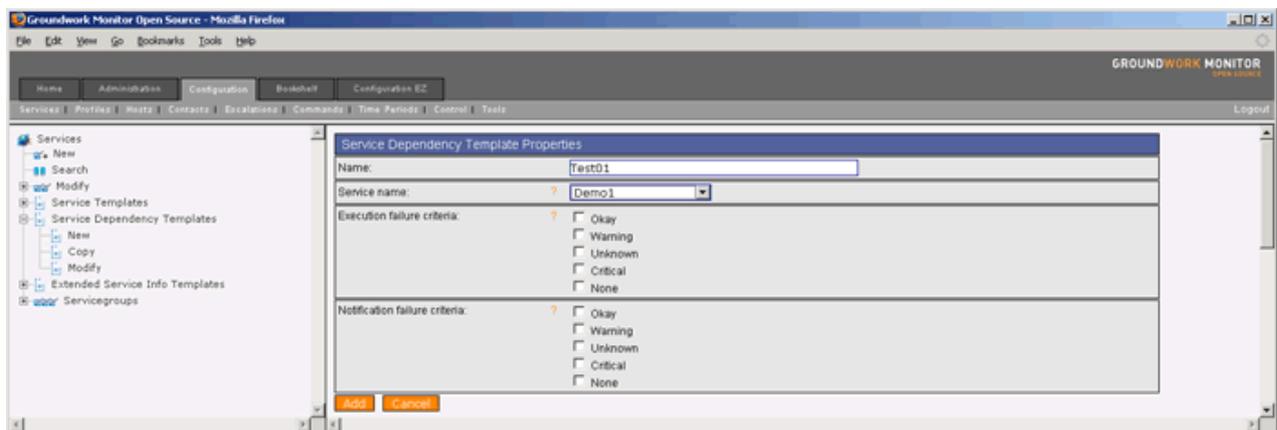


表 5.4.6 Service Dependency Template

Name	[必須] サービス依存性テンプレートの名前
Service Name	[必須] これは、親サービスの名前です
Execution failure criteria	<p>[オプション] このディレクティブは、いつ依存元のサービスが実行されるべきでないかを判断する基準を指定するのに使います。もし、依存しているサービスが、我々が指定したどれかひとつの障害状態になったら、依存元のサービスは実行されません。適用されるオプションは下記のひとつか、組み合わせです: Okay をチェックするとOK(正常)状態での障害、Warning をチェックするとWARNING(ワーニング)状態での障害、UnknownをチェックするとUNKNOWN(不明)状態での障害、Critical をチェックするとCRITICAL(クリティカル)状態での障害です。もし、Noneを選ぶと、実行依存は決して失敗せず、依存サービスのチェックは常に実行されます。例: このフィールドの OK、CRITICALやUNKNOWNを指定すると、依存先のサービスが OK、CRITICALやUNKNOWNのどれになっても、依存元のサービスは実行されません。</p>
Notification failure criteria	<p>[オプション] このディレクティブは、いつ依存元サービスについての通知を送出するかを判断基準を定義するのに使います。もし、サービスが、我々が指定したどれかひとつの障害状態になったら、依存元サービスの通知がコンタクト(連絡先)に送出されます。適用されるオプションは下記のひとつか、組み合わせです: Okay をチェックするとOK(正常)状態での障害、Warning をチェックするとWARNING(ワーニング)状態での障害、UnknownをチェックするとUNKNOWN(不明)状態での障害、Critical をチェックするとCRITICAL(クリティカル)状態での障害です。もし、Noneをオプションとして指定したら、依存性の通知は決して失敗せず、依存元サービスの通知が常に送出されます。例: このフィールドの Warning をチェックすると、依存先(そのホストが依存している)ホストがWARNING状態だった場合、依存元ホストの通知は送出されません。</p>

Monarchを使ったサービス構成設定： どうやって、拡張サービス情報テンプレートを作成するか？

ステップ1 – Extended Service Info Template 画面

1. Configuration メニューオプションから Services を選びます。
2. Extended Service Info Templates を選びます。
3. New を選びます。
4. Extended Service Info Template画面で、プロパティを入力します。
(項目の一覧と説明は、表 5.4.7 を参照してください)
5. Add を選び、新しい拡張サービス情報テンプレートを追加します。
6. 拡張サービス情報テンプレートがサブされたことを知らされます。Continue を選びます。

図 5.4.7. Extended Service Info Template 画面

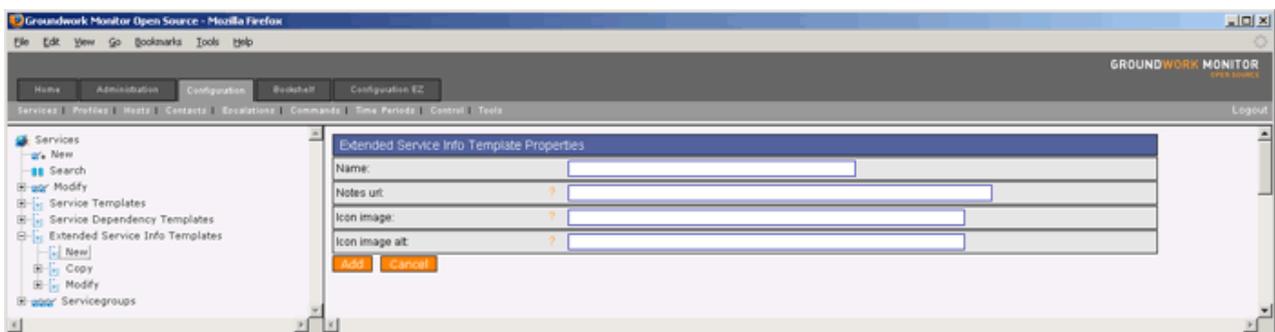


表 5.4.7 Extended Service Info Template

Name	[必須] 拡張サービス情報テンプレートの名前
Notes URL	[オプション] 他のサポートスタッフに、ホスト、たとえば緊急連絡方法などの、より多くの情報を提供するURLを定義します。
Icon Image	[オプション] この変数は、ホストに関連付けられたイメージの名前を定義するのに使います。このイメージはステータスと拡張情報の各CGI の中で表示されます。
Icon Image Alt	[オプション] この変数は、Icon Imageで指定されたイメージのALTタグの中で使われるオプションな文字列を定義するのに使われます。

Monarchを使ったサービス構成設定： どうやって、新しいサービスグループを作成するか？

ステップ1 – Service Group 画面

1. Configuration メニューオプションから Services を選びます。
2. Service Groups を選びます。
3. Select New を選びます。
4. Service Group 画面で、プロパティを入力します。(項目の一覧と説明は、表 5.4.8 を参照してください)
5. Add を選び、Service Group 画面に移ります。
6. Host ドロップダウンボックスを選んで、ホストを選択します。利用可能なサービスがリストされるでしょう。
7. 追加するためにサービスを選び Add Service(s) をクリックします。上のホストとサービスのカラムにあなたの選んだものが表示されます。選んだサービスを削除するには、削除する行に対応したXをクリックしてください。
8. Service ドロップダウンボックスを選び、サービスを選択すると、使用可能なホストがリストされるでしょう。
9. 追加するため、ホストを選んで Add Host(s) をクリックします。上のホストとサービスのカラムにあなたの選んだものが表示されます。選んだサービスを削除するには、削除する行に対応した X をクリックしてください。
10. サービスグループをセーブするため、Save を選びます。Delete は現在のサービスグループを削除し、Rename は現在のサービスグループの名前を変更します。次の画面で、サービス依存性テンプレートが更新されたことを知らさるでしょう。

図 5.4.8. Service Groups

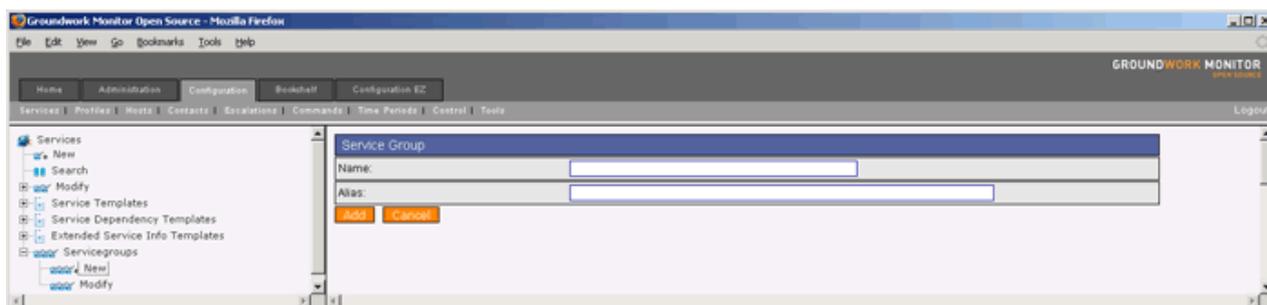


表 5.4.8 Service Groups

Name	[必須] サービスグループの名前。例: dbservices
Alias	[必須] このディレクティブは、サービスグループを識別するための長い名前か説明を定義するのに使います。これは、特定のサービスグループをより容易に識別できるようにするため提供されます。例: Database Services
Host/Service	[必須] これは、このグループに含まれるサービス説明(と、それらの関連するホスト名)のリストです。

追補2. Monarchを使ってプロフィールを構成設定する

目次

どうやって、ホストプロフィールを定義するか？

どうやって、サービスプロフィールを定義するか？

どうやって、プロフィールをインポートするか？

Monarchを使ったプロファイル構成設定： どうやって、ホストプロファイルを定義するか？

ステップ1 – New Host Profile画面

1. Configuration メニューオプションから Profiles を選びます。
2. Host Profiles を選びます。
3. New を選びます。
4. New Host Profile 画面で、プロパティを入力します。(項目の一覧と説明は、表 5.5.1を参照してください)
5. Select Add to go to the Host Profile 画面に行くため、Add を選びます。

図 5.5.1. New Host Profile

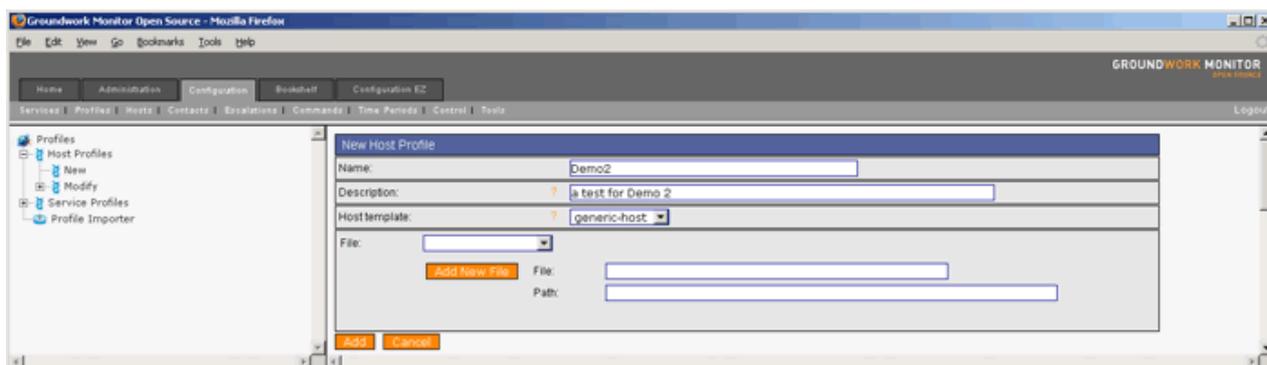


表 5.5.1 New Host Profile

Name	[必須] ホストプロファイルの名前
Description	[オプション] コメントや指示をここに格納します。
Host Template	[必須] このホストプロファイルに適切なホストテンプレートを選びます。
File	[必須] このプロファイルに関連つけたホストのためのデフォルトファイルを選びます。新しいファイルを追加することもできます。

ステップ2- Host Profile画面 – Host Detail タブ

1. Host Detail タブ画面中でプロパティを入力します。(項目の一覧と説明は、表 5.5.2 を参照してください)
2. 新しいホストプロファイルを保存するため Save を選びます。次の画面でプロファイルがセーブされたメッセージが表示されるでしょう。Close を選びます。Delete は、現在のホストプロファイルを削除し、Rename は現在のホストプロファイルの名前を変更し、Export は現在のホストプロファイルを /tmp/host_profile_<hostprofilename>.xml ディレクトリにセーブし、Close はセーブしないでホストプロファイル画面を終了します。
3. Host Profile画面内に一覧になっている他のタブ (Host Detail、Parents、Host Groups、Escalations、Service Profiles、Assing Hosts、Assign Host Groups と Apply) は、オプションなものです。Apply オプションは、どのようにプロファイルが使われるか決めます。プロファイルから親やホストグループ管理するかどうか選ぶことができます。また、適用された設定はセーブされるので、そのプロファイルがどのように適用されたかは、ひとつのセッションから次へと同様に残ります。

図 5.5.2. Host Detail タブ画面

Groundwork Monitor Open Source - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

GROUNDWORK MONITOR
www.groundwork.com

Home Administration Configuration Breakshot Configuration EZ

Services Profiles Hosts Contacts Escalations Commands Time Periods Control Tests Logout

Profiles

- Host Profiles
 - New
 - Modify
 - host_profile_service_ping
 - host_profile_snmp_network
 - host_profile_ssh_unix
- Service Profiles
- Profile Importer

Host Profile

Host Data | Parents | Hostgroups | Escalations | Service Profiles | Assign Hosts | Assign Hostgroups | Apply

Name: host_profile_service_ping

Description: Host profile for ping

Host template: generic-host

Inherit from template

Set Inheritance Inherit all values from template. Set all directives to inherit values from the selected template. Uncheck the left checkbox on the directives below to override the template values.

Process performance data

Retain status information

Flap detection enabled

Low flap threshold

High flap threshold

Retain nonstatus information

Active checks enabled

Passive checks enabled

Obsess over host

Check freshness

Freshness threshold

Check command

Max check attempts: 3

Check interval

Event handler enabled

Event handler

Notifications enabled

Notification interval: 60

Notification period: none

Notification options:

- Down
- Unreachable
- Recover
- Flapping

Stalking options:

- Down
- Up
- Unreachable

Contact groups: nagiosadmin

Extended host info template:

File: hosts.cfg

File: Path:

表 5.5.2 Host Detail

Name	[必須] ホストプロファイルの名前
Description	[オプション] コメントや指示をここに記述します。
Host Template	[必須] このホストに最も適したホストテンプレートを選びます。注意: 下記のディレクトリの継承 (Inheritance: 左のチェックボックス) について。テンプレートの値を上書きする場合は、左側のチェックボックスのチェックをはずしてください。
Inherit All Values From Template	チェックする = すべてのディレクティブの選択したテンプレートから継承した値をセットします。下記で、左のチェックボックスをチェックしていないディレクティブは、テンプレートの値を上書きします。
Process Performance Data	[オプション] このディレクティブは、このテンプレートを使ったホストのパフォーマンスデータの処理が有効かどうかを決めるのに使います。 値: チェックなしはパフォーマンスデータ処理を無効にし、チェックはパフォーマンスデータ処理を有効にします。
Retain Status Information	[オプション] このディレクティブは、ホストに関するステータス関連情報がプログラムの再起動以降も保持されるかどうかを決めるのに使います。これは、retain_state_information ディレクティブを使って状態保持を有効にしたときにだけ便利です。 値: チェックなしはステータス情報保持を無効にし、チェックはステータス情報保持を有効にします。
Flap Detection Enabled	[オプション] このディレクティブは、このテンプレートを使ったホストのフラップ検出を有効にするかどうかを決めるのに使います。 チェックなしはホストフラップ検出を無効にし、チェックはホストフラップ検出を有効にします。
Low Flap Threshold	[オプション] このディレクティブは、このテンプレートを使ったホストのフラップ検出の低しきい値を指定するのに使います。 このディレクティブの値を 0 に設定したら、low_host_flap_threshold ディレクティブで指定されている、プログラム全体に指定されている値を使用します。
High Flap Threshold	[オプション] このディレクティブは、このテンプレートを使ったホストのフラップ検出の高しきい値を指定するのに使います。 このディレクティブの値を 0 に設定したら、high_host_flap_threshold ディレクティブで指定されている、プログラム全体に指定されている値を使用します。
Retain Nonstatus Information	[オプション] このディレクティブは、ホストに関する非ステータス情報がプログラムの再起動以降も保持されるかどうかを決めるのに使います。これは、retain_state_information ディレクティブを使って状態保持を有効にしたときにだけ便利です。 チェックなしは非ステータス情報保持を無効にし、チェックは非ステータス情報保持を有効にします。
Active Checks Enabled	このディレクティブは、このホストのアクティブチェック (定期スケジュールやオンデマンド共、) が有効かどうか決定するために使います。 チェックなしはアクティブホストチェックを無効、チェックはアクティブホストチェックを有効にします。
Passive Checks Enabled	このディレクティブは、このホストのパッシブチェックが有効かどうか決定するために使います。 チェックなしはパッシブホストチェックを無効、チェックはパッシブホストチェックを有効にします。
Obsess Over Host	このディレクティブは、このホストのチェックが (Nagiosのメイン構成設定で定義された ochnp_command を使って、取り付かれる (obsessed) かどうかを定義します。 チェックすると有効になります。
Check Freshness	このディレクティブは新鮮度チェックがこのテンプレートを使っているホストで有効かどうかを定義するのに使います。 チェックすると新鮮度チェックが有効になります。
Freshness Threshold	このディレクティブは、このテンプレートを使っているホストの新鮮度チェックのしきい値 (秒) を指定するのに使います。このディレクティブの値を 0 にセットすると、Nagios は使用する新鮮度のしきい値を自動的に決めます。

Check Command	[オプション] このディレクティブは、ホストが稼働しているか停止しているかをチェックするコマンドのショートネームを指定します。一般的に、このコマンドはホストが“生きている”かどうかを ping をそのホストに発行することによって確認します。このコマンドはOK(0) のステータスを戻す必要があり、そうでなければNagiosはそのホストが停止していると判断します。もしこの引数を空欄にした場合、そのホストはチェックされず、Nagiosは常にそのホストが稼働していると想定します。このことは、プリンタなどの頻繁に電源を落とす機器を監視するのに役立ちます。通知コマンドを発行する回数はhost_check_timeoutオプションで制御します。
Max Check Attempts	[必須] このディレクティブは、OK状態ではない状態値が返ってきたときに Nagios が何回ホストチェックコマンドをリトライするかを指定するのに使います。この値を 1 にセットしたら、Nagiosはホストチェックのリトライをしないで、アラートを発行します。注意：もしそのホストの状態をチェックしたくないのであっても、ここの値は最小値の 1 に設定しなくてはなりません。ホストチェックを行わないようにするには、host_check_command オプションを空のままにするだけです。
Check Interval	[オプション] 注意：それが絶対的に必要になったとき以外は、定期チェックを有効にしないで下さい！ ホストチェックは、既に必要なときにオンデマンドで実行されますので、定期的なチェックが必要になる場合はほとんどありません。定期的なホストチェックはパフォーマンスに負の影響を及ぼします。このディレクティブは、ホストの定期チェックの間の「タイムユニット」の数を定義するのに使います。あなたが interval_length ディレクティブのデフォルトの値 60 を変更していない限り、この数値は分を意味します。
Event Handler Enabled	[オプション] このディレクティブは、このテンプレートを使っているホストのイベントハンドラが有効かどうかを判断するために使用します。値：チェックなし、はイベントハンドラを無効。チェックは、イベントハンドラを有効にする。
Event Handler	[オプション] このディレクティブは、ホストの状態の変化が検出された時(つまり、停止もしくは復旧した時)かならず実行されるコマンドのショートネームを指定するのに使います。イベントハンドラコマンドを最大何回実行できかについては event_handler_timeout オプションで制御します。
Notifications Enabled	[オプション] このディレクティブは、ホストのための通知が有効かどうかを判断するために使用します。値：チェックなしはホスト通知を無効。チェックは、ホスト通知有効にする。
Notification Interval	[必須] このディレクティブは、そのサーバがなおダウンや到達不能であることを、コンタク(通知先)トへの再通知を行うまでに待つ「タイムユニット」の数を定義するのに使います。あなたが interval_length ディレクティブのデフォルト値 60 を変更していない限り、この数値は分を意味します。この値を 0 にセットすると、Nagios は、このテンプレートを使ったホストについて問題は再通知しません - 問題の通知は一度だけ送出されます。
Notification Period	[必須] このディレクティブは、このテンプレートを使ったホストについての通知がコンタク(通知先)に送られる時間帯のショートネームを指定するのに使います。時間帯の範囲にない時刻の間にホストが停止し、到達不能や復旧になったら、通知は送出されません。
Notification Options	[必須] このディレクティブは、ホストに対する通知をいつ送るかを判断するのに使います。適用されるオプションは下記のひとつか、組み合わせです： Down checked は停止 (DOWN) 状態で通知を送り、Unreachable checked は到達不能 (UNREACHABLE) 状態で通知を送り、Recovery checked は復旧 (recoveries:UP) 状態で通知を送ります。もし、n (none) をオプションとして指定したら、ホスト通

	知は送出されません。例: Down と RECOVERY を指定すると、通知はホストが停止した時とダウン状態から回復した時のみ送出されます。
Stalking Options	[オプション] このディレクティブは、どのホスト状態の「追跡 (stalking)」を有効にするかを決定します。適用されるオプションは下記のひとつか、組み合わせです: Up checked は、稼動(UP)状態を追跡、Down checked は、停止 (DOWN)状態を追跡、そして Unreachable checked は到達不能 (UNREACHABLE)状態を追跡します。
Contact Groups	これは、問題(や復旧)が合ったときにいつでも通知を受け取る、コンタクトグループのショートネームのリストです。
Extended Host Info Template	[オプション] このホストプロファイルのための拡張ホスト情報を定義します。このテンプレートは、Nagios 内で見られる、アイコンとURLリンクをコントロールします。
File	[必須] このホストプロファイルを使ってホストに関連付けられたファイル名を選びます。

Monarchを使ったプロファイル構成設定： どうやって、サービスプロファイルを定義するか？

ステップ1 – New Service Profile 画面

1. Configuration メニューオプションから Profiles を選びます。
2. Service Profiles を選びます。
3. New を選びます。
4. New Service Profile 画面でプロパティを入力します。(項目の一覧と説明は、表 5.5.3を参照してください)
5. Add を選ぶと、Service Profile 画面が表示されます。

図 5.5.3. New Service Profile 画面

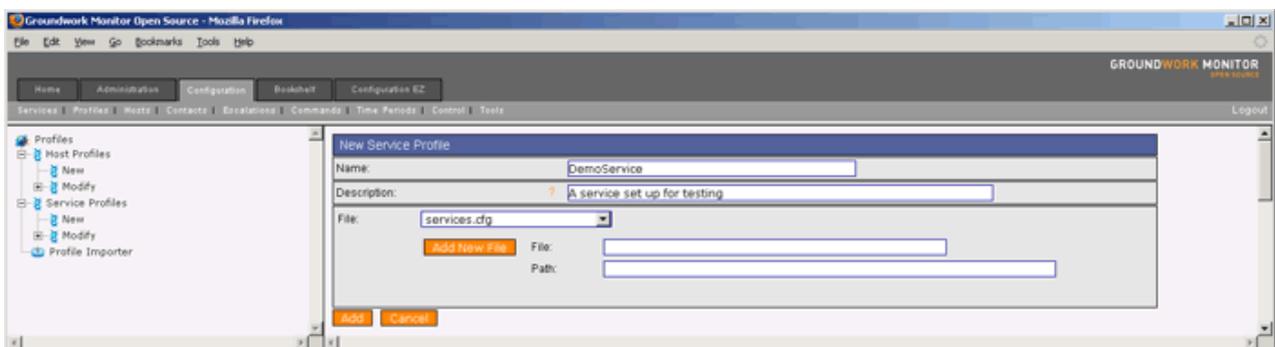


表 5.5.3. New Service Profile

Name	[必須] 新しいサービスプロファイルの名前
Description	[オプション] コメントや指示をここに記述します。
File	[必須] プロファイル中のサービスのためのデフォルトファイルを選びます。
Add New File	[オプション] 新しい構成設定ファイルを作るために選択します。 また Control>Files>File Add と行うことで新規ファイルを追加することができます。

ステップ2 – サービスプロファイルのプロパティ

1. Service Profile 画面中で、プロパティを入力します。(項目の一覧と説明は、表 5.5.3 を参照してください)
2. Save を選び、新しいサービスを追加、保存しに行きます。次の画面でサービスがセーブされたらメッセージが表示されるでしょう。Close を選びます。Delete は、現在のサービスプロファイルを削除し、Rename は現在のサービスプロファイルの名前を変更し、そして Close はセーブしないでサービスプロファイル画面を終了します。

図 5.5.4. Service Profile画面のプロパティ

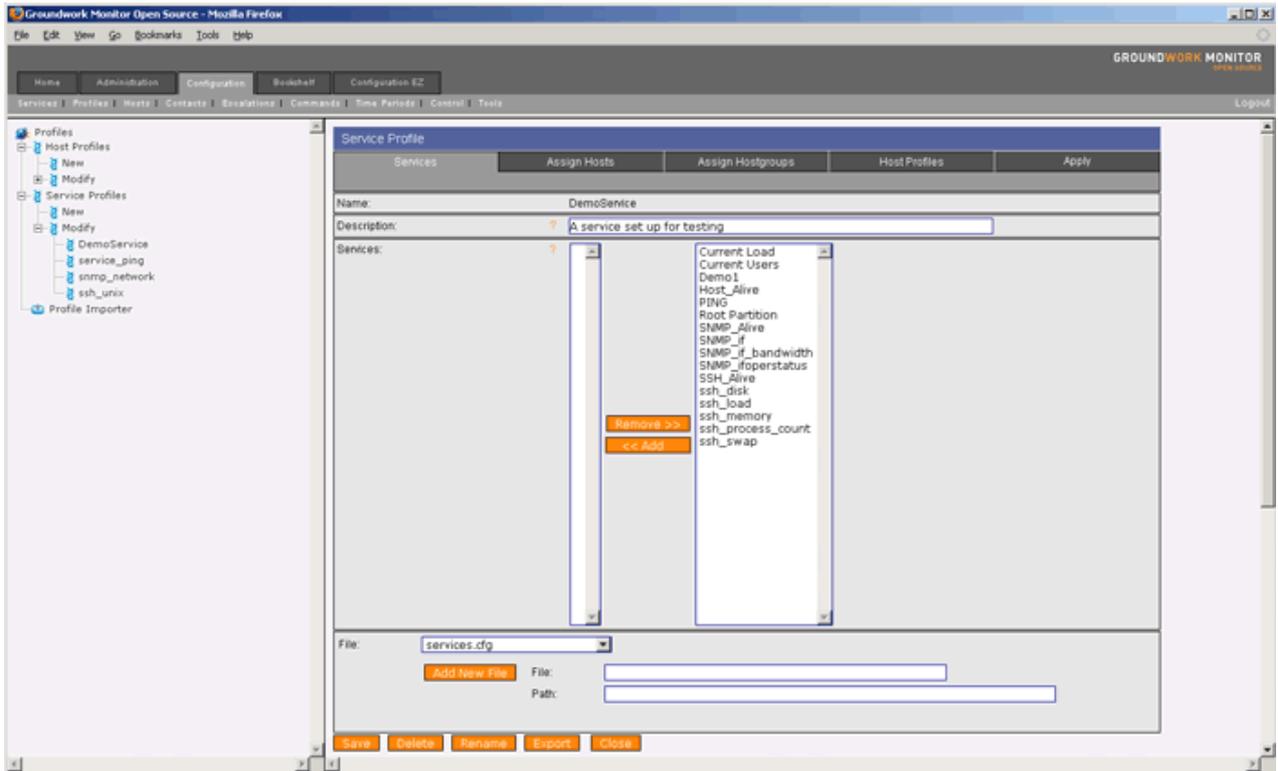


表 5.5.4. Service Profile のプロパティ

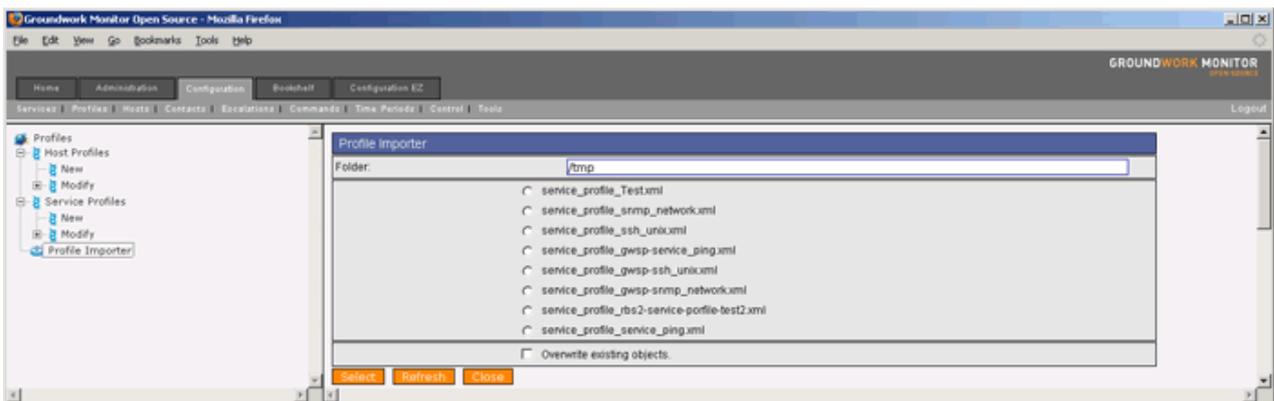
Name	[必須] 新しいサービスプロファイルの名前
Description	[オプション] コメントや指示をここに記述します。
Services	[必須] このプロファイルに含めるサービスを右側の一覧から選択します。サービス依存性をサービス名にアサインしている場合、依存関係テンプレート (dependency templates) で定義されているように親サービスの名前を含めるように注意が必要です。
File	[必須] プロファイル中のサービスのためのデフォルトファイルを選びます。
Add New File	[オプション] 新しい構成設定ファイルを作るために選択します。また Control>Files>File Add と行うことで新規ファイルを追加することができます。

Monarchを使ったプロファイル構成設定： どうやって、プロファイルをインポートするか？

Step 1 – Profile Importer 画面

1. Configuration メニューオプションから Profiles を選びます。
2. Profile Importer を選びます。
3. Profile Importer 画面の Folder オプションはプロファイルのファイルの場所を示します。このディレクトリの中のインポート可能なファイルすべてがリストされます。
4. フォルダーの中の新しいファイルを再読み出しするため、Refresh ボタンを選びます。
5. インポートしたいプロファイルの横にあるラジオボタンをクリックします。
6. ファイル内のオブジェクトによってデータベース内のオブジェクトを上書きするため、リストの最下部にある Overwrite existing objects のボックスをチェックします。
7. 選んだファイルをインポートするため、Select を選びます。Profile Importer を終了するため、Closeを選びます。

図 5.5.4. Profile Importer 画面



追補3. Monarchを使ってホストを構成設定する

目次

- どうやって、新しいホストを定義するか？
- どうやって、ホストテンプレートを構成設定するか？
- どうやって、新しいホストグループを作成するか？
- どうやって、親／子関係を作成するか？
- どうやって、拡張ホスト情報テンプレートを作成するか？
- どうやって、ホスト依存性を作成するか？

Monarchを使ってホスト構成設定： どうやって、新しいホストを定義するか？

ステップ1 – Host Vitals 画面

1. Configuration メニューオプションから Hosts を選びます。
2. New Host Wizard を選びます。
3. New Host Wizard 画面で、プロパティを入力します。(項目の一覧と説明は、表 5.6.1 を参照してください)
4. Next を選び、Host Properties 1 画面に進めます。

注意: 既存のホストをコピーするには、Hosts オプションから Clone を選んでください。それは、新しいホストに名前を付け、IPアドレスを与え、そして既存のホストのプロパティをアタッチします。使える既存のホストを探すには、Hosts の下の Search オプションを使います。ここで、ホスト名かアドレスを入力します。条件にマッチしたホストの一覧が表示されるでしょう。ひとつかいくつかの文字をタイプすると、指定された文字列のすべてのホストが提供されます。

図 5.6.1. Host Vitals 画面

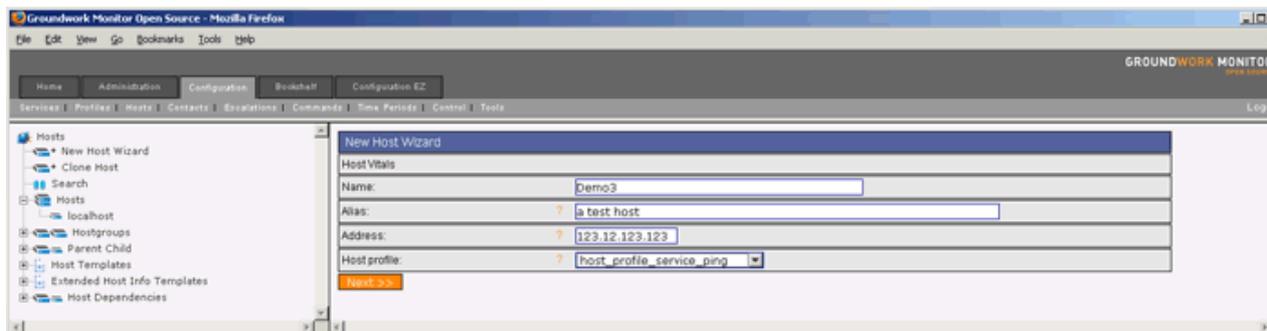


表 5.6.1 Host Vitals

Name	[必須] ホストの名前
Alias	[必須] このディレクティブは、ホストを識別するための長い名前か説明を定義するのに使います。これは、特定のホストをより容易に識別できるようにするため提供されます。
Address	[必須] このディレクティブは、ホストのアドレスを定義するために使われます。通常、これはIPアドレスですが(ホストのステータスをチェックする使うことができる限り)、実際にはどのようなものでもかまいません。IPアドレスの代わりに、ホストの識別のためにFQDNを使うことができますが、DNSサーバが使用できない場合には、問題になります。正しく使われる時には、\$HOSTADDRESS\$ マクロがこのアドレスとして入るでしょう。 注意: Host 定義の中で Address ディレクティブが指定しなかった場合、ホスト名がアドレスとして使用されます。そうした場合の警告 - もしDNSに障害が発生すると、プラグインが、ホスト名を解決できなくなるので、ほとんどのサービスチェックが失敗します。
Host Profile	[オプション] ホストプロファイルは、ホストのデザインや管理の補助情報として使用されます。

ステップ2 – ホストプロパティ1 画面

1. Host Properties 1 画面で、プロパティを入力します。(項目の一覧と説明は、表 5.6.2 を参照してください)
2. Next を選び、Host Properties 1 画面に進めます。

図 5.6.2. Host Properties 1 画面

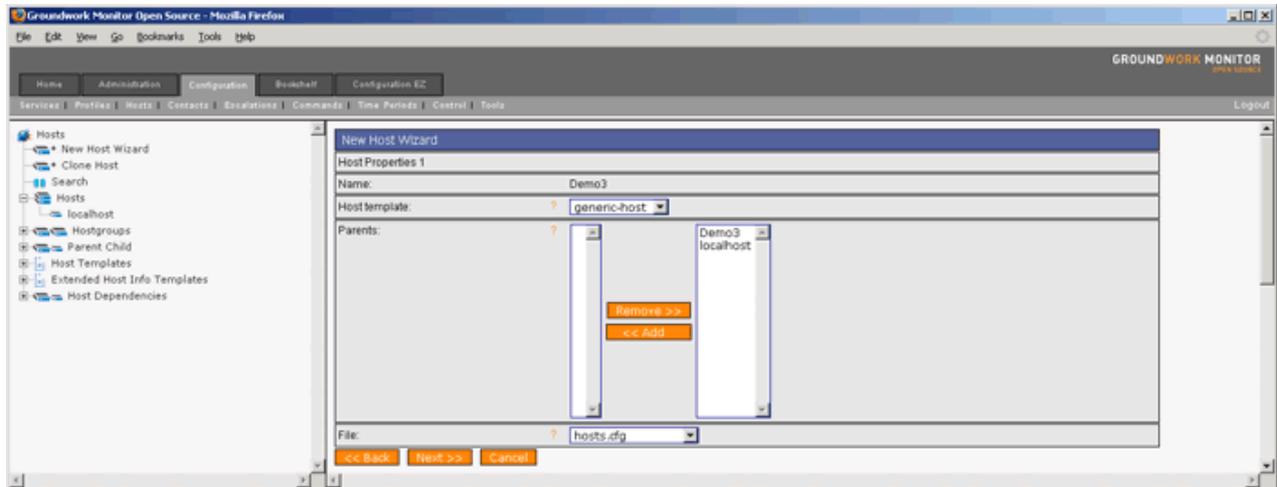


表 5.6.2 Host Properties 1

Name	名前を付けられたホスト
Host Template	[必須] このホストに最も適したホストテンプレートを選びます。
Parents	[オプション] このホストをひとつかそれ以上の親(Parents)にアサインします。親子関係は Hosts>Parent Childで管理することもできます。
File	[必須] このホストのためのファイルを選びます。 ファイルを追加するため、Host/Serviceファイルの Control>Files>Add にいきます。

ステップ3 – Host Properties 2 画面

1. Host Properties 2 画面で、プロパティを入力します。(項目の一覧と説明は、表 5.6.3 を参照してください)
2. Next を選び、Select Services 画面に進めます。

図 5.6.3. Host Properties 2 画面

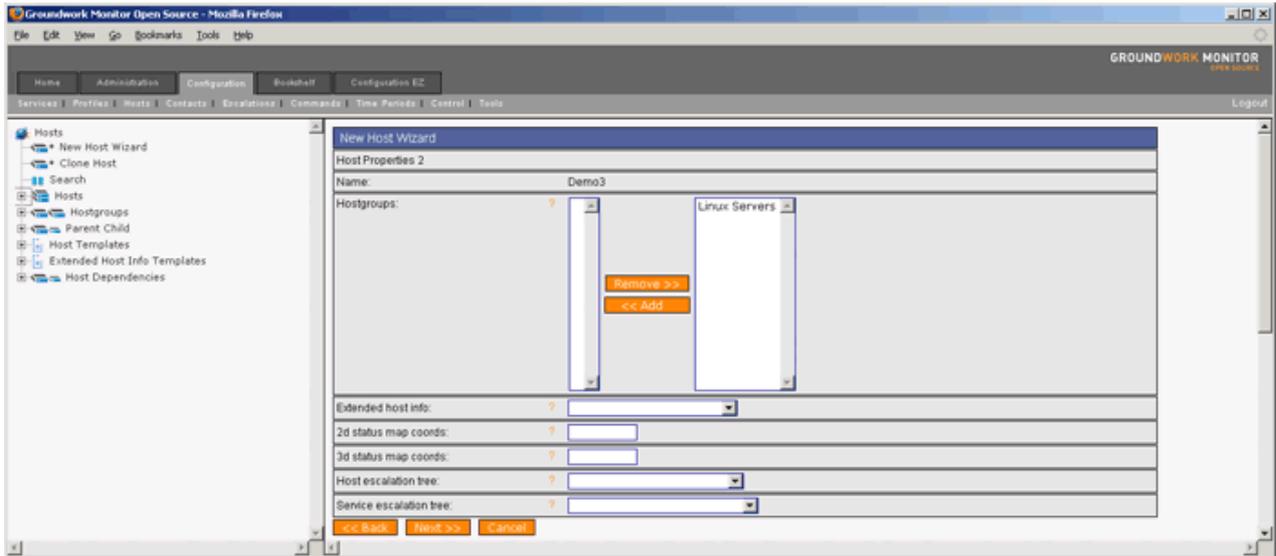


表 5.6.3 Host Properties 2

Host Groups	[オプション] このホストをひとつかそれ以上のホストグループにアサインします。ホストグループは、Hosts>Host Groups から管理できます。
Extended Host Info	[オプション] このホストのための拡張ホスト情報を定義します。そのテンプレートは Nagios 内に見られるアイコンとURLリンクをコントロールします。
2d Status Map Coords	[オプション] Nagios の2番目ステータスマップのホスト位置を定義します。座標は、生成されたイメージの中の物理ピクセルに対応する正の整数で与えられます。描画の基点 (0,0) は、イメージの左上で、イメージの上端で正のX方向(右側へ)に伸び、イメージの左側面で正のY方向(下側へ)に伸びます。参考として、描かれているアイコンは、通常およそ 40x40 ピクセル(テキストはもう少し領域を使います)です。ここで指定した座標は、描かれたホストアイコンの左上端の位置を指定します。注意: 使用するX軸とY軸の最大値を気にする必要はありません。CGI は、あなたが指定したxとy座標を基に、自動的に作成するイメージの最大範囲を計算します。
3d Status Map Coords	[オプション] Nagios の3番目ステータスマップのホスト位置を定義します。座標は、正と負の実数で表します。描画の基点は (0.0,0.0,0.0) です。参考として、ホストの立方体は、各側面が 0.5ユニット(テキストはもう少し領域を使います)です。ここで指定する座標は、ホストの立方体の中心です。
Host Escalation Tree	[オプション] このホストに最適なエスカレーションツリーを選びます。通知の拡大(同じイベントへの複数通知)を回避するため、このホストへのホストエスカレーションは、このホストがメンバーとなっているホストグループにアサインするべきではありません。
Service Escalation Tree	[オプション] このホストに最適なサービスのためのエスカレーションツリーを選びます。サービスエスカレーションがホストにアサインされたら、そのホスト上のすべてのサービスが同じエスカレーションツリーを使います。通知の拡大(同じイベントへの複数通知)を回避するため、このサービス名にアサインされたサービスエスカレーションは、このホストやホストグループにアサインするべきではありません。

ステップ4 – Select Services 画面

1. Select Services 画面で、プロパティを入力します。(項目の一覧と説明は、表 5.6.4. を参照してください)
2. Next を選んで、最終画面に進めます。

図 5.6.4. Select Services 画面

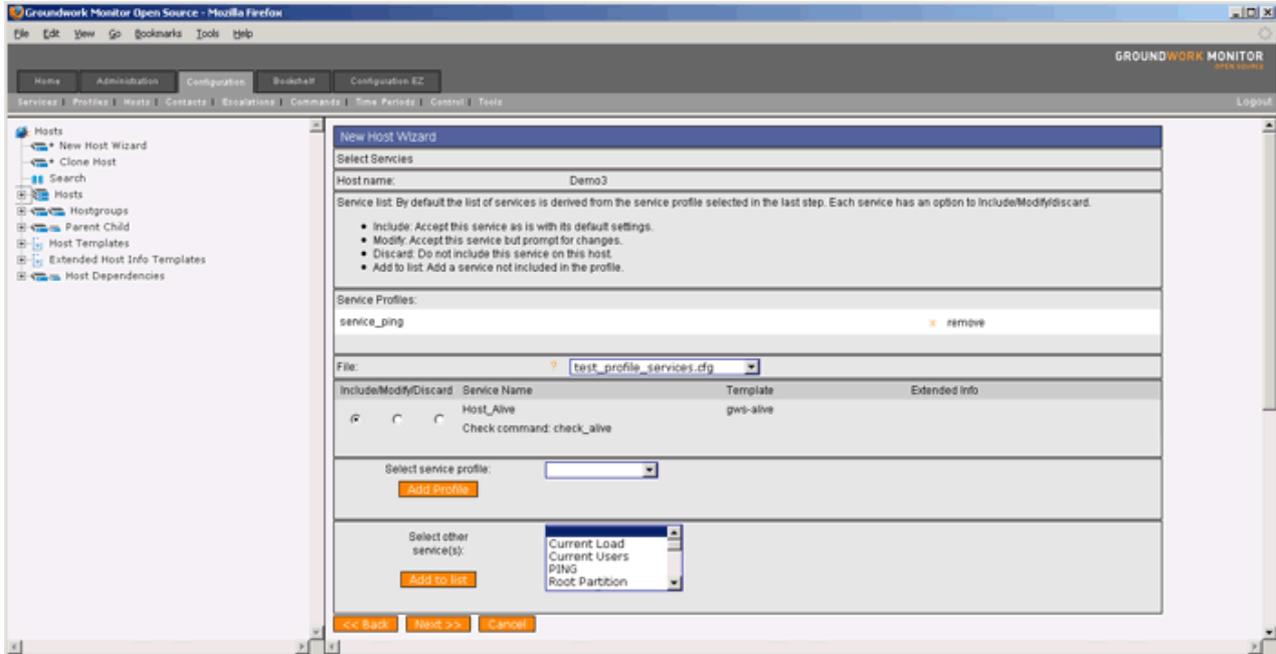


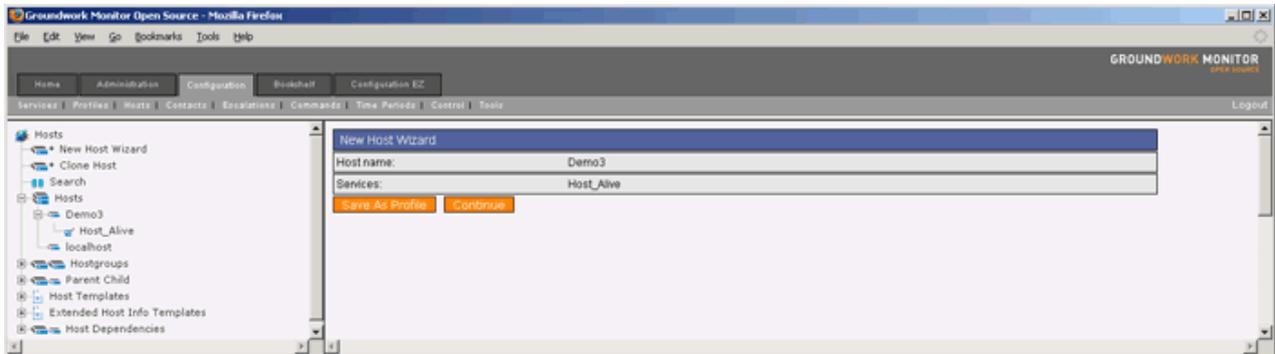
表 5.6.4 Select Services

Service Profiles	これは、このホストに追加されたサービスプロファイルの一覧です。
File	このホストにアサインされたサービスのためのデフォルトファイルを選択する必要があります。Control>Files>File Add でホストやサービスファイルを追加します。
Include/Modify/Discard Service Name	最後のステップで、(ひとつを選択した場合)サービスの一覧がサービスプロファイルから提供されるのを見ることができます。 各サービスでは、Include/Modify/Discard の選択肢があります。Include は、デフォルト設定としてこのサービスを受け入れます。Modify はこのサービスを受け付けますが、変更を促されます。Discard はこのホストにこのサービスを含めません。Add Profile は、このホストに追加のサービスプロファイルを加えます。 Add to list は、このプロファイルに含まれていないサービスを追加します。
Select Service Profile	[オプション] このホストに加える、追加のサービスプロファイルを選びます。
Select Other Services	[オプション] このホストに加える他のサービスを選びます。 サービスをクリックし、Add to List を選びます。

ステップ5 – 最終画面

- 最後の New Host Wizard 画面で、手順を完了して新しいホストをホストプロファイルとしてセーブするため、Save as Profile を選びます。新しいホストを再び構成設定するため、Host Vitals 画面を再開するには、Continue を選びます。

図 5.6.5. 最終画面



Monarchを使ってホスト構成設定： どうやって、ホストテンプレートを構成設定するか？

ステップ1 – Host Template Properties 画面

1. Configuration メニューオプションから Hosts を選びます。
2. ドロップダウンメニュー Host Templates を広げます。
3. New を選びます。
4. Host Template Properties 画面で、プロパティを入力します。
(項目の一覧と説明は、表 5.6.6. を参照してください)
5. 新しいテンプレートを追加するため、Add を選びます。

注意： 既存のテンプレートをコピーするには、Host Template オプションから Copy を選んでください。これで、新しいテンプレートに名前を付けてください。テンプレートを変更や削除するには、Modify を選びます。そうすると、編集したり、変更したテンプレートをセーブすることができます。選んだテンプレートを削除するには、Delete を選びます。名称の変更には Rename、変更しないでの終了は Cancel です。

図 5.6.6. Host Template Properties 画面

The screenshot shows the 'Host Template Properties' configuration page in the Nagios web interface. The page is titled 'Host Template Properties' and contains various configuration options for a host template. The options are organized into sections and include:

- Name: A text input field.
- Process perf data: A checkbox.
- Retain status information: A checkbox.
- Flap detection enabled: A checkbox.
- Low flap threshold: A text input field.
- High flap threshold: A text input field.
- Retain nonstatus information: A checkbox.
- Active checks enabled: A checkbox.
- Passive checks enabled: A checkbox.
- Check period: A dropdown menu.
- Obsess over host: A checkbox.
- Check freshness: A checkbox.
- Freshness threshold: A text input field.
- Check command: A dropdown menu.
- Command line: A large text area.
- Max check attempts: A text input field.
- Check interval: A text input field.
- Event handler enabled: A checkbox.
- Event handler: A dropdown menu.
- Notifications enabled: A checkbox.
- Notification interval: A text input field.
- Notification period: A dropdown menu.
- Notification options: A list of checkboxes for 'Down', 'Unreachable', 'Recovery', and 'Flapping'.
- Stalking options: A list of checkboxes for 'Down', 'Up', and 'Unreachable'.
- Contact Groups: A dropdown menu with 'nagiosadmin' selected, and 'Remove >>' and '<< Add' buttons.

At the bottom of the page, there are 'Add' and 'Cancel' buttons.

表 5.6.6. Host Template Properties

Name	[必須] ホストテンプレートの名前
Process Perf Data	[オプション] このディレクティブは、このテンプレートを使って、ホストのパフォーマンスデータ処理を行うかどうかを決めるのに使います。値：チェックなしは、パフォーマンスデータ処理を無効にする。チェックは、パフォーマンスデータ処理を有効にする。
Retain Status Information	[オプション] このディレクティブは、ホストに関するステータス関連情報が、プログラムの再起動以降も保持されるかどうかを決めるのに使います。これは、retain_state_information ディレクティブを使って、状態保持を有効にしたときにだけ便利です。値：チェックなしは、ステータス情報保持を無効にする。チェックは、ステータス情報保持を有効にする。
Flap Detection Enabled	[オプション] このディレクティブは、このテンプレートを使用するホストのフラップ検出を有効にするかどうかを決めるのに使います。値：チェックなしは、ホストフラップ検出を無効にする。チェックは、ホストフラップ検出を有効にする。
Low Flap Threshold	[オプション] このディレクティブは、このテンプレートを使用するホストのフラップ検出の低しきい値を指定するのに使います。このディレクティブの値を 0 に設定したら、low_host_flap_threshold ディレクティブで指定されている、プログラム全体に指定されている値を使用します。
High Flap Threshold	[オプション] このディレクティブは、このテンプレートを使用するホストのフラップ検出の高しきい値を指定するのに使います。このディレクティブの値を 0 に設定したら、high_host_flap_threshold ディレクティブで指定されている、プログラム全体に指定されている値を使用します。
Retain Nonstatus Information	[オプション] このディレクティブは、非ステータス情報をプログラムが再起動したときに保持するかどうか決定するために使います。これは retain_state_information ディレクティブで状態保存を有効にしている場合に便利です。値：チェックなしは、非ステータス情報保持無効。チェックは、非ステータス情報保持を有効にする
Active Checks Enabled	このディレクティブは、このホストのアクティブチェックが(定期的、要求を受けてかの両方で)有効かどうか決定するために使います。値：チェックなしは、ホストチェックを無効。チェックは、ホストチェックを有効にする。
Passive Checks Enabled	このディレクティブは、このホストのパッシブチェックが有効かどうか決定するために使います。値：チェックなしは、パッシブホストチェックを無効。チェックは、パッシブホストチェックを有効にする。
Obsess Over Host	このディレクティブは、このホストのチェックが(Nagiosのメイン構成設定で定義された)OCHPコマンドを使って、取り付かれる(obsessed)かどうかを定義します。チェックすると有効になります。
Check Freshness	このディレクティブは、新鮮度チェックがこのテンプレートを使っているホストで有効かどうかを定義するのに使います。チェックすると新鮮度チェックが有効になります。
Freshness Threshold	このディレクティブは、このテンプレートを使っているホストの新鮮度チェックのしきい値(秒)を指定するのに使います。このディレクティブの値を 0 にセットすると、Nagios は使用する新鮮度のしきい値を自動的に決めます。
Check Command	[オプション] このディレクティブは、ホストが稼働しているか停止しているかをチェックするコマンドのショートネームを指定します。一般的に、このコマンドはホストが“生きている”かどうかを ping をそのホストに発行することによって確認します。

	このコマンドはOK(0) のステータスを戻す必要があり、そうでなければNagiosはそのホストが停止していると判断します。もしこの引数を空欄にした場合、そのホストはチェックされず、Nagiosは常にそのホストが稼働していると想定します。このことは、プリンタなどの頻繁に電源を落とす機器を監視するのに役立ちます。通知コマンドを発行する回数は host_check_timeout オプションで制御します。
Max Check Attempts	[必須] このディレクティブは、OK状態ではない状態値が返ってきたときに Nagios が何回ホストチェックコマンドをリトライするかを指定するのに使います。この値を 1 にセットしたら、Nagiosはホストチェックのリトライをしないで、アラートを発行します。注意：もしそのホストの状態をチェックしたくないのであっても、ここの値は最小値の 1 に設定しなくてはなりません。ホストチェックを行わないようにするには、host_check_command オプションを空のままにするだけです。
Check Interval	[オプション] 注意：それが絶対的に必要になったとき以外は、定期チェックを有効にしないで下さい！ ホストチェックは、既に必要なときにオンデマンドで実行されますので、定期的なチェックが必要になる場合はほとんどありません。定期的なホストチェックはパフォーマンスに負の影響を及ぼします。このディレクティブは、ホストの定期チェックの間の「タイムユニット」の数を定義するのに使います。あなたが interval_length ディレクティブのデフォルトの値 60 を変更していない限り、この数値は分を意味します。
Event Handler Enabled	[オプション] このディレクティブは、このテンプレートを使っているホストのイベントハンドラが有効かどうかを判断するために使用します。値：チェックなし、は イベントハンドラを無効。チェックは、イベントハンドラを有効にする。
Event Handler	[オプション] このディレクティブは、ホストの状態の変化が検出された時(つまり、停止もしくは復旧した時)かならず実行されるコマンドのショートネームを指定するのに使います。イベントハンドラコマンドを最大何回実行できかについては event_handler_timeout オプションで制御します。
Notifications Enabled	[オプション] このディレクティブは、このテンプレートを使っているホストのための通知が有効かどうかを判断するために使用します。値：チェックなしはホスト通知を無効。チェックは、ホスト通知有効にする。
Notification Interval	[必須] このディレクティブは、そのサーバがなおダウンや到達不能であることを、コンタクト(通知先)への再通知を行うまでに待つ「タイムユニット」の数を定義するのに使います。あなたが interval_length ディレクティブのデフォルト値 60 を変更していない限り、この数値は分を意味します。この値を 0 にセットすると、Nagios は、このテンプレートを使ったホストについて問題は、再通知しません - 問題の通知は一度だけ送出されます。
Notification Period	[必須] このディレクティブは、このテンプレートを使ったホストについての通知がコンタクト(通知先)に送られる時間帯のショートネームを指定するのに使います。時間帯の範囲にない時刻の間にホストが停止し、到達不能や復旧になったら、通知は送出されません。
Notification Options	[必須] このディレクティブは、ホストに対する通知をいつ送るかを判断するのに使います。適用されるオプションは下記のひとつか、組み合わせです： Down checked は停止(DOWN)状態で通知を送り、Unreachable checked は到達不能(UNREACHABLE)状態で通知を送り、Recovery checked は復旧(recoveries : 正常(OK))状態で通知を送ります。もし、n (none) をオプションとして指定したら、ホスト通知は送出されません。例：Down と RECOVERY を指定すると、通知はホストが停止した時とダウン状態から回復した時のみ送出されます。
Stalking Options	[オプション] このディレクティブは、どのホスト状態の「追跡(stalking)」を有効にするかを決定します。適用されるオプションは下記のひとつか、組み合わせです：Up checked は、稼動(UP)状態を追跡、Down checked は、停止(DOWN)状態を追跡、そして Unreachable checked は到達不能(UNREACHABLE)状態を追跡します。

Contact Groups

これは、問題(や復旧)が合ったときにいつでも通知を受け取る、コンタクトグループのショートネームのリストです。

Monarchを使ってホスト構成設定： どうやって、新しいホストグループを作成するか？

ステップ1 – Host Group Properties 画面

1. Configuration メニューオプションから Hosts を選びます。
2. ドロップダウンメニュー Host Groups を広げます。
3. New を選びます。
4. Host Group Properties画面で、プロパティを入力します。
(項目の一覧と説明は、表 5.6.7. を参照してください)
5. 新しいホストグループを追加するため、Add を選びます。

注意： 既存のホストグループをコピーするには、Host Groups オプションから Copy を選んでください。これに新しい名前を付け、編集します。ホストグループを変更や削除するには、Modify を選びます。そうすると、編集したり、変更したホストグループをセーブすることができます。選んだホストグループを削除するには、Delete を選びます。名称の変更には Rename、変更しないでの終了は Cancel です。

図 5.6.7. Host Group Properties 画面

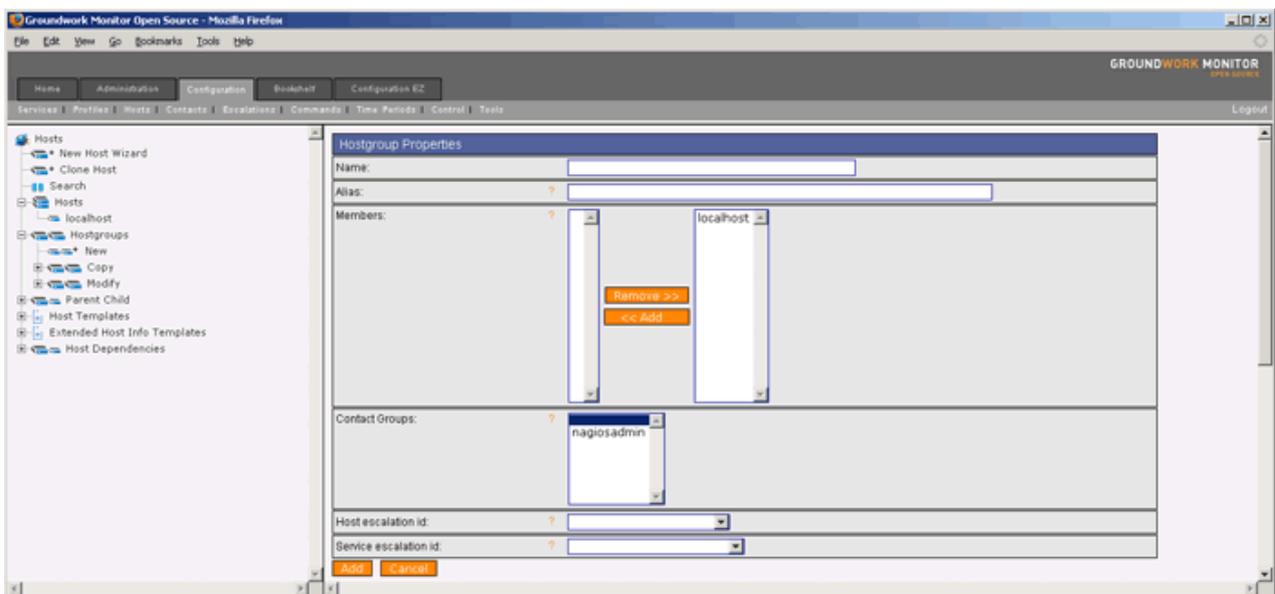


表 5.6.7. Host Group Properties

Name	[必須] ホストグループの名前
Alias	[必須] このディレクティブは、ホストグループを識別するための長い名前や説明を定義するのに使います。それは、このホストグループをより容易に識別できるように提供されます。
Members	[必須] これは、ホストグループに含まれるべきホストのショートネームのリストです。
Contact Group	[必須] これは、このホストグループのホストで問題(あるいは復旧)が発生した場合に通知を受けるべきコンタクトグループのリストです。
Host Group Escalation ID	[オプション] このホストグループに適用できるホストグループエスカレーションツリーを選びます (Escalations>Escalation Trees を参照)。 (注意 : Nagios 2.x のユーザは、同じ結果を得るため、このディレクティブを使うこと

	も、Host escalation id ディレクティブを使うこともできます。)
Host Escalation ID	[オプション] このホストグループに適用できるホストエスカレーションツリーを選びます (Escalations>Escalation Trees を参照)。このホストグループ内の全ホストが同じホストエスカレーションを受けます。
Service Escalation ID	[オプション] このホストグループに適用できるサービスエスカレーションツリーを選びます (Escalations>Escalation Trees を参照)。このホストグループ内の全ホストが同じホストエスカレーションを受けます。

Monarchを使ってホスト構成設定： どうやって、親／子関係を作成するか？

ステップ1 – Parent Child Properties 画面

1. Configuration メニューオプションから Hosts を選びます。
2. ドロップダウンメニュー Parent Child を広げます。
3. New を選びます。
4. Parent Host 画面で、プロパティを入力します。(項目の一覧と説明は、表 5.6.8. を参照してください)
5. 新しい親／子関係を作成するため、Save を選びます。

注意： 既存の親／子関係を変更や削除するには、Modify を選びます。そうすると、編集したり、変更した親／子関係をセーブすることができます。 選んだ親／子関係を削除するには、Delete を選び、変更しないでの終了は Cancel です。

図 5.6.8. Parent Child Properties

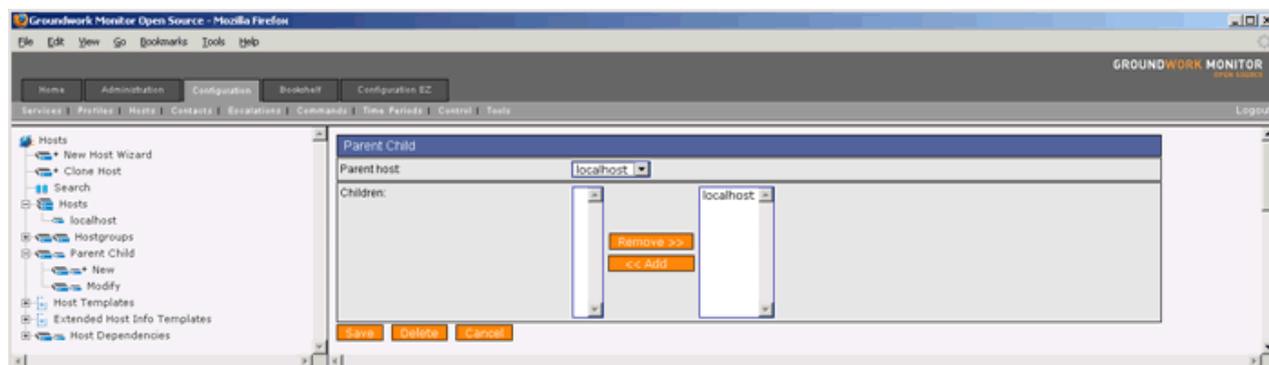


表 5.6.8. Parent Child Properties

Parent Host	[必須] 親ホストの名前。指定可能なホストは、ドロップダウンリスト内にリストされます。
Children	親ホストに対して依存関係のある子供ホスト。ホストを選び、Add ボタンを押します。

Monarchを使ってホスト構成設定： どうやって、拡張ホスト情報テンプレートを作成するか？

ステップ1 – Extended Host Info Template Properties 画面

1. Configuration メニューオプションから Hosts を選びます。
2. ドロップダウンメニュー Extended Host Info Templates を広げます。
3. New を選びます。
4. Extended Host Info Templates 画面で、プロパティを入力します。
(項目の一覧と説明は、表 5.6.9. を参照してください)
5. 新しい拡張ホスト情報テンプレートを作成するため、Add を選びます。
6. テンプレートが保存されたとのメッセージが表示されます。Continue を選びます。

注意： 既存のテンプレートをコピーするには、Extended Host Info Template オプションから Copy を選んでください。ここで、コピーしたプロパティのためのテンプレートに名前を付け、編集してください。テンプレートを変更や削除するには、Modify を選びます。そうすると、編集したり、変更したテンプレートをセーブすることができます。選んだテンプレートを削除するには、Delete を選びます。名称の変更には Rename、変更しないでの終了は Cancel です。

図 5.6.9. Extended Host Info Template Properties 画面

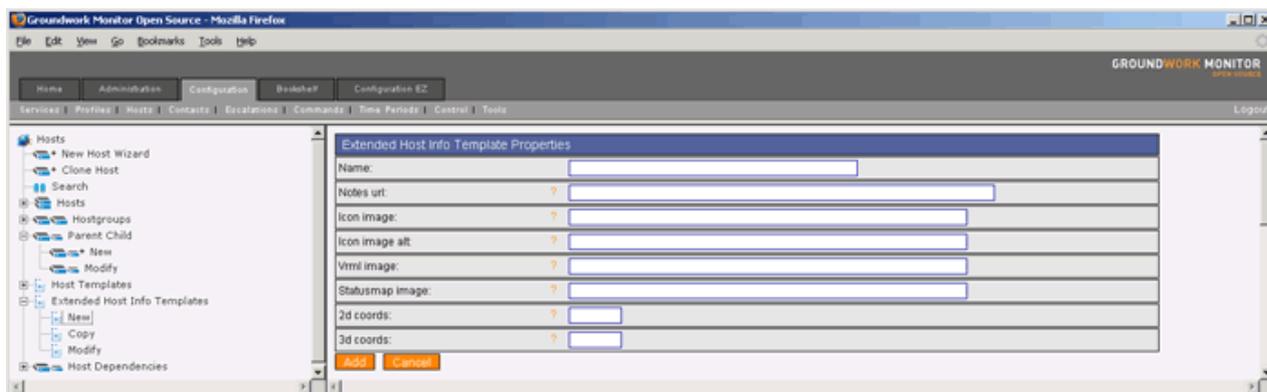


表 5.6.9. Extended Host Info Template Properties

Name	[必須] 拡張ホスト情報テンプレートの名前
Notes URL	[オプション] このオプションはホストに関するより多くの情報を提供するURLを定義するのに使います。 URLを指定すると（指定したホストの情報を参照したとき）'Notes About This Host' というリンクが拡張情報 CGI で見られるようになります。正しければどんなURLでも指定可能です。相対パスを使いたい場合、基点のパスは CGI アクセスで使用しているパス（つまり、/cgi-bin/nagios/）と同じです。ホスト、緊急連絡先、方法などの詳しい情報を他のサポートスタッフに提供したい場合に、この機能は非常に便利です。また、これはテンプレートなので、\$HOSTNAME\$ と \$HOSTADDRESS\$ マクロがURL内で使用できます。
Icon Image	[オプション] この変数は、ホストに関連付けらるべき GIF、PNG、と JPG イメージの名前を定義するのに使います。このイメージは、ステータスと拡張情報 CGI で表

	示されます。イメージは、サイズで 40x40 ピクセルだと見栄えが良いです。ホストのイメージは、HTMLのイメージディレクトリ下の logos/ サブディレクトリにあると想定されます(つまり、/usr/local/nagios/share/images/logos)。
Icon Image Alt	[オプション] この変数は、<icon_image> 引数で指定される ALT タグのイメージの中で使われるオプション文字列を定義するのに使います。ALT タグはステータス、拡張情報、およびステータスマップの各 CGI で使用されます。<icon_image> 変数で使用するイメージと異なり、これはたいていの場合透過性をもたせないものです。もし透過性があると、ホストのオブジェクトは少し風変わり感じになります。
VRML Image	[オプション] この変数は、このホストに関係付けられるべき GIF、PNG、と JPG イメージの名前を定義するのに使います。このイメージは、statuswrl CGI の中で指定されたホストのテキストマップとして使われます。ホストのイメージは、HTMLのイメージディレクトリ下の logos/ サブディレクトリにあると想定されます(つまり、/usr/local/nagios/share/images/logos)。
Statusmap Image	[オプション] この変数は、statusmap CGI 中のホストに関連付けられたイメージの名前を定義するのに使います。好きな JPEG、PNG や GIFイメージが指定できますが、ステータスマップのイメージを生成するとき、他のフォーマットでは多くのCPU時間が無駄に消費されるので、GD2フォーマットの使用を強く推奨します。GD2フォーマットは Thomas Boutell の gd ライブラリから提供される pngtogd2 を使って、PNGイメージから作成することができます。GD2イメージは、statusmap CGI がネットワークマップのイメージを生成するとき、CPU負荷を最小化するために非圧縮形式で作成されるべきです。イメージは、サイズで 40x40 ピクセルだと見栄えが良いです。ホストのイメージは、HTMLのイメージディレクトリ下の logos/ サブディレクトリにあると想定されます(つまり、/usr/local/nagios/share/images/logos)。
2d Coords	[オプション] この変数は、statusmap CGI 中でホストを描くときに使う座標を定義するのに使います。座標は、生成されたイメージの中の物理ピクセルに対応する正の整数で与えられます。描画の基点 (0,0) は、イメージの左上で、イメージの上端で正のX方向(右側へ)に伸び、イメージの左側面で正のY方向(下側へ)に伸びます。参考として、描かれているアイコンは、通常およそ 40x40 ピクセル(テキストはもう少し領域を使います)です。ここで指定した座標は、描かれたホストアイコンの左上端の位置を指定します。注意: 使用するX軸とY軸の最大値を気にする必要はありません。CGI は、あなたが指定したxとy座標を基に、自動的に作成するイメージの最大範囲を計算します。
3d Coords	[オプション] この変数は、statusmap CGI 中でホストを描くときに使う座標を設定するのに使います。座標は、正と負の実数で表します。描画の基点は (0.0,0.0,0.0) です。参考として、ホストの立方体は、各側面が 0.5ユニット(テキストはもう少し領域を使います)です。ここで指定する座標は、ホストの立方体の中心です。

Monarchを使ってホスト構成設定： どうやって、ホスト依存性を作成するか？

ステップ1 – Host Dependency Properties 画面

1. Configuration メニューオプションから Hosts を選びます。
2. ドロップダウンメニュー Host Dependencies を広げます。
3. New を選びます。
4. Host Dependency Properties 画面で、プロパティを入力します。
(項目の一覧と説明は、表 5.6.10. を参照してください)
5. 新しいホスト依存性を作成するため、Add を選びます。
6. 次の画面でホスト依存性が保存されたとのメッセージが表示されますので、Continue を選びます。

注意：Host Dependencies オプションから、ホスト依存性を変更や削除するには、Modify を選びます。そうすると、編集したり、変更したテンプレートをセーブすることができます。選んだテンプレートを削除するには、Delete を選び、変更しないでの終了は Cancel を選びます。

図 5.6.10. Host Dependency Properties 画面

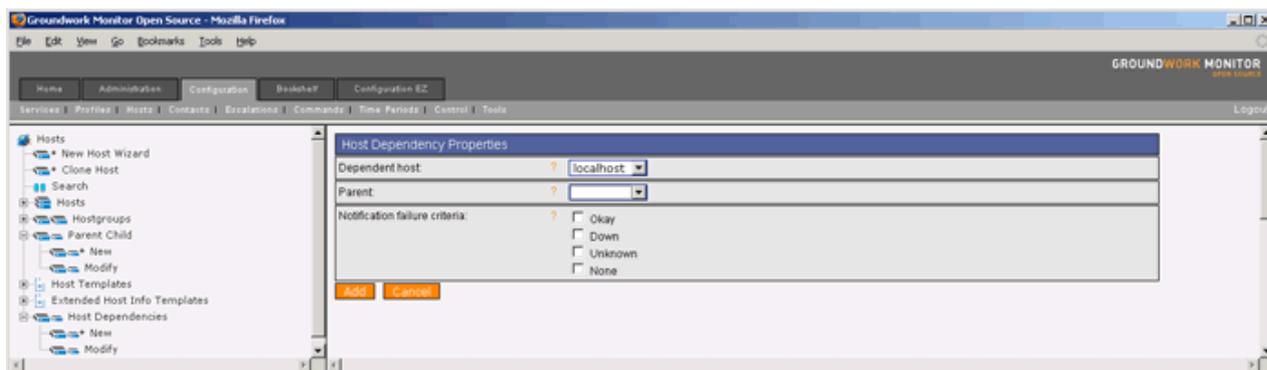


表 5.6.10. Host Dependency Properties

Dependent Host	[必須] このディレクティブは依存元ホストのショートネームを特定するのに使います。
Parent Host	[必須] このディレクティブは依存対象ホストのショートネームを特定するのに使います。
Notification Failure Criteria	[必須] このディレクティブは、いつ依存元ホストについての通知を送出するかの判断基準を定義するのに使います。もし、ホストが、我々が指定したどれかひとつの障害状態になったら、依存元ホストの通知がコンタクト(連絡先)に送出されます。適用されるオプションは下記のひとつか、組み合わせです：o = 稼動(UP)状態での障害、d = 停止(DOWN)状態での障害、u = 到達不能(UNREACHABLE)状態での障害。もし、n (none) をオプションとして指定したら、依存性の通知は送出されず、依存元ホストの通知が常に送出されます。例：d = 停止(DOWN)状態での障害を指定すると、依存先(そのホストが依存している)ホストがダウン状態だった場合、依存元ホストの通知は送出されません。

追補4. Monarch を使ってホストを管理する

目次

ホスト詳細を管理する

ホストプロファイルを管理する

ホスト-親を管理する

ホスト-ホストグループを管理する

ホスト-エスカレーションを管理する

ホスト-サービスを管理する

Monarch を使ったホスト管理： ホスト詳細を管理する

1. Configuration メニューオプションから Hosts を選びます。
2. ドロップダウンメニュー Hosts を開きます。
3. 管理するためホストを選択します。
4. Host Detail 画面の中で、プロパティを入力します。(項目の一覧と説明は、表 5.6b.1 を参照してください)
5. 変更を加えたら、Save を選びます。ホストを削除するには Delete を選びます。ホストの名前を変更するには Rename を選びます。

注記： 既存のホストをコピーするには、Hosts オプションから Clone を選びます。これで新しいホストに名前をつけ、IPアドレスを付与して既存ホストのプロパティをアタッチします。ここで新ホストの編集と保存ができます。既存のホストを見つけるには、Hosts の下の Search オプションが使えます。ここで、ホスト名やアドレスを入力できます。マッチングしたホストが表示されます。一文字か文字列を入力しただけで、指定した文字列で始まるすべてのホストの一覧が得られます。

図 5.6b.1. Host Detail 画面

The screenshot shows the 'Manage Host' configuration page in Groundwork Monitor. The interface includes a navigation menu on the left with options like 'New Host Wizard', 'Clone Host', 'Search', and 'Hosts'. The main content area is titled 'Manage Host' and contains several tabs: 'Host Detail', 'Profile', 'Parents', 'Hostgroups', 'Escalations', and 'Services'. The 'Host Detail' tab is active, displaying the following fields and options:

- Host name: zinal
- Alias: best
- Address: 172.20.113.201
- Host template: generic-host
- Inherit from template:** A section with a 'Set Inheritance' button and a note: 'Inherit all values from template. Set all directives to inherit values from the selected template. Uncheck the left checkbox on the directives below to override the template values.'
- Process performance data:
- Retain status information:
- Flap detection enabled:
- Low flap threshold: [input field]
- High flap threshold: [input field]
- Retain nonstatus information:
- Active checks enabled:
- Passive checks enabled:
- Obsess over host:
- Check freshness:
- Freshness threshold: [input field]
- Check command: [dropdown menu]
- Max check attempts: 3
- Check interval: [input field]
- Event handler enabled:
- Event handler: [dropdown menu]
- Notifications enabled:
- Notification interval: 60
- Notification period: none
- Notification options: Down, Unreachable, Recovery, Flapping
- Stalking options: Down, Up, Unreachable
- Contact groups: nagiosadmin [dropdown menu]
- Extended host info template: [dropdown menu]
- 2d status coords: [input field]
- 3d status coords: [input field]
- File: hosts.cfg

At the bottom of the form, there are three buttons: 'Save', 'Delete', and 'Rename'.

表 5.6b.1 Host Detail

Name	[必須] ホストの名前
Alias	[必須] このディレクティブは、ホストを識別するための長い名前か説明を定義するのに使います。これは、特定のホストをより容易に識別できるようにするため提供されます。
Address	[必須] このディレクティブは、ホストのアドレスを定義するために使われます。通常、これはIPアドレスですが(ホストのステータスをチェックする使うことができる限り)、実際にはどのようなものでもかまいません。ホストの識別のため、IPアドレスの代わりにFQDNを使うことができますが、DNSサーバが使用できない場合には、問題になります。正しく使われる時には、\$HOSTADDRESS\$ マクロがこのアドレスとして入るでしょう。注意:Host 定義の中で Address ディレクティブが指定しなかった場合、ホスト名がアドレスとして使用されます。 そうする場合の警告 - もしDNSに障害が発生すると、プラグインが、ホスト名を解決できなくなるので、ほとんどのサービスチェックが失敗します。
Host Template	[必須] このホストに最も適したホストテンプレートを選びます。
Set Inheritance	テンプレートからすべての値を継承します: チェックする = 選択したテンプレートから全ディレクティブの値が継承されます。テンプレートの値を変える(上書きする)には、ディレクティブの左側にあるチェックボックスをチェックしないでください。

Monarch を使ったホスト管理： ホストプロファイルを管理する

1. Configuration メニューオプションから Hosts を選びます。
2. ドロップダウンメニュー Hosts を開きます。
3. 管理するためホストを選択します。
4. Profile タブを選びます。
5. Profile 画面の中で、プロパティを入力します。(項目の一覧と説明は、表 5.6b.2 を参照してください)

図 5.6b.2. Profile 画面

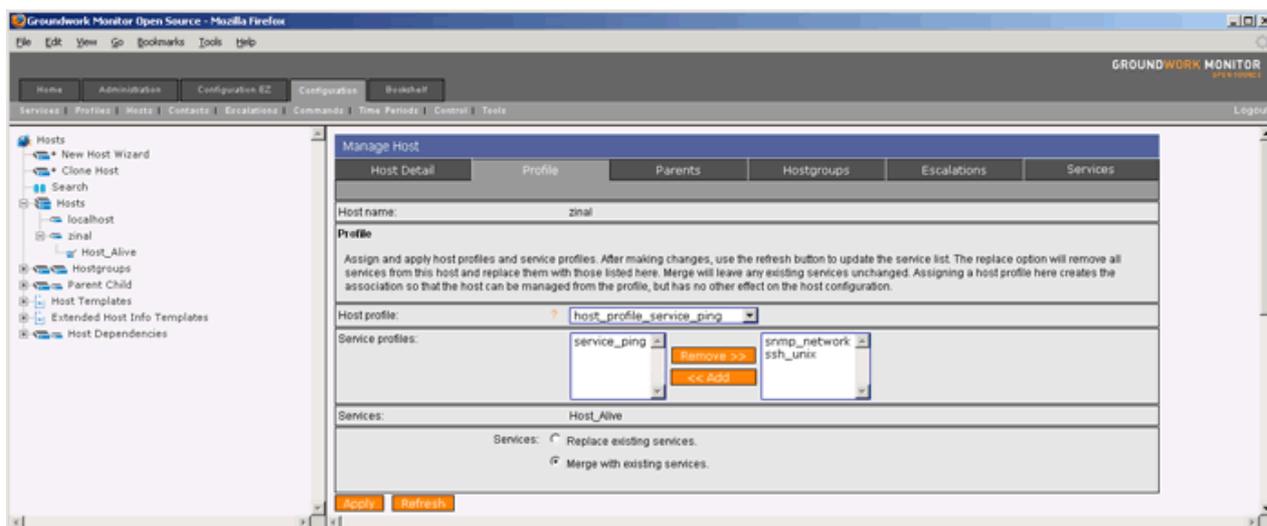


表 5.6b.2. Profile

Host Name	[必須] ホストの名前
Host Profile	[オプション] ホストプロファイルは、ホストのデザインや管理の補助情報として使用されます。
Service Profiles	このホストのためのサービスプロファイルを選びます。これは、デフォルトのサービスセットをセットアップします。
Services	変更を施した後、Refresh ボタンを使ってサービスリストを更新します。 Replace Existing Services は、このホストからすべてのサービスを削除し、リストから外します。 Merge with Existing Services は、既存のサービスをそのままにします。ここでホストプロファイルをアサインすると関連付けが生成されるので、ホストはプロファイルから管理できるようになりますが、ホスト構成設定には他の影響は及ぼしません。

Monarch を使ったホスト管理: ホスト親を管理する

1. Configuration メニューオプションから Hosts を選びます。
2. ドロップダウンメニュー Hosts を開きます。
3. 管理するためホストを選択します。
4. Parents タブを選びます。
5. Parents 画面の中で、プロパティを入力します。(項目の一覧と説明は、表 5.6b.3 を参照してください)
6. 新しい関係を作成するため Save を選びます。

図 5.6b.3. Parents 画面

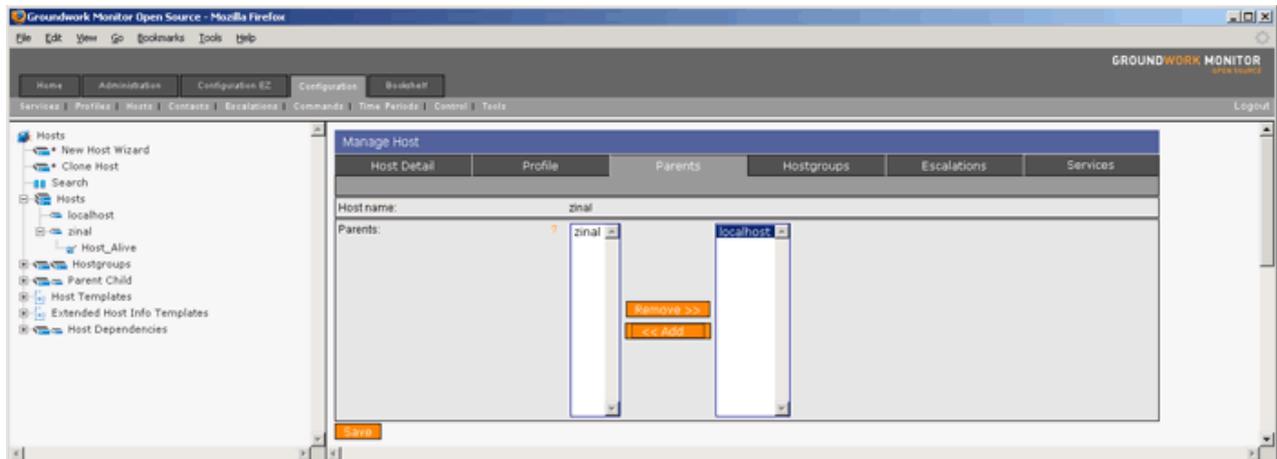


表 5.6b.3. Parents

Host Name	[必須] 選択されたホストの名前
Parents	[オプション] このホストをひとつかそれ以上の親 (Parents) にアサインします。親子関係は Hosts>Parent Child で管理することもできます。

Monarch を使ったホスト管理： ホスト-ホストグループを管理する

1. Configuration メニューオプションから Hosts を選びます。
2. ドロップダウンメニュー Hosts を開きます。
3. 管理するためホストを選択します。
4. Hostgroups タブを選びます。
5. Hostgroups 画面の中で、プロパティを入力します。(項目の一覧と説明は、表 5.6b.4 を参照してください)
6. 新しいホストグループを作成するため Save を選びます。

図 5.6b.4. Hostgroups 画面

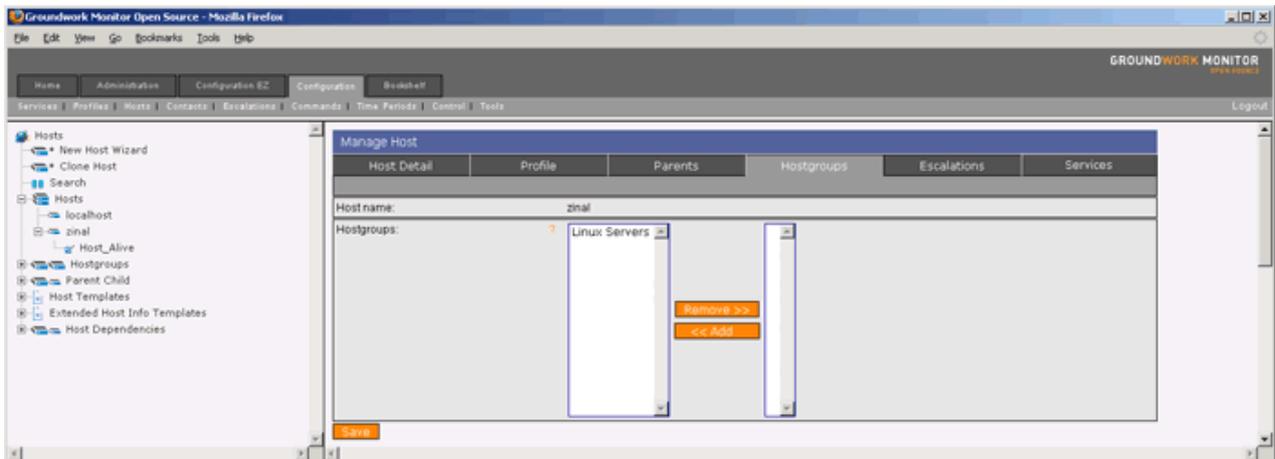


表 5.6b.4. Hostgroups

Host Name	[必須] 選択されたホストの名前
Hostgroups	[オプション] このホストをひとつかそれ以上のホストグループにアサインします。ホストグループは、Hosts>Host Groups から管理できます。

Monarch を使ったホスト管理: ホスト-エスカレーションを管理する

1. Configuration メニューオプションから Hosts を選びます。
2. ドロップダウンメニュー Hosts を開きます。
3. 管理するためホストを選択します。
4. Escalations タブを選びます。
5. Escalations 画面の中で、プロパティを入力します。(項目の一覧と説明は、表 5.6b.4 を参照してください)
6. ホストグループを追加や削除するため Save を選びます。

図 5.6b.5. Escalations 画面

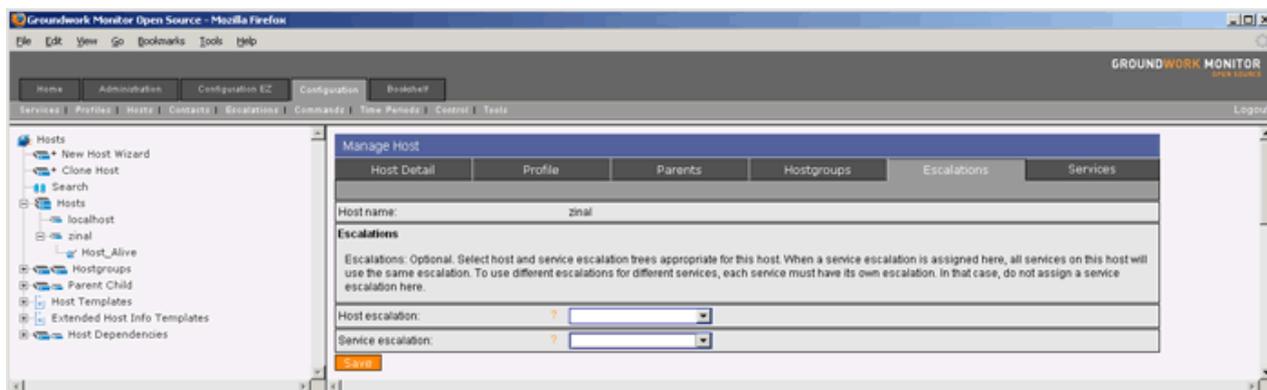


表 5.6b.5. Escalations

Host Name	[必須] 選択されたホストの名前
Host Escalation	[オプション] このホストに最適なエスカレーションツリーを選びます。通知の拡大(つまり、同じイベントへの複数通知)を回避するため、このホストもアサインされたホストエスカレーションは、このホストがメンバーとなっているホストグループにアサインするべきではありません。
Service Escalation	[オプション] このホストに最適なサービスのためのエスカレーションツリーを選びます。サービスエスカレーションがホストにアサインされると、そのホスト上のすべてのサービスが同じエスカレーションツリーを使います。通知の拡大(つまり、同じイベントへの複数通知)を回避するため、このサービス名にアサインされたサービスエスカレーションは、このホストやホストグループにアサインするべきではありません。

Monarch を使ったホスト管理： ホスト-サービスを管理する

1. Configuration メニューオプションから Hosts を選びます。
2. ドロップダウンメニュー Hosts を開きます。
3. 管理するためホストを選択します。
4. Services タブを選びます。
5. Services 画面の中で、プロパティを入力します。(項目の一覧と説明は、表 5.6b.5 を参照してください)
6. 新しいホスト依存性を生成するため、Save を選びます。
7. 次の画面で、ホスト依存性がセーブされたとのメッセージが表示されます。Continue を選びます。

図 5.6b.5. Services 画面

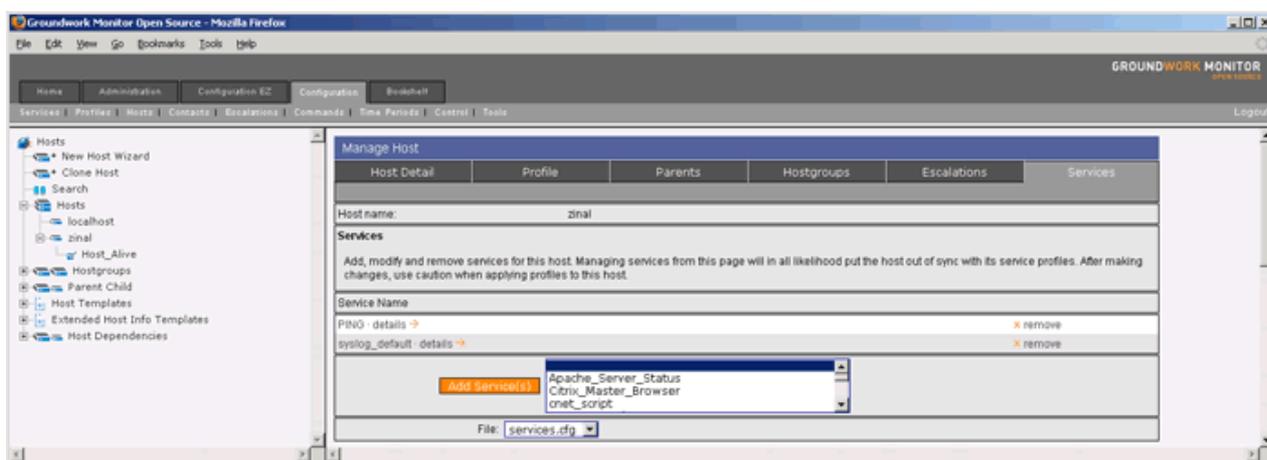


表 5.6b.5. Services

Host Name	[必須] 選択されたホストの名前
Service Name	現在のホストのためのサービスのリスト。これらのサービスは、サービスをドロップダウンボックスから選択して Add Services を選んだ後で、表示されます。
Add Services	現在のホストに追加するサービスを選びます。Ctrl と Shift キーを使って、多数のサービスを一度で選択して Add Serviceをクリックすることで、Service Name のリストに追加することができます。
File	サービスプロファイルをセーブする構成設定ファイル

追補5. コンタクト(連絡先)を定義する

目次

どうやって、新しいコンタクトを定義するか？

どうやって、新しいコンタクトグループを定義するか？

どうやって、コンタクトテンプレートを定義するか？

コンタクトの定義： どうやって、新しいコンタクトを定義するか？

ステップ1 – Contact Properties 画面

1. Configuration メニューオプションから Contacts を選びます。
2. ドロップダウンメニュー Contacts を広げます。
3. New を選びます。
4. Contact Properties 画面で、プロパティを入力します。(項目の一覧と説明は、表 5.7.1を参照してください)
5. Add を選びます。
6. コンタクトが追加されたとのメッセージが表示されるでしょう。Continueを選びます。

記： 既存のコンタクトをコピーするには、Configuration メニューオプションから Copy を選びます。そうすると新プロファイルに名前を付け、コピーされたプロパティを編集できるようになります。コンタクトを変更したり削除する場合は、Modify を選びます。そうすると編集したり変更したコンタクトをセーブすることができます。コンタクトの削除には Delete、名称変更には Rename、変更しないでの終了は Cancel を選びます。

図 5.7.1. Contact Properties 画面

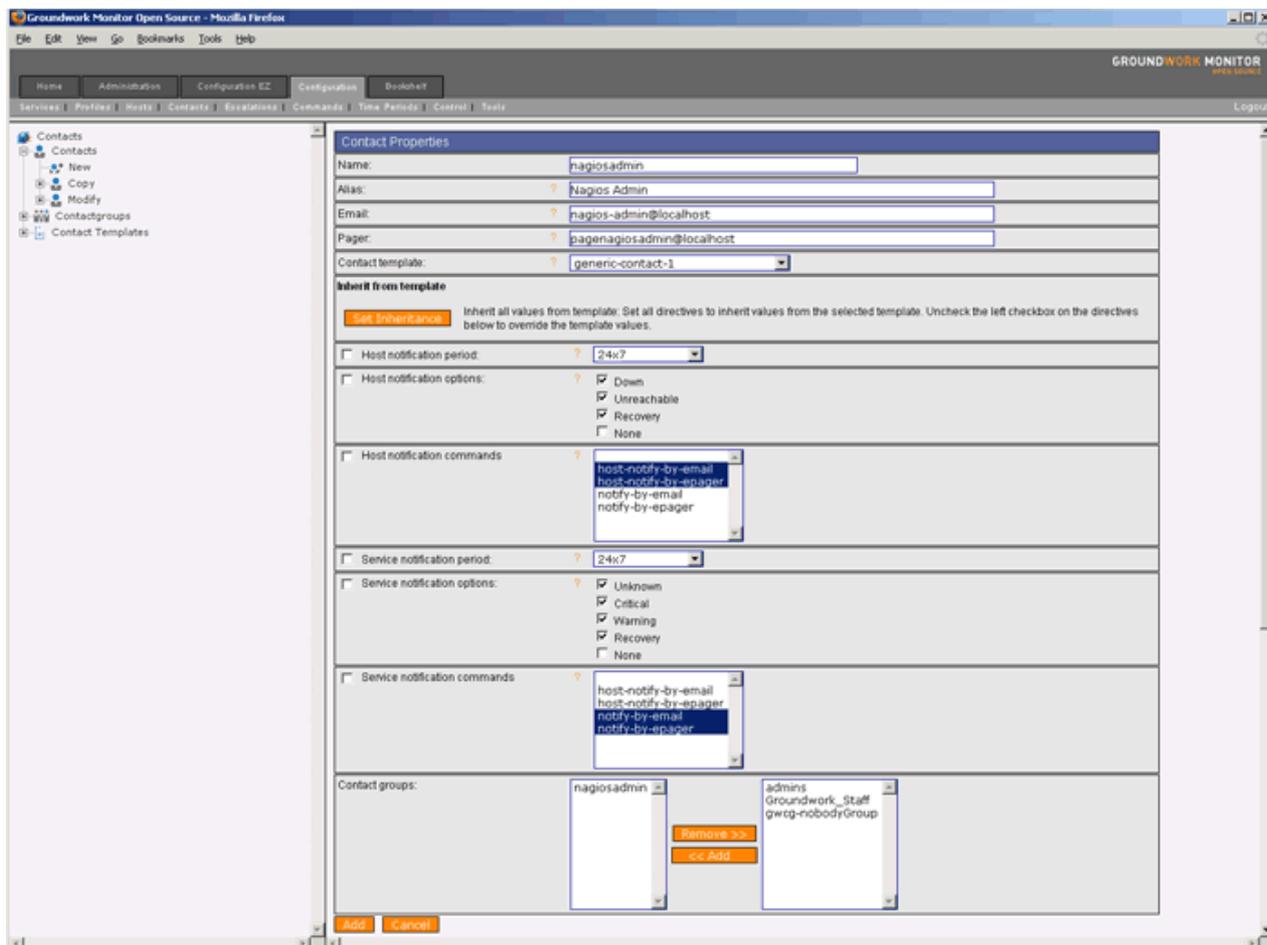


表 5.7.1. コンタクトの属性

Name	[必須] コンタクトの名前
Alias	[必須] このディレクティブは、コンタクトの長い名称を定義したり、説明を定義するのに使われます。正しい環境下では、\$CONTACTALIAS\$ マクロがこの値となります。
Email	[オプション] このディレクティブは、Contactのための電子メールアドレスを定義するのに使われます。通知コマンドを設定することにより、アラートのメールをコンタクトとに送ることができます。正しい環境下では、\$CONTACTEMAIL\$ マクロがこの値となります。
Pager	[オプション] このディレクティブは、Contactのためのページ番号を定義するのに使われます。ページのゲートウェイの電子メールアドレスでも良いです(例: pagejoe@pagenet.com)。通知コマンドを設定することにより、アラートのページコールをコンタクトとに送ることができます。正しい環境下では、\$CONTACTPAGER\$ マクロがこの値となります。
Contact Template	[必須] このコンタクトに最適なコンタクトテンプレートを選びます。注意: 下記のディレクトリの継承 (Inheritance : 左のチェックボックス) について。テンプレートの値を上書きする場合は、左側のチェックボックスのチェックをはずしてください。
Set Inheritance	このボタンを選ぶと、すべてのディレクティブについて選択したテンプレートから継承した値をセットします。下記で、左のチェックボックスをチェックしていないディレクティブは、テンプレートの値を上書き(変更)します。
Host Notification Period	[必須] このディレクティブは、そのコンタクトにホストの障害や復旧について通知することができる時間帯のショートネームを指定するのに使います。これは、コンタクト(連絡先)がホストの通知に対して「オンコール:待機中」にあると考えることができます。これがどのように働き、誤って使った場合の潜在的な問題に関する時間帯についてのより詳しい情報は、ドキュメントを参照してください。
Host Notification Options	[必須] このディレクティブは、ホストのステータスによってコンタクトにどの通知を送るかを定義するのに使います。適用されるオプションは下記のひとつか、組み合わせです: Down チェックは停止 (DOWN) 状態で通知を送り、Unreachable チェックは到達不能 (UNREACHABLE) 状態で通知を送り、Recovery チェックは復旧 (recoveries : UP) 状態で通知を送ります。もし、Noneをオプションとして指定したら、ホスト通知は送出されません。
Host Notification Commands	[オプション] このディレクティブは、ホストの障害や復旧をコンタクトに通知するのに使用するコマンドのショートネームのリストを定義するのに使います。すべての通知コマンドは、コンタクトに通知が必要なときに使用されます。通知コマンドの最大実行時間は、notification_timeout オプションで制御されます。
Service Notification Period	[必須] このディレクティブは、そのコンタクトにサービスの障害や復旧について通知することができる時間帯のショートネームを指定するのに使います。これは、コンタクト(連絡先)がサービス通知に対して「オンコール:待機中」にあると考えることができます。これがどのように働き、誤って使った場合の潜在的な問題に関する時間帯についてのより詳しい情報は、ドキュメントを参照してください。
Service Notification Options	[必須] このディレクティブは、サービスのステータスによってコンタクトにどの通知を送るかを定義するのに使います。適用されるオプションは下記のひとつか、組み合わせです: Warningチェックはワーニング (WARNING) 状態で通知を送り、Unknownチェックは不明 (UNKNOWN) 状態で通知を送り、Critical チェックはクリティカル (CRITICAL) 状態で通知を送ります。 Nagios 2.xでは、ホストやサービスのフラッピングの開始と停止についての通知をすることもきるでしょう。もし、Noneをオプションとして指定したら、ホスト通知は送出されません。
Service Notification Commands	[オプション] このディレクティブは、サービスの障害や復旧をコンタクトに通知するのに使用するコマンドのショートネームのリストを定義するのに使います。すべての通知コマンドは、コンタクトに通知が必要なときに使用されます。通知コマンドの最大

	実行時間は、notification_timeout オプションで制御されます。
Contact Groups	[オプション] どのコンタクトグループが、アラートや復旧通知のために使われるかを定義します

コンタクトの定義： どうやって、新しいコンタクトグループを定義するか？

ステップ1 – Contact Group Properties 画面

1. Configuration メニューオプションから Contacts を選びます。
2. ドロップダウンメニュー Contactgroups を広げます。
3. New を選びます。
4. Contactgroup Properties 画面で、プロパティを入力します。
(項目の一覧と説明は、表 5.7.2 を参照してください)
5. Add を選びます。
6. コンタクトグループが追加されたとのメッセージが表示されるでしょう。Continueを選びます。

記： 既存のコンタクトグループをコピーするには、Configuration メニューオプションから Copy を選びます。そうすると新コンタクトグループに名前を付け、コピーされたプロパティを編集できるようになります。コンタクトグループを変更したり削除する場合は、Modify を選びます。そうすると編集したり変更したコンタクトグループをセーブすることができます。コンタクトグループの削除には Delete、名称変更には Rename、変更しないでの終了は Cancel を選びます

図 5.7.2. Contact Group Properties 画面

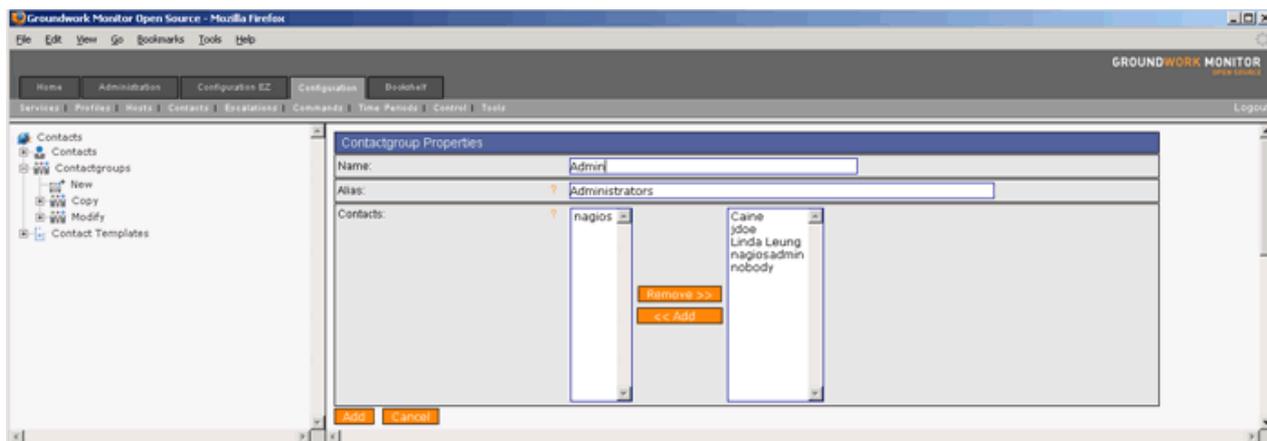


表 5.7.2. コンタクトグループのプロパティ

Name	[必須] コンタクトグループの名前
Alias	[必須] このディレクティブは、コンタクトグループの長い名称を定義したり、説明を定義するのに使われます。
Contacts	[必須] 予め定義されたコンタクトをコンタクトグループにアサインします。

コンタクトの定義： どうやって、コンタクトテンプレートを定義するか？

ステップ1 – Contact Template Properties 画面

1. Configuration メニューオプションから Contacts を選びます。
2. ドロップダウンメニュー Contact Templates を広げます。
3. New を選びます。
4. Contactgroup Properties 画面で、プロパティを入力します。
(項目の一覧と説明は、表 5.7.3 を参照してください)
5. 新しいコンタクトテンプレートを追加、保存するため、Add を選びます。
6. コンタクトテンプレートが追加されたとのメッセージが表示されるでしょう。Continueを選びます。

記：既存のコンタクトテンプレートをコピーするには、Configuration メニューオプションから Copy を選びます。そうすると新コンタクトテンプレートに名前を付け、コピーされたプロパティを編集できるようになります。コンタクトテンプレートを変更したり削除する場合は、Modify を選びます。そうすると編集したり変更したコンタクトテンプレートをセーブすることができます。コンタクトテンプレートの削除には Delete、名称変更には Rename、変更しないでの終了は Cancel を選びます

図 5.7.3. Contact Template Properties 画面

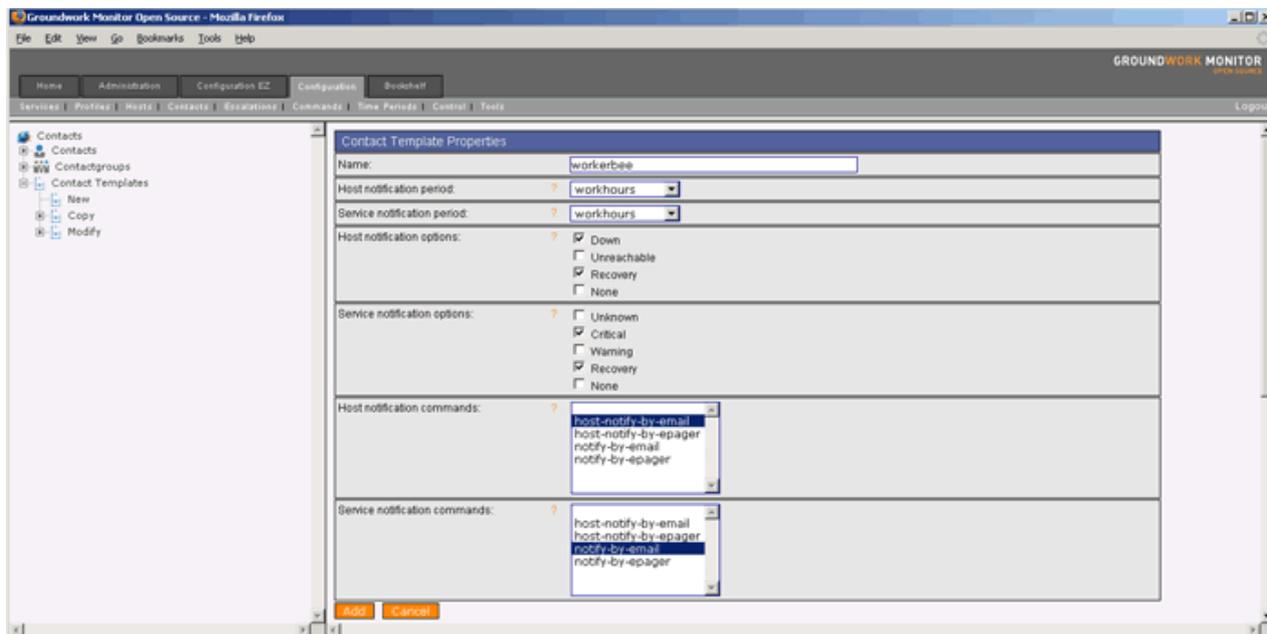


表 5.7.3. コンタクトテンプレートのプロパティ

Name	[必須] コンタクトテンプレートの名前
Host Notification Period	[必須] このディレクティブは、そのコンタクトにホストの障害や復旧について通知することができる時間帯のショートネームを指定するのに使います。これは、コンタクト(連絡先)がホストの通知に対して「オンコール:待機中」にあると考えることができます。これがどのように働き、誤って使った場合の潜在的な問題関する時間帯についてのより詳しい情報は、ドキュメントを参照してください。
Service Notification Period	[必須] このディレクティブは、そのコンタクトにサービスの障害や復旧について通知することができる時間帯のショートネームを指定するのに使います。これは、コンタクト(連絡先)がサービス通知に対して「オンコール:待機中」にあると考えることができます。これがどのように働き、誤って使った場合の潜在的な問題関する時間帯についてのより詳しい情報は、ドキュメントを参照してください。
Host Notification Options	[必須] このディレクティブは、ホストのステータスによってコンタクトにどの通知を送るかを定義するのに使います。適用されるオプションは下記のひとつか、組み合わせです: Downチェックは停止(DOWN)状態で通知を送り、Unreachable チェックは到達不能(UNREACHABLE)状態で通知を送り、Recovery チェックは復旧(recoveries:UP)状態で通知を送ります。もし、Noneをオプションとして指定したら、ホスト通知は送出されません。
Service Notification Options	[必須] このディレクティブは、サービスのステータスによってコンタクトにどの通知を送るかを定義するのに使います。適用されるオプションは下記のひとつか、組み合わせです: Warningチェックはワーニング(WARNING)状態で通知を送り、Unknownチェックは不明(UNKNOWN)状態で通知を送り、Critical チェックはクリティカル(CRITICAL)状態で通知を送ります。Nagios 2.xでは、ホストやサービスのフラッピングの開始と停止についての通知をすることもきるでしょう。もし、Noneをオプションとして指定したら、ホスト通知は送出されません。
Host Notification Commands	[オプション] このディレクティブは、ホストの障害や復旧をコンタクトに通知するのに使用するコマンドのショートネームのリストを定義するのに使います。すべての通知コマンドは、コンタクトに通知が必要なときに使用されます。通知コマンドの最大実行時間は、notification_timeout オプションで制御されます。
Service Notification Commands	[オプション] このディレクティブは、サービスの障害や復旧をコンタクトに通知するのに使用するコマンドのショートネームのリストを定義するのに使います。すべての通知コマンドは、コンタクトに通知が必要なときに使用されます。通知コマンドの最大実行時間は、notification_timeout オプションで制御されます。

追補6. Monarchを使ってエスカレーションを構成設定する

目次

どうやって、ホストやサービスのエスカレーションを構成設定するか？

どうやって、ホストやサービスのエスカレーションツリーを定義するか？

Monarch を使用したエスカレーション構成設定： どうやって、ホストやサービスのエスカレーションを構成設定するか？

ステップ1 - ホストやサービスエスカレーションを構成設定する

このステップでは、Monarch の構成設定ツールを使ってどのようにエスカレーションを追加するかを説明します。

1. Configuration メニューオプションから Escalations を選びます。
2. ドロップダウンメニュー Escalations を広げます。
3. ドロップダウンメニュー Host (かService) を広げます。
4. New を選びます。
5. Host (か Service) Escalation Properties 画面で、プロパティを入力します。
(項目の一覧と説明は、表 5.8.1を参照してください)
6. 新しいホスト(か サービス)エスカレーションを追加するために、Add を選びます。
7. ホスト(か サービス)エスカレーションをが追加されたとのメッセージが表示されるでしょう。
Continueを選びます。

図 5.8.1. Host (Service) Escalation Properties 画面

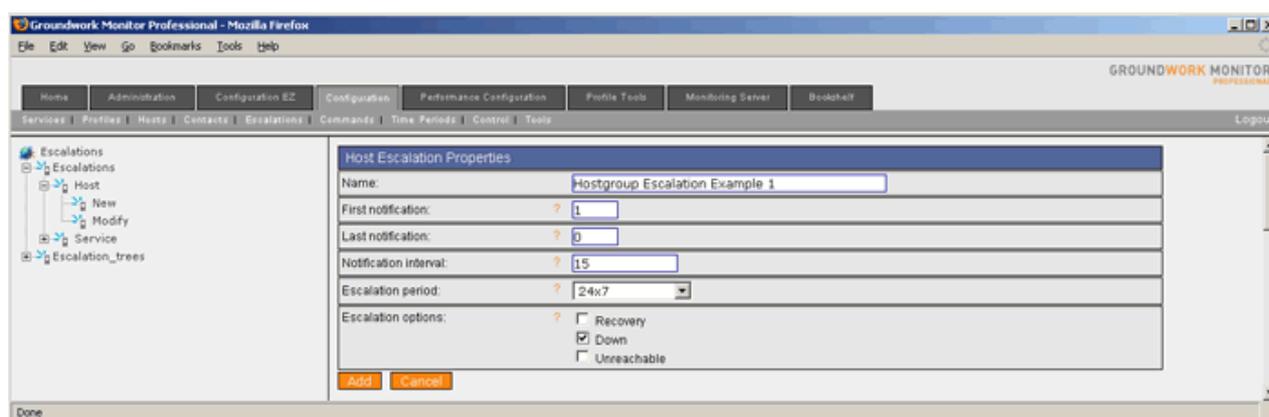


表 5.8.1 ホスト(やサービス)のエスカレーションのプロパティ

Name	ホスト(かサービス)のエスカレーションの名前
First Notification	[必須] このエスカレーションを、どの通知から始めるか？ このディレクティブはこのエスカレーションのための最初の通知を識別する数値です。たとえば 1、最初の通知から始めるとすると、このエスカレーションは最初の通知から通知のコントロールを行います。また、この値に 3 をセットしたら、ホストが停止(ダウン)か到達不能で3度目の通知を送出してから、あるいは、サービスがOKでない状態で3度目の通知を送出してからのみ、このエスカレーションが使用されます。
Last Notification	[必須] このディレクティブはこのエスカレーションが有効な最後の通知を識別する数値です。たとえば、この値を 5 にせつとしたら、このエスカレーションは、ホストやサービスの通知が5回以上送出されたらエスカレーションは使用されなくなります。この値 0 は、エスカレーションが通知のすべてを制御することを意味します。
Notification Interval	[必須] いつ通知を送出するべきか？ このディレクティブはこのエスカレーションが有効な間、通知を行うべき時間間隔(インターバル、interval_length ディレクティブの

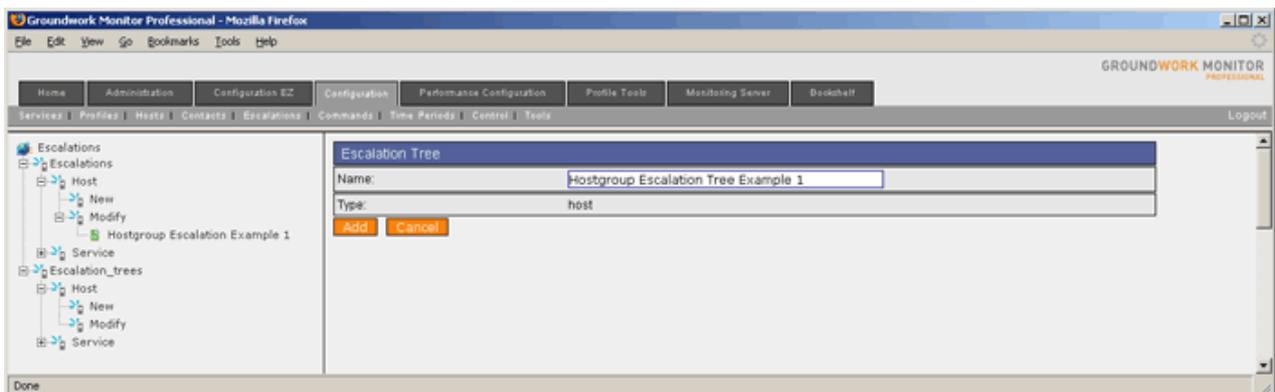
	<p>デフォルトの値 60 を変えない限り、この数値は「分」数を意味します。)を定義します。平均のインターバルは 15分です。インターバルとしてこの値を 0 に指定すると、Nagios は最初の通知のみ送出し、ホストからのそれ以上の障害通知送出手も抑制します。それ以外の値では、指定した間隔で連続した通知を行います。注意：ひとつのホストに複数のエスカレーションエントリがあって、ひとつ以上の通知のレンジが重複している場合、すべてのエスカレーション内の最小の通知インターバルが使われます。</p>
Escalation Period	<p>[オプション] このディレクティブは、このエスカレーションが有効な時間帯の名前(例: Work Hours, 24x7 no limit)を指定するのに使います。もし、このディレクティブが指定されないと、エスカレーションは常時有効であると見なされます。</p>
Escalation Options	<p>[オプション] いつ通知を創出すべきか? このディレクティブは、このホスト(あるいは、サービス)のエスカレーションがいつ(例: 復旧、停止、到達不能)使用されるかの基準を定義するのに使用します。エスカレーションは、ホスト(あるいは、サービス)がこのディレクティブで指定された状態になった場合にのみ使用されます。ホスト(あるいは、サービス)のエスカレーション内でこのディレクティブが指定されないと、すべてのホスト(あるいは、サービス)の状態でもエスカレーションが有効であると見なします。適用されるオプションは下記のひとつか、組み合わせです:</p> <p>Recovery = 復旧 (UP)状態になったときエスカレーションします、 Down = 停止 (DOWN)状態になったときエスカレーションします、 Unreachable = 到達不能 (UNREACHABLE)になったときエスカレーションします。</p> <p>例: このフィールドのDown を指定すると、エスカレーションはホスト(あるいは、サービス)が停止 (DOWN)状態になった場合にのみ使われます。</p>

Monarchを使用したエスカレーション構成設定： どうやって、ホストやサービスのエスカレーションツリーを定義するか？

ステップ1 - ホストやサービスのエスカレーションツリーを定義する

1. Configuration メニューオプションから Escalations を選びます。
2. ドロップダウンメニュー Escalation Trees を広げます。
3. ドロップダウンメニュー Host (かService) を広げます。
4. New を選びます。
5. ホストやサービスのエスカレーションツリーの名前を入力します。
6. 新しいホスト(か、サービス)エスカレーションを追加するために、Add を選びます。新しいエスカレーションツリーの名前がナビゲーションツリーのリストの中に現れるでしょう。この時点でエスカレーションツリーはデータベースに追加されますが、まだ実際のエスカレーションはなにもアサインされていません。ステップ2 - ホストのエスカレーションツリー変更、を続けてください。

図 5.8.2. ホスト(や サービス)の Escalation Tree画面での定義

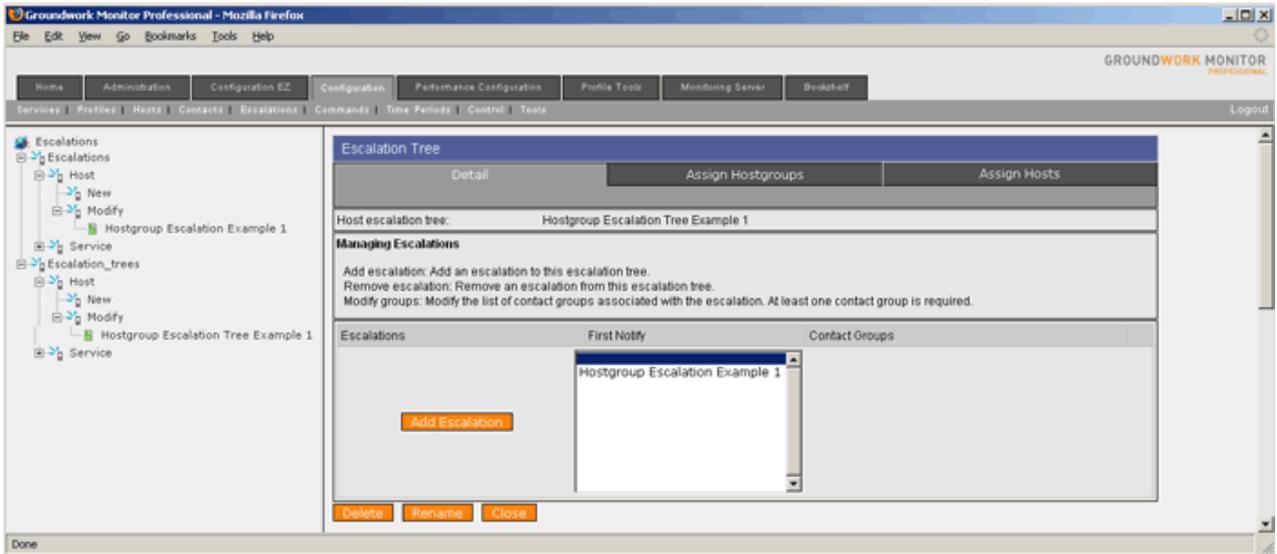


ステップ2 - ホストのエスカレーションツリー変更 - エスカレーションの追加

Modify を選んでエスカレーションツリーのエントリを選択することで、精巧な通知スキームを実施するための多くのエスカレーションで作りに上げたツリーに、エスカレーションを追加することができます。たとえば、テクニカルコンタクトグループは、最初から最後の通知に送られるエスカレーションにアサインされるべきです。ところが、異なるエスカレーションは5回目の通知で始まって6回目で終わり、事前の4回の上承(Acknowledge)されなかった場合にのみ通知を受けたいシニアマネージャへに行きます。

1. ドロップダウンメニューModify を開き、エスカレーションを追加するため、エスカレーションツリー名を選びます。
2. ここで、定義されたエスカレーションのリストが表示されます。エスカレーションを選び、Add Escalation ボタンをクリックします。ステップ3 - ひとつ以上のコンタクトグループの追加 に続きます。

図 5.8.3. ホストエスカレーションツリーの変更

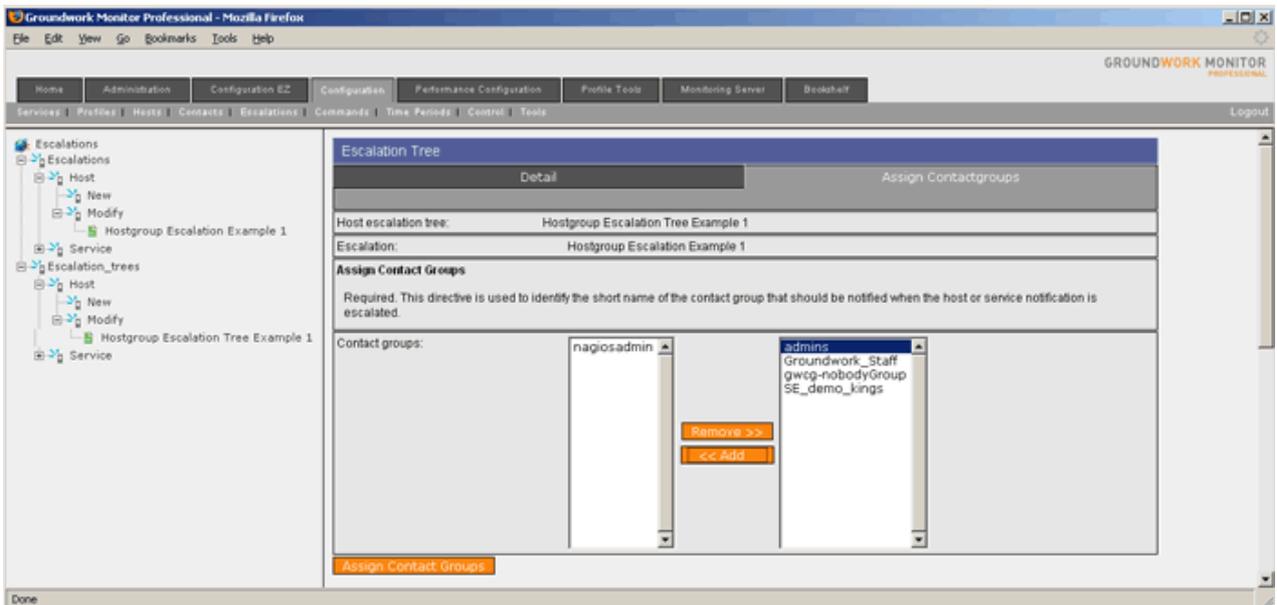


ステップ3 - ホストのエスカレーションツリー変更 - ひとつ以上のコンタクトグループの追加

選択されたコンタクトグループは、ホスト通知がエスカレートされたとき通知を受けます。

1. エスカレーションを追加すると、ひとつ以上のコンタクトグループの追加する必要がある、次の画面に移ります。
2. Assign Contact Groups を選びます。

図 5.8.4. コンタクトグループのアサイン

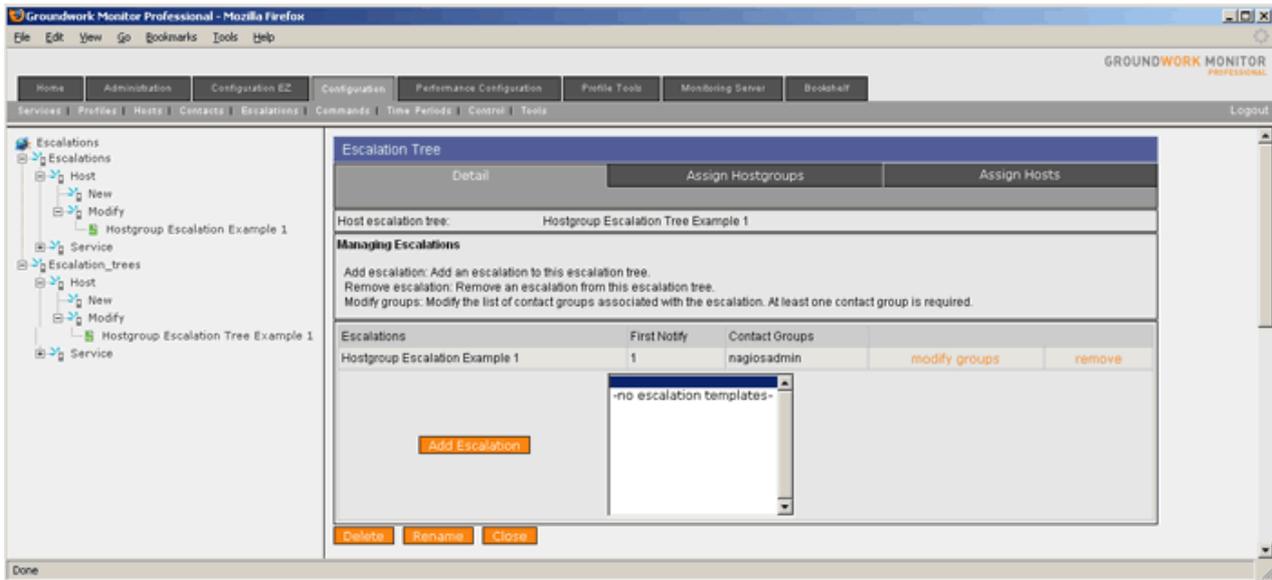


ステップ4 - ホストのエスカレーションツリー変更 - 追加のエスカレーションを加える(エスカレーションテンプレート)

ここで、エスカレーションツリーにホストグループやホストを追加します。

1. 選択したエスカレーションの、名前、最初の通知とアサインされたコンタクトグループが一覧表示されます。
2. また、ここで、どのエスカレーションにでも、Modify Groups を選んでエスカレーションに関係付けられたコンタクトグループのリストを変更するや、Removeを選んでエスカレーションツリーからエスカレーションを削除することができます。ステップ5 - ホストグループのアサイン に進みます。

図 5.8.5. ホストエスカレーションツリーの変更

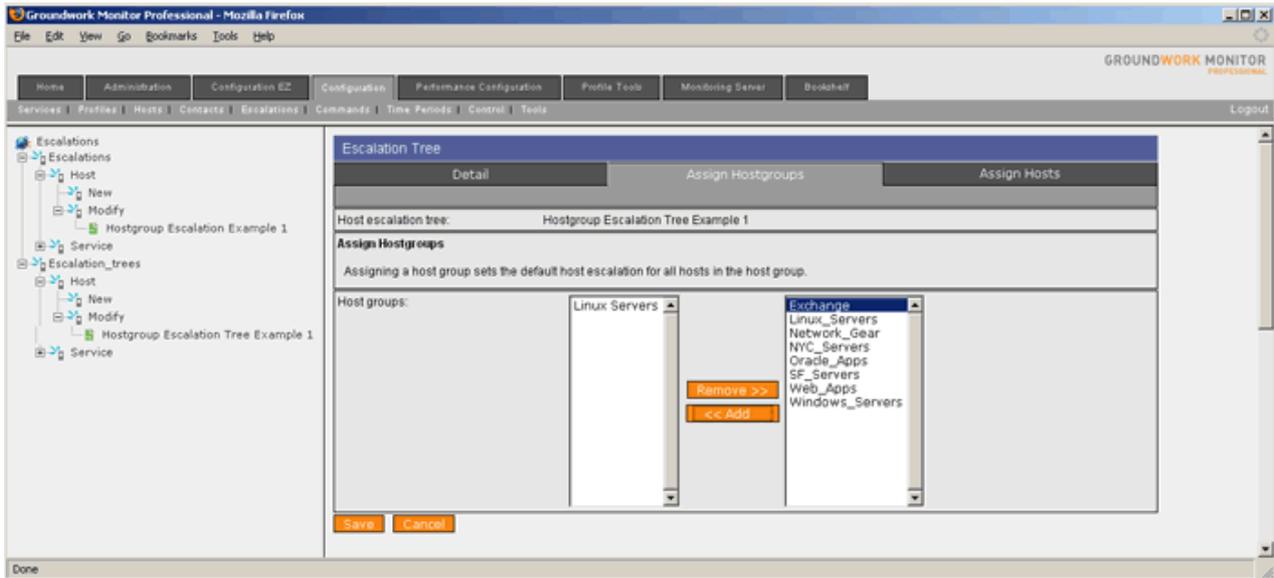


ステップ5 - ホストのエスカレーションツリー変更 - ホストグループのアサイン

いまや、ホストグループの中の全ホストのためにホストエスカレーションのデフォルトを設定する、ホストグループのアサインをしても良いです。また、ホストエスカレーションをホストにセットしてよいです。

1. Assign Host Groups タブを選びます。
2. エスカレーションツリーにアサインするため、ホストグループを選択します。
3. Save と Close を選びます。

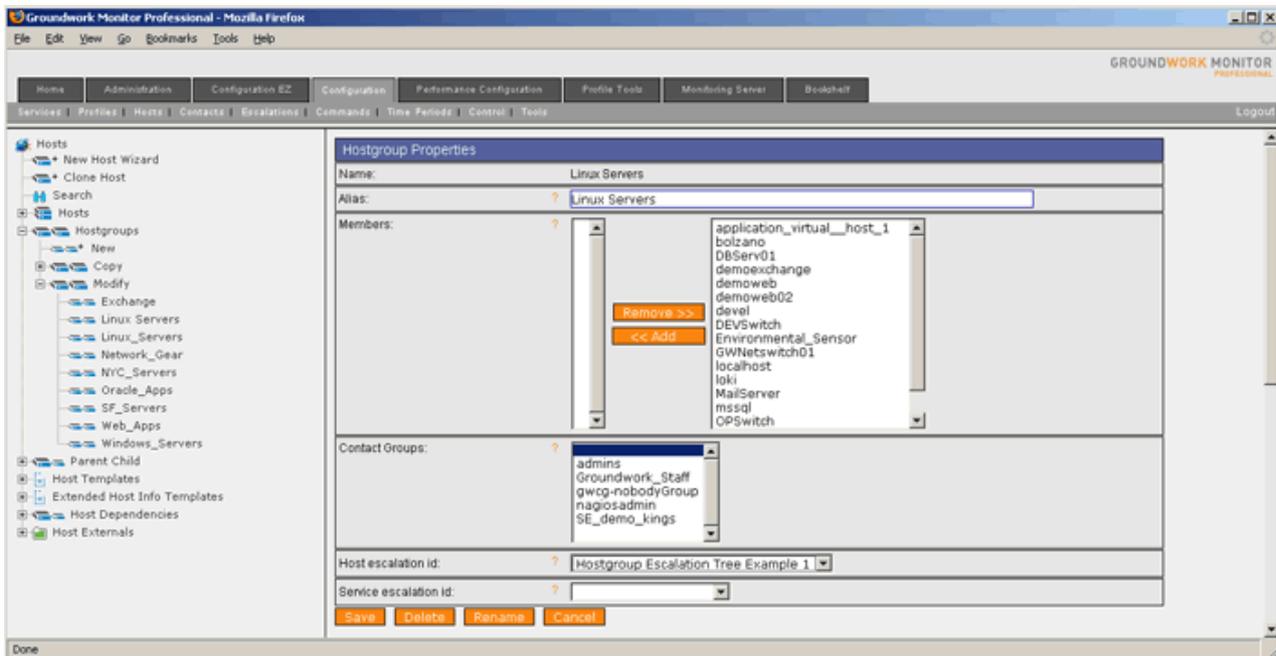
図 5.8.6. Assigning Host Groups 画面



注記 - エスカレーションをホストグループに追加するのは Host > Host Groups > Modify からでも実施可能です。

1. 注記として、Hosts > Host Groups > Modify として Linux Serverを選び、そこでドロップダウンリスト Host Escalation ID から Escalation Tree Example 1 を選択することで、このエスカレーションをホストグループ Linux Server に追加することができます。

図 5.8.7. Host>Host Groups>Modify からエスカレーションをホストグループに追加する



Nagios 構成設定オブジェクト定義

エスカレーションツリーが Nagios エンジンの cfg ファイルにコミットされたら、コンフィギュレータは 下記の Nagios 構成設定オブジェクトを生成します：

```
define hostescalation {
    use Hostgroup Escalation Example 1
    hostgroup_name Linux Servers
    contact_groups nagiosadmin
}

define hostescalation {
    name                Hostgroup Escalation Example 1
    first_notification  1
    last_notification   0
    notification_interval 15
    escalation_period   24x7
    escalation_options   d
    register            0
}
```

ホストエスカレーション定義についてより詳しくは、

http://nagios.sourceforge.net/docs/2_0/xodtemplate.html#hostescalation を参照してください。

追補7. Monarchを使ってコマンドを定義する

目次

どうやって、新しいコマンドを定義するか？

コマンド定義: どうやって、新しいコマンドを定義するか?

ステップ1 - リソース選択

1. Configuration メニューオプションから Commands を選びます。
2. New を選択します。
3. Select Resource 画面の中で構成設定を行うリソースマクロをチェックします。
4. 画面が変わり、選択したマクロのオプションが画面の上部に表示されます。
5. ドロップダウンリストからリソースマクロを関係付けるため、Plugin を選びます。
6. この時点で、リソースマクロを編集したりコメントしたるできるでしょう。
7. 続けるため、Next を選びます。

図 5.9.1. Select Resource 画面

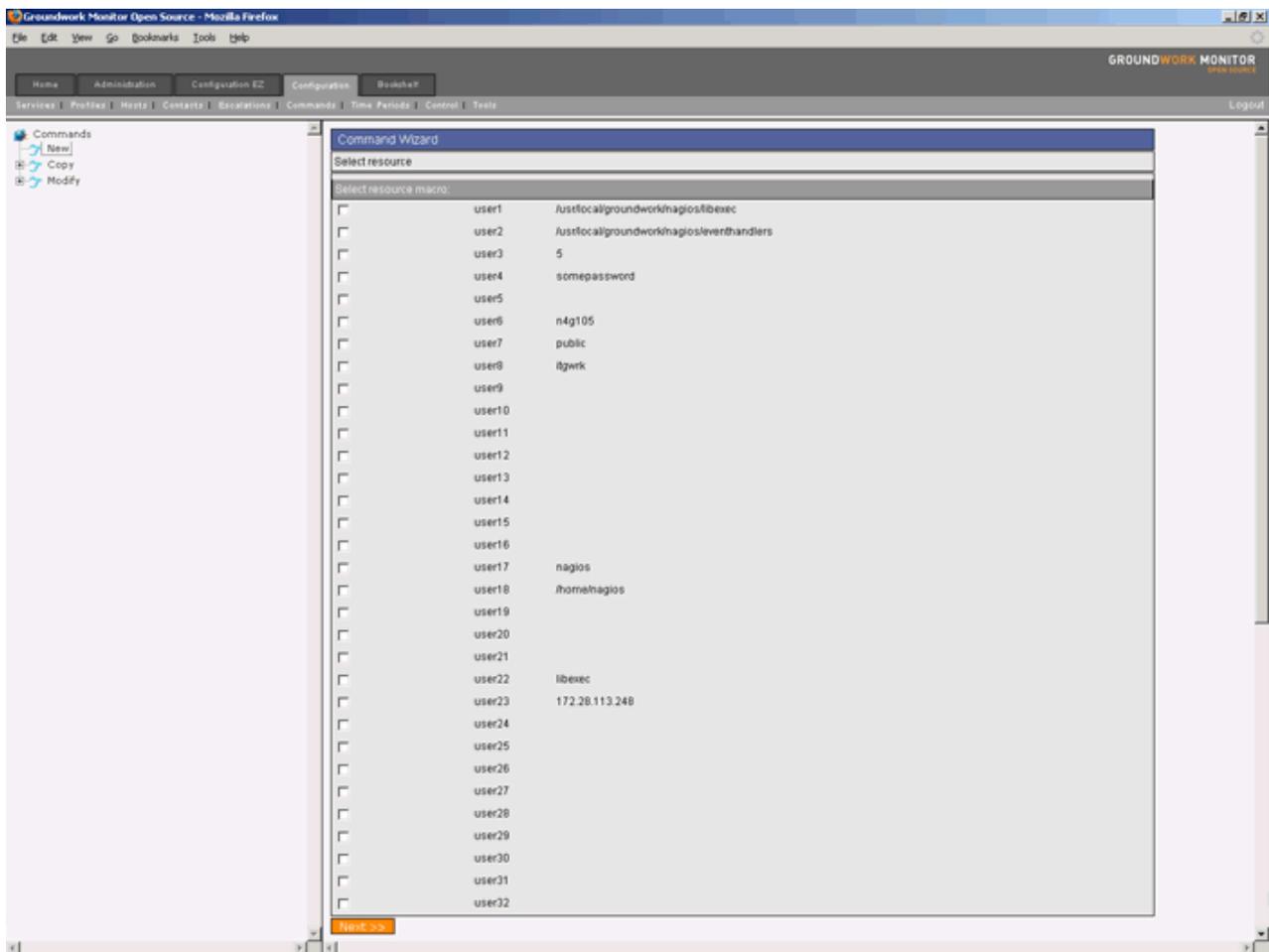


表 5.9.1. Select Resource 画面

Select Resource	これは、任意のリソースファイルを指定するのに使用します。リソースファイルには、\$USERn\$ 定義を含めることができます。\$USERn\$ マクロはユーザ名、パスワードと、(ディレクトリパスのような)コマンド定義で共通に使われる定義(ディレクトリパスのような)を格納するのに便利です。ユーザマクロは、ひとつ以上のリソースファイルで定義することができます。Nagios は、最大32のユーザマクロ (\$USER1\$ から \$USER32\$) をサポートします。
Plugin	ローカルやリモートのリソースあるいはある種のサービスをチェックする選択したプラグイン。これはコマンドラインに取り込まれます。
Comment	このマクロのための、任意のコメント領域です。

ステップ2 - 新規コマンド

1. Command Wizard 画面の中で、プロパティを入力します。
(項目の一覧と説明は、表 5.9.2 を参照してください)
2. 新規コマンド追加のため Add を選びます。Cancel はセーブしないで Command Wizard を終了します。
3. コマンドがセーブされたとのメッセージが表示されます。Continue を選びます。

図 5.9.2. 新規コマンド (Command Wizard 画面)

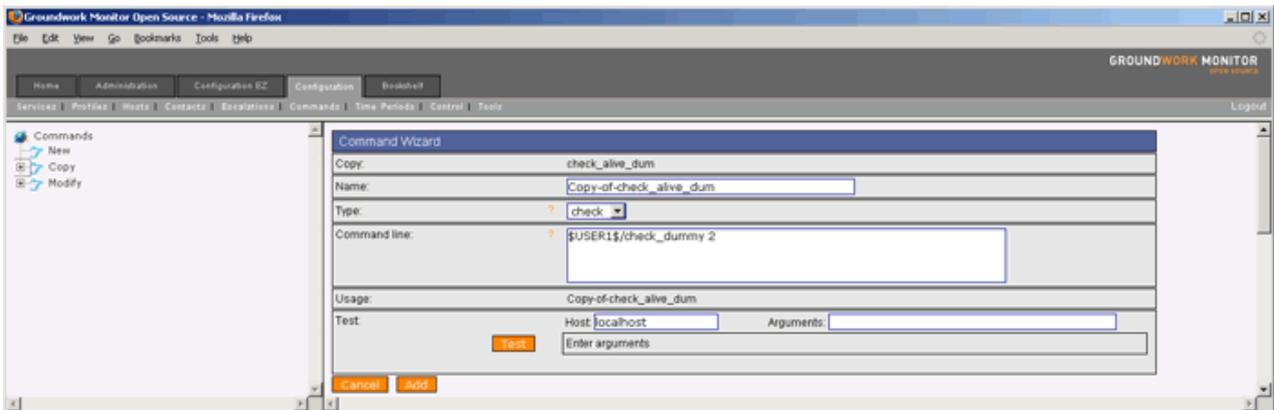


表 5.9.2. 新規コマンド (Command Wizard 画面)

Name	コマンド名
Type	コマンドのタイプを選びます: Check = サービス/ホストチェックのコマンド、ハンドラ; Notify = 通知コマンド; Other = 他のコマンド (つまり、パフォーマンス・データ・コレクタ)
Command Line	このディレクティブは、サービスやホストチェック、通知やイベントハンドラのためにコマンドが使用されたとき、Nagios によって実際に何が実行されるのかを定義するのに使います。編集可能なフィールドは、コマンドが実行される前に、すべての使用可能なマクロをそれぞれの値に置き換えます。異なるマクロを使う場合はマクロについてのドキュメントを参照してください。注意: コマンドラインは、コーテーション(')で囲まれてはいけません。また、ドルサイン (\$) をコマンドラインに渡したい場合は、もうひとつのドルサインをエスケープします。(つまり、\$\$ で、\$ 文字ひとつを表します)
Usage	これで、コマンドがサービス上でどのように定義されるか示します。
Test	コマンドラインのためのホストや引数を入力します。“Test” ボタンをクリックすると、これがコマンドラインに構文解析され、マクロ置換、Host ボックスからのホスト名の挿入、その結果のコマンドのシェルコマンドラインへの挿入および実行、そして 実行結果が右側のボックスに戻されます。

追補8. Monarchを使って時間帯を定義する

目次

どうやって、新しい時間帯を定義するか？

時間帯の定義： どうやって、新しい時間帯を定義するか？

ステップ1 – Time Period Properties 画面

1. Configuration メニューオプションから Time Periods を選びます。
2. New を選びます。
3. Time Period Properties 画面の中で、プロパティを入力します。
(項目の一覧と説明は、表 5.10.1 を参照してください)
4. 新しい時間帯を追加するため Add を選びます。次の画面で時間帯が追加されたとのメッセージが表示されるでしょう。

注意：既存の時間帯をコピーするには、Configuration メニューオプションから Copy を選んでください。すると、新しい時間帯に名前を付け、新しい時間帯のためにコピーされたプロパティを編集できるようになります。時間帯を削除や変更する場合は、Midify を選んでください。そうすると編集したり、変更した時間帯をセーブできます。選択した時間帯を削除するには Delete を選び、名称変更にはRename、変更なしに終了するにはCancel を選びます。

図 5.10.1. Time Period Properties 画面

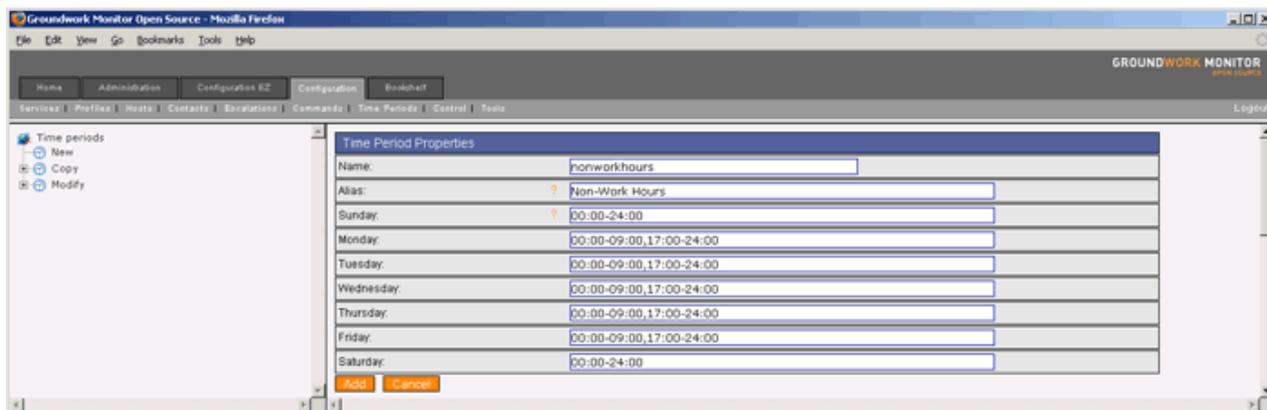


表 5.10.1. Time Period Properties

Name	通知やチェックのための時間帯の名前
Alias	[必須] このディレクティブは時間帯を識別するための、長めの名称か説明です。
Sunday-Saturday	[オプション] Sunday から Saturday のディレクティブは、特定の曜日の適用時間のためのコンマで区分した時間範囲のリストです。注意として、7つの異なる日(日曜から土曜)があり、それぞれで時間範囲を指定できます。 各時間範囲は HH: MM-HH: MM の形式で表現され、時間は24時間時計表されます。たとえば 00: 15-24: 00 は早朝(深夜)12時15分から真夜中の12時(総時間範囲、23時間と45分)を表します。この日のすべての時間を、時間帯から除きたい場合は、単に時間帯定義に何も入れないだけでよいです。

追補9. Monarch の Control オプションを使用する

目次

- どうやって、セットアップの中の設定を参照/編集するか？
- どうやって、Nagios CGI 構成のための設定を参照/編集するか？
- どうやって、Nagios メイン構成のための設定を参照/編集するか？
- どうやって、Nagios リソースマクロを参照するか？
- どうやって、Nagios 構成設定をロードするか？
- どうやって、ファイルとファイルの関連を追加するか？
- どうやって、飛行前チェック (Pre Flight Test) を実行するか？
- どうやって、Nagios 構成設定をコミットするか？

Control オプションを使用する：
 どうやって、セットアップの中の設定を参照/編集するか？

ステップ1 – セットアップ

1. Configuration メニューオプションから Control を選びます。
2. Setup を選びます。
3. Setup 画面が表示されます。(項目の一覧と説明は、表 5.11.1 を参照してください)
4. 編集した後、Save を選びます。次の画面で、Setupの変更が承認されたことが示されます。Close を選びます。Cancel は、変更をセーブしないで、Setup 画面を終了します。

図 5.11.1. Setup 画面

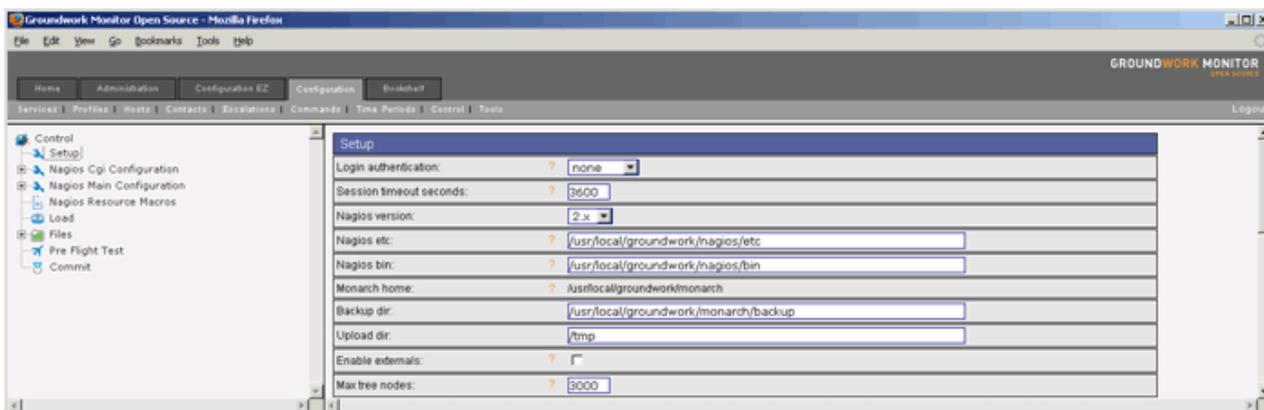


表 5.11.1. Setup

Login Authentication (ログイン認証)	none、active か passive を選びます。none = 認証なし-すべてのユーザは全コントロールを有します。active = ユーザは、アクセスチェックが可能なログインを促されます。passive = ユーザアカウントがURL内で渡されたがアクセスチェックが有効だったら、ログインなし(シングル・サイン・イン方式)。
Session Timeout (セッションのタイムアウト時間)	ログイン認証がアクティブの場合、これはユーザがログインを促されるまでにインアクティブになる秒数となります。
Nagios Version	使用している Nagios のバージョン
Nagios etc	nagios.cfg があるフォルダまでのパス。
Nagios Bin	nagios のバイナリがあるフォルダまでのパス
Monarch Home	Monarch のインストールパス。あなたのWebサーバは、書き込みと読み出しの権限を持っている必要があります。
Backup dir	このフォルダは、Nagios ファイルと Monarch データベースをバックアップするのに使用されます。あなたのWebサーバは、書き込みと読み出しの権限を持っている必要があります。
Max Tree Nodes	Hosts や Services の一覧をセグメント化しないで左枠のメニューツリーに表示できる数。この数が多いとページのロードに時間が掛かります。デフォルトは、500。

Control オプションを使用する：
 どうやって、Nagios CGI 構成のための設定を参照/編集するか？

ステップ1 – Nagios の CGI 構成設定

1. Configuration メニューオプションから Control を選びます。
2. Nagios CGI Configuration のドロップダウンメニューを広げます。
3. Load Nagios CGI. を選びます。
4. Setup 画面内で、プロパティを入力します。(項目の一覧と説明は、表 5.11.2 を参照してください)
5. Save and Next を選び、次の画面に進めます。

図 5.11.2. Nagios CGI Configuration Page 1 画面

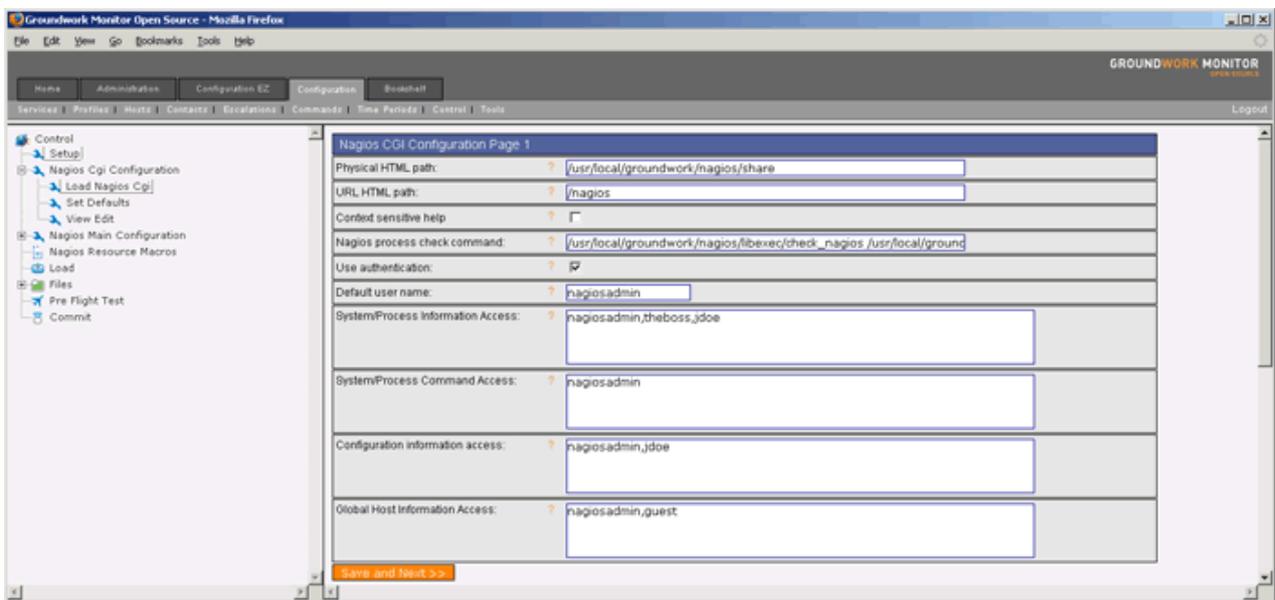


表 5.11.2. Nagios CGI Configuration Page 1

Physical HTML Path	これは、Nagios のためのHTMLファイルが置かれる場所を示します。この値は、statusmap と statuswrl の CGI に必要なロゴイメージの場所を指定するのに使われます。
URL HTML Path	これは、Nagios のためのHTMLファイル(上記で定義されている)の物理位置に一致するURLのパスの部分です。この値は、オンラインドキュメントとグラフィックの位置を示すためにCGIに使用されます。 あなたが http://www.myhost.com/nagios のようなURLで Nagios ページをアクセスするなら、この値は '/nagios' (' は除きます)となるべきです。
Context Sensitive Help	このオプションは、ほとんどのCGIで表示される、コンテキストセンシティブな(内容によって変わる)ヘルプアイコンかどうかを指定します。 値は: チェック = コンテキストセンシティブなヘルプを無効にします チェックなし=コンテキストセンシティブなヘルプを有効にします。
Nagios Check Command	これは、Nagios プロセスの状態をチェックするのに使われるプログラムのフルパスとファイル名です。これはCGI が使用するのみで、完全にオプション的なもので

	<p>す。しかしながら、それを使用しないと、CGI 内で、Nagios が稼動していないで、Webインタフェースからコマンドを実行できないという、ワーニングメッセージを見ることになるでしょう。そのプログラムは、プラグインと同じルールに従う必要があります:リターンコードはプラグインと同様であること、タイムアウト保護を持っていること、標準出力に出力すること、などです。(注意:もし、あなたがここで check_nagios のプラグインを使っているとすると、最初の引数はステータスログへの物理パス、二番目の引数はステータスログの内容がそれよりも新しくなるべき分数、三番目の引数は実行中の Nagios プロセスを見つけるための 'ps' コマンドの出力と一致する文字列となります。そのプロセスの文字列は、どのように Nagios が起動されたかによって変わります。Nagios プロセスでどのようなコマンドラインが入力されているかを見るには、'ps' コマンドを手動で起動してください。</p>
Use Authentication	<p>このオプションは、CGI がホストとサービス情報を表示する、また処理するために Nagios にコマンドをコミットするときに、何らかの認証を使うかどうかを制御します。HTML のドキュメントを読んで、どのように認証が働くかを学んでください。(注意:コマンドの CGI (cmd.cgi) を削除する考えでもない限り、認証を無効にするのは、大変良くない考えです。)間違えて指定しないと、誰でも Nagios を台無しにできるように開放したままになり、あなたのコマンドファイルへの継続書き込みによってドライブを満杯にする、サービス拒絶攻撃にあう可能性があります! この値の設定をチェックせずにすると、CGI が認証を使用しない(良くない考え)ことになり、チェックすると CGI が認証機能を使用します(デフォルト)。</p>
Default User	<p>この値は、認証をしないでページをアクセスできるデフォルトのユーザを定義します。これは、セキュアドメイン内(例、ファイアーウォール内)の人々が認証無しに現在状態を見ることができるようになります。基本認証はパスワードが分かる形で転送しますので、セキュリティサーバを使用しない場合、基本認証をしないで済むよう、この使用を望むかもしれません。重要:セキュアWebサーバを稼働させ、CGI にアクセスしたすべての人がある形で認証されていることを確認するまでは、デフォルトユーザを定義しないようにしましょう! この値を定義すると、Webサーバに認証されていない誰でもが、このユーザにアサインされた権限を継承してしまいます!</p>
System/Process Information Access	<p>このオプションは、拡張情報 CGI (extinfo.cgi) によって提供された Nagios プロセス情報を参照するためのアクセス権を有するすべてのユーザ名のコマンド区分リストです。デフォルトでは、あなたが認証を使用しないと選択しない限り、「誰も」アクセス権を持ちません。Webサーバへの認証をされた誰にでも、権利を与える場合は、アスタリスク(*)が使えます。</p>
System/Process Command Access	<p>このオプションは、コマンド CGI (cmd.cgi) から Nagios へのシャットダウンやリスタートコマンドを発行できるユーザ名のコマンド区分リストです。このリストのユーザは、プログラムのモードをアクティブやスタンバイに変えることができます。デフォルトでは、あなたが認証を使用しないと選択しない限り、「誰も」アクセス権を持ちません。Webサーバへの認証をされた誰にでも、権利を与える場合は、アスタリスク(*)が使えます。</p>
Configuration Information Access	<p>このオプションは、すべての構成設定情報(Hosts、commandsなど)を参照できる全ユーザ名のコマ区分リストです。デフォルトでは、ユーザは自分がコンタクト可能なホストとサービスの構成設定情報のみ参照できます。Webサーバへの認証をされた誰にでも、権利を与える場合は、アスタリスク(*)が使えます。</p>
Global Host Information Access	<p>このオプションは、すべてのホストのステータスと構成設定情報を参照できる、認証されたユーザ名のコマ区分リストです。このリスト内のユーザは、すべてのサービスの情報を見ることができるよう自動的に権限付与されます。このリスト内のユーザは、すべてのホストとサービスに対しコマンドを実行できるようには、自</p>

動的に権限付与されません。ユーザがすべてのホストとサービスに対しコマンドも実行できるようにしたいのであれば、`authorized_for_all_host_commands` 変数をそれらに追加する必要があります。

ステップ2 – Nagios の CGI 構成設定

1. Nagios CGI Configuration Page 2 画面内で、プロパティを入力します。
(項目の一覧と説明は、表5.11.3 を参照してください)
2. 変更を終わるため、Save と Done を選択します。
3. セーブ確認画面が表示されます。Continueを選びます。

図 5.11.3. Nagios CGI Configuration Page 2 画面

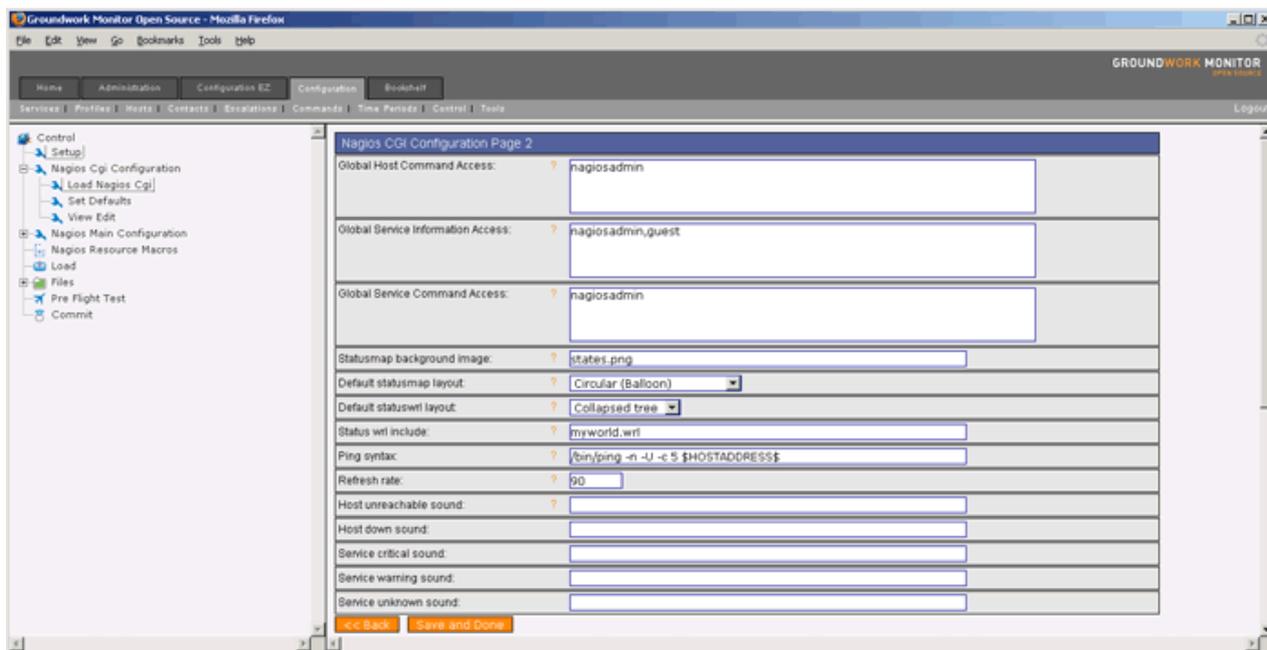


表 5.11.3. Nagios CGI Configuration Page 2

Global Host Command Access	このオプションは、コマンドCGI を介してすべてのホストへのコマンドを実行できる、認証されたユーザ名のコマ区分リストです。このリストのユーザは、すべてのサービスへのコマンド実行ができるよう自動的に権限付与されます。このリスト内のユーザは、すべてのホストとサービスのステータスや構成設定情報を参照できるようには、自動的に権限付与されません。ユーザがすべてのホストとサービスのステータスや構成設定情報を参照できるようにしたいのであれば authorized_for_all_hosts 変数をそれらに追加する必要があります。
Global Service Information Access	このオプションは、すべてのホストとサービスのステータスや構成設定情報を参照できる、認証されたユーザ名のコマ区分リストです。このリストのユーザは、すべてのホストの情報を見ることができるよう自動的に権限付与されません。このリスト内のユーザは、すべてのサービスへのコマンドを実行できるようには、自動的に権限付与されません。ユーザがすべてのサービスへのコマンドを実行できるようにしたいのであれば authorized_for_all_service_commands 変数をそれらに追加する必要があります。
Global Service Command Access	このオプションは、コマンドCGI を介してすべてのサービスへのコマンドを実行できる、認証されたユーザ名のコマ区分リストです。このリストのユーザは、すべてのホストへのコマンド実行ができるよう自動的に権限付与されません。このリスト内のユーザは、すべてのホストのステータスや構成設定情報を参照できるようには、自動的に権限付与されません。ユーザがすべてのサービスのステータスや構成設定情報を参照できるようにしたいのであれば authorized_for_all_services 変数をそれらに追加する必要があります。

Statusmap Background Image	このオプションで、statusmap CGI の中の背景として使われるイメージを指定することができます。イメージは、HTMLのイメージパス（例、/usr/local/nagios/share/images）にあると想定されています。このパスは、'physical_html_path' ディレクティブによって、自動的に '/images' を付けて定義されます。注意：イメージファイルは GD2 フォーマットでなければいけません。
Default Statusmap Layout	このオプションで、statusmap CGI がホストを描画するのに使うデフォルトのレイアウト方式を指定することができます。このオプションを使用しない場合、デフォルトは user-defined coordinates になります。適用できるオプションは下記です：User-defined coordinates、Depth layers、Collapsed tree、Balanced tree、Circular、Circular (Marked Up)
Default Statuswrl Layout	このオプションで、statuswrl (VRML) CGI がホストを描画するのに使うデフォルトのレイアウト方式を指定することができます。デフォルトは user-defined coordinates になります。適用できるオプションは下記です：User-defined coordinates、Collapsed tree、Balanced tree、Circular
Statuswrl Include	このオプションにより、生成したVRML世界の中にあなた自身のオブジェクトを入れることができます。それは、HTMLパス（例、/usr/local/nagios/share）の中にあるファイルと見なされます。
Ping Syntax	このオプションは、(statuswml CGI)を使って) WAPフェースからホストにPingを試みる時、どのシンタックスが使用されるかを決めます。pingのバイナリへのフルパスと必要なすべてのオプションを含める必要があります。\$HOSTADDRESS\$ マクロは、コマンド実行前にホストのアドレスに置き換えられます。
Refresh Rate	このオプションは、さまざまのCGI (status, statusmap, extinfo と outages) でのリフレッシュレートを秒数で指定します
Audio Alerts	これらのオプションで、ネットワーク上で問題があった時にブラウザのウィンドウで再生される任意のオーディオ（音声）ファイルを指定できます。オーディオファイルは、status CGI の中でのみ使用されます。最も重大な問題に対応する音のみが再生されます。重要性の順番（高いか低いか）は、下記のようになります：到達不能ホスト (unreachable Hosts)、ホストダウン (down Hosts)、クリティカル・サービス (critical Services)、ワーニング・サービス (warning Services)、未知・サービス (unknown Services)。明確な問題がなければ、オプションとして 'normal_sound' 変数で指定される音声ファイルが再生されます。注意：すべてのオーディオファイルは、HTMLパス（例 /usr/local/nagios/share/media/）の下の /media サブディレクトの中に置く必要があります

Control オプションを使用する：
どうやって、Nagios メイン構成のための設定を参照/編集するか？

ステップ1 – Nagios Main Configuration を使う

1. Configuration メニューオプションから、Control を選びます。
2. Nagios Main Configuration ドロップダウンメニューを開きます。
3. Load Nagios Cfg オプションを選びます。
4. Nagios Main Configuration Page 1 画面から、プロパティを入力します。
(項目の一覧と説明は、表 5.11.4 を参照してください)
5. Save and Next を選んで、次の画面に進めます。

図 5.11.4. Nagios Main Configuration Page 1 画面

The screenshot shows the 'Nagios Main Configuration Page 1' in the Groundwork Monitor web interface. The page contains the following configuration fields and options:

Log file:	<input type="text" value="/usr/local/groundwork/nagios/var/nagios.log"/>
Object cache file:	<input type="text"/>
Resource file:	<input type="text" value="/usr/local/groundwork/nagios/etc/resource.cfg"/>
Temp file:	<input type="text" value="/usr/local/groundwork/nagios/var/log/nagios.tmp"/>
Status file:	<input type="text" value="/usr/local/groundwork/nagios/var/status.log"/>
Aggregated status updates option:	<input checked="" type="checkbox"/>
Aggregated status data update interval:	<input type="text" value="15"/>
Nagios user:	<input type="text" value="nagios"/>
Nagios group:	<input type="text" value="nagios"/>
Enable notifications:	<input checked="" type="checkbox"/>
Execute service checks:	<input checked="" type="checkbox"/>
Accept passive service checks:	<input checked="" type="checkbox"/>
Execute host checks:	<input type="checkbox"/>
Accept passive host checks:	<input type="checkbox"/>
Enable event handlers:	<input checked="" type="checkbox"/>
Log rotation method:	<input type="text" value="d"/>
Log archive path:	<input type="text" value="/usr/local/groundwork/nagios/var/archives"/>
External command check option:	<input checked="" type="checkbox"/>
External command check interval:	<input type="text" value="-1"/>
External command file:	<input type="text" value="/usr/local/groundwork/nagios/var/spool/nagios.cmd"/>
Downtime file:	<input type="text" value="/usr/local/groundwork/nagios/var/downtime.dat"/>
Comment file:	<input type="text" value="/usr/local/groundwork/nagios/var/comments.dat"/>
Lock file:	<input type="text" value="/usr/local/groundwork/nagios/var/nagios.lock"/>
State retention option:	<input checked="" type="checkbox"/>
State retention file:	<input type="text" value="/usr/local/groundwork/nagios/var/retention.dat"/>
Retention update interval:	<input type="text" value="50"/>
Use retained program state:	<input checked="" type="checkbox"/>
Use retained scheduling info:	<input type="checkbox"/>

At the bottom of the form, there is a button labeled "Save and Next >>".

表 5.11.4. Nagios Main Configuration Page 1

Log File	これは、ヒストリカルな記録のため、サービスとホストイベントがロギングされるメインのログファイルです。これは、コンフィグ(構成)ファイルで最初に指定するべきオプションです！
Object Cache File	このディレクティブは、オブジェクト定義のキャッシュコピーが格納されるファイルを指定するのに使用します。キャッシュファイルは、Nagiosが起動されて、CGIによって使用される度に作成されます。これは、CGI内でのコンフィグファイルのキャッシングをスピードアップを目的とし、Nagios が実行中にCGI中出力表示に影響を及ぼさないで、元のコンフィグファイルを編集することができるようにします。
Resource File	これは、\$USERx\$ マクロ定義が含まれるオプションなリソースファイルです。複数の resource_file 定義を行うことで、複数のリソースファイルを指定することができます。CGI はリソースファイルの中身を読もうとしないので、慎重を期する情報(ユーザ名やパスワードなど)は、ファイル中ではマクロとして定義し、制限されたパーミッション(600)をこのファイルに適用することができます。
Status File	これは、すべての監視対象サービスとホストの現在状態を格納している場所です。この中身は、CGIによって読み出され処理されます。ステータスファイルの中身は、Nagios の再起動の度ごとに削除されます。
Aggregated Status Updates Option	このオプションは、Nagios がホスト、サービスおよびプログラムのステータスデータの更新を集中させるかどうかを決定します。このオプションをチェック(有効に)しないと、ステータスデータはホストやサービスチェックが発生する度毎に更新されます。これは、多くのサービスを監視すると、高いCPUとファイルI/O負荷を生じさせることとなります。もし Nagios に(ステータスファイル中の)ステータスデータを数秒(status_update_interval オプションで指定)毎にアップデートするだけいしたい場合、このオプションをチェック(有効に)します。これをしない正当な理由がない限り、(たとえ短い間隔であっても)この集中アップデートオプションの使用を強く推奨します。
Aggregated Status Data Update Interval	この設定は、Nagios がどの程度の間隔(秒数)でステータスファイルの中のステータスデータを更新するか決定します。最小の更新間隔は5秒です。Aggregated Status Updates をチェックしていない(aggregate_status_updates オプションで無効にしている)場合、このオプションは無効です。
Nagios User	これは Nagios を実行することのできるユーザを決定します。ユーザ名やUIDのどちらでも指定できます。
Nagios Group	これは Nagios を実行することのできるグループを決定します。グループ名やGIDのどちらでも指定できます。
Enable Notifications	このオプションで、Nagios が(再)起動された最初の時点で、通知を送出するかどうかを決定します。このオプションがチェックされない(無効だ)と、Nagios はどんなホストやサービスについても通知を送出しません。 注意: 状態保存(state retention)が有効にされていた場合、use_retained_program_state オプションを無効にしない限り、Nagios は(再)起動時にはこのセッティングを無視し、以前に知っていたこのオプションの設定を使います。状態保存(state retention)が有効(で、use_retained_program_state が enabled)の場合に、このオプションを変更したいなら、適当な外部コマンドをWebインタフェースから実行して変更する必要があります。
Execute Service Checks	このオプションで、Nagios が(再)起動された最初の時点で、サービスチェックを行うかどうかを決定します。このオプションがチェックされない(無効だ)と、Nagios は積極的にサービスチェックを行わず、いわゆる休眠(Sleep)モードのままとなります。(パッシブチェックは、それを無効にしない限り、実施できます。) このオプションは、冗長構成のドキュメントで説明されるように、監視サーバの構成をバックアップする場合か、分散監視環境のセットアップする時に、最もよく使用されます。注意: 状態保存(state retention)が有効にされていた場合、

	<p>use_retained_program_state オプションを無効にしない限り、Nagios は(再)起動時にはこの設定を無視し、以前に知っていた(状態保存ファイルに格納された)このオプションの設定を使います。状態保存(state retention)が有効(で、use_retained_program_state が enabled)の場合に、このオプションを変更したいなら、適当な外部コマンドをWebインタフェースから実行して変更する必要があります。</p>
Accept Passive Service Checks	<p>このオプションで、Nagios が(再)起動された最初の時点で、パッシブサービスチェックを行うかどうかを決定します。</p> <p>注意: 状態保存(state retention)が有効にされていた場合、use_retained_program_state オプションを無効にしない限り、Nagios は(再)起動時にはこの設定を無視し、以前に知っていた(状態保存ファイルに格納された)このオプションの設定を使います。状態保存(state retention)が有効(で、use_retained_program_state が enabled)の場合に、このオプションを変更したいなら、適当な外部コマンドをWebインタフェースから実行して変更する必要があります。</p>
Execute Host Checks	<p>このオプションで、Nagios が(再)起動された最初の時点で、オンデマンドと定期ホストチェックを行うかどうかを決定します。このオプションがチェックされない(無効だ)と、Nagios は積極的にホストチェックを行いませんが、パッシブチェックは、それを無効にしない限り、実施できます。このオプションは、冗長構成のドキュメントで説明されるように、監視サーバの構成をバックアップする場合か、分散監視環境のセットアップする時に、最もよく使用されます。</p> <p>注意: 状態保存(state retention)が有効にされていた場合、use_retained_program_state オプションを無効にしない限り、Nagios は(再)起動時にはこの設定を無視し、以前に知っていた(状態保存ファイルに格納された)このオプションの設定を使います。状態保存(state retention)が有効(で、use_retained_program_state が enabled)の場合に、このオプションを変更したいなら、適当な外部コマンドをWebインタフェースから実行して変更する必要があります。チェック = Host checksを実施。</p>
Accept Passive Host Checks	<p>このオプションで、Nagios が(再)起動された最初の時点で、パッシブホストチェックを受け付けるかどうかを決定します。このオプションがチェックされない(無効だ)と、Nagios はパッシブホストチェックを受け付けません。</p> <p>注意: 状態保存(state retention)が有効にされていた場合、use_retained_program_state オプションを無効にしない限り、Nagios は(再)起動時にはこの設定を無視し、以前に知っていた(状態保存ファイルに格納された)このオプションの設定を使います。状態保存(state retention)が有効(で、use_retained_program_state が enabled)の場合に、このオプションを変更したいなら、適当な外部コマンドをWebインタフェースから実行して変更する必要があります。チェック = イベントハンドラを有効にする。</p>
Enable Event Handlers	<p>このオプションで、Nagios が(再)起動された最初の時点で、イベントハンドラを起動するかを否かを決定します。このオプションがチェックされない(無効だ)と、Nagios はいかなるホストやサービスのイベントハンドラも実行しません。</p> <p>注意: 状態保存(state retention)が有効にされていた場合、use_retained_program_state オプションを無効にしない限り、Nagios は(再)起動時にはこの設定を無視し、以前に知っていた(状態保存ファイルに格納された)このオプションの設定を使います。状態保存(state retention)が有効(で、use_retained_program_state が enabled)の場合に、このオプションを変更したいなら、適当な外部コマンドをWebインタフェースから実行して変更する必要があります。</p>
Log Rotation Method	<p>これは、Nagios がメインログファイルのログローテーションに使用する方式です。値は下記です : n = None(なし) - ログをローテーションしない、h = Hourly : 時間単位ローテーション(毎時の頭で)、d = Daily: 日次ローテーション(毎日深夜12時)、w = Weekly : 週次ローテーション(土曜日深夜12時)、m = Monthly 月次ローテーション(月末の深夜12時)</p>
Log Archive Path	<p>これは、アーカイブ(ローテート)されたログファイルが置かれる場所です(ログローテーションを選択したとして)。</p>

Check External Commands	このオプションで、Nagios が外部コマンド(下記で指定するコマンドファイル)をチェックするかどうかを指定することができます。デフォルトでは、Nagios は外部コマンドをチェックしません、ただ警戒しているだけです。CGI コマンドインタフェースを使えるようにしたいならば、これを有効にしなければなりません。この値をチェックしない設定(デフォルト)はコマンドチェックを無効にし、チェックすると有効になります。
Command Check Interval	これは、Nagios が外部コマンドをチェックする時間間隔です。この値は、後で指定する interval_length を合わせて働きます。これをデフォルトの60(秒)のままにし、この値が1だと Nagios は毎1分ごとに外部コマンドをチェックします。数字の後ろに 's' を付け(例、15s)で指定すると、それは interval_length との掛算することなく、直接秒数に翻訳されます。注意:外部コマンドファイルを定常スケジュール間隔で読むのに加えて、Nagios はイベントハンドラの実行終了後も外部コマンドをチェックします。「注意」: 値として「-1」を指定すると、Nagios は外部コマンドをできるだけ頻繁にチェックします。
External Command File	これは、Nagios が外部コマンド要求のためにチェックするファイルです。また、コマンドCGI がユーザから提出されたコマンドを書き込むところでもありますので、Webサーバを起動させるユーザ(通常 'nobody')が書き込み権限を有している必要があります。その内容が処理されるたびにファイルが削除されますので、パーミッションはファイルレベルでなくディレクトリレベルに掛けておく必要があります。
Downtime File	これは、Nagios がホストとサービスの停止時間を格納するのに使うファイルです。
Comment File	これは、Nagios がホストとサービスのコメントを格納するのに使うファイルです。
Lock File	これは、Nagios がデーモンモードで実行しているときに、そのPID番号を格納するのに使うロックファイルです。
State Retention Option	このオプションで、Nagios がシャットダウンする前にサービスとホスト状態情報をセーブするかどうかを指定することができます。Nagios の開始時に、セーブされたすべてのサービスとホスト状態情報をモニターをスタートする前にリロードします。これは、状態統計の長期間データの保守などには便利ですが、Nagios の(再)起動時のダウンが少し長引きます。これは、単に1回だけの不利益なので、多少の起動遅れがあっても価値があると考えられます。チェック=有効にする。
State Retention File	これは、Nagios がホストとシャットダウン前にサービスとホスト状態情報をセーブするのに使うファイルです。このファイル中のサービスとホスト状態情報は、Nagios がリスタートした時にネットワークをモニターする直前に読まれます。このファイルは preserve_state_information 変数がチェックされている場合のみ使用されます。
Retention Update Interval	この設定で、通常オペレーションの間、どのくらいの頻度(分)でNagios が保存データを自動的にセーブするかを否かを決めます。この値を 0 にすると、Nagios は定期的に保存データをセーブしませんが、シャットダウンやリスタート時にセーブします。 上記の State Retention Option を無効(チェックなし)にすると、このオプションは機能しません。
Use Retained Program State	この設定で、保存(retention)ファイルの中にセーブされた値に従って、Nagios がプログラム状態変数をセットするかを否かを決めます。保存されたプログラム状態を使いたいなら、値を 1 にし、使いたくない場合は 0 にします。
Use Retained Scheduling Info Option	このオプションで、Nagios が(再)起動された最初の時点で、スケジュール情報を保持するかどうかを決定します。このオプションがチェックされない(無効だ)と、Nagios スケジュール情報を保持しません。 注意:状態保存(state retention)が有効にされていた場合、use_retained_program_state オプションを無効にしない限り、Nagios は(再)起動時にはこのセッティングを無視し、以前に知っていた(状態保存ファイルに格納された)このオプションの設定を使います。状態保存(state retention)が有効(で、use_ret

ained_program_state が enabled) の場合に、このオプションを変更したいなら、適当な外部コマンドをWebインタフェースから実行して変更する必要があります。チェック = スケジュール情報を保持する

ステップ2 – Nagios Main Configuration を使う

1. Nagios Main Configuration Page 2 画面で、プロパティを入力します (項目の一覧と説明は、表 5.11.5 を参照してください)。
2. Save and Next を選んで、次の画面に進みます。

図 5.11.5. Nagios Main Configuration Page 2 画面

表 5.11.5. Nagios Main Configuration Page 2

Syslog Logging Option	シスログ機構部や同様にNetAlarmログファイルにメッセージログを残したい場合、このオプションをチェックします。そうでなければ、チェックしません。
Log Notifications	通知をログに残したくない場合、この値をチェックしません。通知をログに残す場合、この値をチェックします。
Host Check Retry Logging Option	ホストチェックのリトライをログに残したくない場合、この値をチェックしません。リトライをログに残す場合、この値をチェックします。
Service Check Retry Logging Option	サービスチェックのリトライをログに残したくない場合、この値をチェックしません。リトライをログに残す場合、この値をチェックします。

Event Handler Logging Option	ホストとサービスのイベントハンドラをログに残したくない場合、この値をチェックしません。 イベントハンドラをログに残す場合、この値をチェックします。
Initial States Logging Option	Nagios にすべてのホストとサービスの初期状態(最初のサービスやホストのチェック)をメインログファイルにロギングさせたい場合、このオプションの値をチェックにし、有効にすることができます。 もし、あなたが長期の状態統計レポートを行う外部アプリケーションを使用しない場合、このオプションを有効にする必要はありません。 その場合は、値はチェックなしです。
External Command Logging Option	Nagios に外部コマンドのログをさせたくない場合、このオプションをチェックしないようにします。 外部コマンドのログをさせたい場合、このオプションをチェックします。 注意:このオプションは、パッシブサービスチェックのロギングを含んでいません - 下記のオプションでパッシブチェックをログするかどうかを確認してください。
Passive Check Logging Option	Nagios にパッシブサービスチェックのログをさせたくない場合、このオプションをチェックしないようにします。 パッシブサービスチェックのログをさせたい場合、このオプションをチェックします。
Global Host Event Handler	このオプションで、ホスト状態の変化毎に実行するイベントハンドラのコマンドを指定することができます。グローバルイベントハンドラは、各ホスト定義の中で任意に指定したイベントハンドラが実行される直前に、実行されます。 このコマンドの引数は、ホスト構成設定ファイルの中で定義した、コマンド定義のショートネーム(short name)です。 詳しくは、NagiosのHTMLドキュメントを参照ください。
Global Service Event Handler	このオプションで、サービス状態の変化毎に実行するイベントハンドラのコマンドを指定することができます。グローバルイベントハンドラは、各サービス定義の中で任意に指定したイベントハンドラが実行される直前に、実行されます。 このコマンドの引数は、ホスト構成設定ファイルの中で定義した、コマンド定義のショートネーム(short name)です。 詳しくは、NagiosのHTMLドキュメントを参照ください。
Sleep Time	これは、システムイベントのチェックと、実行する必要なあるサービスチェックの間にスリープする秒数です。この値はデフォルトの値 1 から変更しないことを推奨します。
Inter Check Delay Method	これは、Nagios がモニタリングを開始するとき、最初にサービスチェックを '分散させる'のに使用する方式です。デフォルトでは、全サービスチェックをCPU負荷を平均して最小になるように間を空けて実行しようとする、スマート(Smart)な遅延計算方式です。愚かな(Dumb)セッティングをするとすべてのチェックが同時に(それらの間に時間間隔を空けずに)スケジュールされます。それは、運用上望ましくありませんが、並行機能をテストする場合には便利です。注: なにも選択しない - チェックの間に遅延を入れない。Dumb: を選択 = チェックの間に 1秒遅延を設ける 'dumb' 遅延を使う。Smart: を選択 = 'smart' なチェック遅延計算を使う。Delay: を選択 = x.xx秒の遅延をチェック間に入れる。
Max Service Check Spread	このオプションは、Nagios スタート時からすべてのチェックを終えるまでの(定期チェックとしてスケジュールされる)最大の分(minutes)の数を定義します。このオプションは、最初のサービスチェックが指定された時間フレームの中で終わるように、(必要であれば)自動的にサービスのチェック間隔遅延を調整します。 use_retained_scheduling_info オプションを使ってスケジュール情報が保存されていると、通常は、このオプションはサービスチェックのスケジュールに影響を及ぼしません。
Service Interleave Factor	この変数は、サービスチェックがインターリーブされるかどうかを定義します。サービスのインターリーブはより均等なサービスチェックの分散とリモートホストを負荷低減ができるようにします。この値を 1 にセットすることは、Nagios の0.0.5までのバージョンと同様にサービスチェックを行います。値を s (smart)にすると、あなたが特別な理由で変更しない限りインターリーブの係数を自動計算します。Smart: を選択 = スマートな自動インターリーブ係数計算を使う。 数値を選択 = インターリーブ係数を x とする、この x は1以上の数字。

Service Reaper Frequency	これは、Nagios がチェックされたさーびすの結果を処理する頻度(秒)です。
Host Inter Check Delay Method	このオプションは、定期的にチェックするようスケジュールされたホストチェックを、どのように最初にイベントキューの中で '分散させる'かを制御できるようにします。'smart' 遅延(デフォルト)を使うことで、Nagios が平均のチェック間隔を計算し、すべてのホストに亘っての最初のチェックを分散し、これによりCPU負荷のスパイク除去を助けます。遅延をさせないことは、通常、推奨できません。遅延なしでは、ホストチェックがすべて同時に行われるようスケジュールされてしまいます。ホストチェックのスケジュールリングに影響するチェック間の遅延をどのように見積もるかについての追加情報がここにあります。この値は以下となります： None = チェックの間に遅延を入れない - すべてのホストチェックを直ちに(同時に)実行します！ Dumb = チェックの間に1秒遅延を設ける 'dumb' 遅延を使う。Smart =ホストチェックが均等に分散して実施されるよう 'smart' なチェック遅延計算を使う(デフォルト)。inter-check delay として x.xx 秒を指定する。
Max Host Check Spread	このオプションは、Nagios のスタート時からすべてのチェックを終えるまでの(定期チェックとしてスケジュールされる)最大の分(minutes)の数を定義します。このオプションは、最初のサービスチェックが指定された時間フレームの中で終わるように、(必要であれば)自動的にホストのチェック間隔遅延を調整します。 use_retained_scheduling_info オプションを使ってスケジュール情報が保存されていると、通常は、このオプションはサービスチェックのスケジュールに影響を及ぼしません。
Max Concurrent Service Checks	このオプションで、ある時点で並行して実行するサービスチェックの最大数を指定できます。値を 1 と指定するとどのサービスチェックも並行して行わないよう抑制されます。値の 0 では実行される並行チェックの数を制限しません。
Timing Interval Length	これは、ホスト/コンタクト/サービスの構成設定で使用されるユニット毎のインターバルの秒数です。これを 60 にすると、各インターバルが1分間(60秒)になることを意味します。他のセッティングはあまりチェックされませんので、どのくらいできるかはさまざまです...
Auto-rescheduling Option	このオプションは、Nagios がホストとサービスのアクティブチェックを長い間スムーズに行えるように、自動的に再スケジュールリングを実行するかどうかを定義します。より硬直したスケジュールでのチェック実行コストで、連続したチェックの間の時間を一定にしようとするので、これは監視サーバの負荷をバランスするのに役立ちます。チェック = 有効です。警告：これは実験的な機能であり、将来のバージョンで削除される可能性があります。このオプションを有効にすると - 不適切な使用によって - パフォーマンスを上げるよりも、低下させかねません。
Auto-rescheduling Interval	このオプションは、Nagios がどれほどの頻度(秒数)で、自動的にチェックの再スケジュールリングを行うかを定義します。このオプションは、auto_reschedule_checks オプションが有効の場合のみ有効です。警告：これは実験的な機能であり、将来のバージョンで削除される可能性があります。このオプションを有効にすると - 不適切な使用によって - パフォーマンスを上げるよりも、低下させかねません。
Auto-rescheduling Window	このオプションは、Nagios 自動的にチェックの再スケジュールリングを行う時に、調べる時間の'ウィンドウ'(秒数)を定義します。次の(この変数で指定された) X 秒間に起こる ホストとサービスチェックだけ、再スケジュールされます。このオプションは、auto_reschedule_checks オプションが有効の場合のみ有効です。警告：これは実験的な機能であり、将来のバージョンで削除される可能性があります。このオプションを有効にすると - 不適切な使用によって - パフォーマンスを上げるよりも、低下させかねません。

ステップ3 – Nagios Main Configuration を使う

1. Nagios Main Configuration Page 3 画面で、プロパティを入力します
(項目の一覧と説明は、表 5.11.6 を参照してください)。
2. Save and Next を選んで、次の画面に進みます。

図 5.11.6. Nagios Main Configuration Page 3 画面

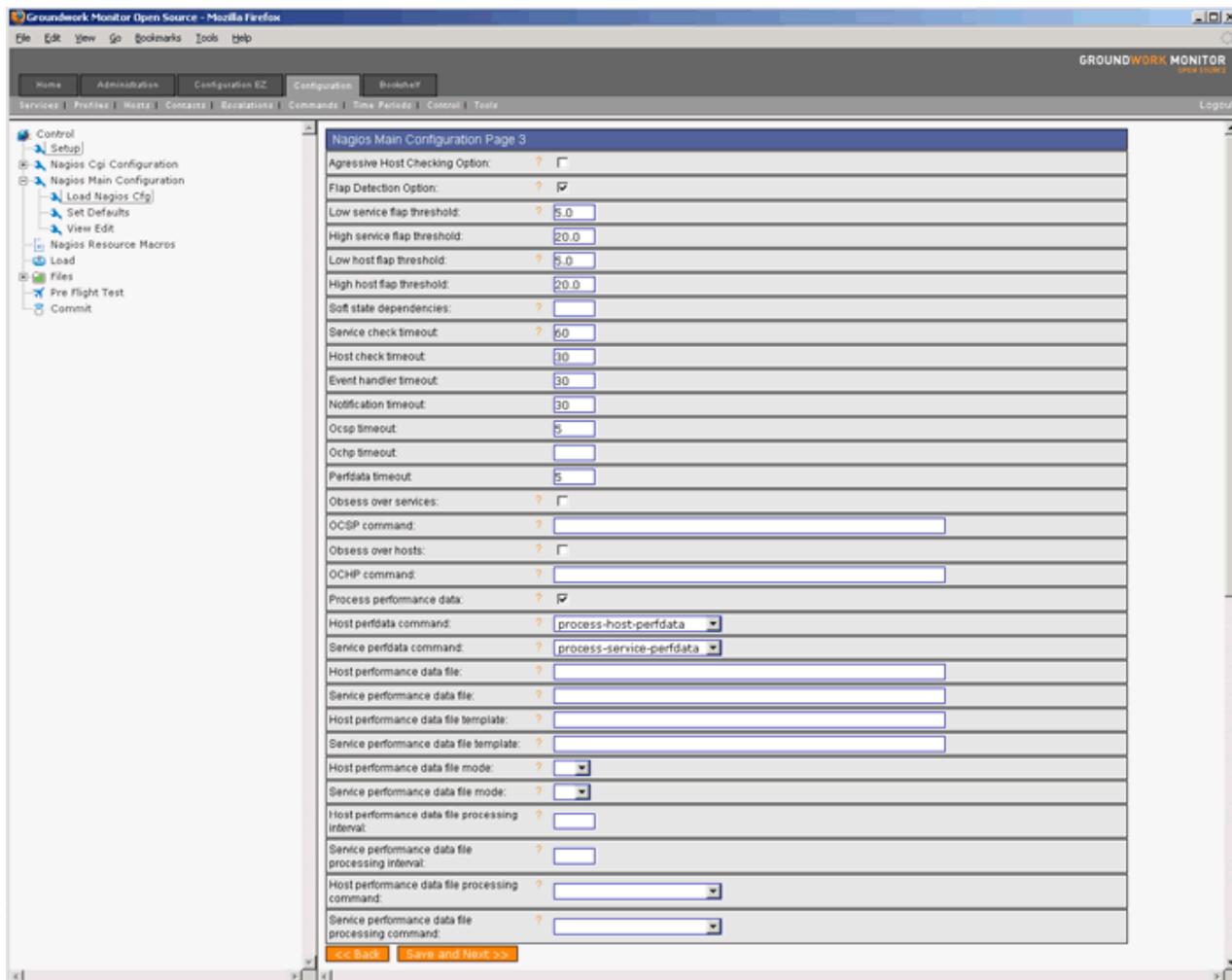


表 5.11.6. Nagios Main Configuration Page 3

<p>Aggressive Host Checking Option</p>	<p>Nagios はホストのステータスをいつ、どのようににチェックするかについて賢明に行おうとします。通常、このオプションを無効にし、Nagios により賢明な判断をさせて少しばかり早くチェックさせます。このオプションを有効にすると、ホストチェックに掛かる時間は増加しますが信頼性は少し向上します。Nagios がホストの復旧を認識しない問題が発生しない限り、このオプションを有効にしないことを強く推奨します。 値：チェック=Aggressive Host checkingを有効にする。 チェックなし = Aggressively check Hosts にしない。</p>
<p>Flap Detection Option</p>	<p>このオプションは、Nagios がホストとサービスのフラッピング (flapping) を検出しようとするか否かを定義します。フラッピングは、ホストやサービスのステータス変化が非常に激しい時に発生します。Nagios がホストとサービスのフラッピングを検出した時、ホスト/サービスのフラッピングが止まるまで一時的に通知を抑制します。フラップ検出は、非常に実験的なので、この機能を有効にする前に Nagios の HTMLドキュメントを読んでください。値：チェック=フラップ検出を有効にする。チェックなし = フラップ検出を無効にする(デフォルト)。</p>

Service Flap Detection Thresholds	このオプションについては、Nagios の HTML ドキュメントのフラップ検出についての説明を読んでください。このオプションはFlap Detection が無効の場合有効ではありません。
Host Flap Detection Thresholds	このオプションについては、Nagios の HTML ドキュメントのフラップ検出についての説明を読んでください。このオプションはFlap Detection が無効の場合有効ではありません。
Soft State Dependencies	このオプションは、Nagios がサービス依存性をチェックする際に、ソフトサービス状態を使うかどうかを定義します。通常、Nagios は依存性をチェックする際に、最新のハードサービス状態を使用します。(ハードかソフトかのタイプに関わらず)最新の状態を使わせたい場合は、このオプションを有効してください。
Timeout Values	このオプションは Nagios が、さまざまなタイプのコマンドについて、どのくらいの時間実行を許容するかをコントロールします。その時間が経ったら、killで実行を停止します。下記の実行最長時間をコントロールするオプションがあります : サービスチェック、ホストチェック、イベントハンドラ、通知、OCSPコマンド、パフォーマンスデータコマンド。すべての値は、秒単位です。
Obsess Over Services	これは、Nagios がサービスチェックを obsess over し、下記で定義する OCSPコマンドを実行するかどうかを定義します。分散モニタリングを行おうとしない限り、このオプションは、有効にしないで下さい。分散モニタリングについて詳しくは、NagiosのHTMLドキュメントを参照してください。値: チェック = obsess over サービスを有効にする。チェックなし = obsess over サービスを無効にする(デフォルト)。
OCSP Command	これは、Nagios によって処理されるサービスチェック毎に実行されるコマンドです。このコマンドは obsess_over_service オプションが、1 にセットされている場合のみ実行されます。コマンドの引数は、あなたのホスト構成設定ファイルで定義されたコマンドにショートネームです。分散モニタリングについて詳しくは、NagiosのHTMLドキュメントを参照してください。
Process Performance Data	これで、Nagios サービスとホストチェックから返されたパフォーマンスデータを処理するかどうかを定義します。もしこのオプションが有効だと、ホストパフォーマンスデータは Host perfddata command (下記で定義)で処理され、サービスパフォーマンスデータは Service perfddata command (これも、下記で定義)で処理されます。パフォーマンスデータについて詳しくは、NagiosのHTMLドキュメントを参照してください。値: チェック = パフォーマンスデータを処理にする。チェックなし = パフォーマンスデータを処理しない。
Host Perfddata Command	このオプションは、チェックから返されるであろうホストパフォーマンスデータを処理するため、すべてのホストチェックの後で実行するコマンドを指定します。コマンドの引数は、あなたのホスト構成設定ファイルで定義されたコマンドにショートネームです。このコマンドは、process_performance_data option がグローバルに有効で、ホスト定義の process_perf_data ディレクティブが有効である場合のみ実行されます。
Service Perfddata Command	このオプションは、チェックから返されるであろうサービスパフォーマンスデータを処理するため、すべてのサービスチェックの後で実行するコマンドを指定します。コマンドの引数は、あなたのオブジェクト構成設定ファイルで定義されたコマンドにショートネームです。このコマンドは、process_performance_data option がグローバルに有効で、サービス定義の process_perf_data ディレクティブが有効である場合のみ実行されます。
Host Performance Data File	このオプションは、各ホストチェック後にホストパフォーマンスデータが書き込まれるファイルを指定します。データは、host_perfddata_file_template オプションで指定されたように、パフォーマンスファイルに書かれます。パフォーマンスデータは process_performance_data オプションがグローバルに有効で、ホスト定義の process_perf_data ディレクティブが有効である場合のみ書き込まれます。
Service Performance Data	このオプションは、各サービスチェック後にサービスパフォーマンスデータが書き込

File	まれるファイルを指定します。 データは、service_perfdata_file_template オプションで指定されたように、パフォーマンスファイルに書かれます。パフォーマンスデータは process_performance_data オプションがグローバルに有効で、サービス定義の process_perf_data ディレクティブが有効である場合のみ書き込まれます。
Host Performance Data File Template	このオプションは、どの(また、どのように)データがホストパフォーマンスデータファイルに書き込まれるかを定義します。 そのテンプレートは、マクロ、特殊文字 ((\t :タブ、\r:キャリッジリターン、\n :改行)とテキストを含みます。 改行(NewLine)がパフォーマンスデータファイルへの書き込み毎に自動的に追加されます。
Service Performance Data File Template	このオプションは、どの(また、どのように)データがサービスパフォーマンスデータファイルに書き込まれるかを定義します。 そのテンプレートは、マクロ、特殊文字 ((\t :タブ、\r:キャリッジリターン、\n :改行)とテキストを含みます。 改行(New Line)が、パフォーマンスデータファイルへの書き込み毎に自動的に追加されます。
Host Performance Data File Mode	このオプションは、ホストパフォーマンスデータファイルが、書き込み(Write)モードでオープンされるか、追加(Append)モードでされるかを定義します。ファイルが名前付きパイプでない限り、おそらくデフォルトの追加(append)を使うでしょう。 a = 追加(Append)モードでファイルを開く、w = 書き込み(Write)モードでファイルを開く。
Service Performance Data File Mode	このオプションは、サービスパフォーマンスデータファイルが、書き込み(Write)モードでオープンされるか、追加(Append)モードでされるかを定義します。ファイルが名前付きパイプでない限り、おそらくデフォルトの追加(append)を使うでしょう。 a = 追加(Append)モードでファイルを開く、w = 書き込み(Write)モードでファイルを開く。
Host performance Data File Processing Interval	このオプションは、ホストパフォーマンスデータファイルがHost Performance Data File Processing Command(下記)で処理されるインターバル(秒)を指定します。 値 0 はパフォーマンスデータファイルは定期的に処理されないことを示します。
Service Performance Data File Processing Interval	このオプションは、サービスパフォーマンスデータファイルがService Performance Data File Processing Command(下記)で処理されるインターバル(秒)を指定します。 値 0 はパフォーマンスデータファイルは定期的に処理されないことを示します。
Host Performance Data File Processing Command	このオプションで、ホストパフォーマンスデータファイルを処理する際に実行するコマンドを指定できます。 そのコマンド引数は、あなたのオブジェクト構成設定ファイルで定義されたコマンドのショートネームです。 このコマンドを実行するインターバルは、host_perfdata_file_processing_interval ディレクティブで定義されます。
Service Performance Data File Processing Command	このオプションで、サービスパフォーマンスデータファイルを処理する際に実行するコマンドを指定できます。 そのコマンド引数は、あなたのオブジェクト構成設定ファイルで定義されたコマンドのショートネームです。 このコマンドを実行するインターバルは、service_perfdata_file_processing_interval ディレクティブで定義されます。

ステップ4 - Nagios Main Configuration を使う

1. Nagios Main Configuration Page 4 画面で、プロパティを入力します (項目の一覧と説明は、表 5.11.7 を参照してください)。
2. Save and Done を選び、変更を保ちます。
3. あなたのセーブを確認する画面が表示されます。Continueを選びます。

図 5.11.7. Nagios Main Configuration Page 4 画面

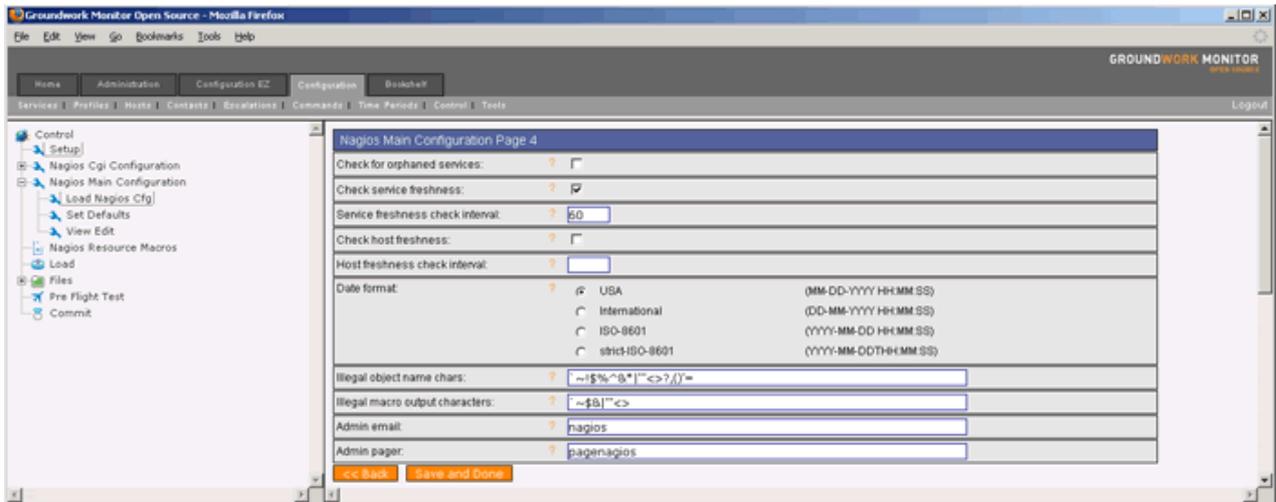


表 5.11.7. Nagios Main Configuration Page 4

Check for Orphaned Services	これはNagiosが見捨てられた(Orphaned)サービスを定期的チェックするかどうかを決めます。サービスチェックは、以前の実行インスタンスの結果が処理されるまで再スケジュールされないため、いくつかのチェックは二度と再スケジュールされなくなるかもしれません。これは稀な問題で通常の状況では起こりえないはずですが。もしサービスチェックが再スケジュールされないという問題があったら、このオプションを有効にしてみたいかなるでしょう。値: チェック = チェックを有効にする、チェックなし = チェックを無効にする。
Check Service Freshness Option	このオプションは Nagios が定期的にサービス結果の「新鮮度 (freshness)」をチェックするかどうかを定義します。このオプションを有効にすると、パッシブチェックが時間どおりに受信されていることを確実にするのに便利です。値: チェック = 新鮮度チェックを有効にする、チェックなし = 新鮮度チェックをディスエイブルにする。
Service Freshness Check Interval	このセッティングは、Nagios がどのくらいの頻度(秒数)で、サービスチェック結果の「(新鮮度 (freshness))」をチェックするかを定義します。 Service freshness checkingが無効の場合、このオプションは無効です。
Check Host Freshness Option	このオプションは Nagios が定期的にホストチェック結果の「新鮮度 (freshness)」をチェックするかどうかを定義します。このオプションを有効にすると、パッシブチェックが時間どおりに受信されていることを確実にするのに便利です。 値: チェック = 新鮮度チェックを有効にする
Host Freshness Check Interval	このセッティングは、Nagios がどのくらいの頻度(秒数)で、ホストチェック結果の「(新鮮度 (freshness))」をチェックするかを定義します。Host freshness checkingオプションが無効の場合、このオプションは無効です。
Date Format	このオプションは、短い日付をどのように表示するかを定義します— 米国: MM-DD-YYYY HH:MM:SS、欧州: DD-MM-YYYY HH:MM:SS、 ISO8601 YYYY-MM-DD HH:MM:SS、厳密-ISO8601 YYYY-MM-DDTHH:MM:SS
Illegal Object name Chars	このオプションで、ホスト名、サービス説明、オブジェクトタイプの名称で使用できない、不正キャラクタを指定できます。
Illegal Macro Output Chars	このオプションで、通知、イベントハンドラなどで使用される前に、マクロから剥ぎ取られる不正キャラクタを指定できます。これは、サービスやホストチェックコマンド内で使用されるマクロには影響しません。以下のマクロであなたが指定したキャラクタが剥ぎ取られます: \$OUTPUT\$, \$PERFDATA\$.
Admin Email	(監視を行っている)、このマシンの管理者eメールアドレスです。 Nagios 自体では

	この値を使いませんが、\$ADMINEMAIL\$ マクロで、通知コマンドの中でこの値をアクセスすることができます。
Admin Pager	このマシンの管理者のページャ番号/アドレスです。 Nagios 自体ではこの値を使いませんが、\$ADMINPAGER\$ マクロで、通知コマンドの中でこの値をアクセスすることができます。

Control オプションを使用する： どうやって、Nagios リソースマクロを参照するか？

ステップ1 -Nagios リソースマクロ

1. Configuration メニューオプションから Control を選びます。
2. Nagios Resource Macrosを選びます。
3. Nagios Resource Macros 画面から、構成設定するマクロを選びます。
4. 変更をセーブするため、Update を選びます。

図 5.11.8. Nagios Resource Macros画面

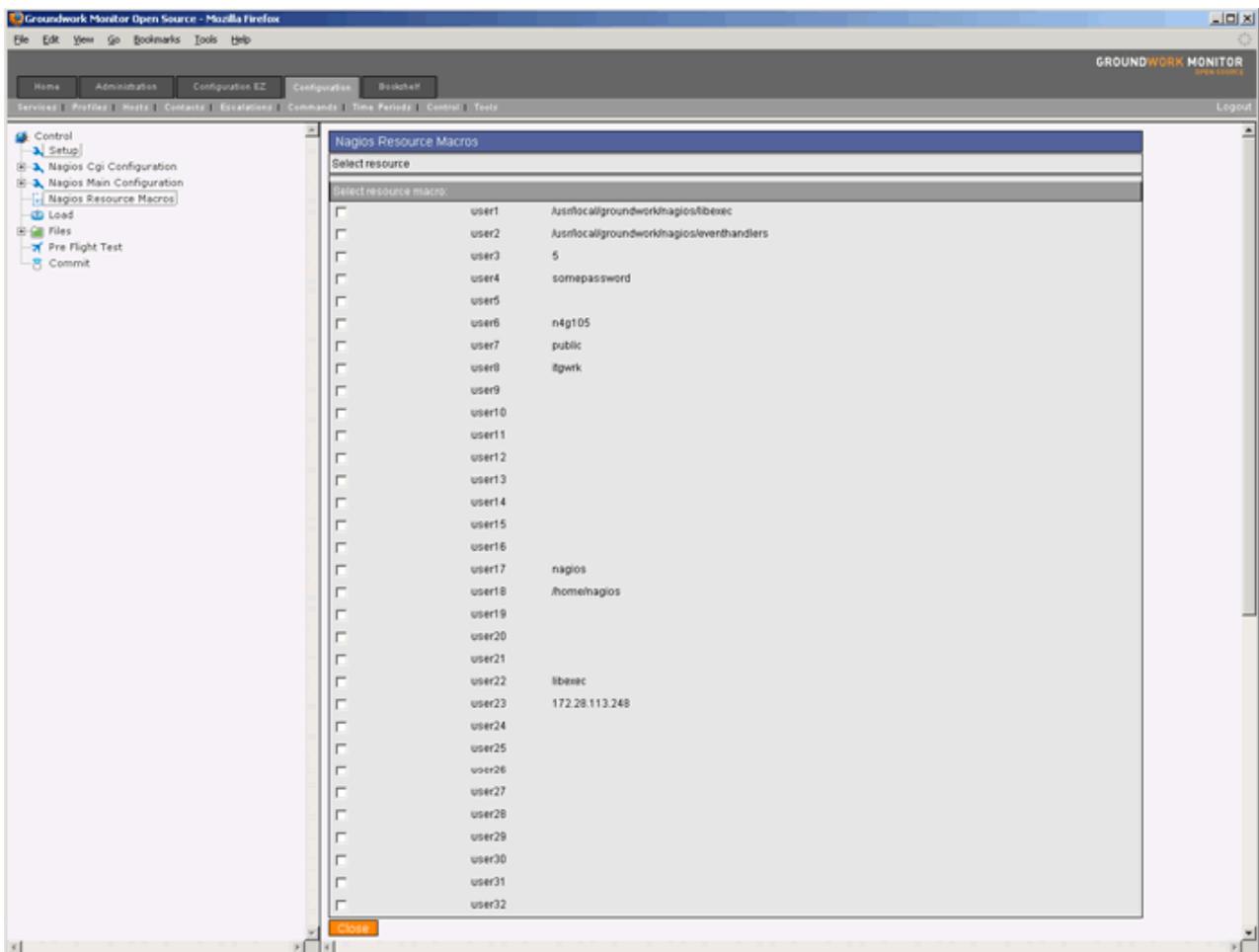


表 5.11.8. Nagios Resource Macros

Select Resource Macro

作成/編集するリソースマクロのチェックボックスをチェックします。そうすると、マクロは、画面のトップに表示されます。選んだユーザーのためのマクロを入力します。また、リソースマクロのリストにマウスを乗せると表示されるコメントを追加してもよいです。変更をセーブするため、Update を選びます。変更が下の画面に表示されます。

Control オプションを使用する： どうやって、Nagios 構成設定をロードするか？

ステップ1 - Nagios 構成設定をロードするか

1. Configuration メニューオプションから Control を選びます。
2. Load を選びます。
3. Load Nagios Configuration 画面中に、Nagios 構成設定をアップデートする前に Backup オプションを使うよう提案が表示されます。
4. Backup を選んだ後、次の画面で、Backup の確認とどこにおくかのリストが表示されます。続けて Load を選びます。
5. Load Nagios Object Files 画面の中に Nagios のオブジェクトファイルがリストされます。Next を選ぶと Nagios のオブジェクトファイルが処理されます。
6. 次のLoad Nagios Object Files 画面で、オブジェクトファイル処理が終了したことがほのめかされます。ホストを処理するため、Next を選びます。
7. 次のLoad Nagios Object Files 画面で、ホストファイル処理が終了したことがほのめかされます。サービスを処理するため、Next を選びます。
8. 次のLoad Nagios Object Files 画面で、サービスファイル処理が終了したことがほのめかされます。エスカレーションを処理するため、Next を選びます。
9. 次のLoad Nagios Object Files 画面で、エスカレーションを保持するか、置き換えるかを確認されます。
10. 最後の画面で、Nagios ファイルのロードが完了したことが報告されます。Continue を選んで終了します。

図 5.11.9. Load Nagios Configuration 画面

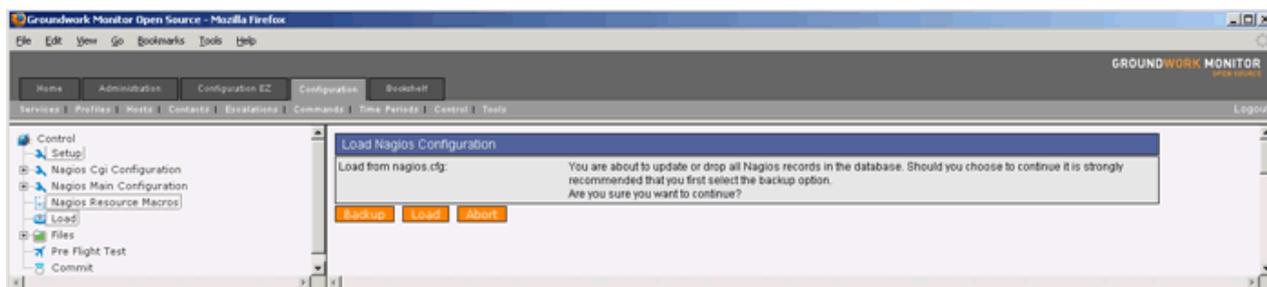


表 5.11.9. Load Nagios Configuration

Load	あなたが確かに、データベース内の Nagios レコードを更新あるいはドロップしたい場合、Load を選びます。続けることを選ぶには、先に Backup オプションを選ぶことを強く勧めます。
Backup	データベース内の Nagios レコードをバックアップするため、Backup を選びます。
Abort	Nagios レコードのローディングを中止もしくはキャンセルします。

Control オプションを使用する： どうやって、ファイルとファイルの関連を追加するか？

ファイルとファイル関連

1. Configuration メニューオプションから Control を選びます。
2. ドロップダウンメニューFiles を開きます。
3. File Add を選びます。
4. File Name と File path を入力します。ドロップダウンオプションから File Typeを選びます。
5. 新しいファイルを含めるため、Add を選びます。
6. すべてのファイル関連のリストに、今作成されたものがあるのを見るため、File Assn を選びます

図 5.11.10. Add File 画面

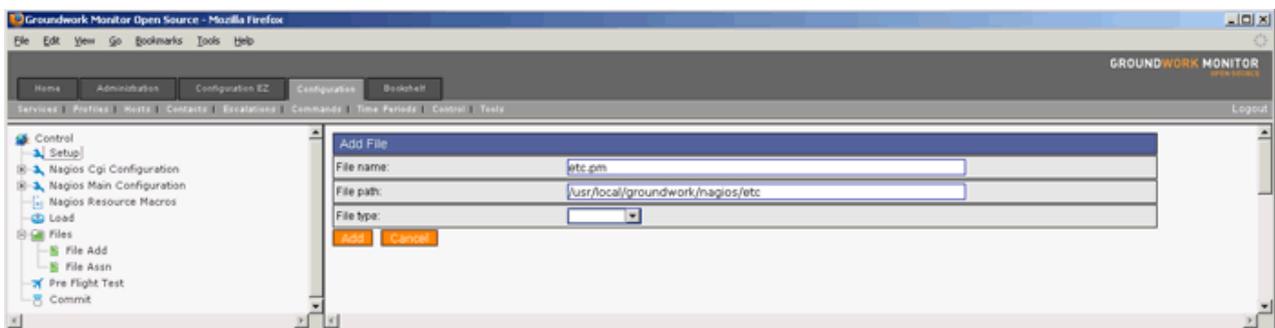


表 5.11.10. Add File

File Name	構成設定ファイルの名前
File Path	構成設定ファイルのパス。デフォルトがあなたのためにセットされています。
File Type	ファイルのタイプを入力します。構成設定ファイルでは Hosts か Services です。

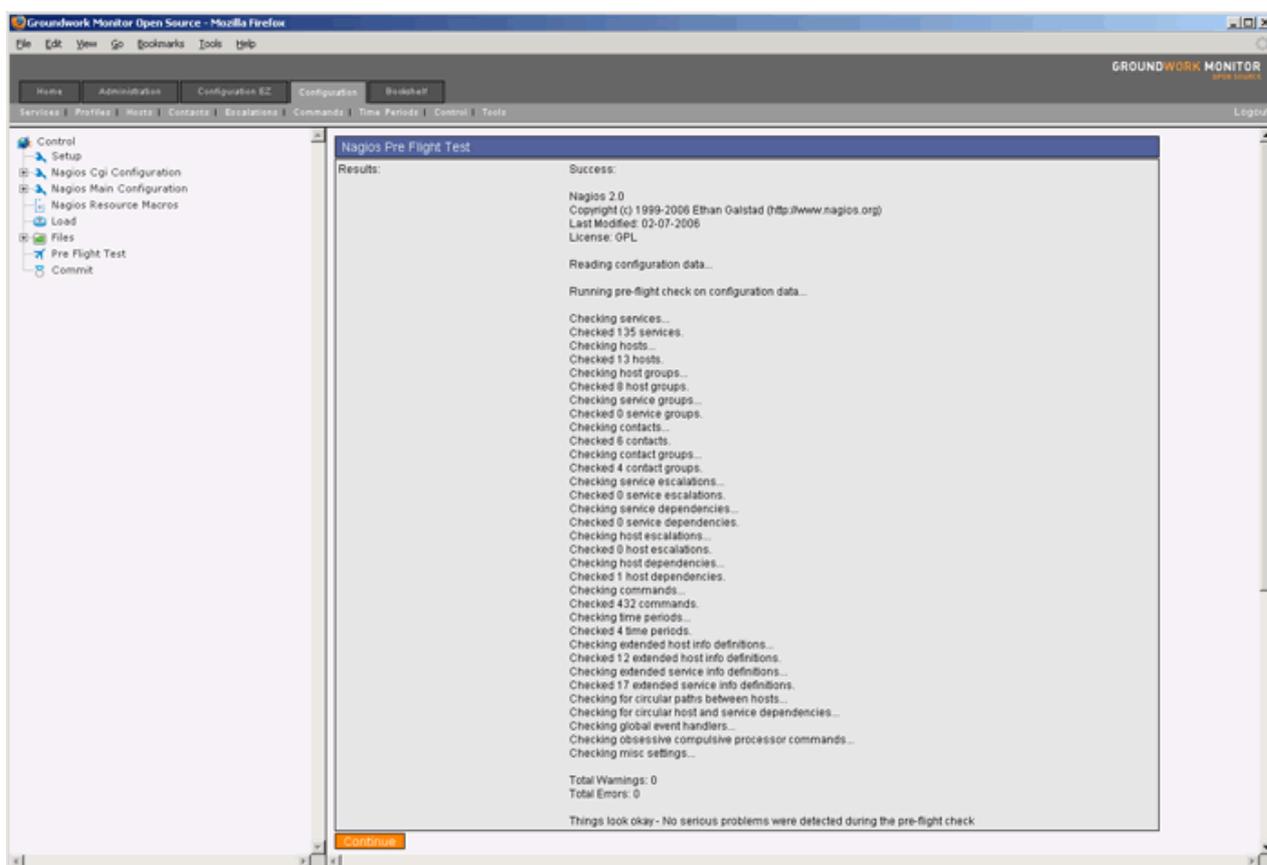
Control オプションを使用する:

どうやって、飛行前チェック (Pre Flight Test) を実行するか？

ステップ1 - Nagios の飛行前テスト (Pre Flight Test)

1. Configuration メニューオプションから Control を選びます。
2. Pre Flight Test を選びます。
3. 飛行前テストが実施され、その結果が次の画面に表示されます。
4. Continue を選び、終了します。

図 5.11.11. Nagios Pre Flight Test 画面



Control オプションを使用する： どうやって、Nagios 構成設定をコミットするか？

ステップ1 – Nagiosのコミット

1. Configuration メニューオプションから Control を選びます。
2. Commit を選びます。
3. Nagios Commit 画面で、稼働中の Nagios 構成設定をバックアップするよう促されます。
4. 稼働中のNagios構成設定を上書きし、Nagios を再起動するため、Commit を選びます。

図 5.11.12. Nagios Commit 画面

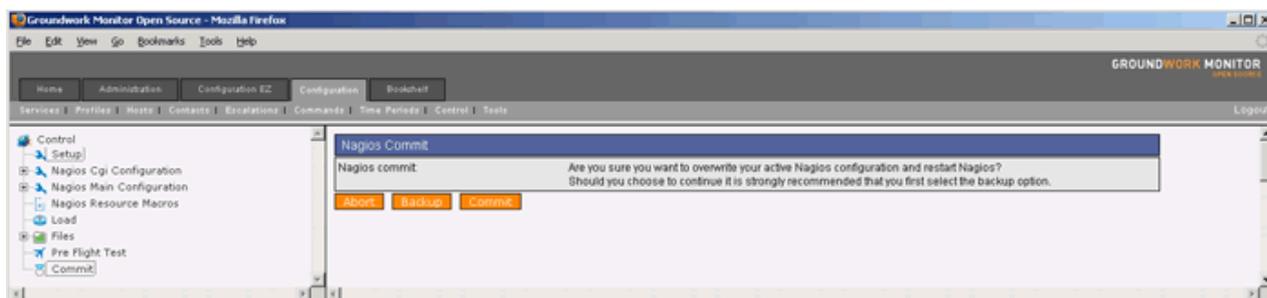


表 5.11.12. Nagios Commit

Commit	稼働中のNagios構成設定を上書きして Nagios を再起動するには、Commit を選びます。
Backup	データベース中の Nagios レコードをバックアップするには、Backup を選びます。
Abort	Nagios レコードのロードを中止あるいはキャンセルする場合、Abort を選びます。

追補10. Monarch内の Tools オプションを使う

目次

どうやって、構成設定ファイルをエクスポートするか？

Tools オプションの使用: どうやって、構成設定ファイルをエクスポートするか？

ステップ1 - ファイルのエクスポート

1. Configuration メニューオプションから Tools を選びます。
2. Export を選びます。
3. 次の画面で、usr/local/groundwork/apache2/htdocs/monarch/download ディレクトリにダウンロードされた構成ファイルのリストが表示されます。それぞれのファイルで右クリックし、Save Link As を選んで、あなたの好きなディレクトリを指定するかブラウザの中で表示されるファイルリンクをクリックすることができます。

図 5.12.1. ファイルのエクスポート

